

Андрей Прахов

Самоучитель Blender 2.6

Санкт-Петербург

«БХВ-Петербург»

2013

УДК 681.3.06
ББК 32.973.26-018.2
П70

Прахов А. А.

П70 Самоучитель Blender 2.6. — СПб.: БХВ-Петербург, 2013. — 384 с.: ил. — (Самоучитель)

ISBN 978-5-9775-0823-0

Книга предназначена для самостоятельного освоения трехмерного моделирования и анимации в свободно распространяемой программе Blender 2.6. Описано простое моделирование с помощью примитивов Mesh, использование кривых, поверхностей NURBS, материалов и текстур, создание анимации. Рассмотрены различные физические системы для симуляции движений частиц (гравитация, силовые поля, жидкости, дым, волосы и мех, ткани), свет, камеры и окружение. Описаны система рендеринга, особые функции Blender (Node Editor, NLA Editor, Grease Pencil, редактор видео VSE, плагины). Изложение сопровождается как простыми и наглядными примерами, так и расширенными уроками.

На сайте издательства находятся файлы всех рассмотренных в книге примеров.

Для широкого круга пользователей

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Елена Васильева</i>
Редактор	<i>Екатерина Капалыгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 31.07.12.
Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 30,96.
Тираж 1000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.
Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-0823-0

© Прахов А. А., 2013
© Оформление, издательство "БХВ-Петербург", 2013

Оглавление

Введение.....	7
Глава 1. Знакомство с интерфейсом.....	11
1.1. Оконная система	11
1.2. Устройства ввода и "умное меню"	16
1.3. Концепция экранов и сцен	16
1.4. Объекты в Blender.....	19
1.5. Ориентация в 3D-пространстве	21
1.6. Базовые манипуляции объектами.....	24
1.7. Иерархия сцены: группы, связи, слои	33
1.8. Работа с файлами	39
Глава 2. Простое моделирование с Mesh.....	43
2.1. Примитивы и их структура	43
2.2. Основные инструменты редактирования.....	46
2.3. Симметричное моделирование	57
2.4. Булевы операции.....	66
2.5. Вспомогательная решетка <i>Lattice</i>	69
2.6. Высокополигональное моделирование	73
2.7. Практика. Модель веера	81
2.8. Практика. Кусочек сыра	85
Глава 3. Кривые, поверхности <i>NURBS</i>	95
3.1. Основные понятия	95
3.2. Простейшие операции со сплайнами	97
3.3. Деформация объектов с помощью кривой	102
3.4. Создание объемных моделей	105
3.5. Знакомимся с поверхностями <i>NURBS</i>	110
3.6. Работа с текстом.....	115
3.7. Практика. Как сделать смайлик	122
3.8. Практика. Модель лодки	126
Глава 4. Материалы и текстуры.....	135
4.1. Что такое материал	135
4.2. Создание и настройка материала.....	136

4.3. Базовый цвет и отражение	141
4.4. Рамповые шейдеры	144
4.5. Эффекты <i>Halo</i>	147
4.6. Мультиматериалы	153
4.7. Отражение и преломление	155
4.8. Создание и настройка текстур	160
4.9. Процедурные текстуры.....	170
4.10. Карты <i>Normal</i> и <i>Displacement</i>	180
4.11. Наложение текстуры по развертке UV.....	194
4.12. Ручная окраска текстуры и вершин.....	199
4.13. Практика. Замшелый камень	202
4.14. Практика. Сочное яблоко	208

Глава 5. Анимация..... 215

5.1. Основы анимации в Blender	215
5.2. Простое управление с <i>Timeline</i>	216
5.3. Точная настройка анимации с <i>Graph Editor</i>	220
5.4. Движение объекта по кривой.....	227
5.5. Анимация и деформация	233
5.6. Основы анимации персонажа	237
5.7. Создание и редактирование скелета.....	239
5.8. Нарастиваем "мясо"	247
5.9. Для чего нужны "ограничители"	253
5.10. Работа с <i>Action Editor</i>	261
5.11. <i>NLA Editor</i> — заключительный аккорд.....	264
5.12. Практика. Жарим яичницу	268

Глава 6. Физика..... 279

6.1. Физический мир Blender	279
6.2. Создание и настройка частиц.....	281
6.3. Моделирование волос и меха.....	290
6.4. Работа с <i>Soft Body</i>	297
6.5. Создание ткани.....	302
6.6. Силовые поля	304
6.7. Имитация жидкости.....	307
6.8. Как сделать дым	313
6.9. Практика. Создание торнадо.....	316
6.10. Практика. Следы на воде.....	319

Глава 7. Свет, камеры и окружение..... 323

7.1. Источники света.....	323
7.2. Солнце и атмосфера.....	328
7.3. Работа с камерой.....	331
7.4. Окружение: звезды, туман, глобальный свет	334
7.5. Практика. Закат солнца	341

Глава 8. Система рендеринга Blender..... 345

8.1. Основы обработки	345
8.2. Дополнительные возможности	350

Глава 9. Что еще умеет Blender	353
9.1. Изучаем <i>Node Editor</i>	353
9.2. Встроенный редактор видео	357
9.3. Восковой карандаш.....	359
9.4. Скрытые возможности	362
9.5. Практика. Приемы работы с нодами.....	367
 ПРИЛОЖЕНИЯ	 375
Приложение 1. Инсталляция Blender	377
П1.1. Установка программы в Windows	377
П1.2. Установка программы в Linux	378
 Приложение 2. Файловый архив	 379
Предметный указатель	381

Введение

Пару десятилетий назад трехмерная графика была недостижимой и вызывала восхищение. Сейчас она стала привычной обыденностью. С экранов телевизоров, компьютеров и даже мобильных телефонов льется информация, насыщенная трехмерными элементами. Реклама, художественные фильмы и мультфильмы, компьютерные игры, виртуальные студии новостей — этот список можно продолжать до бесконечности. Мир 3D стал популярен.

Во время просмотра какого-нибудь популярного фильма, вы наверняка восхищались спецэффектами и реалистичной трехмерной графикой. Возможно, даже завидовали тем специалистам, которые работали над ним. Многие считают, что создание трехмерной графики — это удел избранных. Ничего подобного! Капля усидчивости, немного воображения, а главное горячее желание — и мир 3D вам покорится.

Для создания трехмерной графики нужно специальное программное обеспечение. Сейчас имеется широкий выбор соответствующих приложений, вот только большинство профессиональных пакетов стоят очень и очень дорого. Но почему не воспользоваться бесплатной альтернативой?

"Бесплатный сыр бывает только в мышеловке" — это изречение вполне могло прийти вам в голову, но не все так просто. Многие считают, что свободные и бесплатные программы в чем-то уступают своим проприетарным собратьям. Firefox, Chrome, Open Office, Gimp, Mplayer — всего лишь несколько известных приложений, которые в корне опровергают приведенную поговорку.

Трехмерный редактор Blender — это жемчужина в коллекции свободных программ. У него удивительная история, с которой, право, стоит ознакомиться.

В далеких 80-х годах XX века, когда трехмерная графика была в новинку и ей занимались всего несколько известных компаний, появилась небольшая голландская анимационная студия NeoGeo. Вскоре она заняла лидирующее положение. Прошли годы, и один из ее основателей Тон Розендаль (Ton Roosendaal), отвечавший за разработку внутреннего программного обеспечения, пришел к выводу, что используемый 3D-инструментарий устарел и требует замены. Так в недрах компании зародился программный продукт, который вскоре стал известен всему миру под названием Blender.

Изначально эта программа использовалась только для внутренних целей студии, но со временем Тон пришел к выводу, что Blender созрел для широкой аудитории. В 1998 году появляется дочерняя фирма Not a Number (NoN), которая занимается исключительно разработкой и продвижением Blender.

Новая компания продвигала по тем временам революционный подход к распространению профессионального программного продукта, предлагая его бесплатно, в то время как его собратья стоили многие тысячи долларов. Бизнес-модель NaN строилась на коммерческом сопровождении программы.

В 1999 году компания демонстрирует свое детище на всемирно известной конференции Siggraph. Успех был ошеломляющий! На волне подъема после конференции NaN получает громадное вливание в виде 4,5 млн евро. Это позволяет существенно ускорить работу над программой, и вскоре появляется новая версия Blender со встроенным игровым движком. Всего через полгода количество зарегистрированных пользователей на официальном сайте компании перевалило за полмиллиона. Будущее казалось радужным...

Но неумелое управление, большие амбиции и рыночные реалии того времени поставили NaN на грань банкротства. У компании меняется инвестор, производится реструктуризация, а главное меняется подход к распространению продукта. Через несколько месяцев появляется первая коммерческая версия Blender Publisher. Это не спасает положение, и инвестор прекращает функционирование NaN, а соответственно, разработку Blender.

Такая новость взбудоражила тысячи пользователей, успевших приобрести и полюбить эту программу. Тон Розендаль принимает решение о создании некоммерческой организации Blender Foundation, главной задачей которой был поиск возможностей для продолжения разработки и продвижения Blender. Вскоре ему удается договориться с инвесторами NaN о продаже исходных кодов программы и передачи права интеллектуальной собственности за 100 тысяч евро.

Была организована уникальная компания по сбору денег "Free Blender" (Свободный Blender), которая на удивление всего мира, всего за семь недель собрала требуемую сумму. С 2002 года Blender обрел новую жизнь и стал полностью свободной программой.

Сейчас Blender — это уникальный программный комплекс, позволяющий создавать реальный и красочный трехмерный мир. Его возможности сравнимы с популярными коммерческими пакетами, такими как Maya, 3Ds Max и даже больше того. С помощью свободного инструментария вы можете создавать модели, работать с анимацией, использовать законы физики для имитации природных явлений.

- ◆ Если вы работаете в сфере телевидения, Blender станет неоценимым помощником в создании эффектов. Кроме того, он имеет встроенный видеоредактор, который позволяет с легкостью объединять трехмерную сцену с видео.
- ◆ Если вы разработчик игр, то наверняка знаете, что многие игровые движки поддерживают модели, созданные в Blender. Вас интересует эта программа как игровой конструктор? Blender имеет мощный, встроенный игровой движок, ко-

торый в совокупности со средствами моделирования и анимации представляет собой законченное решение для создания игр.

- ◆ Если вы дизайнер интерьера, программа пригодится и тут. Мощные средства разработки, большое количество плагинов для импортирования объектов, созданных в других приложениях, realtime-движок, который позволит клиенту тут же прогуляться в созданной комнате.
- ◆ И таких "если" может набраться великое множество. Прибавьте к этому бесплатность и кроссплатформенность. Blender умеет работать в разных операционных системах: Windows, Linux, Mac Os X, FreeBSD. Посетите сайт **www.blender.org**, чтобы просмотреть все варианты и скачать нужную версию. Кстати, размер дистрибутива программы весит всего около 30 Мбайт!
- ◆ В 2009 году вышла моя книга по Blender: "3D-моделирование и анимация". Она основывалась на версии программы 2.48, которая в тот момент была актуальной. Но за прошедшие годы Blender сильно изменился. Начиная с версии 2.5x, разработчики выполнили колоссальную работу по оптимизации и расширению функционала программы. Изменился подход к использованию интерфейса, появились новые инструменты и возможности. К сожалению, первый учебник устарел.
- ◆ Книга, которую вы держите в руках, не просто рассматривает мажорную версию программы (на момент написания 2.62), здесь в корне изменен принцип подачи информации. Теперь главный упор сделан на практичность материала и излишней перегруженности теорией нет.

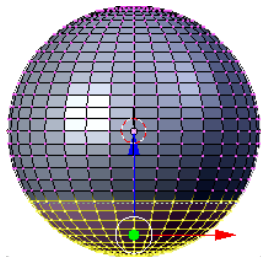
В ней имеется большое количество несложных примеров, которые необходимо выполнять на практике. Кроме того, в конце глав приводятся дополнительные, масштабные уроки для закрепления материала. Все практические примеры, описываемые в книге, вы можете найти в файловом архиве (см. приложение 2).

Если все же у вас останутся вопросы, вы можете обратиться к дружелюбному сообществу пользователей Blender в сети Интернет:

- ◆ **www.blender.org** — официальный сайт Blender Foundation. Здесь размещены новости от разработчиков, свежие сборки программы, документация;
- ◆ **www.blenderartists.org** — крупнейший англоязычный форум, посвященный программе;
- ◆ **www.b3d.mezon.ru** — русскоязычная wiki с большим количеством переведенной документации. Там же есть ссылки на русскоязычные ресурсы.

Итак, открывайте следующую страницу и окунайтесь в безбрежный океан возможностей Blender. Удачи!

ГЛАВА 1



Знакомство с интерфейсом

Когда приходится запускать неизвестное приложение, первым делом обращаешь внимание на то, как оно выглядит. Не зря одна известная поговорка гласит: "Встречают по одежке...". А вот "проводить по уму" такую программу, как Blender, вам и не захочется. Если вы любитель или профессионал в области 3D-графики, то мощь и простота этого редактора вас покорают.

"Но, простите!, — воскликнет дотошный читатель. — Ведь о Blender ходит слава сложной программы с запутанным интерфейсом". Резонно замечу, что любое серьезное приложение требует вдумчивого изучения. По мере накопления опыта все кажущиеся сложности уйдут на задний план. Будь то Blender или Maya, или 3Ds Max — в умелых руках любая программа станет послушным инструментом.

Год назад разработчики Blender решились на беспрецедентный шаг и в корне изменили интерфейс программы. Элегантный внешний вид, полностью настраиваемая рабочая область, гибкое конфигурирование горячих клавиш — все это призвано облегчить работу пользователя, дать возможность ему сконцентрироваться исключительно на творчестве. Если вы ранее работали с другими программами 3D-моделирования, то первоначально многое будет казаться неудобным и несуразным. Однако это покажется только на первый взгляд. Ведь Blender обладает собственным подходом к созданию 3D-контента, своего рода мировоззрением. Постигнув его, начинаешь понимать, как удобно и продуманно все в нем устроено. В конце концов, ничто не мешает настроить программу на свой вкус, благо разработчики предлагают для этого широчайшие возможности.

Эта глава является базовой в понимании работы с Blender. Прочитав ее, вы изучите интерфейс программы, познакомитесь с основными объектами и даже сделаете свою первую модель.

1.1. Оконная система

Итак, вы установили Blender, запустили программу и готовы идти в бой. Что ж, давайте начинать.

При запуске программа всегда встречает приветственным окном с какой-либо занимательной картинкой. От версии к версии изображение меняется, но это и неважно. Важно то, что данное окошко имеет ссылки на полезные интернет-ресурсы, посвященные Blender, а также содержит список последних рабочих проектов. Кроме того, есть еще два интересных пункта:

- ♦ **Recover Last Session** (Восстановить последний сеанс) — вызывает проект, из которого был произведен выход из программы. Дело в том, что когда вы закрываете Blender, происходит автоматическое сохранение файла с именем `quit.blend` в локальной директории `Temp` текущего пользователя. Эта функция чрезвычайно полезна, если вы забыли сохранить проект перед выходом;

ЗАПОМНИТЕ!

Blender никогда не предупреждает о необходимости сохранения, даже если оно необходимо!

- ♦ **Interaction** (Взаимодействие) — данная ссылка будет интересна тем, кто ранее работал с пакетом Maya. Просто выберите из предлагаемого списка пресетов нужный пункт и получите привычную раскладку горячих клавиш.

А теперь давайте посмотрим на скриншот стандартного окна программы (рис. 1.1).

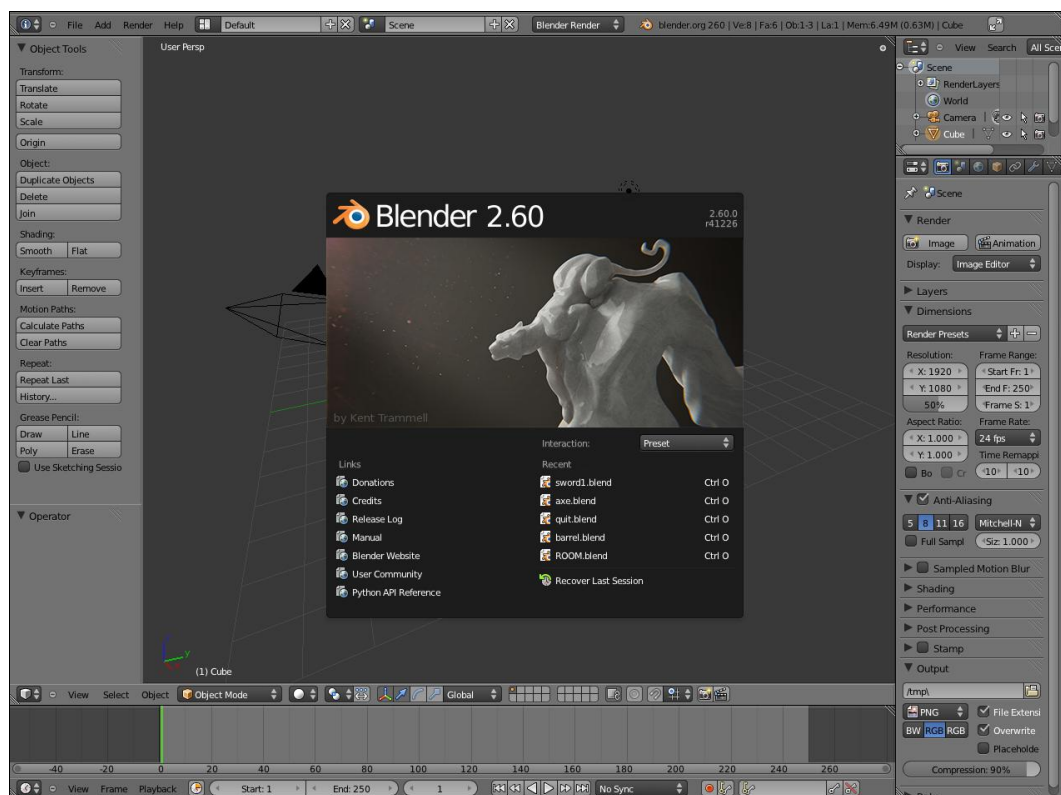


Рис. 1.1. При первом запуске Blender выглядит именно так

В действительности здесь не одно окно, а целых пять. И вы можете делать с ними, что захотите: изменять размеры, добавлять новые, удалять ненужные и даже отсоединять в виде отдельного окна. Последняя функция пригодится тем, у кого несколько мониторов.

Попробуем немного поиграть с окнами. Щелкните в любом месте программы, чтобы закрыть приветствие Blender. Если вы подведете курсор к какой-либо границе между панелями, то он поменяется на характерный двухсторонний указатель. Теперь вы можете, зажав левую кнопку мыши, перетянуть границу в нужную сторону.

СОВЕТ

Экспериментируй, проверяй, используй! Вот ваш лозунг для лучшего усвоения материала. Все несложные упражнения, приведенные в книге, старайтесь повторять на практике.

Весьма оригинально в Blender выполнено управление окнами: удаление, объединение и отсоединение. Для этих операций служит небольшой уголок в верхнем правом углу каждого окна (рис. 1.2).

Уцепитесь курсором за любой понравившийся уголок и попробуйте немного перетянуть влево. Как видите, появился дубликат окна. Чтобы объединить их, достаточно перетянуть тот же уголок вправо. При этом вы увидите схематичные стрелки, указывающие, куда именно "вольется" окно. Можете поменять направление, перемещая вправо или влево мышью.

Запомните два простых правила:

- ◆ для *добавления* окна — движение мышью влево или вниз;
- ◆ для *объединения* окон — движение мышью направо или вверх.

Отсоединение окна выполняется еще проще:

1. Нажмите и удерживайте клавишу <Shift>.
2. Уцепитесь курсором за уголок.
3. Перетяните мышью в любую сторону.

Такое управление окнами достаточно необычное, но на практике оказывается весьма эффективным.



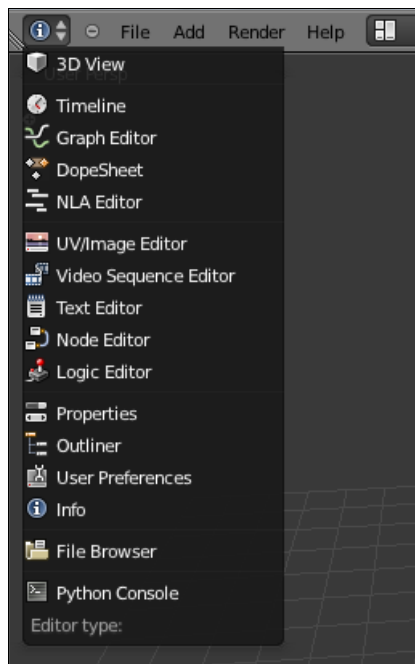
Рис. 1.2. Ключевой элемент для управления окном

Blender — программа многофункциональная, позволяет моделировать, выполнять анимацию, создавать видеопоследовательности, поддерживает скриптование. Вполне понятно, что разработчики озаботились надлежащим образом сгруппиро-

вать инструментарий отдельными блоками для оптимального использования. Давайте познакомимся с глобальными окнами программы.

Присмотритесь внимательно к окнам приложения. С правой стороны, вверху, у каждой области имеется небольшая кнопка, при нажатии которой выскакивает меню (рис. 1.3). Именно в нем вы можете выбрать, какое окно будет находиться в данной области.

Рис. 1.3. Это меню содержит все возможные окна программы



Приведем список возможных окон.

- ◆ **3D View** (3D Вид) — основное окно для создания и просмотра сцены.
- ◆ **Timeline** (Шкала времени) — временная шкала для контроля анимации.
- ◆ **Graph Editor** (Редактор графов) — редактор анимационных ключей. Позволяет с помощью кривых точно корректировать поведение конкретного объекта и его свойств.
- ◆ **DopeSheet** (Таблица ключей) — многофункциональный редактор анимации, имеющий три режима работы:
 - **DopeSheet** — управление ключевой анимацией всей сцены;
 - **Action Editor** (Редактор действий) — специальный режим для управления только скелетной анимацией;
 - **Shape Key Editor** (Редактор ключей формы) — настройка внутренней анимации конкретного объекта.
- ◆ **NLA Editor** (Редактор NLA) — нелинейный редактор последовательностей. Его удобно использовать для глобального контроля анимации всей сцены.
- ◆ **UV/Image Editor** (Редактор изображений) — создание, наложение текстур на объект.
- ◆ **Video Sequence Editor** (Редактор видеопоследовательностей) — полнофункциональная монтажная система, позволяющая объединять 3D-сцену с видеофайлами, использовать переходы, работать со звуковыми дорожками.

- ◆ **Text Editor** (Текстовый редактор) — встроенный текстовый редактор для написания сопроводительного текста к сцене или создания скриптов на языке Python. При этом данные сохраняются в самой сцене.
- ◆ **Node Editor** (Редактор узлов). Уникальный инструмент, позволяющий в графическом виде настраивать свойства объектов, сцены, результата рендера (окончательная визуализация сцены).
- ◆ **Logic Editor** (Редактор логики) — окно для создания игровой логики.
- ◆ **Properties** (Свойства) — основное окно, позволяющее настраивать свойства объектов, сцены, рендера.
- ◆ **Outliner** (Планировщик) — вспомогательный инструмент для просмотра и организации сцены.
- ◆ **User Preferences** (Пользовательские настройки) — настройки программы.
- ◆ **Info** (Информация) — глобальное меню программы и полезная информации о сцене или выделенном объекте.
- ◆ **File Browser** (Браузер файлов) — файловый менеджер для загрузки и сохранения файлов. Имеет немало функций, оптимизированных под нужды Blender.
- ◆ **Python Console** (Консоль) — интерпретатор Python.

Не правда ли, впечатляющий список инструментария для программы объемом всего 30 Мбайт? Конечно, многие термины вам пока непонятны, но по мере прочтения книги все станет ясно. А теперь давайте вернемся к теме раздела.

Любое окно в Blender имеет стандартные области: кнопка выбора режима (рассматривалась ранее), текстовое или графическое меню, рабочая область. Интересно, что и вид окна можно подстраивать на свой вкус. К примеру, вы можете переместить наверх стандартно размещенное внизу меню или вообще его спрятать. Щелкните правой кнопкой мыши в любом месте по заголовку, и появится небольшая панель **Header** (Заголовок). На ней всего два пункта:

- ◆ **Flip to Top** (Отразить вверх) — перемещает меню вверх окна. После этого надпись изменится на **Flip to Bottom** (Отразить вниз);
- ◆ **Maximize Area** (Максимальный охват) — распаивает текущее окно на весь экран.

Вы можете полностью скрыть меню, если оно не нужно. Подведите курсор к границе между меню и рабочей областью. Он должен поменяться на изображение вертикальных стрелок. Нажмите левую кнопку мыши и сожмите меню. А вот для восстановления его ищите небольшую круглую кнопку с крестиком (см. рис. 1.2). Заметьте, что такие кнопки появляются для всех скрытых панелей текущего окна.

СОВЕТ

В Blender очень активно используются горячие клавиши. Так, к примеру, для распаивания активного окна (т. е. такого, в котором находится курсор мыши) достаточно нажать комбинацию клавиш <Ctrl>+<Up>.

1.2. Устройства ввода и "умное меню"

Работа с мышью и клавиатурой в Blender имеет свои особенности по сравнению с альтернативными программами. Забудьте про всевозможные контекстные меню, вызываемые правой кнопкой мыши, — их ничтожно мало, существующие относятся к настройке окон. Все манипуляции с интерфейсом осуществляются левой кнопкой мыши, а вот правая служит исключительно для выделения объектов и их элементов. Средняя кнопка (колесико) предназначена для вспомогательных функций: масштабирование и поворот сцены в **3D View**, прокручивание содержимого панелей.

Большое значение в Blender отводится использованию комбинаций горячих клавиш. Разумеется, имеются их альтернативы в виде интерфейсных элементов (кнопок, меню), но на практике гораздо проще пользоваться только ими. На многие горячие клавиши повешены и контекстные меню. Для быстрого выбора необходимого пункта в них можно нажать соответствующую цифровую клавишу на основной клавиатуре.

Однако гордостью Blender является так называемое "умное меню". В прошлых версиях программы для создания новых объектов существовало специальное меню, вызываемое по клавише <Space>. Это значительно убистряло работу над сценой, ведь не приходилось лишний раз посещать главное меню приложения. Сейчас разработчики пошли еще дальше. Новая версия "умного меню" позволяет вызывать любую функцию Blender, буквально набрав несколько ключевых символов в поисковой строке. Причем в предлагаемый список попадут команды, исключительно относящиеся к выделенному элементу сцены (рис. 1.4). Так для поиска команды **Add UV Sphere** (создание примитива сферы) достаточно набрать `sphe`.

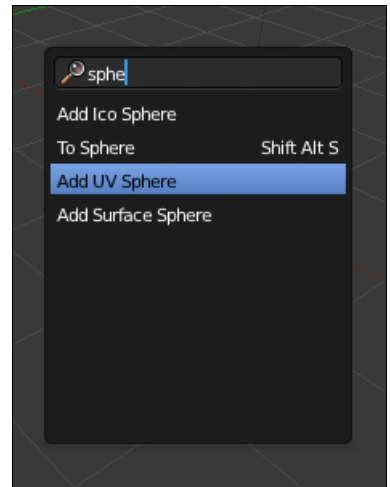


Рис. 1.4. Пара набранных букв — и список готов

1.3. Концепция экранов и сцен

Работая над сценой, приходится выполнять множество действий: моделировать, текстурировать, настраивать анимацию и физику, компоновать видеопоследовательности. Уже понятно, что Blender предоставляет широчайшие возможности для настройки рабочих окон. Вот только постоянно манипулировать окнами в зависимости от текущей задачи как-то нецелесообразно. Гораздо проще заранее создать

различные раскладки окон под все нужды. Разумеется, такая возможность в Blender имеется. Мало того, разработчики озаботились созданием нескольких заготовок под определенные задачи. Этот механизм получил название — **Screen Layout** (Макет экрана).

Пользоваться экранами очень и очень просто. Взгляните на рис. 1.5. Несложно определить, что здесь виден кусочек главного меню программы (окно **Info**).



Рис. 1.5. Элементы управления **Screen Layout**

Рассмотрим элементы окна **Info**:

1. Схематичное изображение окон — выбор имеющейся раскладки из выпадающего меню.
2. Название текущего экранного макета.
3. Добавление новой раскладки.
4. Удаление активной раскладки.

Для выбора желаемого макета достаточно щелкнуть по кнопке открытия пресетов и выбрать подходящий, благо разработчики предусмотрели несколько вариантов:

- ◆ **Animation** (Анимация) — набор и расположение окон; оптимально подходит для работы с анимацией;
- ◆ **Compositing** (Композиция) — настройка материалов и текстур в режиме **Node Editor** (Редактор узлов);
- ◆ **Default** (По умолчанию) — вариант по умолчанию, предлагаемый программой при первом запуске или создании нового проекта;
- ◆ **Game Logic** (Игровая логика) — оптимально для настройки логики игры или презентации;
- ◆ **Scripting** (Скриптование) — используйте эту раскладку, если необходимо отредактировать или написать скрипт на Питоне;
- ◆ **UV Editing** (Развертка UV) — создание, окраска и применение текстур;
- ◆ **Video Editing** (Редактирование видео) — подготовка видеопоследовательности.

СОВЕТ

Вы можете быстро переключать экранные макеты с помощью горячих клавиш **<Ctrl>+<Left>** или **<Ctrl>+<Right>**.

Иногда имеющихся вариантов макетов просто недостаточно. К примеру, во время моделирования необходимо просматривать рабочий объект с разных ракурсов. Для этой цели целесообразно создать экранную раскладку с несколькими окнами **3D View**.

Последовательность действий следующая:

1. Создать новый профиль путем нажатия кнопки с изображением плюса.
2. Переименовать объект (щелкните по названию и введите новое значение).
3. Расположить окна в нужном порядке.

Заметим, что все созданные макеты, равно как и встроенные, сохраняются в файле текущего проекта. Так что, при загрузке своего проекта вы опять вернетесь к любимым настройкам.

СОВЕТ

Создали уникальные экранные раскладки и желаете их использовать для всех проектов? Нет ничего проще. Нажмите <Ctrl>+<U> или выберите пункт меню **File | Save User Settings** (Файл | Сохранить настройки) для сохранения пользовательских настроек. Теперь при создании нового проекта вы сможете работать с собственными заготовками.

В Blender имеется уникальная возможность одновременной работы с несколькими сценами. Это не говорит о том, что происходит запуск разных экземпляров программы. Все происходит в одном и том же проекте. Управление сценами осуществляется на панели главного меню (рис. 1.6).

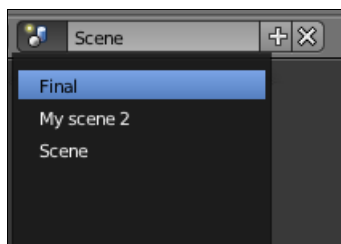


Рис. 1.6. Элементы управления Scene Selectors

В принципе, работа с этой панелью равноценна ранее рассмотренной **Screen Layout**. Но в отличие от последней, при нажатии кнопки добавления — появится меню **New Scene** с несколькими пунктами:

- ◆ **New** (Новая) — создание новой сцены без каких-либо объектов. Значения настроек рендера берутся по умолчанию;
- ◆ **Copy Setting** (Копирование настроек) — создание чистой сцены с текущими настройками рендера;
- ◆ **Link Objects** (Связь объектов) — копирование существующей сцены. При изменении расположения и свойств объектов в одной сцене результаты проявятся в другой;
- ◆ **Link Object Data** (Связь данных объектов) — в этом случае возможно независимое перемещение объектов, но изменения в свойствах фиксируются во всех сценах;
- ◆ **Full Copy** (Полная копия) — создание полностью независимой сцены с копиями объектов и настроек.

Для наглядности представим ситуацию, когда возможности одновременной работы с несколькими сценами нет или она не используется. Допустим, идет работа над созданием интерьера гостиной комнаты. По просьбе заказчика вы расставили виртуальную мебель и с гордостью демонстрируете результат. Заказчик в целом удовлетворен, но просит добавить эпизод, где эта мебель располагается несколько по-иному. Нет ничего проще, сохраняете проект в другой файл, меняете расположение объектов и обрабатываете результат. Заказчик просматривает оба эпизода и загорается желанием посмотреть, что произойдет со стеклянным столиком, если на него упадет дубовый шкаф. Желание клиента — закон. Создаете уже третий файл, где красиво вдребезги разлетается столик. Внезапно появляется партнер заказчика и заявляет, что его не устраивает форма шкафа. Все, весь труд насмарку, а впереди переделка трех проектов.

Смешная ситуация? Но она вполне реальна. Давайте посмотрим, что можно было бы сделать, используя механизм сцен Blender:

1. Создать сцену с полностью готовыми объектами.
2. Выполнить копирование сцены при помощи функции **Link Object Data** (Соединить с данными объекта). Теперь возможно независимое перемещение объектов в обеих сценах, но любые изменения свойств объектов проявятся во всех связанных сценах.
3. Создать третью копию, используя функцию **Full Copy** (Полная копия). Эта сцена уже независима от первых двух, зато содержит те же объекты и настройки.

1.4. Объекты в Blender

Когда смотришь на сложную, красивую трехмерную модель, то поневоле задумываешься, как ее делали. Трудно представить, что изначально она представляла собой какой-нибудь несложный объект, например куб. Такие объекты, которые служат первородными кирпичиками для создания моделей, называют *примитивами*.

Но примитивами не ограничивается состав сцены. В ней присутствует множество других объектов, без которых работа над проектом была бы невозможной. Одни служат для создания моделей, другие обеспечивают освещение, третьи являются вспомогательными при работе с анимацией или физикой. Blender предлагает большой набор объектов для реализации самой сумасшедшей идеи.

Для добавления нового объекта в сцену существует специальное меню **Add** (Добавить) в главном меню программы (рис. 1.7).

Blender имеет различные примитивы, подходящие для выполнения той или иной задачи. Причем для качественного результата нужно заранее правильно определиться, какой объект будет лежать в основе. Например, если нужно создать модель яблока, то оптимально использовать примитив сферы.

В некоторых случаях разные виды примитивов могут с равным успехом служить первоосновой. Так модель лодки можно сделать из того же куба, но более удобно

для этой цели использовать кривые **NURBS**. Следовательно, нужно ориентироваться в обилии примитивов Blender и знать их потенциальные возможности.

Все примитивы Blender разделены на две большие группы:

- ◆ объекты **Mesh** (плоскость, куб, сфера, цилиндр, конус, тор);
- ◆ математические объекты (кривые, поверхности, метаобъекты).

Отличие между ними заключается в способах генерации их программой. Так, если структура первых объектов представляет собой набор вершин, ребер и плоскостей (рис. 1.8), то вторые создаются на основе специальных математических функций (рис. 1.9). Это обеспечивает необычайную гибкость при редактировании форм.

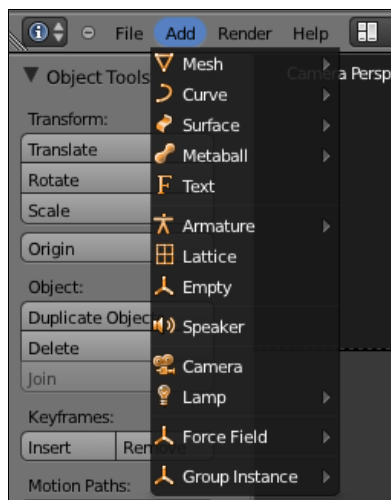


Рис. 1.7. Это меню содержит все объекты Blender



Рис. 1.8. Для создания меча использовался простой куб



Рис. 1.9. А этот бокал был создан с помощью кривых

Особняком от этих примитивов стоит объект **Text**. Как вы уже догадались, он служит для создания букв и текстов в сцене. Предлагаемые программой возможности чрезвычайно широки: от выбора и настройки шрифта до манипулирования независимыми текстовыми блоками. Объект **Text** является двухмерным и служит только для создания текста. В дальнейшем его обычно конвертируют либо в **Mesh**, либо в кривые.

Продолжим рассмотрение меню **Add**.

- ◆ Группы **Curve**, **Surface**, **Metaball** содержат примитивы, создаваемые с помощью математических функций. Благодаря этому редактирование таких объектов является более точным и удобным. Обычно их используют при высокополигональном моделировании.
- ◆ **Armature** (Арматура) — данный объект необходим при создании скелетной анимации.
- ◆ **Lattice** (Решетка) — вспомогательный объект, как правило, используемый для деформации модели.
- ◆ **Empty** (Пустышка) — вспомогательный объект, игнорируемый рендером программы при визуализации сцены.
- ◆ **Speaker** (Динамик) — используется для позиционирования источника звука в 3D-пространстве.
- ◆ **Camera** (Камера) — при первом запуске Blender в сцене уже есть одна камера, но вы можете добавить еще несколько.
- ◆ **Lamp** (Свет) — данный пункт содержит пять различных типов источников света, начиная точечным и заканчивая солнечным.
- ◆ **Force Field** (Силовые поля) — объекты, используемые для имитации физических явлений, таких как ветер.
- ◆ **Group Instance** (Группы) — иногда несколько объектов в сцене выгодно объединять в одну группу. После создания такой группы ее название появится в этом пункте.

СОВЕТ

Для быстрого вызова меню добавления объектов в сцену нажмите клавиши <Shift>+<A> в окне **3D View**.

1.5. Ориентация в 3D-пространстве

Основная работа в Blender происходит в окне **3D View**. Именно в нем создаются, редактируются и размещаются объекты. При первом запуске в окне отображается сцена с минимальным количеством необходимых объектов: камерой, источником света и кубом.

Окно **3D View** имеет большое количество интерфейсных элементов, управляющих режимами отображения и редактирования (рис. 1.10).

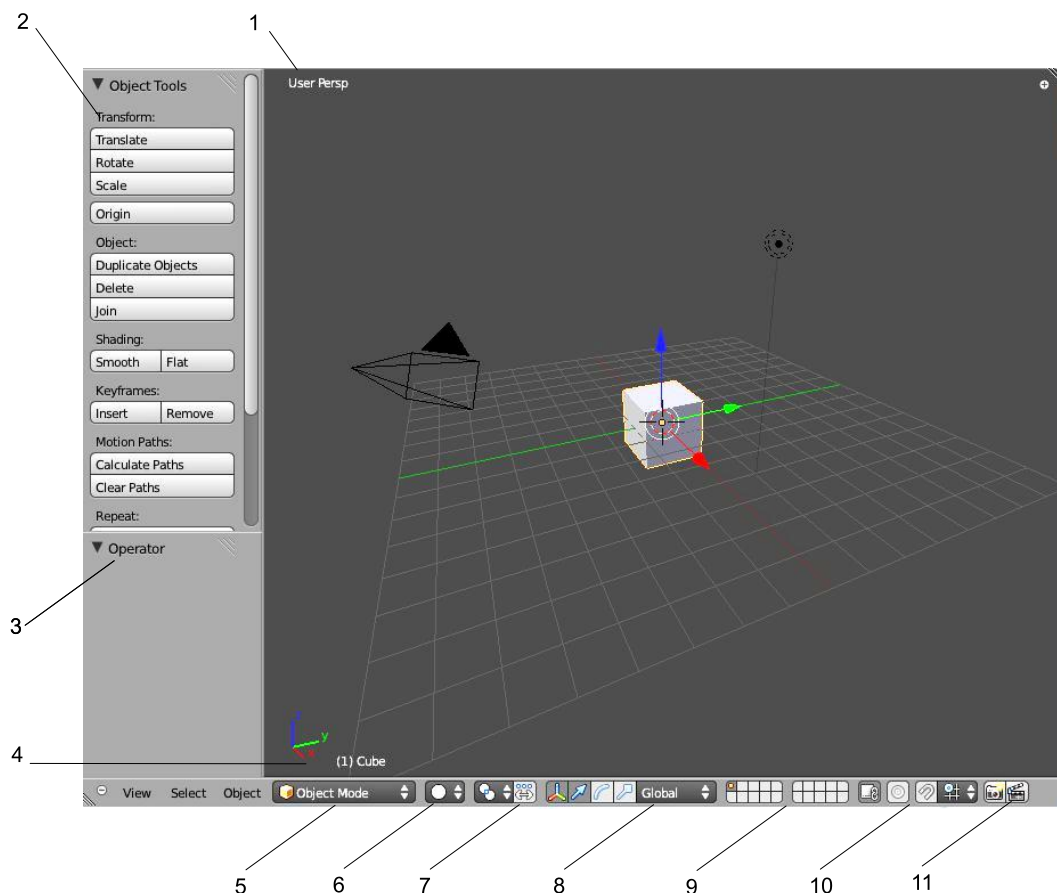


Рис. 1.10. Окно 3D View

Описание основных элементов 3D View:

1. Информация о режиме просмотра с точки зрения пользователя (вид сцены спереди, сбоку, сверху и т. д.).
2. **Tool Shelf** (Панель инструментов). Содержит основные функции, характерные для выделенного объекта и режима редактирования.
3. Эта панель содержит настройки выбранной функции или созданного объекта. К примеру, если добавить в сцену примитив **Circle**, то на данной панели отобразятся значения: **Vertices** (Количество вершин), **Radius** (Радиус), **Fill** (Заливка).
4. Название выделенного объекта в сцене.
5. **Mode** (Режим). Выбор режима работы 3D View.
6. **Viewport Shading** (Режим отрисовки). Качество отображения объектов на экране, от схематичного до текстурного.
7. **Pivot Point** (Точка центра). Используется для указания точки в пространстве, по отношению к которой будет выполняться поворот или масштабирование объекта.

8. Кнопки выбора вида манипуляции объектом (перемещение, масштабирование, поворот).
9. Управление отображением слоев в сцене.
10. **Snap** (Привязка). Вспомогательный режим, для более точного манипулирования объектом.
11. Кнопки "мгновенного" черного рендеринга.
 - Нужно заметить, что некоторые элементы заголовка окна **3D View** могут изменяться в зависимости от выбранного режима (**Mode**). Более подробно они будут изучаться в следующих главах.
 - При моделировании очень важно рассматривать модель или сцену со всех сторон. Blender позволяет это делать различными способами: с использованием мыши и горячих клавиш.

Рассмотрим основные способы просмотра сцены.

- ◆ *Вращение*. Осуществляется путем нажатия и удержания средней кнопки мыши (колесика). Точкой вращения служит центр окна. Таким образом, чтобы детально рассмотреть нужный объект, достаточно поместить его в центр экрана.
- ◆ *Панорамирование* (перемещение по вертикали или горизонтали). Используется все та же средняя кнопка мыши с нажатой клавишей <Shift>.
- ◆ *Масштабирование*. Для увеличения или уменьшения масштаба сцены используется колесико мыши. Альтернативой служит комбинация клавиши <Ctrl> и средней кнопки мыши.

Кроме этих основополагающих манипуляций, Blender предлагает заготовки просмотра, вызываемые горячими клавишами дополнительной цифровой клавиатуры:

- ◆ <NumPad 1> (Front View) — вид сцены спереди;
- ◆ <NumPad 3> (RightView) — вид сцены справа;
- ◆ <NumPad 7> (Top View) — вид сцены сверху;
- ◆ <NumPad 0> (Front View) — проекция из камеры;
- ◆ <NumPad 5> — переключение отображения из перспективы в ортогональную проекцию и наоборот;
- ◆ <NumPad 4> и <NumPad 6> — вращение сцены по вертикальной оси;
- ◆ <NumPad 2> и <NumPad 8> — вращение сцены по горизонтальной оси;
- ◆ <NumPad +> и <NumPad -> — масштабирование;
- ◆ <NumPad .> — выделенный объект помещается в центр и заполняет всю рабочую область окна.

Итак, вся дополнительная цифровая клавиатура в Blender полностью завязана на управление просмотром окна **3D View**. Запомните эти сочетания клавиш, в будущем они не раз пригодятся.

Теперь рассмотрим меню **Viewport Shading** (Режим отрисовки). Как уже было сказано, в нем можно выбрать способ прорисовки содержимого окна **3D View**. При

нажатии кнопки **Viewport Shading** (Режим отрисовки) (см. № 6 на рис. 1.10) программа предложит следующий список:

- ◆ **Textured** (Текстурные) — максимально приближенный к реальности режим прорисовки объектов: показываются текстуры, тени, сложные шейдеры;
- ◆ **Solid** (Сплошной) — используется по умолчанию: сами объекты, основные цвета материалов, простые шейдеры;
- ◆ **Wireframe** (Каркас) — простое, схематичное представление объектов, этот режим удобно использовать для моделирования;
- ◆ **Bounding Box** (Границы объекта) — реальные объекты заменяются параллелепипедами — минимальное представление, но зато максимальная скорость перерисовки сцены.

1.6. Базовые манипуляции объектами

В этом разделе вы познакомитесь с основными способами управления объектами в сцене, такими как: перемещение, масштабирование, ротация, добавление и удаление, дублирование.

Запустите Blender и обратите внимание на объекты, находящиеся в окне **3D View**. По умолчанию их должно быть три: камера, источник света и куб (см. рис. 1.10).

Чтобы управлять объектом, его необходимо предварительно выделить. Для этого служит правая кнопка мыши. При этом края активного объекта отмечаются оранжевым цветом. Кроме того, появляются вспомогательные элементы в виде разноцветных стрелок с окружностью. Так Blender отмечает центр объекта (окружность) и по умолчанию включает манипулятор перемещения. Попробуйте выделить куб, затем ухватиться за любую из стрелок и, нажав левую кнопку мыши, осторожно переместить ее в сторону — объект послушно последует за курсором. Заметьте, что в этом случае перемещение осуществляется строго по выделенной координатной оси. Для произвольного перемещения вне отдельной оси наведите курсор на объект и, нажав правую кнопку мыши, двигайте манипулятор.

На заголовке окна **3D View** имеется специальная группа кнопок, активирующих разные типы манипуляторов (рис. 1.11):

1. Включение или отключение визуального отображения в окне активного манипулятора. Это не значит, что объект нельзя будет перемещать или масштабировать. Просто придется использовать горячие клавиши.
2. Режим перемещения (**Grab**).
3. Режим вращения (**Rotate**).

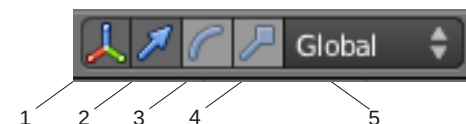


Рис. 1.11. Панель управления манипуляторами

4. Режим масштабирования (**Scale**).

5. Выбор координатной системы.

При выборе режима вид вспомогательных элементов у объекта соответствующим образом изменится. Также имеется возможность включения сразу нескольких манипуляторов. Просто выбирайте нужные из них, удерживая нажатой клавишу <Shift>.

В большинстве случаев работать с манипуляторами эффективнее с использованием горячих клавиш:

- ◆ <G> — перемещение;
- ◆ <S> — масштабирование;
- ◆ <R> — вращение.

Причем имеется возможность движения объекта строго по нужной координатной оси:

- ◆ <X> — координата X;
- ◆ <Y> — координата Y;
- ◆ <Z> — координата Z.

Например, необходимо переместить объект по координате Z. Соответственно, нажимаем клавишу <G>, затем <Z> и перемещаем объект. Для закрепления результата служит левая кнопка мыши. Чтобы отменить действие, достаточно щелкнуть правой кнопкой мыши или нажать клавишу <Esc>.

СОВЕТ

Вы можете использовать клавишу <Shift> для замедления движения объекта при перемещении, масштабировании или ротации. А вот если нажать клавишу <Ctrl>, то движение объекта будет привязано к вспомогательной сетке окна **3D View**.

Немаловажным моментом является выбор подходящей координатной системы. По умолчанию Blender работает в глобальной системе, т. е. в такой, где координатные оси объектов совпадают с осями самой сцены. Для выбора иной ориентации служит соответствующее меню на панели манипуляторов (см. № 5 на рис. 1.11).

Всего программа предлагает пять вариантов:

- ◆ **View** (Оконная) — зависит от текущей проекции окна **3D View (Front, Top, Right)**;
- ◆ **Normal** (Нормали) — ориентация по нормальям объекта;
- ◆ **Gimbal** (Джимбал) — используется при скелетной анимации;
- ◆ **Local** (Локальная) — собственная координатная система выделенного объекта;
- ◆ **Global** (Глобальная) — ориентация по осям сцены.

Теория без практики ничто, не так ли? А ведь даже того, что вы узнали на этом этапе, вполне достаточно для начала работы над простейшей моделью, например кувалды.

Итак, кувалда состоит из двух деталей: рукояти и бойка. Blender уже имеет подходящие примитивы для создания этих объектов. Для рукояти подойдет **Cylinder** (Цилиндр), а роль бойка возьмет на себя обычный куб.

Начнем с чистого листа. Создайте новый проект, нажав <Ctrl>+<N> или выбрав пункт меню **File | New** (Файл | Новый). На запрос программы **Reload Start-Up File** (Перезагрузить начальный файл) нажмите клавишу <Enter>.

Blender предлагает начать работу с примитивом куба. В данный момент он нам пока не нужен, поэтому удалим его. Для этого выделите куб, нажмите клавишу <X> и подтвердите решение в появившемся окошке.

Сейчас нужно добавить в сцену примитив **Cylinder**. Этот объект появится в месте, на которое указывает вспомогательный элемент **3D Cursor** (рис. 1.12). Для его установки щелкните левой кнопкой мыши в нужной части сцены.

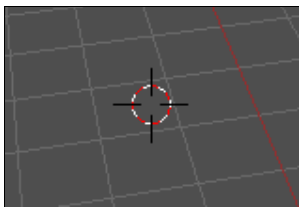


Рис. 1.12. Элемент 3D Cursor

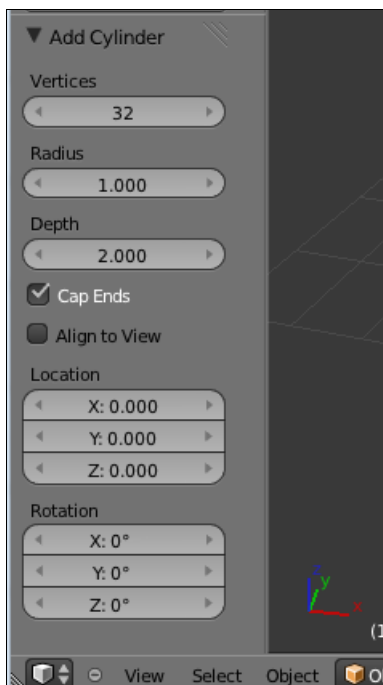


Рис. 1.13. Эта панель содержит настройки последней выполненной команды

После того как вы определились с местом, настало время добавить свой первый примитив. Это можно сделать как из общего меню **Add | Mesh | Cylinder** (Добавить | Сетка | Цилиндр) верхнего окна программы, так и с помощью "умного меню".

Теперь обратите внимание на левую часть окна **3D View**, а именно на вспомогательную панель с названием последней команды **Add Cylinder** (рис. 1.13).

Панель содержит настройки, характерные для созданного примитива **Cylinder**. Особенностью ее является то, что она показывает опции, связанные с последней

вызванной командой. Понятно, что при выборе другой функции содержимое панели утратится. Причем некоторые настройки объекта можно выполнить только на этапе его создания. Рассмотрим их подробнее.

- ◆ **Vertices** (Вершины) — значение этого параметра влияет на внешний вид примитива. Так, если вписать цифру 6, то вы получите заготовку шестигранника. Увеличение этого параметра приведет к более гладкой фигуре, но не переусердствуйте! Blender имеет немало инструментов для сглаживания объектов, без увеличения количества их элементов.
- ◆ **Radius** (Радиус) — радиус объекта.
- ◆ **Depth** (Глубина) — размер объекта по координате Z.
- ◆ **Cap Ends** (Закрывать концы) — важный параметр, отвечающий за заливку примитива. Если снять галочку, то получите сквозную трубу.
- ◆ **Align to View** (Выравнивание) — очень полезная функция, которая позволяет ориентировать объект по координате Z в оконной системе координат. Проще говоря, как бы вы не повернули сцену, установка этой галочки заставит объект "смотреть" вершиной на вас.
- ◆ **Location** (Размещение) — размещение объекта.
- ◆ **Rotation** (Вращение) — поворот объекта.

Не будем пока мудрить с этими параметрами и оставим как есть. Итак, первый примитив, основа рукояти, создан!

Теперь придадим ему более подходящую форму. Нужно вытянуть объект по вертикали (координате Z) и немного сузить в диаметре. Конечно, можно воспользоваться манипуляторами, описанными ранее, и вручную придать форму примитиву, но давайте познакомимся с еще одной панелью. С ее помощью можно точно настраивать размещение, масштаб, ротацию объекта и многое другое. Она располагается в правой части окна **3D View**. Если ее нет, то нажмите клавишу <N> (рис. 1.14).

Подправим немного масштаб модели в пункте **Scale** (Масштаб). Для изменения параметра можете воспользоваться стрелками или, щелкнув левой кнопкой мыши в нужном поле, вписать иное значение:

- ◆ X = 0.295;
- ◆ Y = 0.295;
- ◆ Z = 4.700.

С рукоятью разобрались, настало время добавить боек. В качестве заготовки послужит обычный куб. Добавьте его в сцену командой **Add | Mesh | Cube** (Добавить | Сетка | Куб).

Для подгонки его расположения на рукояти будет удобнее работать с ракурсом **Front View** (Вид спереди) в ортогональной проекции. Нажмите клавиши <NumPad 1> и <NumPad 5>.

Придайте бойку вытянутую форму и уменьшите общий масштаб в соответствии с рукоятью:

1. Нажмите клавишу изменения масштаба <S>. Затем <X> для работы с координатой X (объект будет вытягиваться по горизонтали). Щелкните левой кнопкой мыши для закрепления нового масштаба.
2. С помощью общего масштабирования немного уменьшите объект.

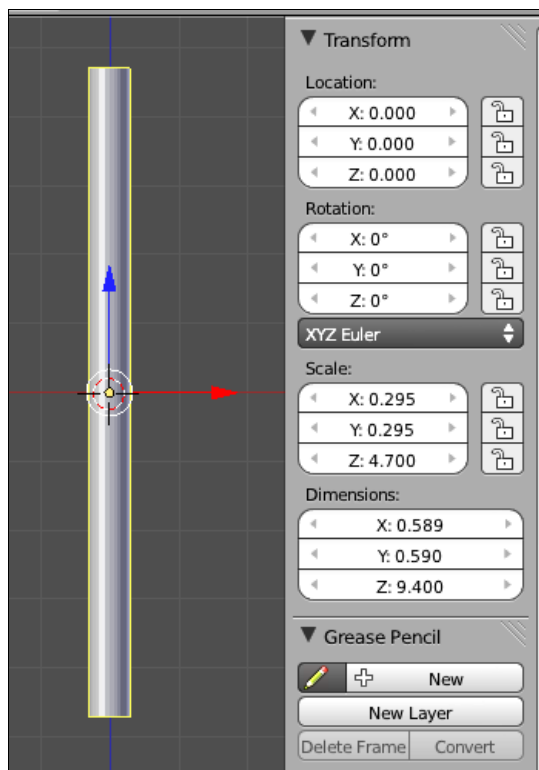


Рис. 1.14. Панель основных настроек выделенного объекта

Теперь установите боек на его место в верхней части рукояти (рис. 1.15).

Добавим небольшое утолщение в нижней части кувалды. Это можно сделать путем добавления еще одного примитива **Cylinder** и подгонки его к рукояти. Можно, но не нужно. Пойдем более сложным путем и заодно познакомимся с еще несколькими полезными инструментами.

Сделаем копию нашей рукояти, которая послужит заготовкой для утолщения. Процедура копирования объекта в Blender называется *дублированием* (Duplicate). Выделите цилиндр, нажмите клавиши <Shift>+<D>, переместите мышь для отделения дубликата и зафиксируйте результат нажатием левой кнопки мыши. Полученный объект является полной копией оригинала. Сожмите по Z дубликат и немного увеличьте общий масштаб (рис. 1.16).

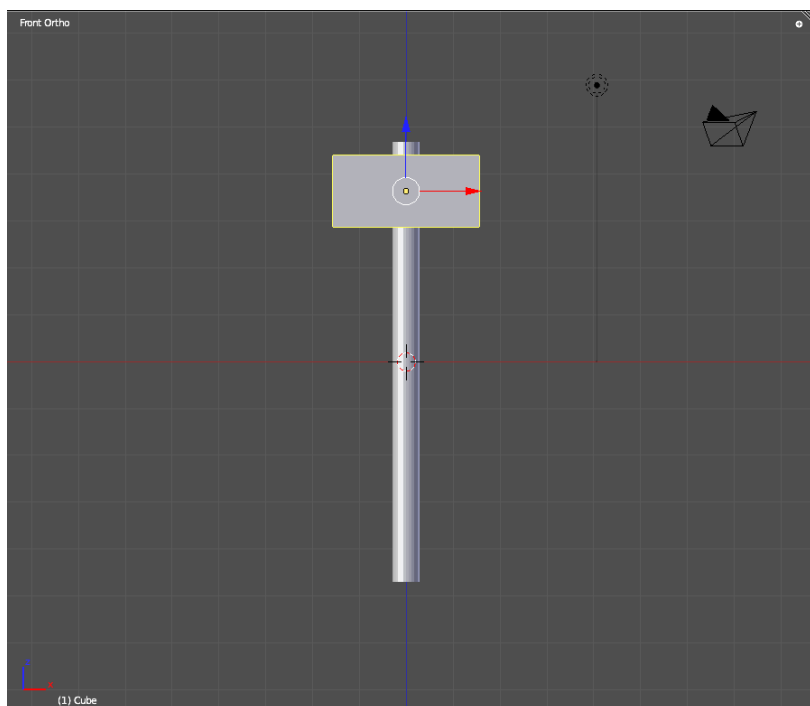


Рис. 1.15. Боек с рукоятью



Рис. 1.16. Утолщение для рукояти — копия примитива **Cylinder**

СОВЕТ

Blender предлагает несколько вариантов вызова своих функций в окне **3D View**: с использованием горячих клавиш, выбор из меню и работа со вспомогательными панелями. С двумя вы уже знакомы — это панель свойств выделенного объекта (в правой части окна) и панель настроек вызванной команды (в левой нижней части экрана). Программа предлагает еще одну "умную" панель с функциями, характерными для выделенного объекта, — *панель инструментов* (располагается в левой верхней части). Ее название меняется в зависимости от объекта и режима работы. В данный момент она называется **Object Tools**. Все рассмотренные инструменты для работы с объектом вы можете найти и там.

Возникает вопрос — как теперь разместить новый объект на рукояти? Blender предлагает ряд несложных функций для точного позиционирования объектов по отношению друг к другу. Все они размещаются в группе **Snap** (Привязка) меню **Object** (Объект) окна **3D View** (рис. 1.17).

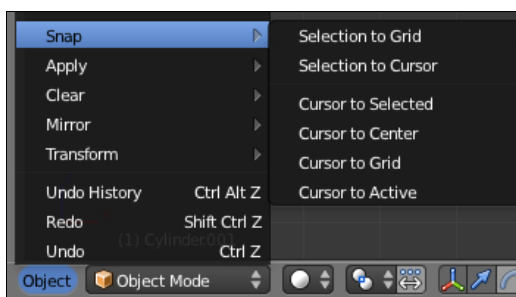


Рис. 1.17. Меню привязки к объектам

Пункты меню **Snap** (Привязка):

- ◆ **Selection to Grid** (Выделение к решетке) — смысл этой функции в том, что выделенный объект будет перемещен в ближайший узел вспомогательной решетки окна **3D View**;
- ◆ **Selection to Cursor** (Выделение к курсору) — объект перемещается к местонахождению **3D Cursor**;
- ◆ **Cursor to Selected** (Курсор к выделенному) — обратная операция — курсор перемещается в центр выделенного объекта. Причем если выделено больше одного объекта, то курсор размещается в центре всей группы;
- ◆ **Cursor to Center** (Курсор к центру) — координаты **3D Cursor** сбрасываются по умолчанию, и он возвращается в центр вспомогательной решетки окна;
- ◆ **Cursor to Grid** (Курсор к решетке) — перемещение курсора к ближайшему узлу вспомогательной решетки;
- ◆ **Cursor to Active** (Курсор к активному) — перемещение курсора в центр активного объекта.

В нашем случае можно поступить так:

1. Переместить **3D Cursor** в центр рукояти. Выделите рукоять и выберите пункт меню **Object | Snap | Cursor to Active**.

2. Переместить новый объект к **3D Cursor**. Выделите дубликат и выберите пункт меню **Object | Snap | Selection to Cursor**.

Теперь осталось немного подправить масштаб утолщения и переместить его по координате Z к низу рукояти (рис. 1.18).

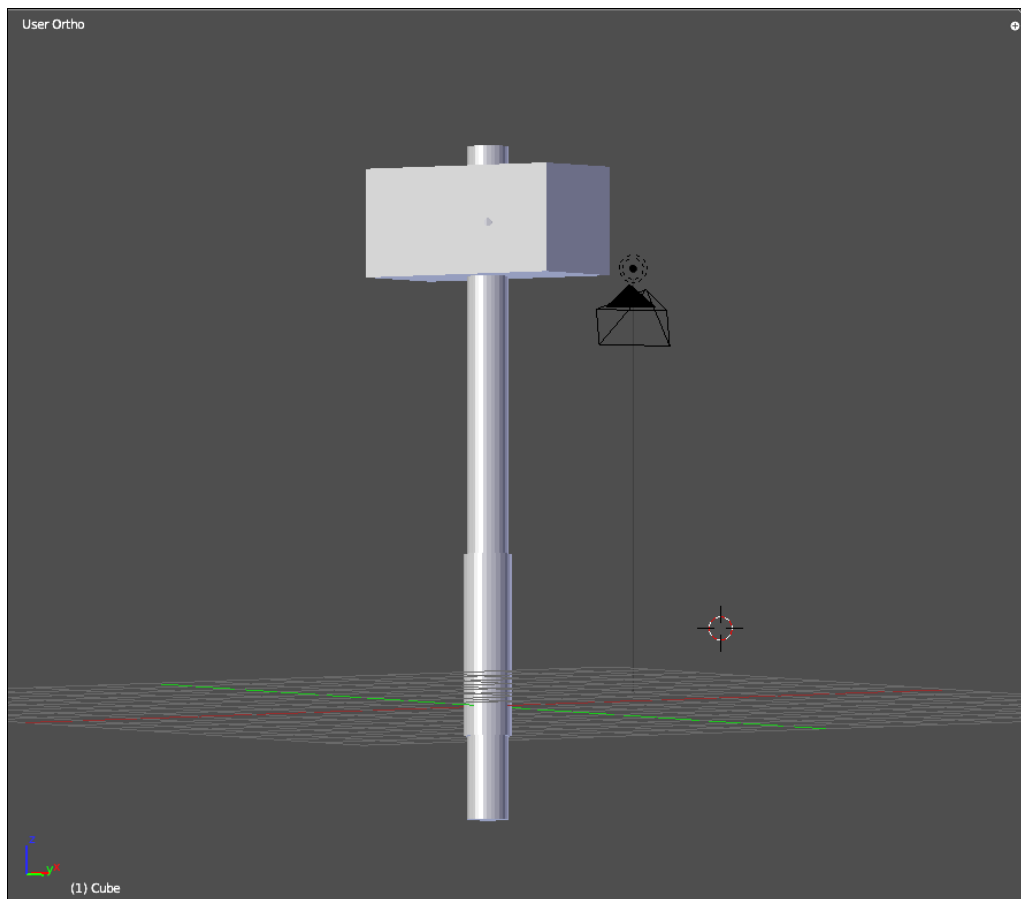


Рис. 1.18. Готовая модель кувалды

Сейчас модель состоит из трех отдельных объектов. В некоторых случаях удобнее было бы объединить их в одно целое. Сделать это можно командой **Join** (Объединение). Выделите все примитивы, удерживая клавишу <Shift> и щелкая правой кнопкой мыши по объектам. Нажмите комбинацию клавиш <Ctrl>+<J> для вызова функции объединения. Теперь кувалда представляет собой цельную модель.

Вроде бы все, но есть один небольшой недостаток. Посмотрите на рис. 1.19, где запечатлен момент поворота модели вокруг оси Y. Как видите, центр вращения находится где-то в середине бойка, а удобнее и правильнее было бы поместить его внизу рукояти.

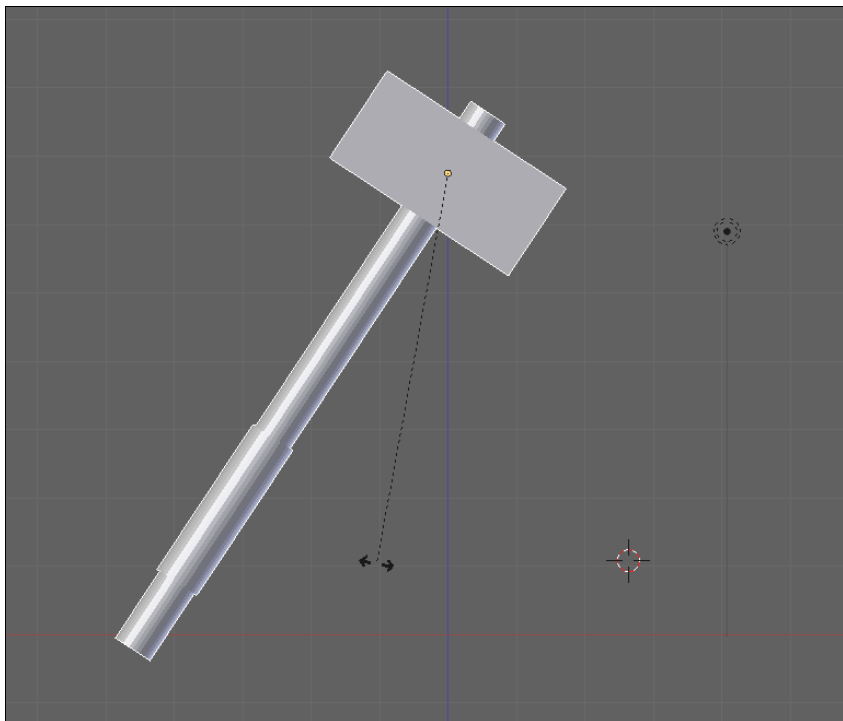


Рис. 1.19. В этой модели неправильно расположен ее центр

СОВЕТ

Если вам нужно больше места для работы со сценой в окне **3D View**, то можно временно спрятать вспомогательные панели.

Дело в том, что при объединении объектов программа берет в качестве центра последний выделенный примитив. Для правильной установки центра нужно было бы установить следующий порядок выделения: боек, рукоять, утолщение. Впрочем, все можно исправить.

Нужные функции находятся в меню **Object | Transform** окна **3D View**:

- ◆ **Geometry to Origin** (Геометрия в оригинал) — высчитывается среднее значение, и объект смещается в новую точку центра;
- ◆ **Origin to Geometry** (Оригинал в геометрию) — высчитывается среднее значение, и центр смещается при неподвижном объекте;
- ◆ **Origin to 3D Cursor** (Оригинал в 3D Cursor) — центр устанавливается в позицию 3D Cursor.

Таким образом, достаточно установить **3D Cursor** в нужном месте, выбрать пункт меню **Object | Transform | Origin to 3D Cursor**, и получим нужный результат (рис. 1.20).

Вот и все, ваша первая модель готова! Конечно, она немного неказиста и требует доработки, но это уже тема следующих глав.

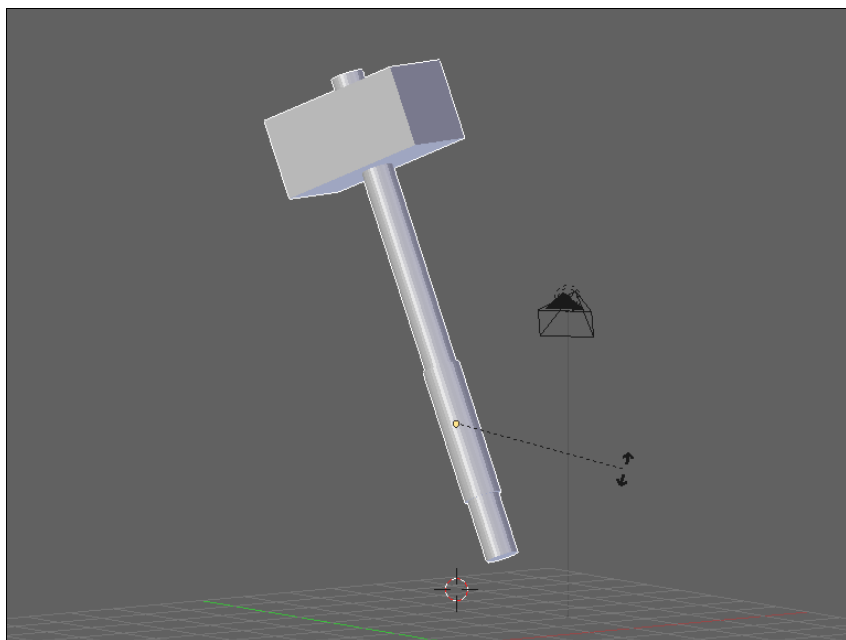


Рис. 1.20. Модель кувалды с правильно установленным центром

1.7. Иерархия сцены: группы, связи, слои

Сложные сцены могут содержать сотни, а то и тысячи объектов. Разобраться в этом сонмище бывает подчас сложно. Поэтому важно поддерживать порядок в своем детище. Blender имеет достаточный набор средств для организации и контролирования иерархии сцены. С одним из таких глобальных инструментов вы уже познакомились — это **Scene Selectors**. Однако есть ряд и локальных функций.

Самый простой вид иерархии — это *связи*. Под этим понимается связка, где один объект является главным, а второй — дочерним. Чтобы было понятно, для чего можно использовать связи, выполним небольшой урок.

Создайте новый проект и удалите имеющийся куб в сцене. Теперь добавьте примитив **UV Sphere**. Для ясности поменяйте стандартное имя объекта "Sphere" на нечто более понятное — "Sun" (Солнце). Это можно сделать на панели свойств окна **3D View**, вызываемой клавишей <N> (рис. 1.21).

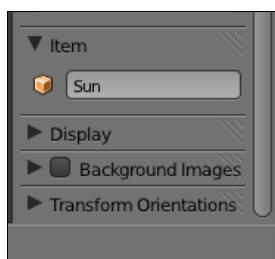


Рис. 1.21. Закладка **Item** позволяет изменить имя объекта

Сделайте дубликат сферы (<Shift>+<D>), отодвиньте его в сторону и немного уменьшите масштаб. Измените имя нового примитива на "Earth" (Земля).

Таким же образом создайте еще одну сферу с именем "Moon" (Луна). В целом сцена должна выглядеть примерно так же, как на рис. 1.22.

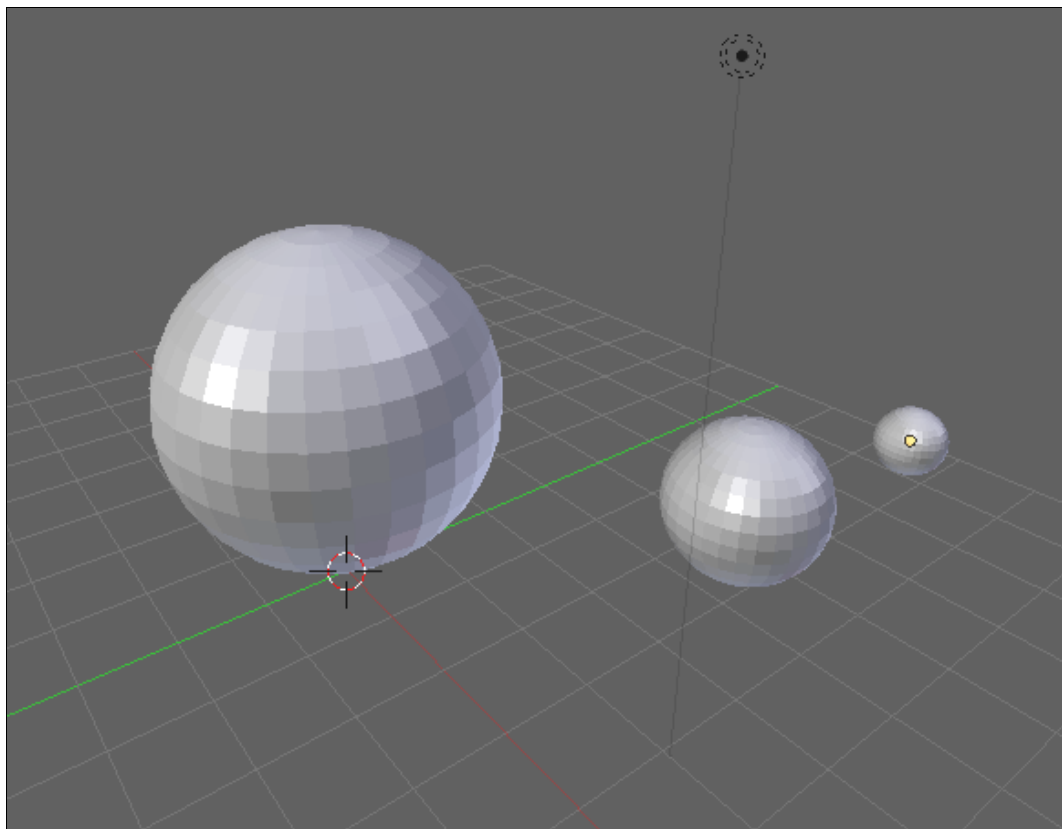


Рис. 1.22. Заготовка для Солнечной системы

Итак, Луна является спутником нашей планеты, которая в свою очередь вращается вокруг Солнца. Вся наша Солнечная система — часть галактики. Вот и попробуем смоделировать движение тел.

Выделите сначала **Moon**, а затем **Earth**, удерживая нажатой клавишу <Shift>. Таким образом, мы указали, что объект **Moon** будет являться дочерним по отношению к **Earth**. Для создания связи можно воспользоваться пунктом меню **Object | Parent | Set | Object** или просто нажать клавиши <Ctrl>+<P>. В последнем случае программа выведет контекстное меню **Set Parent To** с просьбой подтвердить. Если вы присмотритесь внимательнее к сцене, то заметите черную пунктирную линию, соединяющую связанные объекты. Так Blender отмечает связи.

Проверим, что получилось. Выделите объект **Earth** и попробуйте его переместить. Если все сделано правильно, то примитив **Moon** будет послушно следовать за все-

ми манипуляциями родителем, включая даже масштабирование. В то же время перемещение подчиненного объекта никак не сказывается на основном.

Выделяем **Earth**, затем **Sun** и создаем последнюю дочернюю связь. Что же в итоге получилось? Луна может свободно двигаться по отношению к Земле. В свою очередь при выделении Земли оба объекта с успехом перемещаются, не затрагивая Солнце. А вот движение Солнца утягивает за собой все дочерние объекты.

Для контроля над иерархией сцены Blender имеет специальное окно **Outliner**. По умолчанию оно располагается в верхней правой части программы (рис. 1.23).

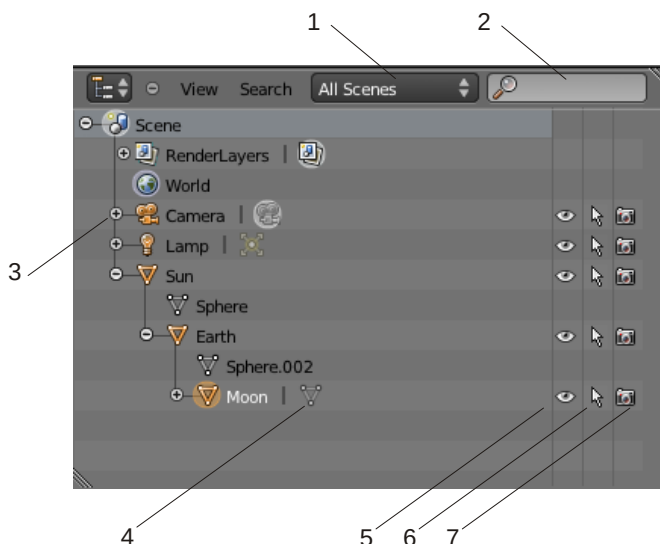


Рис. 1.23. Окно **Outliner**

Это очень удобное средство для просмотра и поиска объектов в сцене. Рассмотрим основные его особенности:

1. **Display Mode** (Режим просмотра). Здесь можно выбрать, какие объекты будут отображаться в окне Outliner, начиная от сцен и до отдельно выделенного объекта.
2. **Строка поиска**. Поиск осуществляется по мере ввода символов. При этом соответствующим образом корректируется вывод в окне.
3. Узловой элемент, позволяющий сворачивать или раскрывать иерархию подобъектов.
4. Если у объекта имеется возможность его редактирования, то нажатие этой кнопки переведет **3D View** в **Edit Mode** (Режим редактирования).
5. Отключение или включение отображения объекта в **3D View**.
6. Этот элемент влияет на возможность выделения объекта в сцене.
7. Если отключить этот элемент, то объект не будет участвовать в финальной визуализации проекта (рендере).

СОВЕТ

Окно **Outliner** позволяет быстро переименовывать объекты. Для этого достаточно нажать клавишу <Ctrl> и щелкнуть левой кнопкой мыши на нужном имени.

Как правило, при создании более-менее сложной модели используется несколько примитивов. В некоторых случаях их можно объединить командой **Join**, но это скорее внутренняя оптимизация объекта. Гораздо удобнее и правильнее пользоваться группами.

Группы — это временное объединение объектов, позволяющее манипулировать ими как одной независимой единицей. Однако в Blender несколько шире трактуется это привычное понятие.

Важно знать, что существует два типа групп: эталон и копии. Разница между ними очень существенна. Рассмотрим их на примере.

Создайте новый проект и добавьте туда еще пару примитивов из главного меню **Add | Mesh**. Для динамичности добавим к этим объектам эффект взрыва. Выделите все примитивы и выберите пункт меню **Object | Quick Effects | Quick Explode** (Объект | Быстрые эффекты | Взрыв). Теперь проверим работу физики, нажав <Alt>+<A> (это сочетание клавиш запускает анимацию). Если все было сделано правильно, то осколки объектов разлетятся по сцене.

Для создания новой группы нужно:

1. Выделить все необходимые объекты. Причем в группе могут быть не только примитивы, но и различные вспомогательные объекты, такие как лампы, камеры и т. д.
2. Выбрать пункт меню **Object | Group | Create New Group** (Объект | Группа | Создать новую группу). Можете воспользоваться горячими клавишами <Ctrl>+<G>. Окантовка у выделенных объектов должна стать зеленого цвета.

Желательно еще дать нормальное имя новоявленной группе. Сделать это можно в окне **Properties** (Свойства). По умолчанию при запуске Blender оно располагается в правой части экрана (рис. 1.24).



Рис. 1.24. Панель кнопок окна **Properties**

Это окно чрезвычайно важное, т. к. именно в нем собрана большая часть настроек объектов сцены. Рассмотрим основные закладки:

1. **Render** (Рендер) — визуализация сцены. Именно здесь можно выбрать, в каком формате будет обработана сцена, с каким качеством, что будет участвовать в обработке и многое другое. Собственно, на этой панели находятся милые сердцу любого моделлера кнопки запуска рендера.

2. **Scene** (Сцена). Здесь можно выбрать метрическую систему, настроить звук, физику.
3. **World** (Окружение). Установка фона сцены, глобального освещения, некоторых эффектов (туман, звезды).
4. **Object** (Объект). Настройка всевозможных параметров выделенного объекта, от его названия до участия в группах.
5. **Constraints** (Ограничители). Выбор и настройка различных вспомогательных инструментов.
6. **Modifiers** (Модификаторы). Выбор и настройка функций, изменяющих состояние и поведение объекта.
7. **Object Data** (Данные объекта). Настройка внутренних параметров объектов.
8. **Material** (Материал). Настройки цветов объекта, отражения, преломления, теней.
9. **Textures** (Текстуры). Создание и настройка текстур.
10. **Particles** (Частицы). Все, что касается работы с частицами.
11. **Physics** (Физика). Работа с физикой Blender.

Для изменения имени группы нужно выделить любой объект в группе, открыть панель **Object** окна **Properties** и поменять название в закладке **Groups** (Группы) (рис. 1.25).

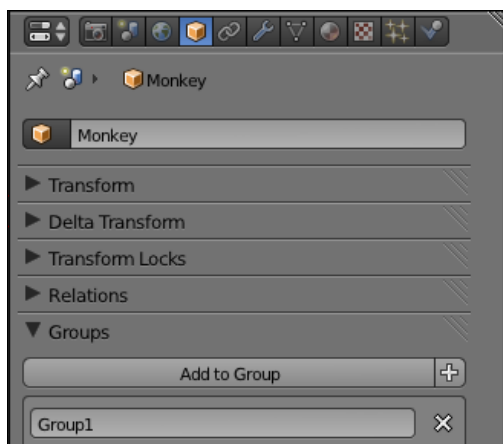


Рис. 1.25. Настройки группы

Здесь вы можете также добавить выделенный объект в новую или существующую группу или, наоборот, удалить из нее. Для добавления объекта в существующую группу служит кнопка **Add to Group** (Добавить в группу). Программа выведет меню с возможными вариантами. Чтобы создать новую группу, достаточно щелкнуть по кнопке с плюсом, а для удаления — по крестику.

Теперь немного поэкспериментируем. Нажмите клавишу <A> для снятия выделения и щелкните правой кнопкой мыши по любому примитиву в группе. Появится

окантовка зеленого цвета. Вот только объект выделился один, хотя по логике должна отметиться вся группа. Да и в окне **Outline** особых изменений не наблюдается. В чем же дело?

Откройте пункт **Add | Group Instance** (Добавить | Группы) в главном меню программы. Здесь должна появиться созданная группа. Щелкните по ней для добавления в сцену.

Вот теперь в окне **Outliner** виден новый объект с названием нашей группы. Причем по поведению он ничем не отличается от одиночных примитивов. Его можно перемещать, масштабировать, вращать. Да и при щелчке выделяется вся группа, как единый объект. Попробуйте нажать клавиши **<Alt>+<A>** для запуска анимации. Обе группы одновременно разлетятся на куски.

Итак, в первом случае мы с вами создали группу-эталон, а уже из меню **Add | Group Instance** добавили в сцену ее копию.

Попробуйте выделить любой объект в эталоне и, скажем, повернуть его. Копия послушно повторит проделанную манипуляцию. В то же время манипуляции с дубликатом никак не отразятся на оригинале. Именно с копиями и происходит основная работа.

Возникает закономерный вопрос, а куда "спрятать" эталон, чтобы он не мешался в сцене? Вот тут-то на помощь придет *слои* (**Layers**).

Сцену в Blender можно представить как многослойный торт с различной начинкой. По умолчанию работа происходит в самом первом слое. Но чем сложнее становится сцена, тем чаще приходится пользоваться слоями. В некоторых случаях без них просто не обойтись.

Layers (Слои) — это способ разграничения отображения объектов сцены. Вы можете управлять размещением объектов в слоях, указывать, какие слои являются активными для показа в окне **3D View**, а также при рендере окончательного результата.

В заголовке окна **3D View** имеется специальная область для управления слоями (рис. 1.26).



Рис. 1.26. Панель **Layers**

Всего программа предлагает 20 слоев для хранения объектов. На панели **Layers** каждый слой представлен в виде квадратика. Визуально они могут выглядеть по-разному:

- ◆ светлая заливка — слой неактивен и не отображается в сцене;
- ◆ темная заливка — слой активен и участвует в отображении;
- ◆ точка в квадрате означает наличие каких-то объектов в данном слое.

Выделите все объекты группы-эталона и нажмите клавишу **<M>**. Появится всплывающее окно **Move to Layer** (Поместить в слой). Пощелкайте левой кнопкой мыши

по квадратикам. Панель **Layers** окна **3D View** должна будет отображать перемещение объектов по слоям. Поместите объект в любой другой слой (по умолчанию).

СОВЕТ

Для быстрого выделения всех объектов группы-эталона щелкните по любому из них мышью и нажмите <Shift>+<G>. В появившемся меню **Select Grouped** выберите пункт **Group**.

Итак, сейчас в окне **3D View** видна только группа-дубликат, что и было необходимо сделать.

Вы можете выбрать любой слой для отображения, щелкнув левой кнопкой мыши по нужному квадратику панели **Layers**. Для множественного выбора используйте клавишу <Shift>.

1.8. Работа с файлами

Как и любая программа, Blender имеет средства для работы с файлами. Основные глобальные функции сосредоточены в главном меню **File** (Файл): сохранение и загрузка, импорт и экспорт, присоединение. Кроме этого меню, некоторые окна также используют файловые операции, например, при работе с текстурами или рендером. Все они вызывают специальное окно **File Browser** (Просмотр файлов) со множеством функций и возможностей (рис. 1.27).

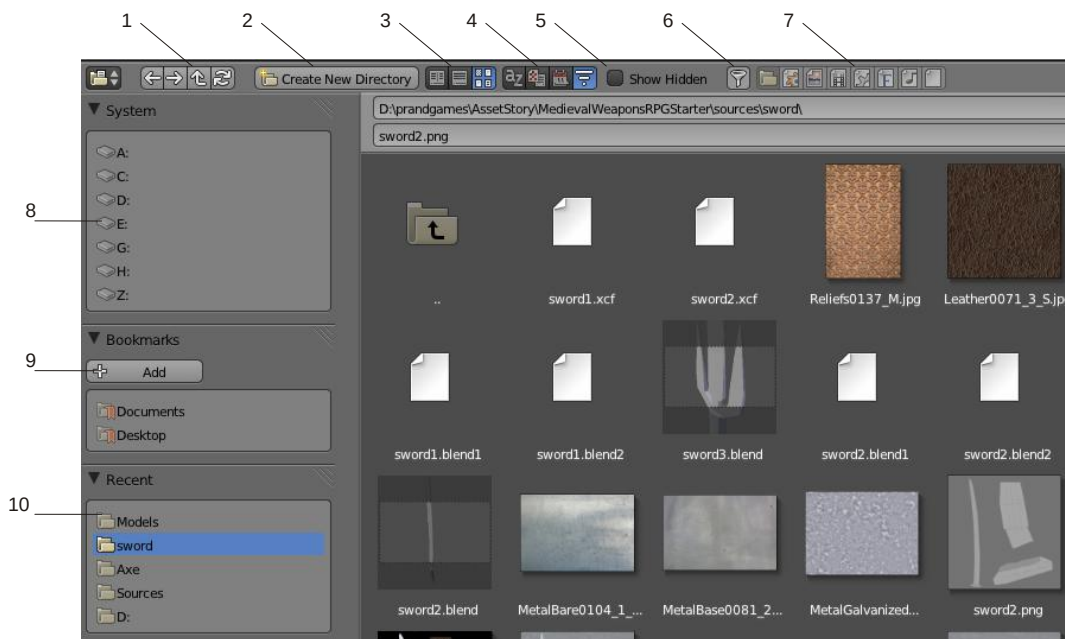


Рис. 1.27. Окно File Browser

Рассмотрим элементы окна **File Browser**:

1. Навигация по просмотренным папкам.
2. Создание новой папки.
3. Режим отображения файлов и папок: краткий, полный и эскизы.
4. Сортировка по алфавиту, расширению, дате и размеру.
5. Активация показа скрытых файлов.
6. Кнопка включения фильтров.
7. Фильтры показа: папки, проекты Blender (.blend), графические, видео, скрипты, шрифт, звуки, тексты.
8. Выбор корневого устройства в системе.
9. **Bookmarks** (Закладки). Для быстрого перехода в нужную директорию вы можете установить закладку. Откройте папку и нажмите кнопку **Add** (Добавить).
10. **Recent** (Последние). Отображение последних открытых папок.

В некоторых случаях добавляются дополнительные интерфейсные элементы. Об этом будет сказано отдельно.

В работе **File Browser** является очень удобным инструментом. Быстрое создание эскизов графических и видеофайлов, просмотр содержимого проектов Blender, использование закладок — все это облегчает работу с файлами. Но есть несколько моментов, о которых нужно знать.

Во-первых, не используйте кириллицу в названиях папок и файлов. В некоторых операционных системах, например Windows, русские буквы выводятся кракозябрами. Программа сможет открыть подобные папки или файлы, но сама навигация весьма затруднена.

Во-вторых, просмотр файловой иерархии происходит по методу "одного щелчка". Для пользователей Windows это немного непривычно, но на практике получается гораздо быстрее.

Теперь рассмотрим меню **File** (Файл) основного окна программы:

- ◆ **New** (Новый) — создание нового проекта;
- ◆ **Open** (Открыть) — загрузка файла проекта. При этом предыдущий проект не сохраняется;
- ◆ **Open Recent** (Открыть последний) — программа предложит список проектов, которые открывались ранее;
- ◆ **Recover Last Session** (Восстановить последнюю сессию) — при выходе Blender автоматически производит запись сцены в файл quit.blend. Эта команда подгружает данный файл;
- ◆ **Recover Auto Save** (Восстановить автосохранение) — по умолчанию программа сохраняет сцену в отдельном файле каждые пять минут;
- ◆ **Save As** (Сохранить как) — сохранение проекта сцены в файл с расширением blend. Если вы просмотрите папку с сохраненным проектом, то увидите еще два

файла с расширениями `blend1` и `blend2`. В них хранятся автоматические сохранения предыдущего состояния до записи сцены;

- ◆ **Save Copy** (Сохранить копию) — сохранение сцены в новый файл, без открытия его в Blender;
- ◆ **User Preferences** (Пользовательские настройки) — вызывается окно с настройками программы;
- ◆ **Save User Settings** (Сохранение пользовательских настроек) — к примеру, если вы хотите сохранить расположение окон и экранов для будущих проектов, то воспользуйтесь этой функцией;
- ◆ **Load Factory Settings** (Загрузка параметров по умолчанию) — сброс параметров Blender в соответствии с настройками разработчика;
- ◆ **Link** (Связь) — связь с объектами чужого проекта;
- ◆ **Append** (Добавить) — импортирование объектов из других проектов Blender;
- ◆ **Import** (Импорт) — импортирование объектов сторонних приложений;
- ◆ **Export** (Экспорт) — экспорт сцены или объектов в другие форматы;
- ◆ **External Data** (Данные проекта) — управление ресурсами сцены, допустим загруженными текстурами:
 - **Pack Into .blend file** (Упаковать в .blend файл) — все сторонние файлы сохраняются в проекте сцены. Используйте эту функцию для переноса сцены на другой компьютер или систему;
 - **Unpack Into Files** (Распаковать файлы) — обратная операция предыдущей функции. Упакованные файлы распаковываются в корневую директорию проекта;
 - **Make All Path Relative** (Сделать все пути относительными) — скопируйте дополнительные ресурсные файлы в папку с проектом, выберите эту функцию и сохраните проект. Теперь программа при открытии проекта всегда будет находить нужные файлы;
 - **Make All Paths Absolute** (Сделать все пути абсолютными) — т. е. с жесткой привязкой к файловой системе компьютера;
 - **Report Missing Files** (Показать пропущенные файлы) — вы можете проверить целостность проекта на наличие подключенных ресурсов. Полезно при запуске проекта на другой машине;
 - **Find Missing Files** (Поиск пропущенных файлов) — сопоставление пропущенных файлов вручную через **File Browser**;
- ◆ **Quit** (Выход) — выход из программы без предупреждения и сохранения.

Если с записью и загрузкой проектов в принципе все понятно, то функции **Append** и **Link** могут привести к замешательству.

- ◆ **Append** (Добавление) — это добавление объектов из другого проекта Blender. В этом случае сторонние объекты становятся "родными" для текущего проекта. Вы можете изменять их и редактировать на свое усмотрение.

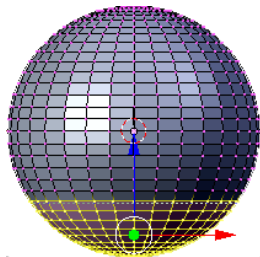
- ◆ **Link** (Связать) — это связь текущего проекта с объектами других проектов. Здесь происходит только использование объектов без возможности их редактирования. В то же время любые изменения с объектами в оригинальных сценах приведут к изменению присоединенного объекта.

Чтобы использовать эти функции, нужно разбираться в иерархии файла сцены. Дело в том, что **Append** или **Link** работают не со сценой целиком, а с отдельными объектами. Попробуйте выбрать пункт **File | Append** (или **File | Link**) и щелкнуть по любому проекту в браузере программы. Откроется целый ряд папок в составе сцены: **Brush** (Кисть), **Camera** (Камера), **Image** (Картинки), **Lamp** (Лампа), **Material** (Материал), **Mesh** (Структура Меш), **Object** (Объект), **Scene** (Сцена), **Texture** (Текстура), **World** (Окружение). В каждой директории находятся файлы, хранящие соответствующие данные. Так для загрузки конкретного объекта его нужно искать в папке **Object**. Заметьте, что при этом сохраняется иерархия данных. К примеру, если модель имеет свой материал и настроенные текстуры, то при выборе ее из **Object** произойдет полное внедрение связанных данных.

СОВЕТ

Часто при сохранении проекта создаются его копии для возможности отката на предыдущие состояния. Этот процесс можно упростить, если в окне **File Browser** использовать клавиши **<+>** и **<->** на дополнительной цифровой клавиатуре. В этом случае к имени файла будет прибавляться или уменьшаться порядковое число. Например: **test.blend**, **test1.blend**, ..., **testN.blend**.

ГЛАВА 2



Простое моделирование с *Mesh*

В первой главе мы рассмотрели интерфейс программы, познакомились с основными объектами Blender и даже сделали простую модель кувалды с помощью примитивов **Mesh**. Хотя внешне она получилась вполне узнаваемой, но качественная ее доработка невозможна без изменения внутренней структуры.

Объекты **Mesh** — это наиболее часто выбираемые примитивы Blender в качестве основы для модели. Наверное, такая популярность объясняется легким редактированием структуры, обилием инструментария и методов моделирования. Чего только Blender не предлагает в помощь пользователю! От инструментов простого низкополигонального редактирования до высокоточной скульптурной лепки.

В этой главе вы узнаете об особенностях строения этих примитивов, основных инструментах редактирования, а также о некоторых интересных методах моделирования.

2.1. Примитивы и их структура

Когда запускается Blender, по умолчанию в сцене уже присутствует один из примитивов **Mesh** — куб. Это сделано не зря, т. к. именно куб чаще всего используют для создания модели. Трудно представить, что из него можно сделать реалистичную фигуру человека, предмета обихода или техники. Причем без добавления иных объектов, просто дорабатывая и редактируя исходный примитив. Путем несложных изменений кубу можно придать совершенно иную форму, например сферы. Но иногда все же выгоднее воспользоваться другими примитивами **Mesh**, благо Blender предлагает на выбор добрый десяток заготовок (рис. 2.1). Все они расположены в главном меню **Add | Mesh** (Добавить | Mesh).

- ◆ **Plane** (Плоскость) — простейший двухмерный примитив, имеющий четыре вершины. Путем применения специальной функции ему можно придать объем. Пример использования: модели стекла, зеркала, стены.
- ◆ **Cube** (Куб) — полноценный трехмерный объект, который создается программой по умолчанию при первом запуске. Широко используется для моделирования самых разных объектов, от коробок до фигур человека.

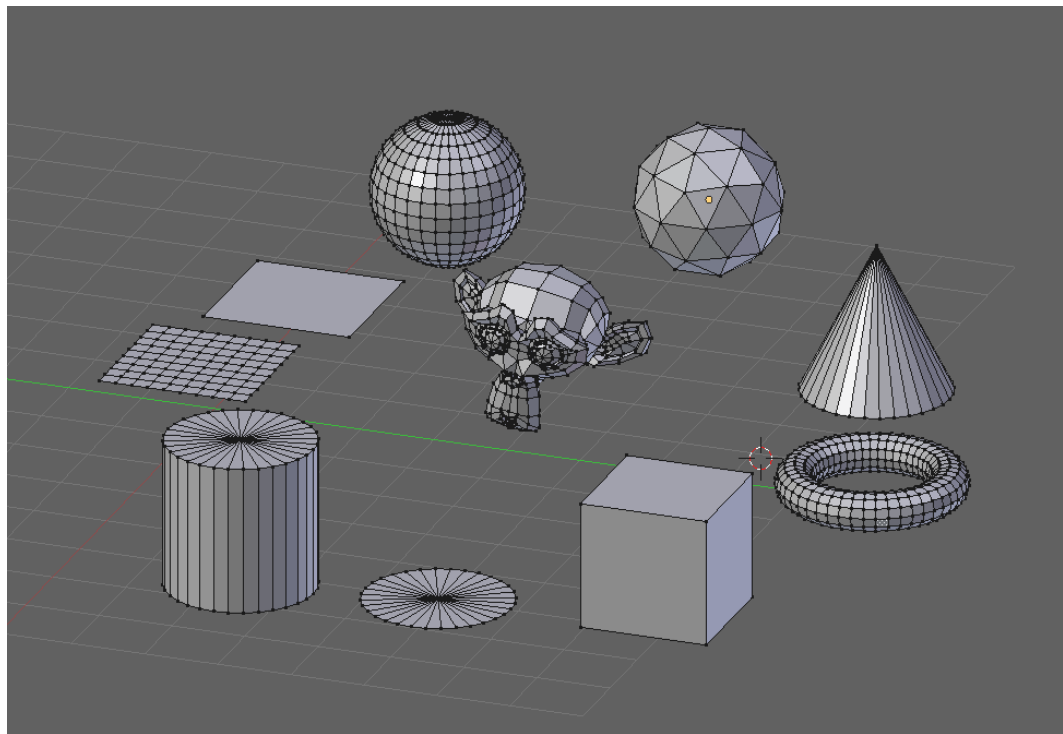


Рис. 2.1. Прimitives Mesh

- ◆ **Circle** (Окружность) — двухмерный объект. При создании можно указать количество вершин (**Vertices**), размер радиуса (**Radius**), использовать ли заливку (**Fill**).
- ◆ **UV Sphere** (Сфера UV) — сфера, состоящая из сегментов (**Segments**) и колец (**Rings**) с равномерно изменяющимся размером к полюсам. Популярный примитив для создания сферичных объектов, например человеческой головы.
- ◆ **Icosphere** (Сфера) — в отличие от **UV Sphere** этот примитив состоит из треугольных плоскостей одинакового размера.
- ◆ **Cylinder** (Цилиндр) — имеет свойство **Cap Ends** (Закрыть концы), позволяющее создавать полые либо сплошные объекты.
- ◆ **Cone** (Конус) — имеет круглую основу, при его генерации на панели **Tool Shelf** (Полка инструментов) можно изменять количество вершин. Большее значение — более гладкий объект.
- ◆ **Grid** (Решетка) — двухмерный примитив, наподобие **Plane**, с возможностью указания дробления структуры на ячейки.
- ◆ **Monkey** (Манки) — во всех популярных 3D-редакторах имеется объект, не несущий практического значения, но являющийся своего рода традиционным (к примеру, в 3Ds Max — это чайник). В Blender таким примитивом стала модель головы обезьянки, названная Манки.

- ◆ **Torus** (Тор) — этот примитив по форме больше напоминает пончик. Его особенность в наличии двух переменных **Radius** (Радиус), которые позволяют управлять размерами внутренней и наружной стенки.

"Mesh" в переводе с английского языка обозначает "петля", "сетка", "ячейка". Такое название эти примитивы получили из-за своего строения. Структура Mesh-объекта состоит из следующих элементов:

- ◆ вершины (Vertices);
- ◆ ребра (Edges);
- ◆ грани (Faces).

Таким образом, вершины — это своего рода узлы, которые соединяются между собой ребрами, а вот заливка между ними является гранью.

Для работы со структурой примитива существует специальный режим окна **3D View**, переключиться в который можно при помощи меню **Mode**. Найти его можно в заголовке окна **3D View** (рис. 2.2).

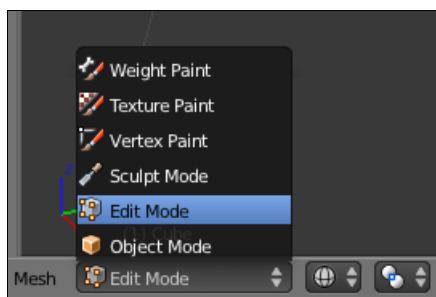


Рис. 2.2. Меню выбора режима работы 3D View

В зависимости от выделенного объекта пункты меню **Mode** могут меняться. В случае с примитивами **Mesh** программа предложит следующий список:

- ◆ **Object Mode** (Режим объекта) — это основной режим работы окна, который позволяет манипулировать объектами в сцене. Мы уже рассматривали его в первой главе;
- ◆ **Edit Mode** (Режим редактирования) — в этом случае работа происходит исключительно с выделенным примитивом. Программа предоставляет доступ к его структуре, и пользователь может редактировать любые элементы;
- ◆ **Sculpt Mode** (Режим скульптуры) — особый вид моделирования, когда редактирование происходит при помощи специальных инструментов — кистей, которые влияют на целые области;
- ◆ **Vertex Paint** (Раскраска вершин) — режим раскраски модели в 3D-окне, без использования сторонних текстур;
- ◆ **Texture Paint** (Раскраска текстуры) — обычно в качестве текстур используются готовые файлы, сделанные в сторонних приложениях (например, Gimp или Photoshop). В этом режиме Blender позволяет рисовать на текстурах непосредственно в окне **3D View**;

- ♦ **Weight Paint** (Установка веса) — специфичный режим для определенных функций. К примеру, для настройки скелетной анимации или физики.

СОВЕТ

Для быстрого перехода в режим редактирования выделите объект и нажмите клавишу <Tab>. Повторное нажатие этой клавиши переведет окно **3D View** обратно в режим **Object Mode**.

2.2. Основные инструменты редактирования

Режим редактирования используется для работы над формой примитива путем изменения его структуры. Вы можете перемещать любые элементы, добавлять новые или удалять ненужные, масштабировать, вращать, копировать и многое другое. Но для этого программе необходимо указать, с чем вы собственно желаете работать.

Blender предлагает большое количество инструментов для выделения структуры. Основные расположены в заголовке окна **3D View** (рис. 2.3).



Рис. 2.3. Режим выделения

Эти три кнопочки позволяют переключаться для выделения (слева направо): вершин, ребер, граней. Теперь немного попрактикуемся.

Выделите куб и нажмите клавишу <Tab> для перехода в режим редактирования. Первое, что бросается в глаза — это то, что он окрашивается в желтый цвет. Таким образом, программа показывает выделенные части. Нажмите клавишу <A> для полного сброса выделения.

Наиболее удобно работать со структурой, когда наблюдаешь все части объекта. Как вы, наверное, помните, окно **3D View** имеет специальный режим прорисовки **Wireframe**. Переключитесь в него, выбрав соответствующий пункт в меню **Draw Mode**, или просто нажмите клавишу <Z>.

Попробуйте выделить одну из вершин объекта правой кнопкой мыши. Программа окрасит ее в белый цвет. Теперь вы можете манипулировать ею, как заблагорассудится (рис. 2.4). Все способы манипуляции, рассмотренные в первой главе, идентичны и для режима редактирования.

Чтобы выделить несколько элементов, нажмите и удерживайте клавишу <Shift>. В этом случае выделенные элементы окрашиваются в розовый цвет, а последний — в белый.

СОВЕТ

Для быстрого выделения нескольких элементов можно воспользоваться рамкой. Нажмите клавишу , переместите курсор в начало области выделения и, удерживая нажатой левую кнопку мыши, охватите рамкой нужные элементы.

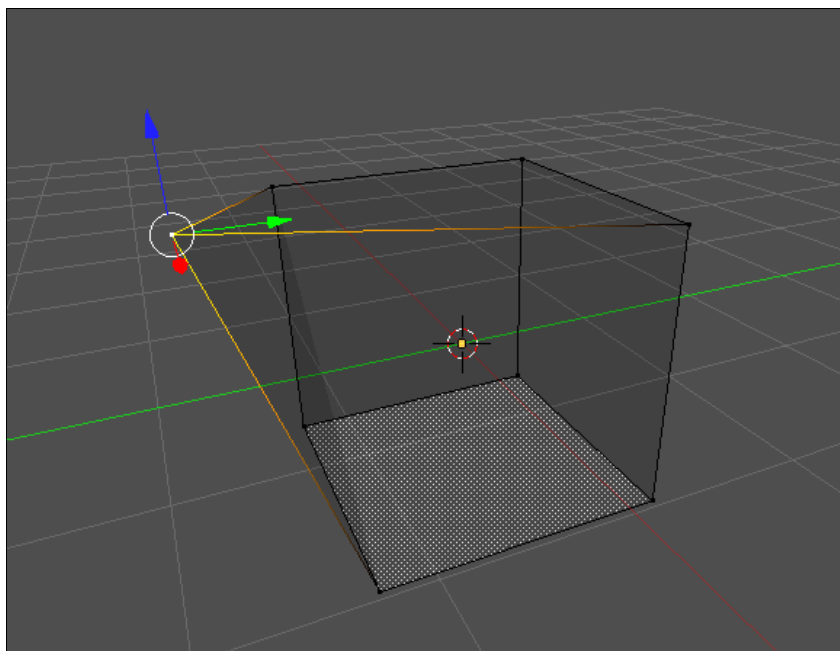


Рис. 2.4. Здесь выделена вершина куба и немного сдвинута в сторону

Равноценно работа происходит и с ребрами, только выберите соответствующий режим на панели окна **3D View** (см. рис. 2.3). А вот для выделения граней служит специальная точка, которая располагается строго по центру плоскости (рис. 2.5).

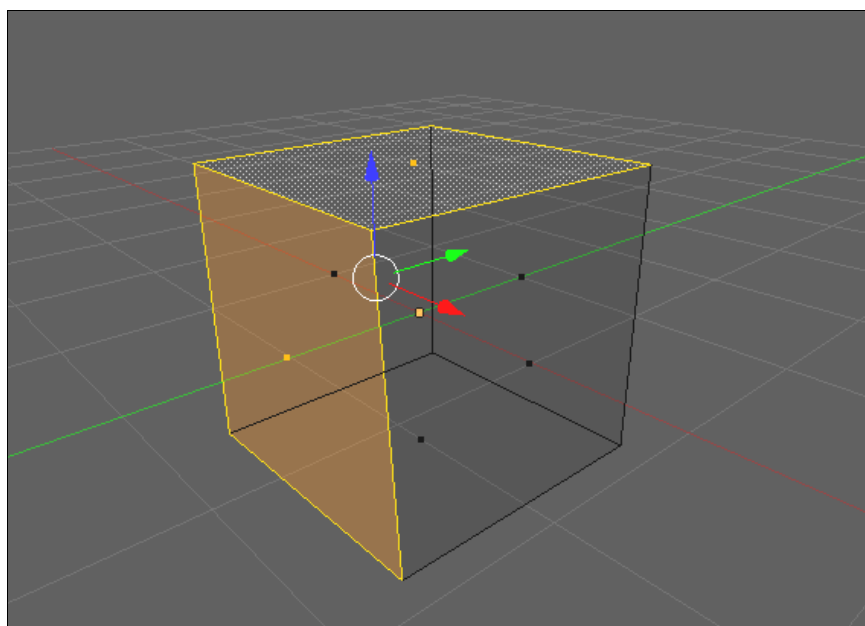


Рис. 2.5. Выделение граней

Лучший способ научиться — это сделать самому. Вот и попробуем смоделировать несложный объект, а именно гриб.

Вроде бы пустяк, взять сферу для шляпки (точнее ее половинку), в качестве ножки использовать **Tube**, а командой **Join** все это объединить. Так можно сделать и даже получится что-то путевое, но мы попытаемся смоделировать гриб из всего одного примитива.

В качестве основы воспользуемся сферой. Создайте новый проект (<Ctrl>+<N>) и удалите куб из сцены. Для этого выделите объект, а затем нажмите клавишу <X>. Программа выведет запрос на подтверждение. Нажмите клавишу <Enter>.

Теперь добавьте примитив **UV Sphere** из меню **Add | Mesh | UV Sphere**. Редактировать объект будет удобнее при просмотре его спереди (**Front View**). Нажмите <NumPad 1> для переключения ракурса и <NumPad 5>, чтобы перейти в режим ортогональной проекции. Последнее действие позволит просматривать объект без искривлений перспективы.

Сначала займемся созданием шляпки. Перейдите в режим редактирования (клавиша <Tab>) и сбросьте выделение клавишей <A>. Ваша задача — сжать нижнюю часть сферы до половины примитива, а поможет в этом режим *пропорционального редактирования*.

Выделите центральную нижнюю точку сферы. Для этого немного разверните сцену так, чтобы точка стала доступной (рис. 2.6). Нажмите <NumPad 1> для возврата просмотра **Front View**.

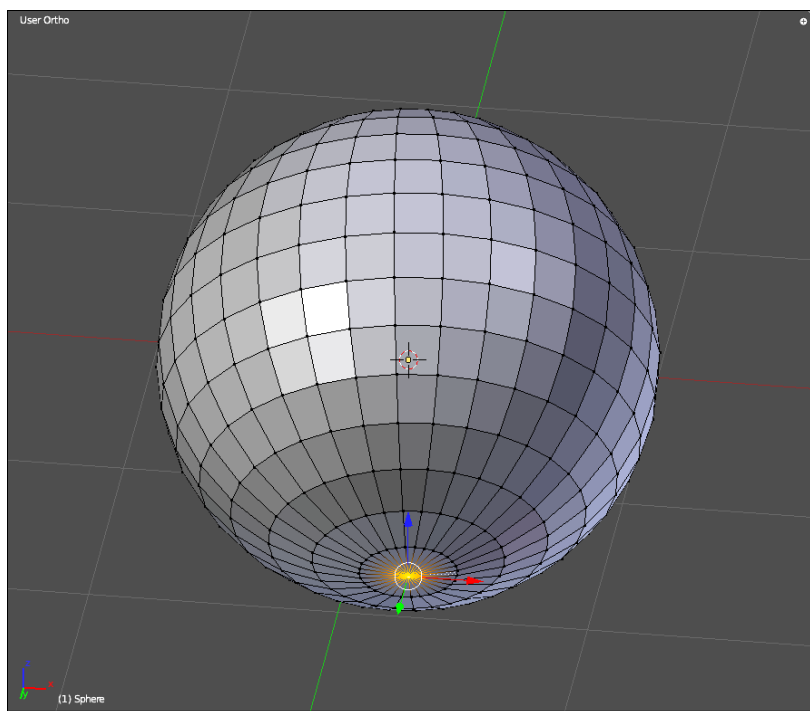


Рис. 2.6. Выделение нижнего полюса объекта

Режим *пропорционального редактирования* — это возможность плавного изменения структуры объекта в пределах отмеченной области.

Нажмите клавишу <O> для включения режима, а затем <G> для перемещения точки. Обратите внимание, что вокруг выделения появился круг, очерчивающий область влияния инструмента. Вы можете управлять его масштабом при помощи колесика мыши. Переместите точку по вертикали вверх и немного поиграйте с масштабом пропорции для создания грибообразной шляпки. После завершения отключите режим пропорционального редактирования клавишей <O>.

СОВЕТ

Во время перемещения объекта вы можете нажать среднюю клавишу мыши для выравнивания по вертикали или горизонтали. Последнее зависит от начального направления движения.

Чтобы создать ножку, вам понадобится выделить грани нижней части сферы (рис. 2.7).

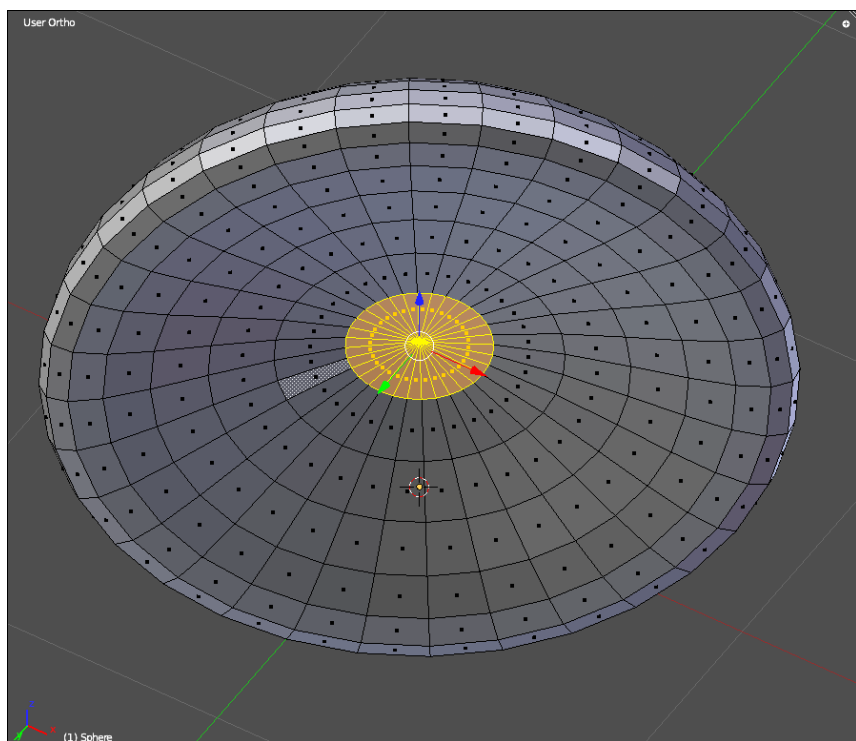


Рис. 2.7. Основа для создания ножки

Если попробовать просто переместить выделение по вертикали вниз, то получится несуразная ножка с утолщением к шляпке. Поэтому мы воспользуемся методом выдавливания.

Extrude (Выдавливание) — это копирование выделенных элементов, без отрыва от основной структуры.

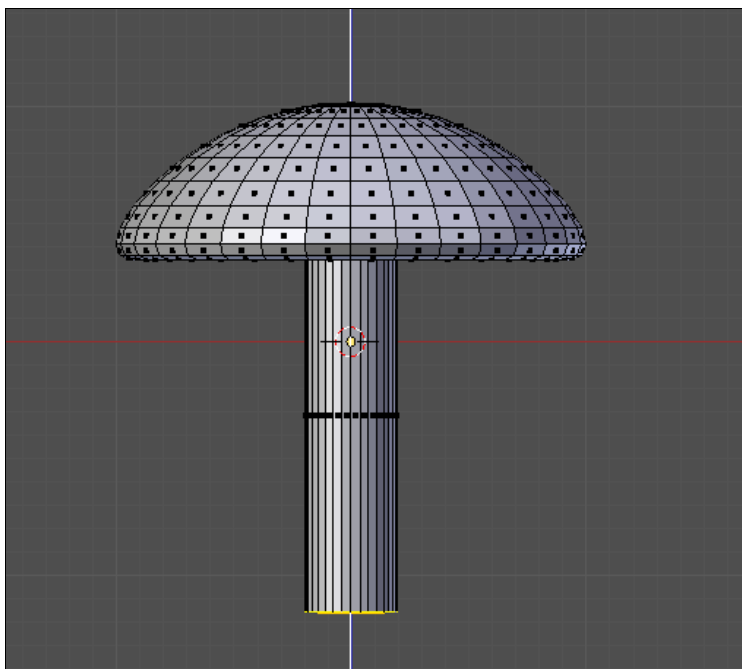


Рис. 2.8. Результат работы **Extrude**

Нажмите клавишу <E> и переместите мышью для выдавливания новых элементов. Зафиксируйте результат левой кнопкой мыши (рис. 2.8).

Уже становится немного похоже на гриб, но давайте сделаем небольшое утолщение середины ножки. Вот только придется разбить ее на несколько частей путем добавления горизонтальных ребер.

Blender имеет замечательный инструмент **Loop Cut and Slide** (Создать петлю и переместить). Вы его можете найти на панели **Tool Shelf** (слева в окне **3D View**) в секции **Add** или просто нажать <Ctrl>+<R>. После активации функции попробуйте подвигать курсор мыши по объекту. В зависимости от местонахождения курсора будут предлагаться фиолетовым цветом возможные варианты добавления ребер. Наведите курсор на ножку. В этом случае фиолетовая полоска появится ровно посередине. Путем вращения колесика можно выбрать количество мест для разбиения. Нам понадобится пять штук. Для фиксирования результата нажмите левую кнопку мыши. Новые ребра окрасятся в желтый цвет. Сбросьте выделение с помощью клавиши <A>.

Сейчас необходимо отметить по кругу центральные ребра и выполнить масштабирование. Выделять по отдельности все нужные элементы просто нецелесообразно. Blender и тут придет на помощь, предоставив способ для циклического выделения. Нажмите <Alt> и щелкните правой кнопкой мыши по любому из ребер. Проверните сцену, чтобы убедиться в полном выделении всех элементов.

Нажмите клавишу <O> для включения режима пропорционального редактирования, затем <S> и немного увеличьте толщину ножки (рис. 2.9).

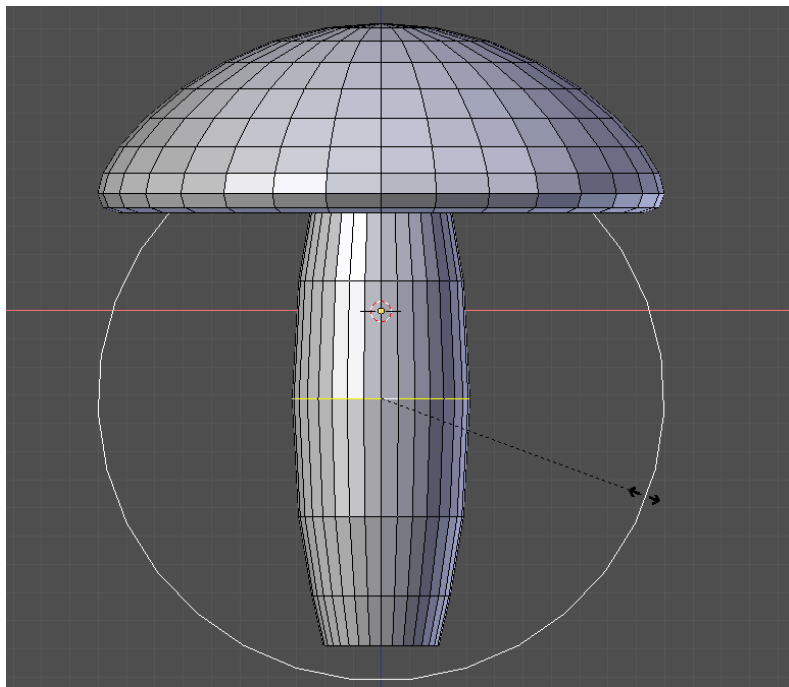


Рис. 2.9. Готовая модель гриба

До этого момента мы с вами занимались созиданием, но не менее важно научиться разрушать.

Допустим, в нашем грибе нужно вырезать парочку симпатичных дырок, скажем, проеденных червяками. Чтобы не портить готовую модель, создайте ее дубликат. Перейдите в режим **Object Mode** (клавиша <Tab>) и нажмите <Shift>+<D>. Полученную копию передвиньте в свободное место сцены.

Для удаления объектов служит команда **Delete** (быстрая клавиша <X>). Та же самая функция используется и в режиме редактирования.

Выделите любой элемент на шляпке гриба и нажмите клавишу <X>. Появится контекстное меню со следующими пунктами:

- ◆ **Verticles** (Вершины) — удаляются только вершины;
- ◆ **Edges** (Ребра) — удаляются ребра;
- ◆ **Faces** (Грани) — удаляются грани;
- ◆ **All** (Все) — полное удаление всей структуры;
- ◆ **Edges & Faces** (Ребра и грани) — удаляются ребра и грани;
- ◆ **Only Faces** (Только грани) — удаление грани;
- ◆ **Edge Loop** (Ребра по кругу) — удаление ребра со стягиванием швов.

Если при использовании функции **Delete** происходит разрушение выделенного участка модели, то **Edge Loop** удаляет ребра с последующим восстановлением струк-

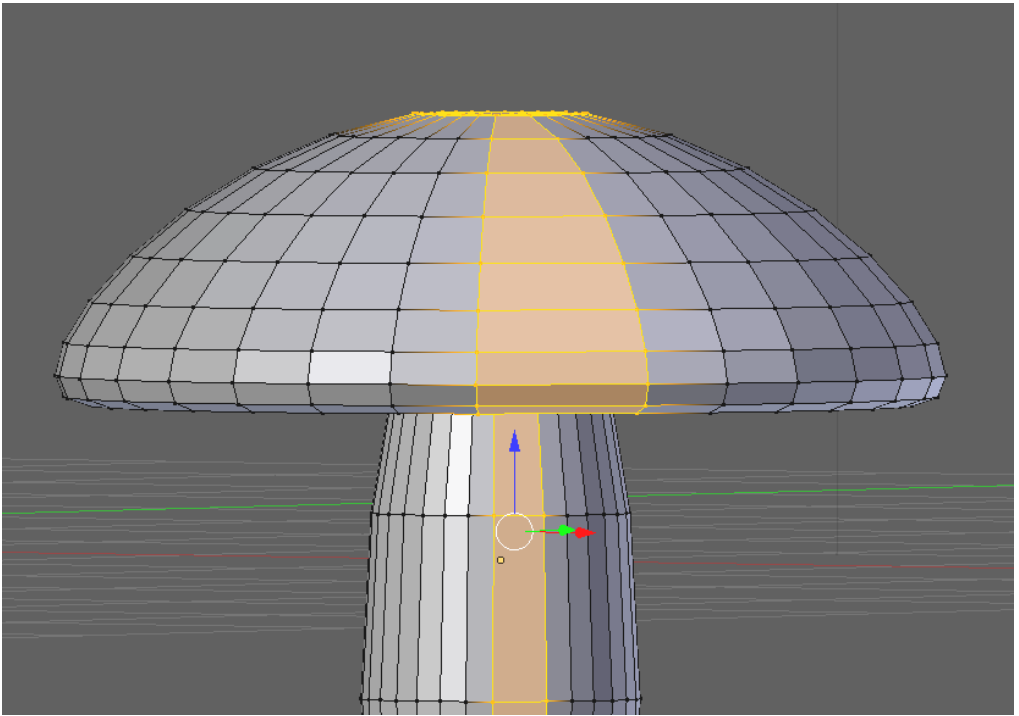


Рис. 2.10. Результат работы удаления функцией **Edge Loop**

туры. Попробуйте выделить с помощью клавиши <Alt> любую группу ребер. Нажмите <X> и выберите пункт **Edge Loop** (рис. 2.10).

Теперь вернемся к основной задаче. Как бы вы не пробовали удалять элементы гриба, в лучшем случае получается прямоугольная дырка. Это и понятно, ведь полигоны шляпки отнюдь не круглые. На помощь придет следующий инструмент **Knife**.

Knife (Нож) — это функция, позволяющая добавлять дополнительные ребра к выделенной части объекта при помощи рисованной фигуры.

Выделите четыре полигона на шляпке гриба. Нажмите и удерживайте клавишу <K>. Теперь при нажатой левой кнопки мыши появится курсор в виде ножа. Очертите область вокруг центральной точки в выделении (рис. 2.11).

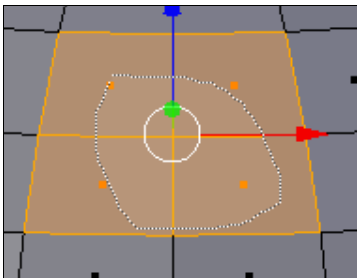


Рис. 2.11. Область, выделенная с помощью **Knife**

Если все сделано правильно, то на месте выделения появятся дополнительные ребра, а на панели **Tool Shelf** (Полка инструментов) в нижней части — настройки **Knife Cut** (Разрез ножом). Обратите внимание на пункт **Type** (Тип). С его помощью можно выбрать, как программа будет переносить наше свободное выделение в строгие рамки структуры:

- ◆ **Exact** (Точный) — разрезы происходят в месте пересечения ребер с пользовательским выделением;
- ◆ **Midpoints** (Середина) — разрезы будут выполнены строго по центру выделенных ребер;
- ◆ **Multicut** (Множественный разрез) — множественное создание копий ребер по принципу матрешки. При выборе этого пункта количество копий можно указать в поле **Number of Cuts** (Количество разрезов).

Все бы хорошо, вот только полученный ромб мало напоминает отверстие (рис. 2.12). Результат может улучшить увеличение полигонов объекта. Откатите назад при помощи функции **Undo** (Отмена) до первоначального состояния модели (используйте <Ctrl>+<Z>).

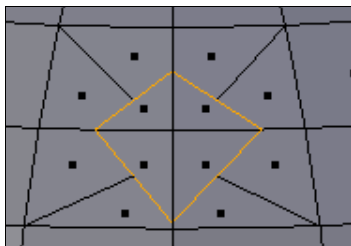


Рис. 2.12. Результат работы Knife

Процесс пропорционального увеличения элементов структуры примитива называется **Subdivide** (Разбиение). Эта функция может работать как с отдельно выделенным фрагментом, так и целиком с объектом. Использование **Subdivide** имеет свои плюсы и минусы. С одной стороны, объект становится высокополигональным (т. е. структура содержит большее количество элементов), а значит, открывается простор для точной реализации задуманного. С другой стороны, увеличение элементов несет дополнительную нагрузку на ресурсы компьютера. Это может вылиться в дополнительное время на обработку сцены. К тому же, нежелательно использовать разбивку выделенной части объекта. В некоторых случаях это приводит к нарушению целостности структуры. Справедливости ради нужно сказать, что Blender имеет и другие инструменты для безболезненного решения этой задачи. Выполнить **Subdivide** можно из панели **Tool Shelf** (Полка инструментов) либо из контекстного меню, вызываемого клавишей <W>.

Итак, нажмите клавишу <A> для выделения всего объекта. Затем <W> и выберите из появившегося меню пункт **Subdivide**. Разбивку примитива нужно выполнить дважды.

Снимите выделение (<A>) и отметьте боксовой рамкой () часть полигонов на шляпке. Теперь вы можете заново воспользоваться инструментом **Knife**.

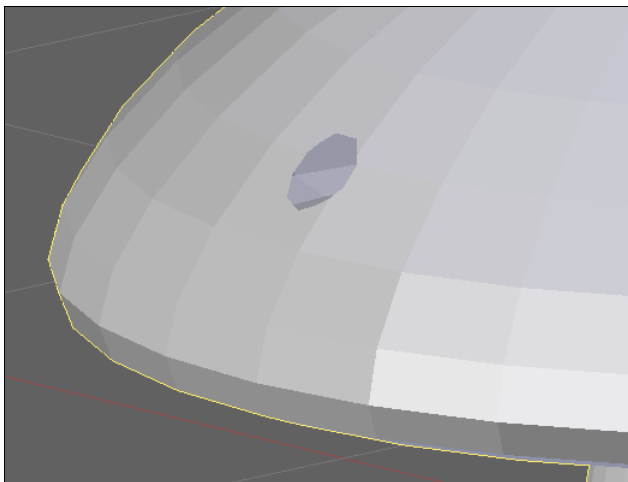


Рис. 2.13. Отверстие в шляпке

Осталось только выделить полученные грани, нажать клавишу <X> и выбрать для удаления пункт **Faces** (рис. 2.13).

Поставим перед собой еще одну задачу — срезать гриб, точнее выполнить разрез и разделение ножки модели. Для этого придется разделить объект на две независимые части. Такая функция в Blender, разумеется, есть и называется **Separate** (Разделение).

Переключите режим отображения окна на **Wireframe** клавишей <Z>. Это позволит выделить нужную часть объекта без вращения сцены. Сейчас нужно отметить половину ножки в режиме ребер.

Для выполнения операции разделения выполните команду **Mesh | Vertices | Separate**. В появившемся контекстном меню выберите пункт **Selection** (Выделенное). Вот и все, получилось два отдельных объекта (рис. 2.14).

Срез получился ровный, гладкий на загляденье. Только модель интереснее получится, если края сделать изломанными, как будто гриб сломали.

Не пугайтесь, вам не придется вручную вырезать изгибы инструментом наподобие **Knife**. Есть путь более быстрый и гораздо более эффективный.

Итак, работать будем с первой нетронутой моделью гриба. Выделите объект и нажмите клавишу <NumPad .> для центрирования его в окне программы (**Front View**). Перейдите в режим редактирования и сбросьте выделение (<A>).

Для начала выделите с помощью клавиши <Alt> по всему диаметру ножки горизонтальные ребра — те, что в середине. Это будет являться линией разлома. Собственно, задача состоит в том, чтобы по координате Z в беспорядочном порядке передвинуть вершины, т. е. создать зубцы.

Однако сначала скроем ненужную часть объекта, чтобы было удобнее возиться с разломом. Для этого инвертируйте выделение с помощью функции **Select | Inverse** и нажмите клавишу <N> (**Object | Show/Hide | Hide Selected**).

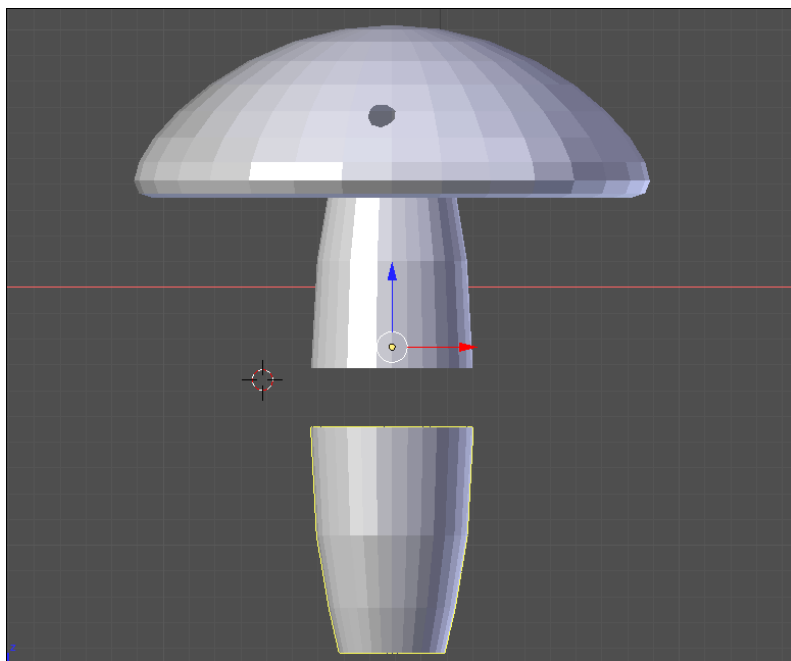


Рис. 2.14. Срез ножом

Итак, от объекта осталось лишь кольцо. Выберите режим выделения вершин и отметьте любую точку на линии.

Чтобы создать изломанную окружность, воспользуемся режимом пропорционального редактирования. При включении этого режима (клавиша <O>) в заголовке окна **3D View** появится дополнительная кнопка с меню (рис. 2.15).

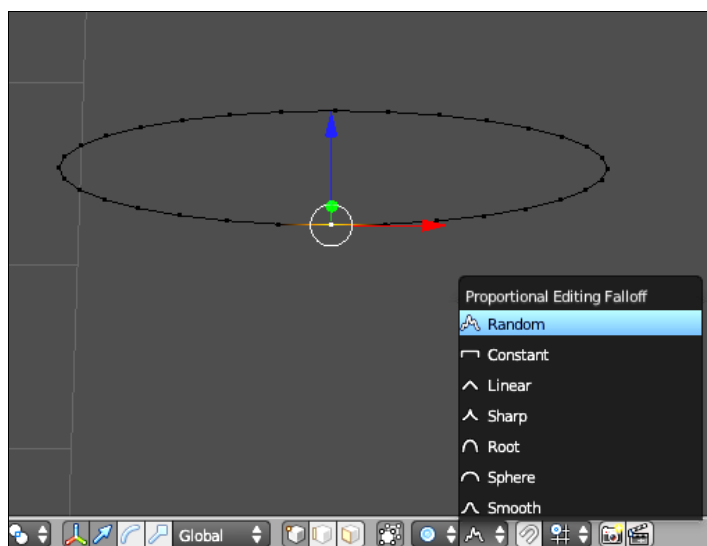


Рис. 2.15. Меню Proportional Editing Falloff

В этом меню можно выбрать тип влияния инструмента пропорционального редактирования на объект. В нашем случае годится пункт **Random** (Случайно). Выберите его.

Нажмите клавишу <G> для включения манипулятора перемещения, а затем <Z> для ограничения движения по оси Z. Немного передвиньте вершину вверх и колесиком мыши отрегулируйте влияние инструмента на всю окружность (рис. 2.16).

Восстановите видимость остальной части объекта путем нажатия клавиш <Alt>+<H> и выделите нижнюю часть примитива с захватом окружности разлома. Осталось только отделить выделение с помощью функции **Separate** (рис. 2.17).

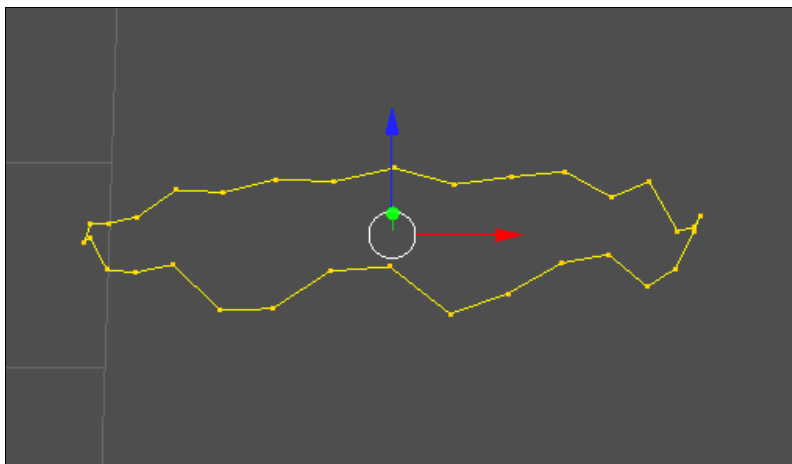


Рис. 2.16. Изломанная окружность

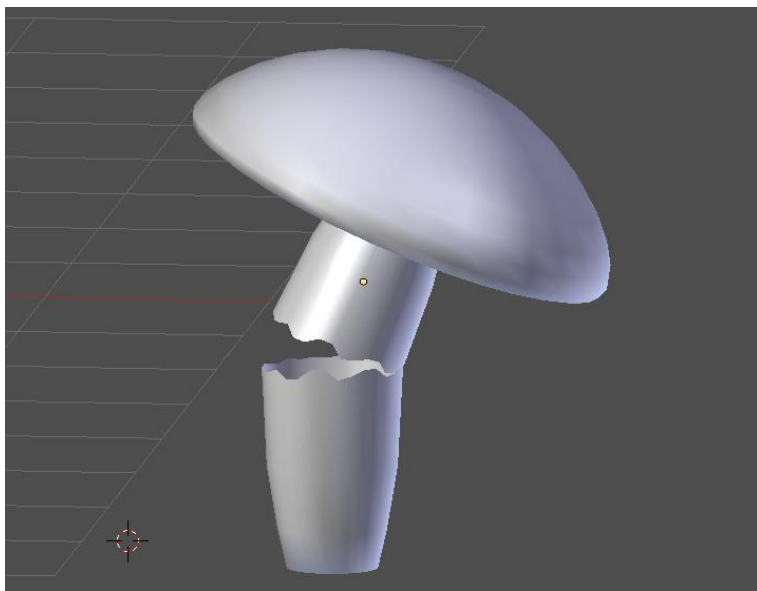


Рис. 2.17. Сломанный гриб

СОВЕТ

Для лучшего результата к модели можно применить сглаживание, которое уберет видимые неровности структуры. Для этого на панели **Tool Shelf** в режиме **Object Mode** нажмите кнопку **Smooth** (Сглаживание). Чтобы вернуть модель к первоначальному виду, нажмите кнопку **Flat** (Плоскость).

2.3. Симметричное моделирование

Оглянитесь вокруг, практически все, что нас окружает, обладает определенной симметрией. Возьмите простой карандаш, проведите мысленно линию посередине и увидите, что две половинки симметричны друг к другу. Это касается и техники, и животных, и даже человека. А если так, то зачем при моделировании выполнять двойную работу?

Симметричное моделирование — это способ создания объекта, где пользователь создает часть модели, а программа зеркально достраивает все остальное.

Рассмотрим этот способ на практике создания модели меча. Удобнее всего работать, имея перед собой образец. Но так как меч достать проблематично, то остановимся на обычном рисунке.

Blender позволяет загрузить в окно **3D View** в качестве фона любой из поддерживаемых графических файлов. Руководствуясь таким образцом, остается только повторить рисунок в 3D.

Откройте программу или создайте новый проект. Настройки фона находятся в окне **3D View** на панели свойств, которая вызывается клавишей <N>. Прокрутите ее в самый низ и найдите закладку **Background Images** (Фоновые изображения) (рис. 2.18). Установите галочку рядом с названием закладки для активации фона. Сейчас там всего одна кнопка **Add Image** (Добавить картинку). Нажмите ее.

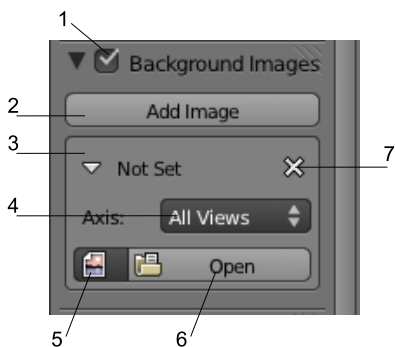


Рис. 2.18. Панель создания фона

Описание элементов панели **Background Images**:

1. Включение или отключение показа картинок в окне.
2. Добавление новой картинки. Одновременно вы можете загрузить в программу несколько файлов.
3. Область настроек отдельной картинки.

4. Указание ракурса окна **3D View**, где эта картинка будет выведена.
5. Добавление картинки из уже загруженных ранее в Blender.
6. Открытие нового файла.
7. Удаление текущего фона.

Особенностью **Background Images** является возможность загрузки нескольких картинок и привязки их к конкретным видам окна **3D View**. Это удобно использовать, если у вас имеется полноценный скетч модели (рисунок) с разных сторон.

Мы будем работать всего с одной картинкой. Поэтому нажмите кнопку **Open** и откройте файл `Scenes\glava2\sword.png` из архива примеров к этой книге (см. приложение 2) (рис. 2.19).

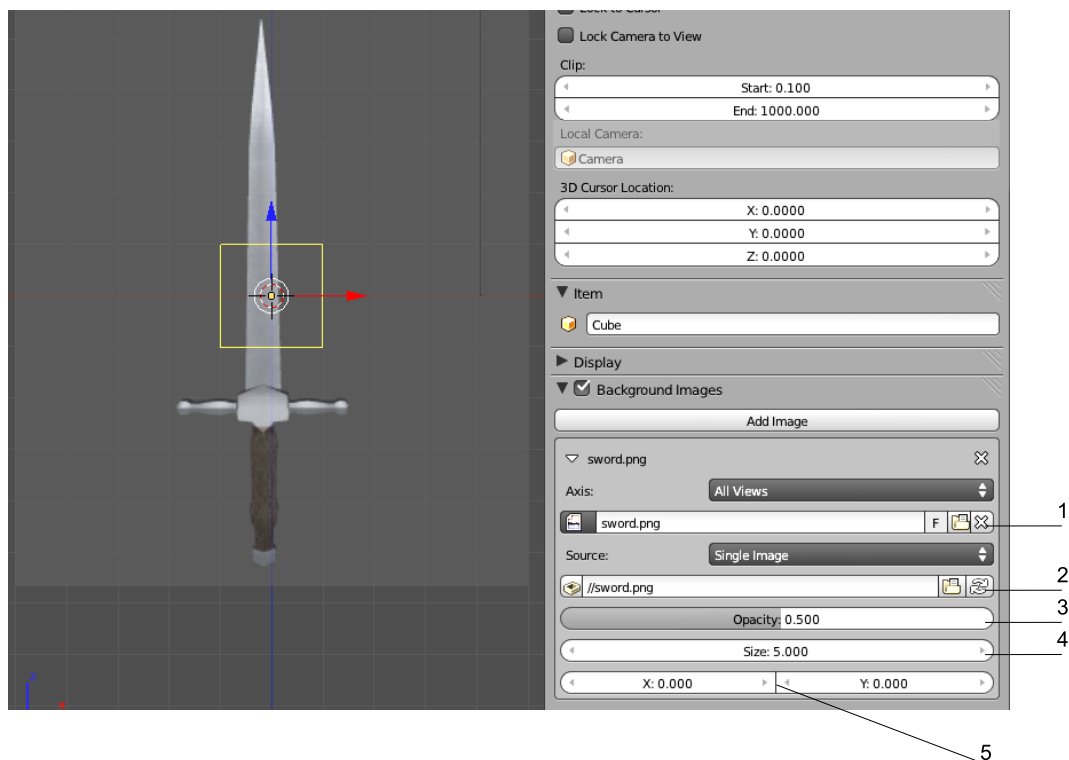


Рис. 2.19. Настройки картинки и само изображение

Описание настроек загруженной картинки:

1. Закрытие файла.
2. Перезагрузка ранее выбранной картинки.
3. Установка прозрачности изображения в окне **3D View**.
4. Масштаб изображения.
5. Смещение по вертикали и горизонтали.

Выделите куб в сцене и перейдите на просмотр **Front View** (клавиша <NumPad 1>) ортографической проекции (<NumPad 5>). Отрегулируйте изображение меча так, чтобы центр куба находился в центре перекладины. Сам куб сожмите (клавиша <S>) до размеров перекладины (рис. 2.20).

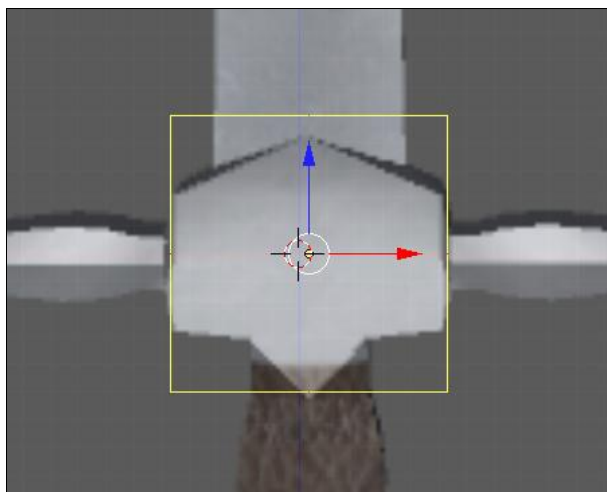


Рис. 2.20. Подгонка куба

Прежде чем начать работу над моделью, необходимо правильно подготовить примитив для симметричного моделирования. Войдите в режим редактирования и выделите всю структуру клавишей <A>. Сейчас нужно разделить объект на две половинки. Для этого разобьем куб с помощью команды **Subdivide** (Разбиение). Нажмите клавишу <W> и выберите пункт меню **Subdivide**. Удобнее всего работать будет в режиме **Wireframe**, поэтому нажмите клавишу <Z> для его активации.

Моделировать вы будете левую сторону меча. Соответственно, вся правая сторона примитива не нужна. Выделите крайние вершины и удалите их (рис. 2.21).

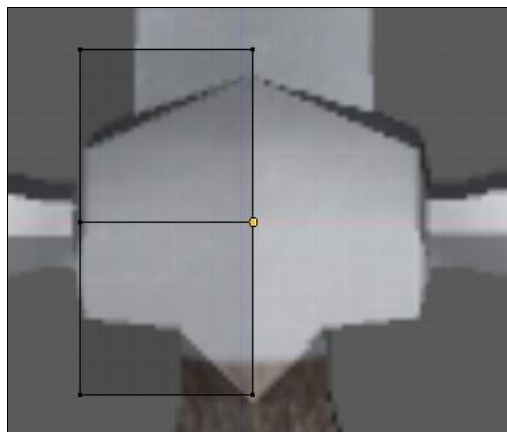


Рис. 2.21. Удаление правой стороны

Откройте вид **Right View** с помощью клавиши <NumPad 3> и удалите вершины справа. В итоге получилась заготовка для меча, где изменяться будет только левая сторона, а Blender построит объект по осям X и Y (рис. 2.22).

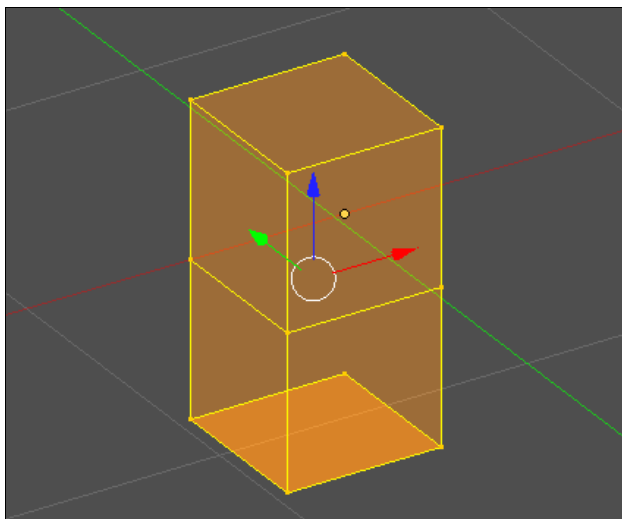


Рис. 2.22. Подготовка примитива

Давайте познакомимся с еще одним важным инструментом Blender — модификатором.

Модификатор — это временная надстройка над объектом, которая позволяет изменять его свойства и возможности. Любые действия, произведенные с помощью модификатора, можно отменить, удалив его самого. В то же время имеется возможность применить модификатор, тем самым окончательно подтвердив все изменения. К объекту может быть прикреплено неограниченное количество модификаторов.

Работа с модификаторами осуществляется в окне **Properties** на вкладке **Object Modifiers** (визуально она выглядит как гаечный ключ). В данный момент к примитиву не было присоединено ни одного модификатора, поэтому на панели должна находиться одна лишь кнопка **Add Modifier** (Добавить модификатор). При ее нажатии программа выведет список модификаторов, доступных для данного типа объекта. Выберите из него **Mirror** (Зеркало). На появившейся панели установите галочку в поле **Axis Y** (рис. 2.23).

Рассмотрим основные элементы панели модификаторов:

1. Если эта кнопка включена, то модификатор участвует в рендере результата.
2. При включении результат работы модификатора отображается в окне **3D View**.
3. Действие модификатора распространяется на режим редактирования.
4. Активация этой кнопки прорисовывает структуру всего меша в режиме редактирования.

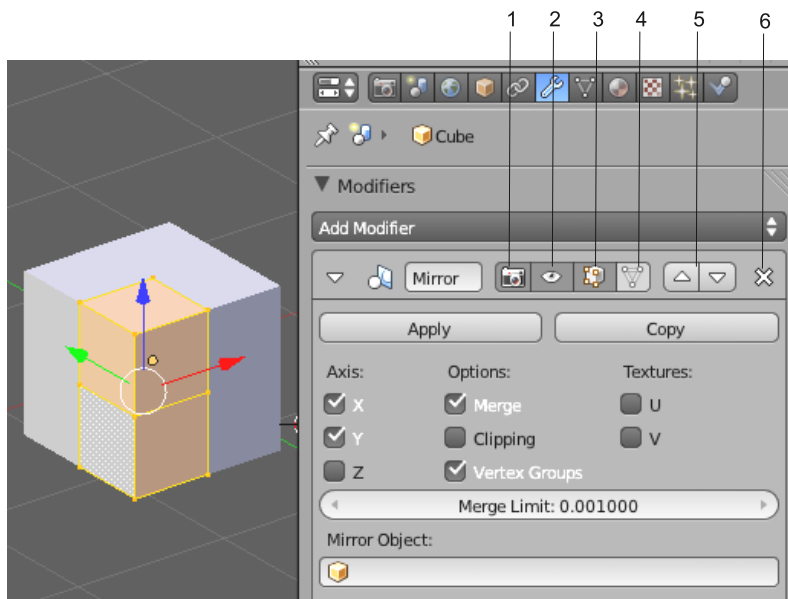


Рис. 2.23. Модификатор Mirror

5. Кнопки управления модификатором в стеке. Дело в том, что модификаторы обрабатываются по очереди и местонахождение их может влиять на конечный результат.
6. Кнопка удаления модификатора.

Кроме этих кнопок имеются еще две: **Apply** (Применить) и **Copy** (Копировать). В первом случае результат работы модификатора применяется, а сам он исчезает из очереди. Вторая кнопка производит копирование модификатора с установленными параметрами.

Модификатор **Mirror**, как уже следует из названия, выполняет зеркалирование структуры объекта по выбранным осям (**Axis**). В нашем случае нужно установить галочки для осей X и Y. Группа **Options** содержит три опции.

- ◆ **Merge** (Слияние). Когда две вершины (обычная и зеркальная) находятся на достаточно близком расстоянии, включение этой опции заставит вершины слиться в одну. Расстояние, на котором это слияние может произойти, устанавливается в поле **Merge Limit**. По умолчанию включена.
- ◆ **Clipping** (Обрезать). При включении данной опции модификатор не позволит произойти пересечениям между зеркальной и обычной вершинами. Выключена по умолчанию.
- ◆ **Vertex Groups** (Группы вершин). Если объект содержит созданную группу вершин, то модификатор будет работать с ней. Нестабильная опция. Включена по умолчанию.

Как видите, после добавления данного модификатора наш куб опять воссоздался в первоначальном виде. Однако редактирование возможно только оригинальной

части, все остальные просто повторяют манипуляции. Попробуйте подвигать любые вершины и посмотрите на реакцию объекта.

Продолжим работу над моделью. Следующая задача — путем использования **Extrude** воссоздать в 3D вертикальное изображение меча. Делается это по такому принципу:

1. Выделить верхние четыре вершины при помощи боксовой рамки (клавиша).
2. Выполнить операцию **Extrude** (<E>) и переместить новые вершины до ключевого момента рисунка.
3. Зафиксировать результат нажатием левой кнопки мыши.
4. То же самое нужно сделать и для нижней части. В целом у вас должна получиться такая заготовка, как на рис. 2.24.

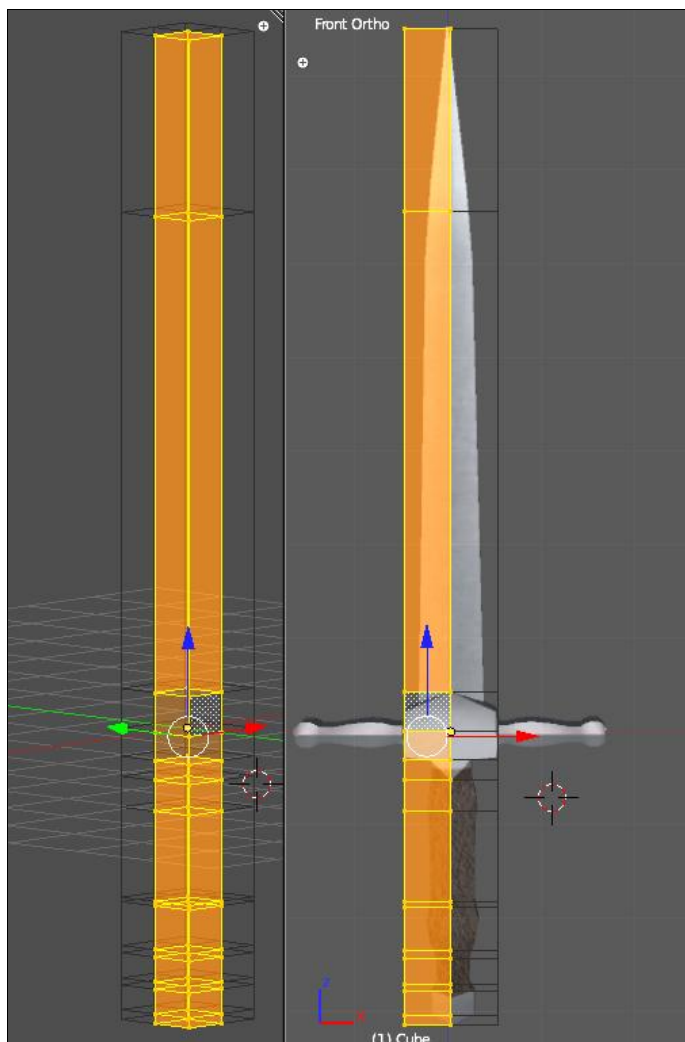


Рис. 2.24. Создание заготовки по координате Z

Для создания выступов на рукояти выделите вершины, как на рис. 2.25, и продолжите выдавливание до получения нужной фигуры (рис. 2.26).

В целом у вас должна получиться болванка, напоминающая перевернутый крест. Вот только из этой грубой формы необходимо выковать изящный меч. Сейчас предстоит работа по приведению модели в нужный вид.



Рис. 2.25. Группа вершин для создания выступов

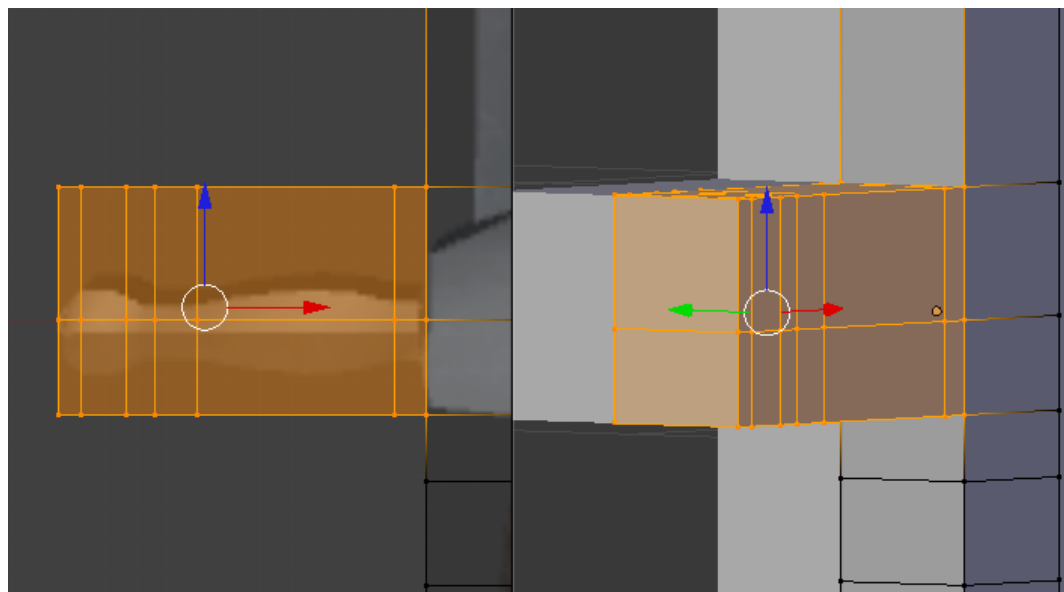


Рис. 2.26. Готовая заготовка для выступов

Алгоритм тонкой доводки:

1. В окне **Front View** выделить точку с помощью клавиши . Боксовая рамка в действительности отметит не одну, а две вершины, в том числе ту, что находится на заднем плане.
2. Переместить выделение, используя клавишу <G>, на ключевой участок изображения. Зафиксировать результат левой кнопкой мыши.
3. Снять выделение с помощью <A> и продолжить работу далее.
4. Должно получиться примерно так, как на рис. 2.27.

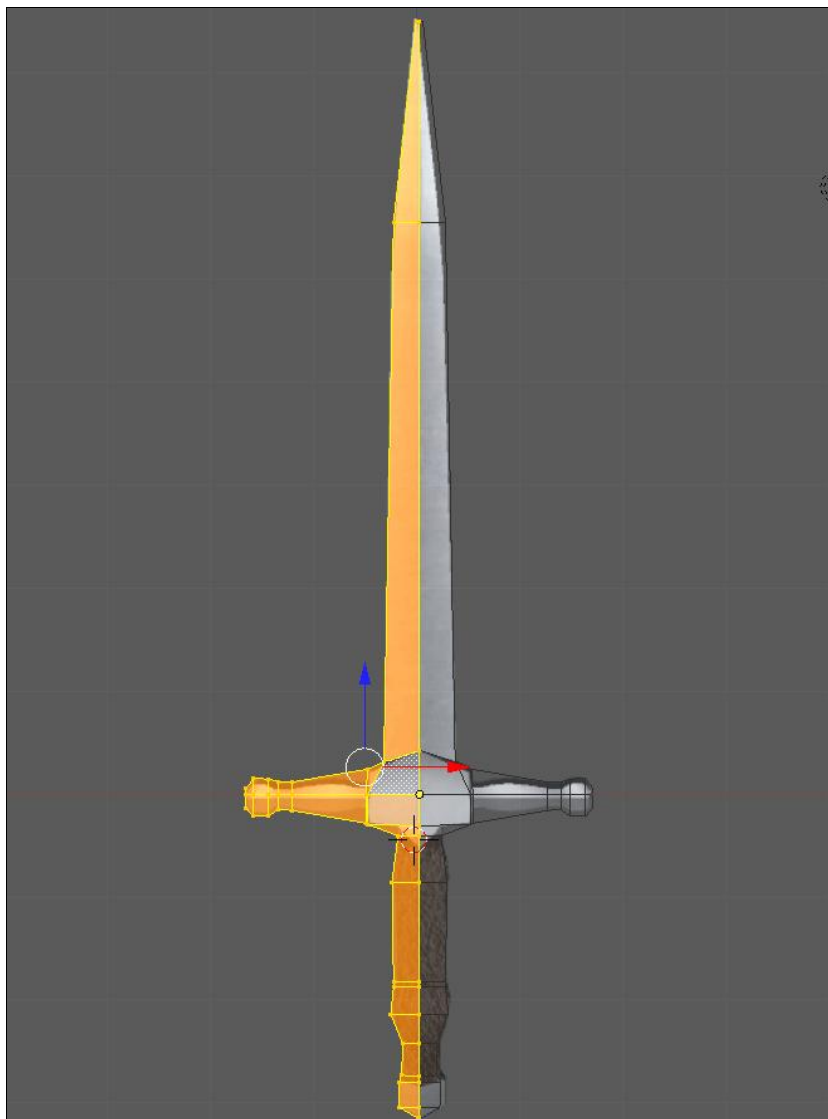


Рис. 2.27. Модель уже выглядит почти как меч

СОВЕТ

Не стремитесь полностью воссоздать структуру, как на рисунках. Вы всегда сможете добавить нужные элементы прямо на ходу. Если вдруг окажется, что для создания контура не хватает ребер, то воспользуйтесь функцией **Loop Cut And Slide** (Кольцевой разрез и скольжение) (<Ctrl>+<R>).

Вот это уже смахивает на настоящий меч. Перейдите в режим **Object Mode** (клавиша <Tab>) и сожмите масштаб модели по координате Y, уж больно она широкая.

Для того чтобы сделать лезвие уже, нежели рукоять, вам придется опять войти в режим редактирования. Разверните модель так, чтобы видеть ее в профиль, выделите вершины лезвия и подтяните их к центру. В данном случае эту операцию лучше выполнить в режиме прорисовки **Solid** (<Z>).

И наконец-то придадим кромке лезвия нужную остроту, путем слияния вершин. Для этого воспользуемся функцией **Merge** (Слияние). Принцип выполнения:

1. Первой выделяется крайняя вершина, затем с помощью клавиши <Shift> добавляется параллельная с ней точка в середине кромки (рис. 2.28). Следуйте именно этому порядку выделения, т. к. он важен для корректного выполнения слияния.
2. Нажмите клавишу <W> для вызова контекстного меню и выберите в нем пункт **Merge** (Объединение). Появится дополнительное меню, где выберите пункт **At Last** (К последней).

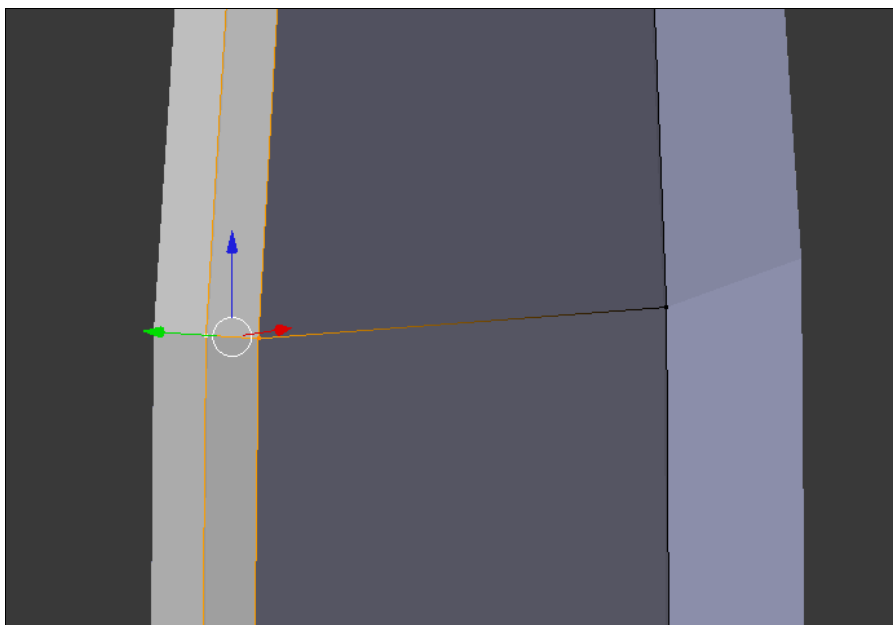


Рис. 2.28. Пример вершин для функции Merge

Рассмотрим варианты слияния, выдаваемые функцией **Merge**.

- ◆ **At First** (К первой). Новая точка образуется на месте первой выделенной вершины.

- ◆ **At Last** (К последней). Новая вершина образуется на месте последнего выделения.
- ◆ **At Center** (К центру). Высчитывается центр между вершинами.
- ◆ **At Cursor** (К курсору). Вершина образуется на месте курсора **3D Cursor**.



Рис. 2.29. Готовая модель меча

Последним завершающим штрихом можно применить модификатор **Mirror** с помощью кнопки **Apply**, если вы, конечно, уверены, что он больше не понадобится.

Модель получившегося меча показана на рис. 2.29.

2.4. Булевы операции

В одном из примеров показывался вариант создания отверстий с помощью инструмента **Knife**, но это кропотливый и ручной способ выполнения данной задачи. В некоторых случаях использование **Knife** может быть нецелесообразным, например, при создании большого количества отверстий. Бывает и обратная ситуация, когда нужно не вырезать, а добавлять выпуклые искажения в существующую модель.

Попробуем представить, как можно сделать пустой ящик. Вроде бы простая задача, но только на первый взгляд. Первое, что приходит в голову — это увеличить количество элементов в кубе и удалить их часть с помощью клавиши **<Delete>**. Правда

делать это придется поэтапно, скрупулезно вгрызаясь внутрь объекта. Вторым вариантом — создать сборный ящик из нескольких примитивов, а затем склеить их с помощью **Join**. Сделать можно, но потратится немало времени на подгонку объектов. А если нужно в кубе вырезать сферическую область?

Blender предлагает иной способ решения подобных задач — булевы операции.

Термин "Boolean" чаще всего используется в программировании для обозначения примитивных логических операций, где результатом является значение True (Правда) или False (Ложь). В компьютерной 3D-графике подобным словом обозначается способ взаимодействия двух объектов и создания на их основе смещенного результата.

Для использования булевой функции в Blender обязательно нужны два объекта. К одному из них присоединяется специальный модификатор **Boolean** (рис. 2.30). Кстати, эта функция доступна только для примитивов **Mesh**.

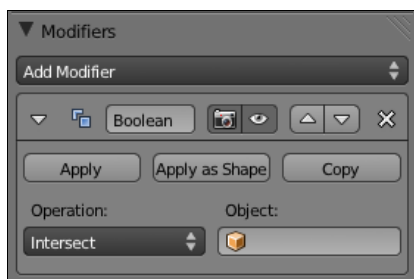


Рис. 2.30. Модификатор Boolean

Настройки модификатора **Boolean**:

- ◆ меню **Operation** (Операция) позволяет выбрать способ взаимодействия двух примитивов:
 - **Intersect** (Пересечение);
 - **Union** (Объединение);
 - **Difference** (Различие);
- ◆ поле **Object** (Объект) — название второго примитива для действия. При щелчке появляется меню, где можно выбрать объект из списка.

Рассмотрим работу с **Boolean** на примере **Cube** и **Cone**. Допустим, к первому примитиву присоединен модификатор, а второй выбран в качестве взаимодействующего объекта. В меню **Boolean** установлена операция **Intersect**. На рис. 2.31 вверху показан первоначальный вид и расположение примитивов, а внизу результат операции.

Итак, результатом **Intersect** являются два объекта, где **Cone** не изменяется, а куб принимает форму обрезанного конуса. Причем, выходящие за его пределы части **Cone** не учитываются.

Следующая операция — **Union**. Эта функция производит сложение двух объектов. Результат ее вы можете увидеть на рис. 2.32.

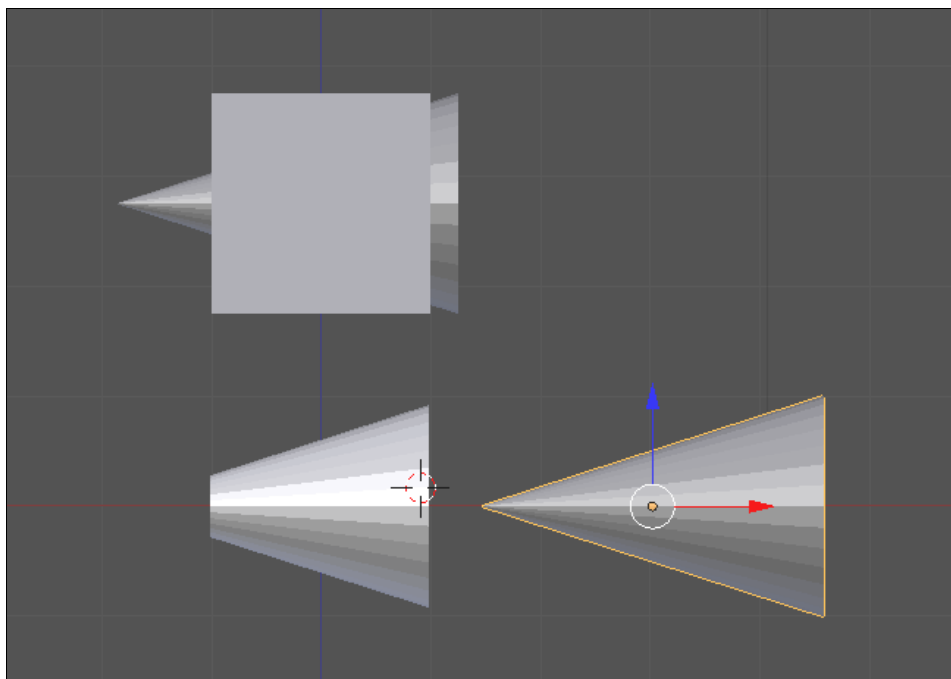


Рис. 2.31. Операция Intersect

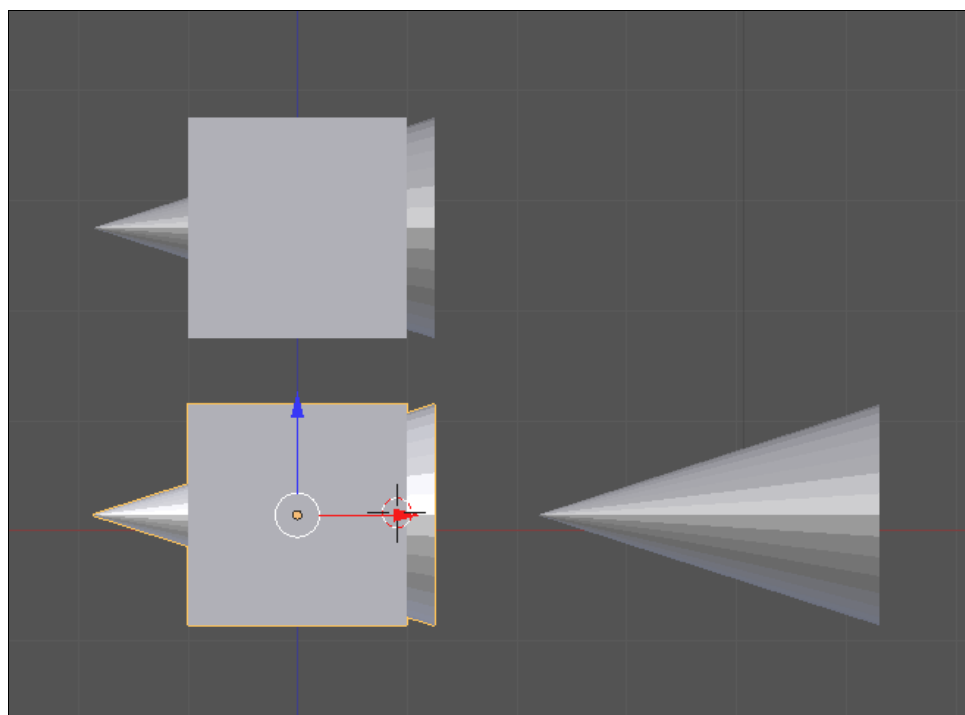


Рис. 2.32. Операция Union

Difference может использоваться для создания отверстий в главном объекте (рис. 2.33).

Работа модификатора **Boolean** происходит в реальном времени. После выбора нужной операции и установки взаимодействующего объекта можно манипулировать примитивами и подгонять их под нужный результат. Следует учитывать, что при использовании **Difference** конечный результат можно увидеть только после применения модификатора.

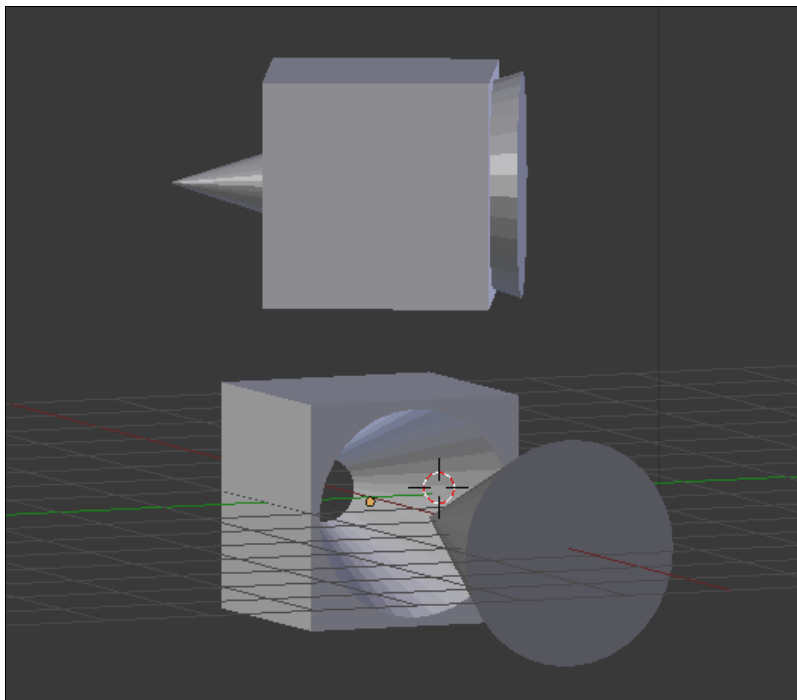


Рис. 2.33. Операция **Difference**

ЭТО ВАЖНО!

Работа с булевыми операциями требует определенной осторожности. С простыми формами объектов обычно не возникает никаких проблем. Более сложные модели могут обрабатываться некорректно. Иногда это вызывает зависание Blender. Наиболее типичной проблемой являются серьезные нарушения структуры готового объекта. Чаще всего они видны при применении сглаживания. Этого можно избежать, если: а) увеличить количество элементов структуры у обоих объектов; б) установить сглаживание до применения модификатора.

2.5. Вспомогательная решетка *Lattice*

В работе со сценой бывают случаи, когда необходимо временно изменить форму модели без нарушения ее структуры. Например, при создании анимации сплющивания мяча при ударе о стену. Разумеется, можно банально воспользоваться масштабированием объекта, но есть вариант лучше.

Lattice (Решетка) — это вспомогательный объект, который не отображается при обработке сцены, но позволяет деформировать модель.

Для использования **Lattice** нужно выполнить следующее:

1. Добавить в сцену сам объект из меню **Add | Lattice**.
2. Добавить модификатор **Lattice** к модели, которую нужно деформировать.
3. Выбрать в модификаторе имеющийся в сцене объект **Lattice**.

Главным условием работы решетки **Lattice** является местонахождение модели внутри нее. По умолчанию **Lattice** выглядит, как обычный куб с 12 ребрами (рис. 2.34).

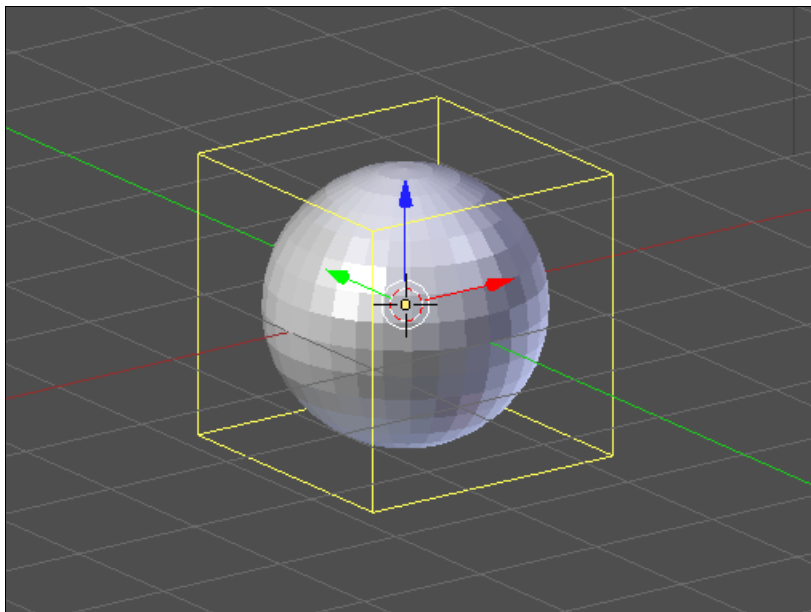


Рис. 2.34. Объект **Lattice** с примитивом внутри

Решетка подчиняется тем же правилам манипуляций, что и другие объекты. Теперь рассмотрим панель модификатора **Lattice**. Всего там находится две опции:

- ◆ **Object** (Объект) — при щелчке по полю откроется меню, где можно выбрать из списка объект **Lattice**;
- ◆ **Vertex Group** (Группа вершин) — если модель имеет созданные группы вершин, то данное поле позволяет выбрать из них. В этом случае **Lattice** будет работать только с группой.

Изменение формы модели осуществляется в режиме редактирования объекта **Lattice**. В отличие от обычных примитивов доступно перемещение, масштабирование и ротация только вершин. Собственно, манипуляции с вершинами **Lattice** влияют на деформацию модели.



Рис. 2.35. Настройки объекта **Lattice**

Объект **Lattice** имеет свои собственные настройки, которые доступны в окне **Properties** (рис. 2.35).

В некоторых случаях восьми точек **Lattice** бывает недостаточно для изменения модели. В настройках объекта можно изменить разбивку структуры с помощью параметров **U**, **V**, **W**. Рядом с этими полями находятся меню, где можно выбрать алгоритм, по которому **Lattice** будет деформировать модель (рис. 2.36—2.38).

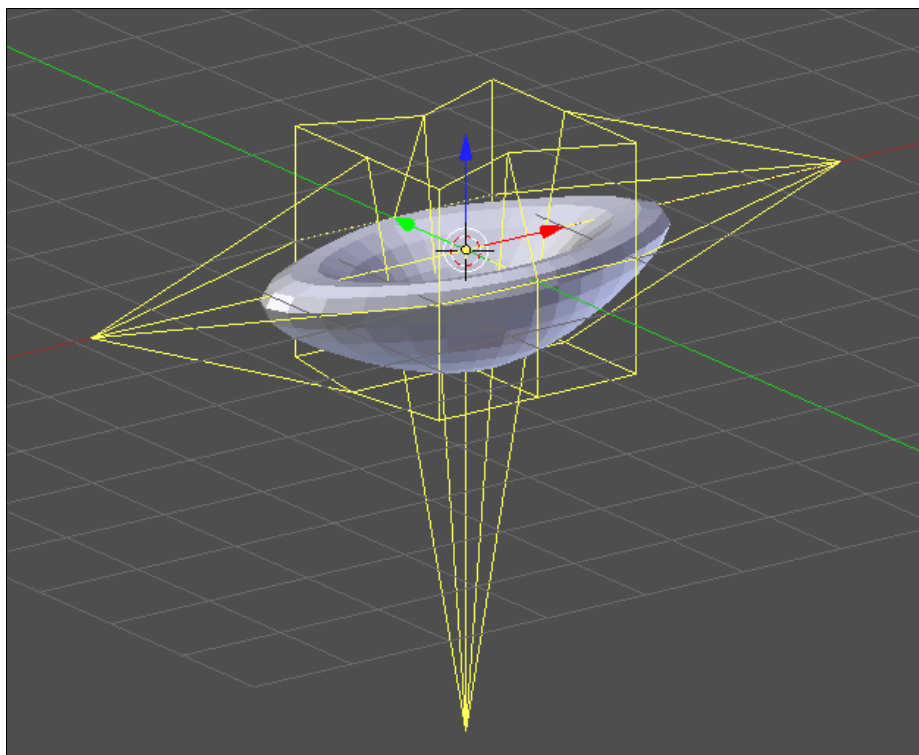


Рис. 2.36. Результат деформации сферы с параметрами **BSpline**

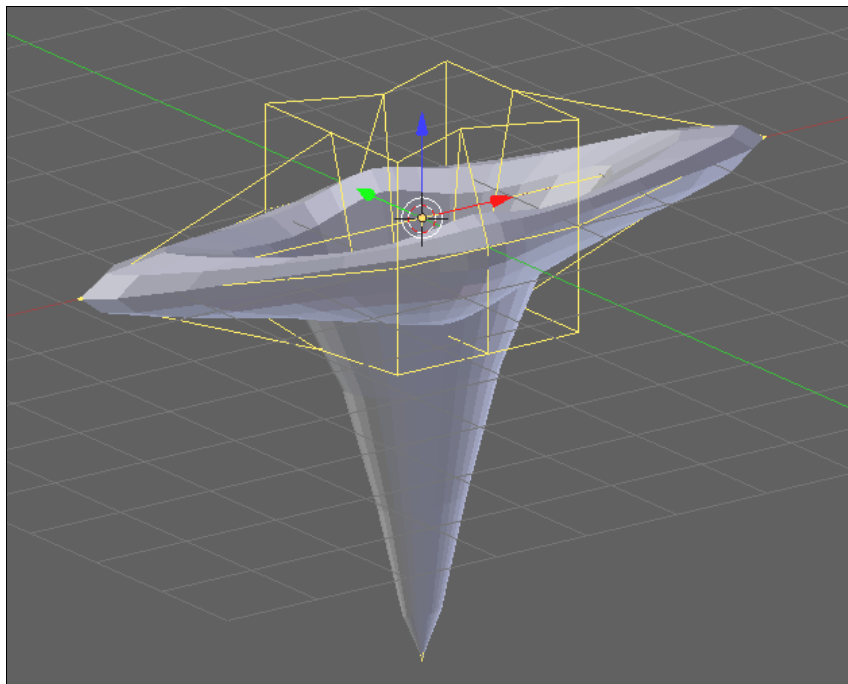


Рис. 2.37. Результат деформации сферы с параметрами **Cardinal**

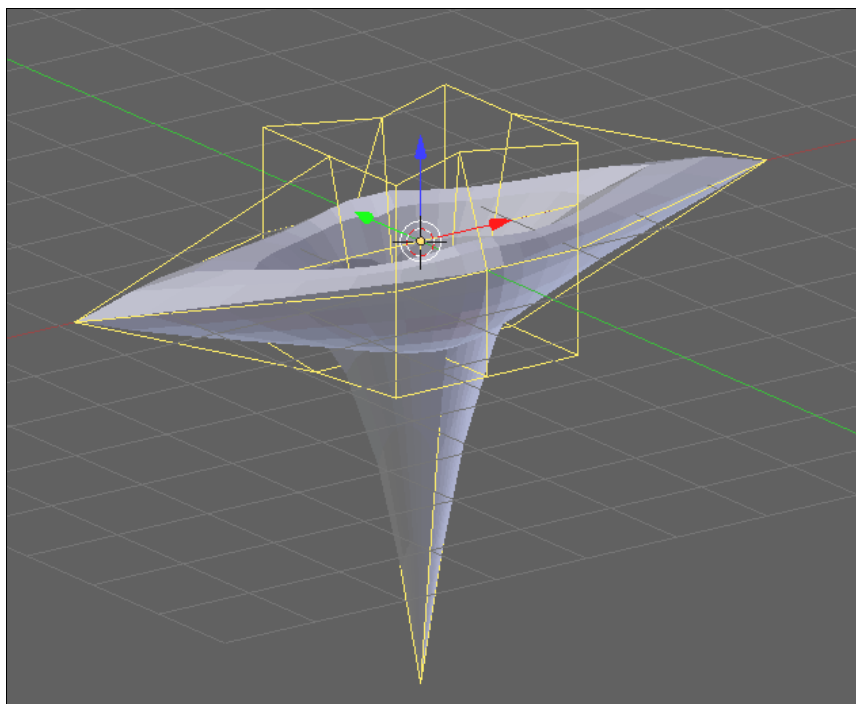


Рис. 2.38. Результат деформации сферы с параметрами **Linear**

2.6. Высокополигональное моделирование

В этот раз разговор пойдет о создании моделей с большим количеством полигонов. Для этого в Blender имеются особые способы редактирования.

Высокополигональные модели обладают двумя связанными свойствами: возможностью максимально точно передать форму задуманного объекта и большой нагрузкой на систему при рендере результата.

Именно высокополигональные модели используются при создании кинематографических эффектов, 3D-мультфильмов и даже в играх. Правда, в последнем случае, они чаще всего играют вспомогательную роль. Дело в том, что для вывода трехмерной графики в реальном времени, да и еще в хорошем визуальном качестве, разработчики вынуждены пускаться на разные ухищрения. Так, в играх используются исключительно низкополигональные объекты, а мелкие детали передаются с помощью различных рельефных карт (normal map, bump map и т. д.). Blender позволяет при наличии двух моделей в разном разрешении и качестве деталей создавать рельефную карту. В дальнейшем такая карта уже накладывается на простую модель и тем самым создается видимый эффект высокополигонального объекта. Эти карты, равно как и способы создания, мы будем рассматривать чуть позже.

Работу над подобными моделями можно разбить на два этапа: создание базовой формы и доводка ее до ума мелкими деталями.

Итак, все начинается с низкополигонального объекта, который в общих чертах представляет собой форму модели. Для более качественной отделки требуется увеличить количество элементов структуры объекта. Мы уже рассматривали функцию **Subdivide**, которая пропорционально разбивает структуру примитива. Но вот использовать исключительно только ее для высокополигонального моделирования категорически запрещено. Дело в том, что увеличение элементов выполнить очень просто, но вот уменьшение их же представляет большую сложность. Последнее может понадобиться, если количество элементов структуры затрудняет моделирование или критично по времени рендера.

Для работы над высокополигональной моделью лучше воспользоваться следующим алгоритмом:

1. Создание низкополигональной модели.
2. Разбиение ее с помощью **Subdivide** (Разбиение) на минимально достаточное количество элементов для работы над формой.
3. Использование модификатора **Multires** (Многоуровневая детализация), который позволяет контролировать качество структуры.
4. "Обтесывание" формы с помощью инструментов скульптурного моделирования.

Это, конечно, очень упрощенный список, но в целом, чаще всего так производится работа над качественной **Mesh**-моделью.

Как работать с **Subdivide** (Разбиение), вы уже знаете. Теперь познакомимся с модификатором **Multires** (Многоуровневая детализация).

Создайте новый проект и добавьте к уже имеющемуся кубу этот модификатор на панели **Properties** (рис. 2.39). По умолчанию **Multires** никаких действий с объектом предпринимать не будет. Работа с этим модификатором осуществляется в режиме **Object Mode**.

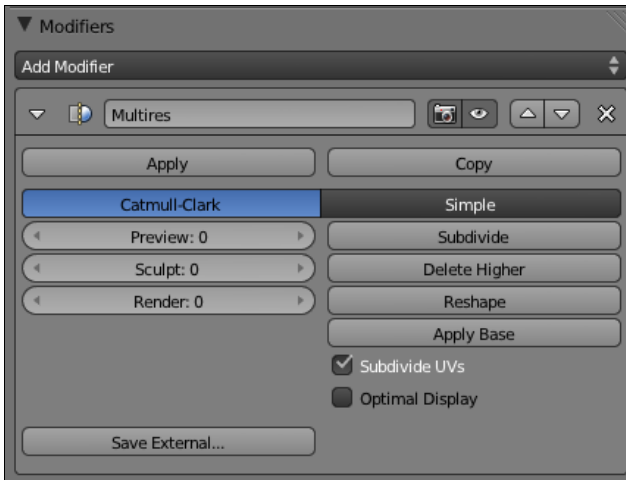


Рис. 2.39. Настройки модификатора **Multires**

Для увеличения количества элементов на панели модификатора имеется кнопка с уже знакомым названием **Subdivide**. Попробуйте ее нажать, но что это? Прimitives **Cube** неожиданно превратился в сферу. В то же время в режиме редактирования примитив выглядит как положено (рис. 2.40).

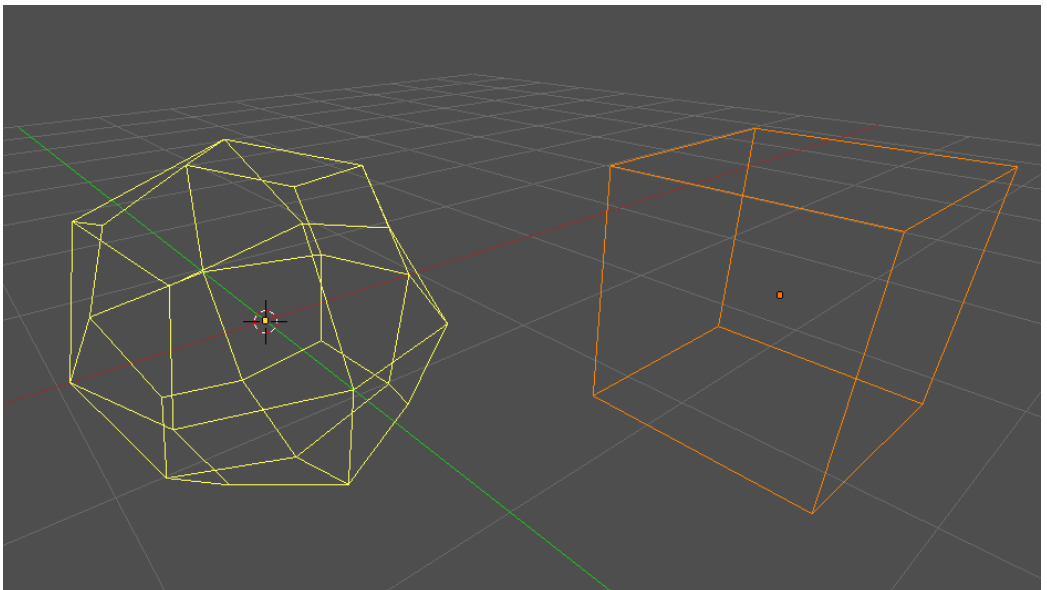


Рис. 2.40. Слева результат работы **Multires**, справа тот же объект в **Edit Mode**

Модификатор **Multires** предлагает для работы два различных алгоритма разбивки: **Catmull-Clark** и **Simple**. Первый создает сглаживание, а второй просто выполняет **Subdivide**. По умолчанию включен всегда **Catmull-Clark**. Для переключения режимов разбивки на панели модификатора имеются одноименные кнопки.

Если вам необходимо создать модель со сглаженными краями и при этом сохранить ее форму, а такое бывает сплошь и рядом, то на помощь придет обычная функция **Subdivide**, вызываемая в режиме редактирования.

Нажмите еще два раза кнопку **Subdivide** на панели модификатора для большего увеличения количества элементов. Теперь перейдите в режим редактирования и выполните несколько раз разбивку структуры с помощью меню, вызываемого клавишей <W> (рис. 2.41).

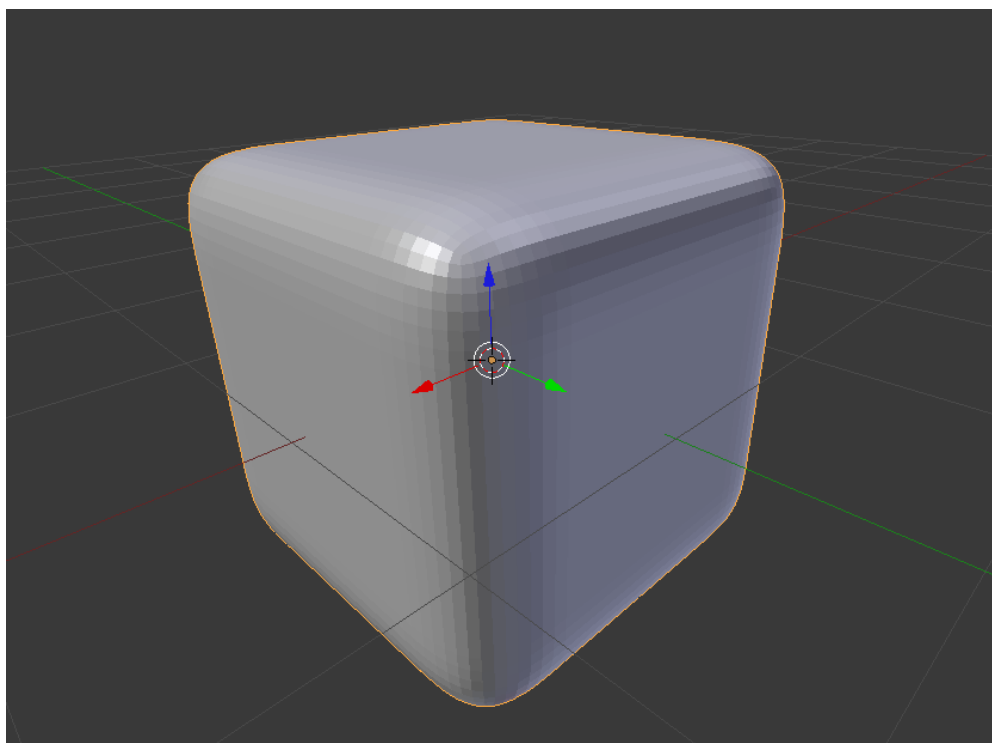


Рис. 2.41. Объект со сглаженными ребрами

Вот теперь получился куб со сглаженными ребрами. Обратите внимание на то, что вне зависимости от степени разбивки на панели модификатора, в режиме редактирования объект выглядит как обычно. Возникает закономерный вопрос, если в **Edit Mode** не произошло видимых изменений и напрямую использовать в работе разбивку **Multires** не удастся, то зачем его собственно использовать?

Во-первых, для придания форме объекта эффекта сглаживания. Здесь нужно понять разницу между обычным сглаживанием, выполняемым функцией **Smooth** из панели **Tool Shelf**, и результатом работы модификатора.

Функция **Smooth** создает эффект сглаживания за счет игры светотеней, не влияя напрямую на структуру объекта. **Multires** соответствующим образом обрабатывает сетку примитива, путем добавления новых полигонов. Кстати, для абсолютной внешней гладкости модели используют одновременно оба варианта.

Во-вторых, **Multires** позволяет в любое время изменить степень разбивки объекта, вплоть до возврата в начальное состояние. Этим удобно пользоваться при работе в режиме скульптурного моделирования. Например, вы можете нанести на объект мелкие детали в режиме скульптуры, а вот общую форму модели изменять в режиме редактирования, причем безо всяких потерь.

Модификатор **Multires** позволяет управлять отображением качества объекта с помощью следующих функций (см. рис. 2.39):

- ◆ **Preview** (Предпросмотр) — уровень детализации объекта в окне **3D View**;
- ◆ **Sculpt** (Скульптура) — уровень детализации в режиме скульптурного моделирования;
- ◆ **Render** (Обработка) — уровень детализации при рендере объекта.

Допустим, вы создали двойную разбивку объекта. Соответственно, во всех этих полях будет стоять цифра 2. Так, для убыстрения прорисовки объекта в окне **3D View** можно указать в поле **Preview** меньшее значение, а вот результат рендера будет просчитываться с максимальным качеством.

Теперь рассмотрим способ удаления уровня разбивки. Конечно, можно просто уменьшить значения в полях **Preview** и **Render**. Но в некоторых случаях лучше полностью удалить детализацию. Для этого служит функция **Delete Higher** (Удалить высший), вызываемая одноименной кнопкой на панели модификатора (см. рис. 2.39).

Delete Higher удаляет уровни, стоящие после указанного в поле **Preview**. Рассмотрим небольшой пример. Пусть было создано три уровня разбивки. Нужно оставить только первый. Устанавливаем в поле **Preview** цифру 1 и нажимаем кнопку **Delete Higher**.

В этом разделе несколько раз мелькало словосочетание "скульптурное моделирование". Под этим понятием подразумевается особый способ работы над высокополигональными моделями. Со стороны это действительно напоминает работу скульптора, который движениями руки придает глине нужную форму.

Основным инструментом здесь является **Brush** (Кисть). Причем для скульптурного моделирования создатели Blender предлагают специальный режим окна **3D View** — **Sculpt Mode** (рис. 2.42).

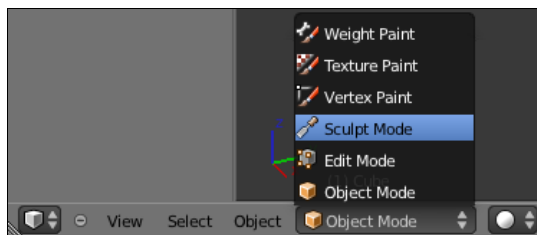


Рис. 2.42. Выбор режима **Sculpt Mode**

Скульптурное моделирование доступно и эффективно только для **Mesh**-объектов высокого разрешения. Это и не удивительно, ведь рабочий инструмент **Brush** представляет собой окружность, влияющую в своих пределах на структуру объекта. Поэтому, чем выше разрешение объекта, тем тоньше и точнее можно работать с кистями. В этом как раз и поможет модификатор **Multires**.

У вас же сохранился примитив куба с несколькими уровнями разбивки модификатором **Multires**? Если да, то перейдите в режим **Sculpt Mode**.

В этот момент стандартный курсор Blender примет форму окружности — это и есть та кисть, о которой шла речь. Пользоваться ею очень просто — наведите на грань куба и, удерживая нажатой левую кнопку мыши, нарисуйте что угодно (рис. 2.43).

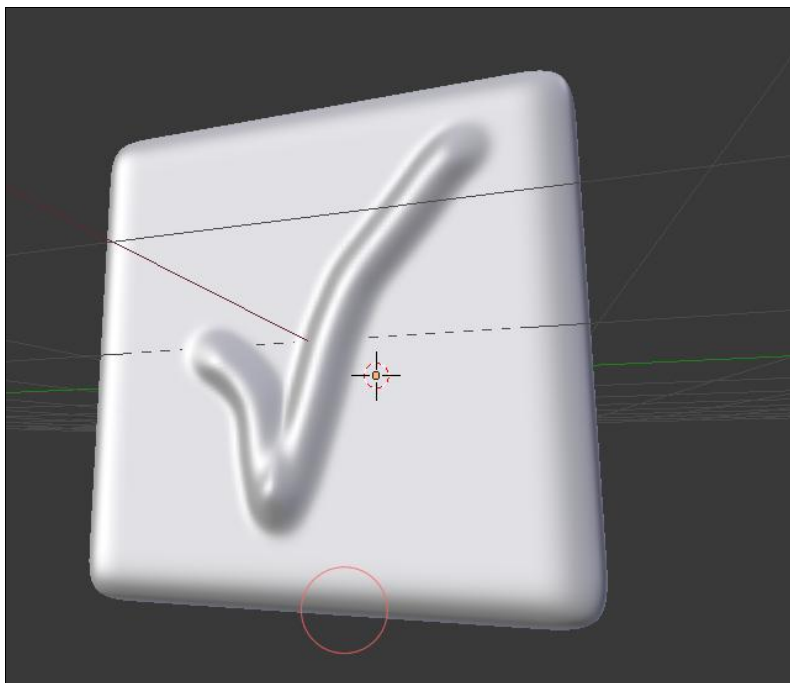


Рис. 2.43. Пробный росчерк кистью

По умолчанию Blender использует кисть с названием **SculptDraw** в режиме выдавливания. Подобных заготовок у него 19! Они доступны на панели **Tool Shelf** (рис. 2.44). Действие их наглядно демонстрируется в окне выбора кисти.

Под выбранной кистью находится набор параметров, влияющий на конечный результат (рис. 2.45).

Первая группа отвечает за поведение кисти:

- ◆ **Radius** (Радиус) — масштаб рабочей области курсора;
- ◆ **Strenght** (Сила) — сила выдавливания;
- ◆ **Autosmooth** (Автоматическое сглаживание) — используйте это значение для регулирования влияния действия кисти за пределами курсора инструмента.

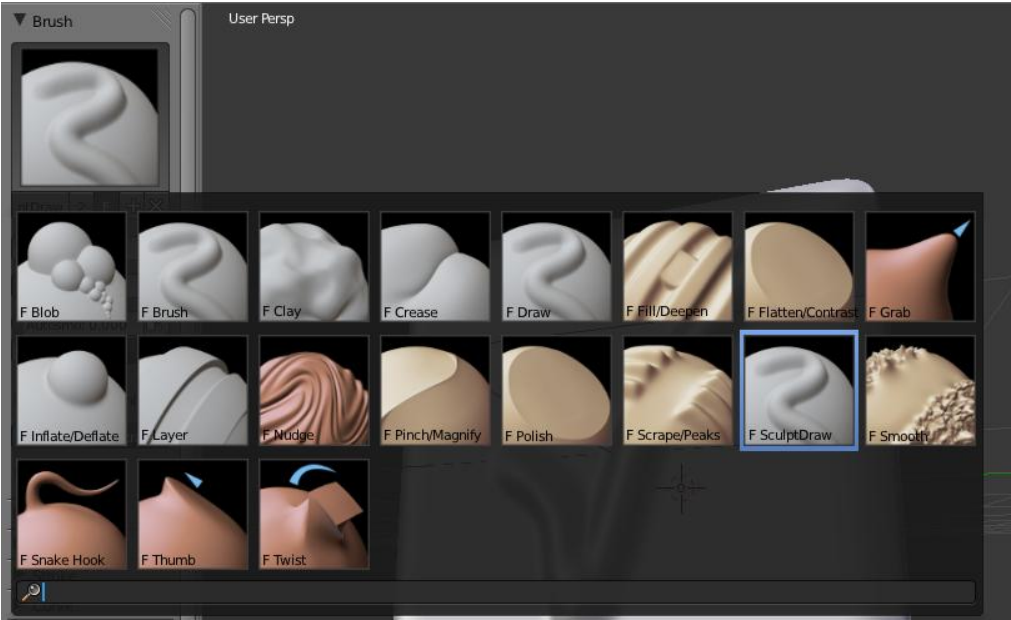


Рис. 2.44. Набор кистей Blender

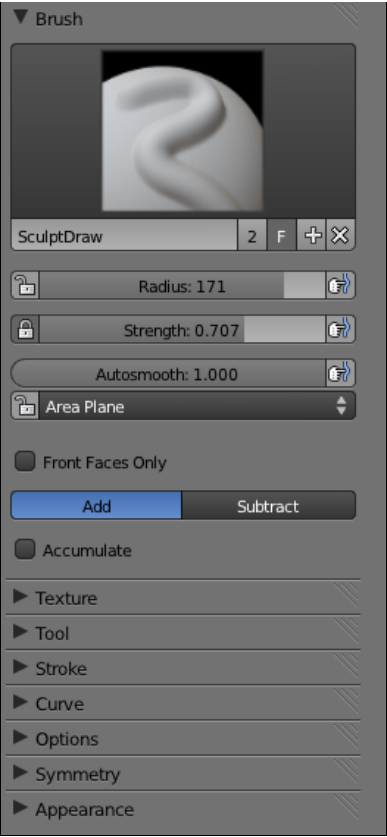


Рис. 2.45. Параметры Brush

Следующее меню **Sculpt Plane** (Область лепки) позволяет выбрать ориентацию кисти:

- ◆ **Area Plane** (Область плоскости) — результат не зависит от точки просмотра объекта или выбора грани и выполняется в соответствии с локальной системой координат модели. Это значение используется по умолчанию;
- ◆ **View Plane** (Видимая плоскость) — кисть всегда "смотрит" с точки зрения пользователя;
- ◆ **X,Y,Z Plane** (Плоскость по координатам) — результат работы зависит от выбранной глобальной координаты.

Для переключения режима работы кисти существуют кнопки: **Add** (Добавить) и **Subtract** (Отнять). Таким образом, если вам необходимо произвести вдавливание, нажмите кнопку **Subtract**.

Это были основные параметры, но есть еще несколько вкладок, которые полностью будут рассмотрены позже, а пока только описание:

- ◆ **Texture** (Текстура) — эта вкладка позволяет выбрать или создать текстуру, которая будет использоваться, как шаблон для кисти;
- ◆ **Tool** (Инструмент) — универсальная вкладка, которая существует для многих режимов работы **3D View**. Так, для кистей дублируется выбор типа из списка, сброс параметров по умолчанию (**Reset Brush**);
- ◆ **Stroke** (Влияние) — предлагается меню, в котором можно выбрать поведение кисти:
 - **Space** (Пространство) — свободное рисование. Кисть равномерно выполняет свою функцию, вне зависимости от скорости и направления движения;
 - **Drag Dot** (Перемещение точки) — в этом случае действие кисти может перемещаться по объекту, пока удерживается нажатой левая кнопка мыши. Отпускание кнопки фиксирует результат;
 - **Dots** (Точки) — результат зависит от скорости перемещения кисти;
 - **AirBrash** (Воздушная кисть) — по умолчанию действие кисти является однократным. Если выбран этот параметр, то при нажатии и удержании левой кнопки мыши, результат действия будет накапливаться;
 - **Anchored** (Привязка) — движение мыши вызывает не перемещение кисти, а увеличение масштаба охвата поверхности, т. е. при нажатии левой кнопкой мыши курсор "привязывается" к выбранной точке;
- ◆ **Curve** (Кривая) — точная настройка интенсивности действия кисти от начала и до конца движения;
- ◆ **Options** (Опции) — настройки режима скульптурного моделирования;
- ◆ **Symmetry** (Симметрия) — полезные опции, позволяющие дублировать по разным осям действие кисти;
- ◆ **Appearance** (Проявление) — здесь можно указать цвет раскраски курсора кисти в режимах **Add** и **Subtract**.

Как и везде в Blender, управление кистями можно осуществлять разными способами: через меню **Sculpt**, кнопками в **Tool Shelf** (Полка инструментов) и, естественно, горячими клавишами.

Продолжим практику работы в режиме скульптурного моделирования. Для быстрого переключения кисти в режим **Subtract** (Вдавливание) нужно использовать клавишу <Ctrl> совместно с левой кнопкой мыши. Попробуйте нарисовать пробную линию данным способом.

Оригинально, а главное удобно сделано "горячее" масштабирование (**Radius**) и управление силой (**Strength**) кисти. Так для изменения радиуса необходимо навести курсор на объект и нажать клавишу <F>. Обратите внимание, что при движении мыши (не колесика!) появляется новая окружность и темная зона. Та часть курсора, что не изменяется, указывает на старый радиус, а вот второй круг позволяет выбрать желаемый радиус. Темная зона, в свою очередь, демонстрирует реальную разрешающую способность кисти при данном масштабе и зависит от ранее выбранного типа кисти. Щелчок левой кнопкой изменит радиус, а правой отменит действие (рис. 2.46).

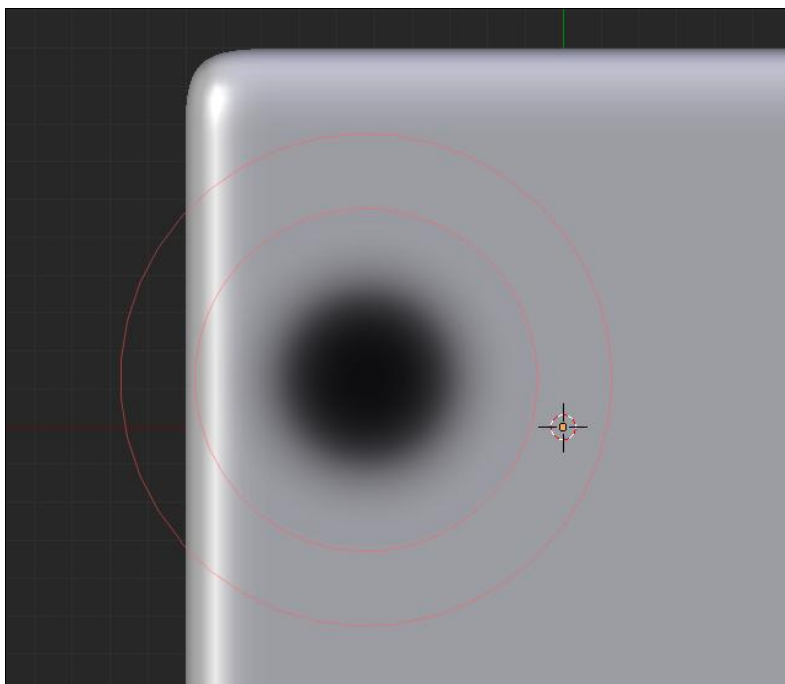


Рис. 2.46. Интерактивное изменение радиуса

Аналогичным образом изменяется и сила нажатия. Для этого используется комбинация клавиш <Shift>+<F>. Причем, чем меньше радиус новой окружности, тем выше значение параметра **Strength**.

Чтобы выбрать тип прорисовки кисти, нужно нажать клавишу <A>. Появится меню со знакомыми уже пунктами: **Drag**, **Drag Dot**, **Space**, **Anchored**, **Airbrush**.

СОВЕТ

Работа со скульптурным моделированием ложится тяжким бременем на систему из-за большого количества полигонов. Разработчики Blender подготовили механизм, несколько облегчающий навигацию в окне **3D View**. Так, можно установить меньшее значение параметра **Sculpt** на панели модификатора **Multires**. Но это решение не оптимально, ведь в реальности окно будет отображать достаточно приблизительную картинку. Есть способ гораздо удобнее. Включите опцию **Fast Navigate** (Быстрая навигация) на панели **Tool Shelf** (группа **Options**) и при вращении сцены Blender автоматически сбросит разрешение объекта для быстрого просмотра.

2.7. Практика. Модель веера

Иногда смотришь на предмет и думаешь, а как перенести его в 3D, да чтоб побыстрее и покачественнее? К примеру, возьмем обычный веер. Простая, несложная безделушка, состоящая из нескольких одинаковых пластин, скрепленных в основании. Первое, что приходит в голову — это создать оригинальную пластинку, размножить ее, совместить дубликаты и немного повернуть под определенным углом. Вот только мороки с этим будет...

Разработчики Blender озаботились созданием многочисленных инструментов, способных значительно облегчить рутинный труд моделлеров. В этом уроке вы познакомитесь с модификатором **Array** (Массив), который как раз и помогает размножить заготовку в любом количестве. Посмотрите на рис. 2.47. Нечто подобное вы сможете сделать буквально за пять минут. Не верите?

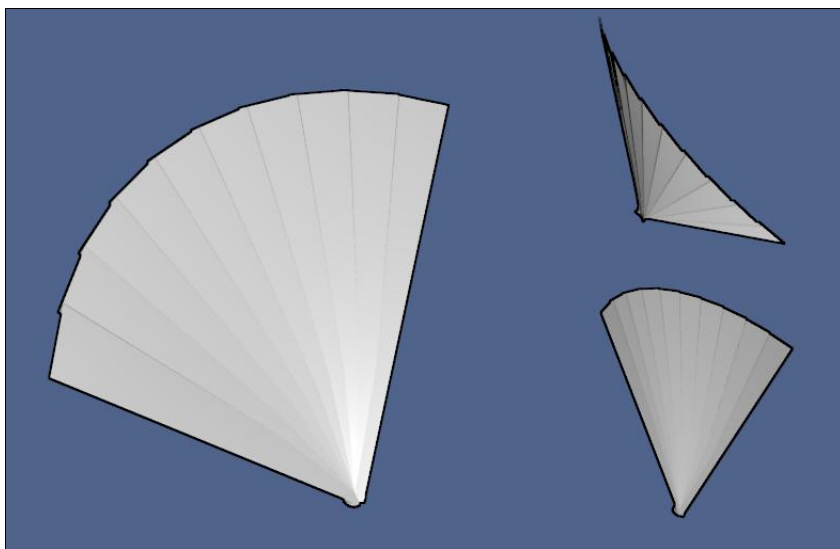


Рис. 2.47. Веер, созданный при помощи **Array**

Создайте новый проект и удалите имеющийся в сцене куб. В качестве заготовки для пластинки подойдет примитив **Plane** (Плоскость). Переключитесь в вид сверху (<NumPad 7>) и добавьте плоскость (**Add | Mesh | Plane**).

Теперь этому примитиву нужно придать форму, как на рис. 2.48. Для этого выполните следующие шаги:

1. Нажмите клавишу <S>, а затем <Y>, чтобы растянуть плоскость по координате Y.
2. Перейдите в режим редактирования (<Tab>) и нажмите клавишу <A> для снятия выделения.
3. Выделите две нижние вершины либо рамкой (), либо с помощью клавиши <Shift>.
4. Нажмите клавишу <S> для включения масштабирования и немного сожмите.

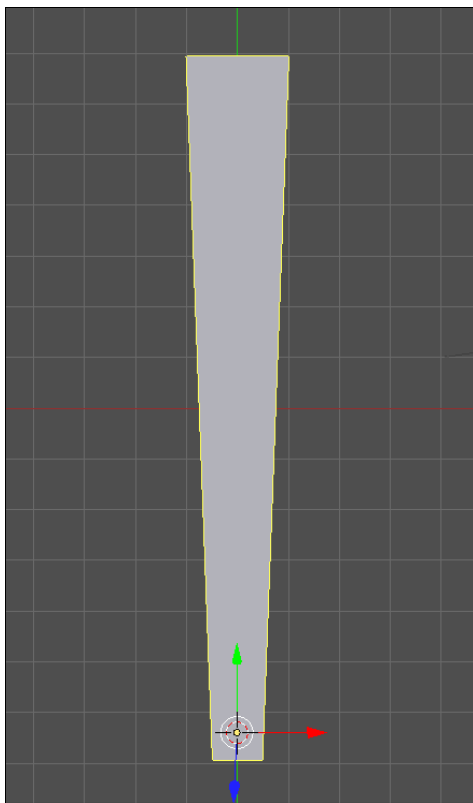


Рис. 2.48. Заготовка пластины для веера

Присмотритесь внимательнее к рис. 2.48. Центр объекта там смещен в нижнюю часть примитива, ведь именно в этой точке будут соединяться дубликаты. В вашем случае, центр находится в середине плоскости.

Для смещения его выполните следующее:

1. Выйдите из режима редактирования (<Tab>) и установите **3D Cursor** в нужном месте объекта, щелкнув левой кнопкой мыши.
2. Выберите пункт меню **Object | Transform | Origin to 3D Cursor** (Объект | Трансформация | Оригинал к 3D Cursor).

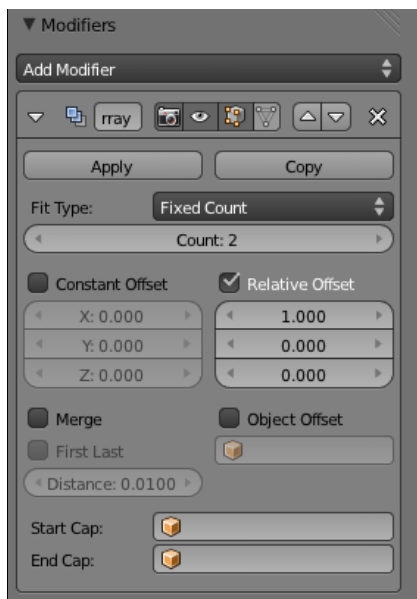


Рис. 2.49. Панель модификатора **Array**

Теперь добавьте к примитиву модификатор **Array** в окне **Properties** (рис. 2.49).

Рассмотрим его параметры.

- ◆ Меню **Fit Type** (Выбор типа) позволяет выбрать способ создания массива, где:
 - **Fixed Count** (Фиксированное количество) — нужное значение копии устанавливается в нижнем поле **Count**;
 - **Fit Length** (По длине) — программа рассчитывает количество дубликатов исходя из размеров образца и значения указанного в поле **Length**;
 - **Fit Curve** (По кривой) — если выбрать в поле **Curve** примитив кривой, то копии будут расположены по ее траектории.
- ◆ **Constant Offset** (Постоянное смещение) и **Relative Offset** (Относительное смещение). Указание смещения для копий по XYZ относительно оригинала или с учетом масштабирования.
- ◆ **Merge** (Слияние). При включении этой опции вершины копий при близком расположении будут объединяться, т. е. можно получить сплошной объект.
- ◆ **First Last** (Первый и последний). Если включена, то первый и последний объекты будут объединены (расстояние для слияния указывается в поле **Distance** (Дистанция)). Таким образом, объект получится полностью бесшовным.
- ◆ **Object Offset** (Относительно объекта). Копии выстраиваются в зависимости от указанного в данном поле стороннего объекта.
- ◆ **Start Cap** (Начальная часть) и **End Cap** (Конечная часть). Здесь можно выбрать объекты, которые будут установлены в начале и в конце массива.

Модификатор **Array** (Массив) с успехом может создать нужное количество копий по любым координатным осям. Вот только скрепить их в одной точке и повернуть

на определенный градус ему не по силам. Точнее, для этого действия понадобится вспомогательный объект, установленный в поле **Object Offset**.

В качестве такого объекта можно использовать любой примитив, но лучше **Empty** (Пустышка). По сути дела, **Empty** — это вспомогательный объект, к которому допустимы все возможные манипуляции, но не отражаемый в результатах рендера. Добавьте его из меню **Add | Empty**.

На панели модификатора включите опцию **Object Offset**, а в поле ниже выберите из меню **Empty**.

Результат может обескуражить, ведь пропорции копии получились искаженными. Исправить положение поможет функция **Object | Apply | Scale** для примитива **Plane**.

Теперь при вращении и перемещении **Empty** копия будет послушно следовать за вспомогательным объектом. Переместите **Empty** влево так, чтобы оригинал и копия совместились, и немного поверните его по координате Z (клавиша <R>, а затем <Z>). Выделите плоскость и увеличьте счетчик **Count** до 10 (рис. 2.50).

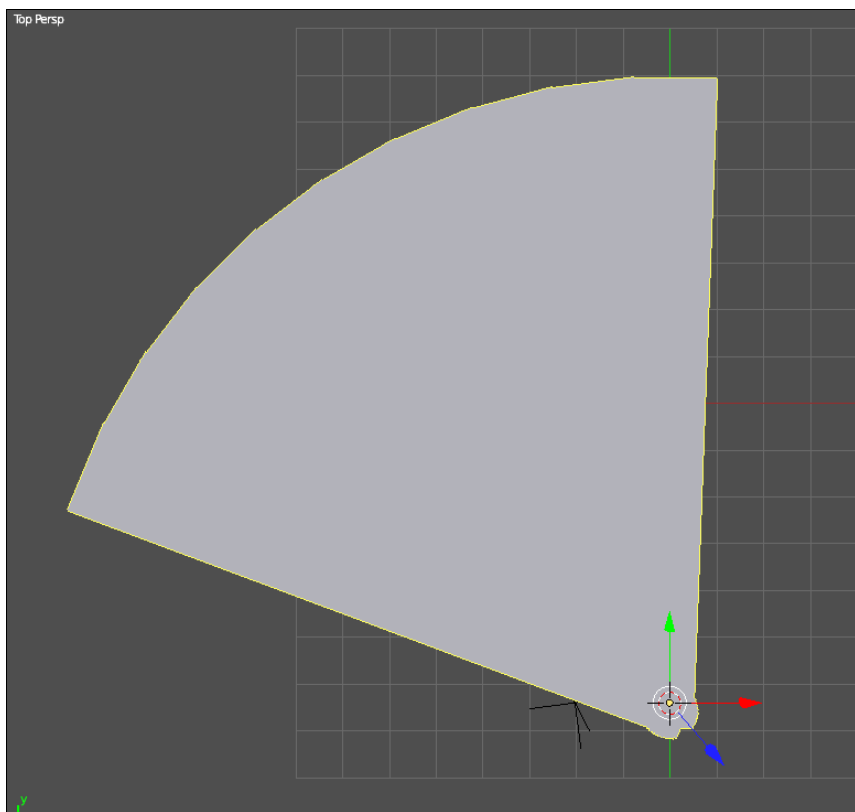


Рис. 2.50. Уже немного похоже на веер

Все хорошо, вот только пластинки веера слишком близко прилегают друг к другу и он выглядит излишне плоским. Это также можно устранить путем вращения

Empty, но уже по другим координатам. Выделите **Empty** и перейдите в просмотр **Front** (<NumPad 1>). Сейчас нужно немного развернуть вспомогательный объект по координате Y (рис. 2.51).

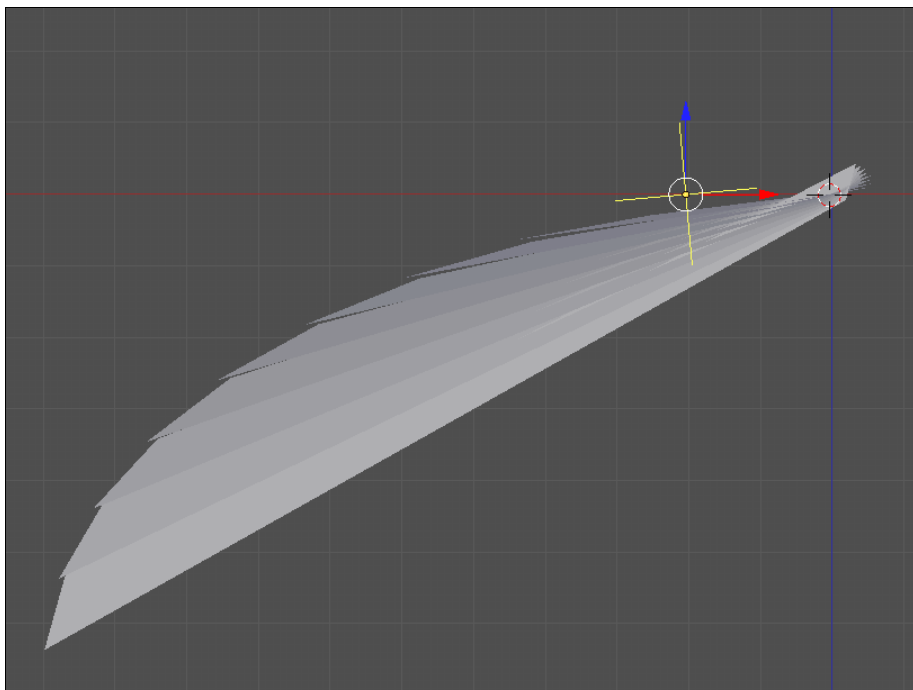


Рис. 2.51. Вращение **Empty** по координате Y придаст округлую форму и слегка раздвинет плоскости

В окне просмотра **Top** возможно вы увидите небольшой артефакт, связанный с пересечением плоскостей в основании. Это можно подправить путем перемещения **Empty** по координате X, а так работа над моделью завершена.

2.8. Практика. Кусочек сыра

Этот урок посвящен работе с булевыми операциями. С одной стороны, уже рассказывалось, как ими пользоваться, с другой стороны, получаемый на практике результат зачастую обескураживает. Бывает, что геометрия сложного **Mesh**-объекта портится. Борьаться с полученными искажениями от булевых операций сложно, но все же имеется ряд инструментов, способных исправить ситуацию. Мы создадим кусок круглого крупноячеистого сыра и будем отрабатывать на этом объекте приемы высокополигонального моделирования.

В качестве основного примитива используем **Circle** (Окружность), который можно найти в группе **Mesh**. По умолчанию этот объект создается с количеством вершин равным 32, нам же нужно получить треугольник. Добавьте объект в сцену и на панели **Tool Shelf** на закладке **Add Circle** установите значение **Vertices** = 3. Таким

образом, получится обычный равнобедренный треугольник. Щелкните опцию **Fill** для включения заливки примитива.

Для большей схожести с куском сыра объект нужно немного удлинить. Это можно сделать в режиме редактирования. Перейдите в этот режим с помощью клавиши <Tab>, переключитесь в вид **Top** (<NumPad 7>) и, выделив одну из вершин, немного перетяните ее в сторону.

У объекта имеется ненужная вершина, которая расположена в центре. Если попытаться просто ее удалить, то произойдет разрушение видимой грани. Поэтому можно объединить центральную точку с какой-нибудь вершиной треугольника (рис. 2.52). Для этого воспользуемся командой **Merge** (Объединить):

1. Снимите выделение с помощью клавиши <A>.
2. Выделите сначала центральную вершину, затем, удерживая клавишу <Shift>, одну из боковых.
3. Нажмите клавишу <W> и в появившемся меню **Specials** выберите пункт **Merge**.
4. Появится дополнительное подменю, где нужно выбрать пункт **At Last** (К последней).

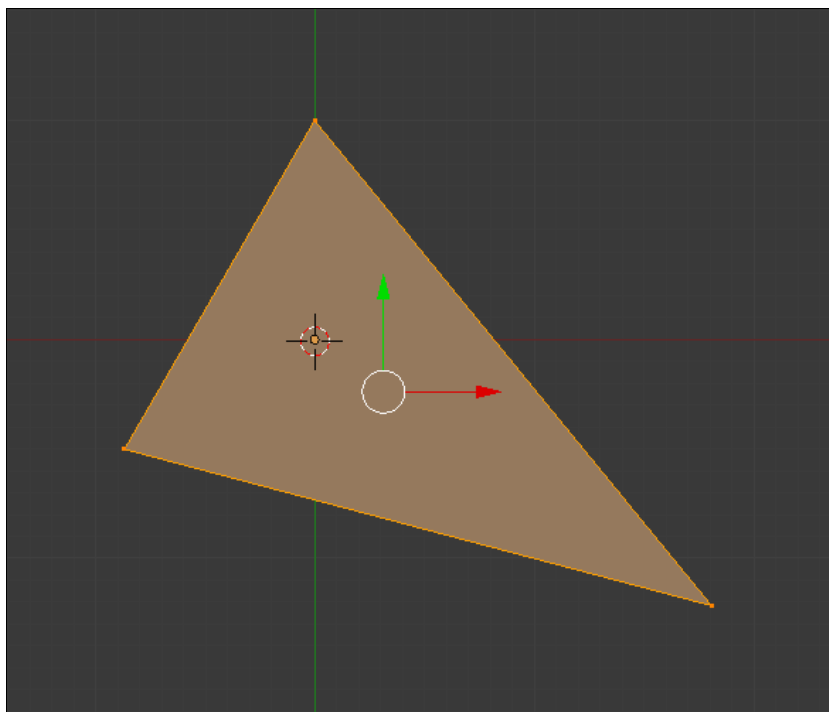


Рис. 2.52. Двухмерная заготовка для модели

Объект **Circle** является плоским. Для придания объема можно использовать уже привычную команду **Extrude**. Это также выполняется в режиме редактирования.

Перейдите для удобства в просмотр **Front** (<NumPad 1>) и выделите все вершины. Нажмите клавишу <E> для включения режима выдавливания. Затем немного приподнимите новую плоскость.

Итак, получился уже объемный треугольник. Вот только выглядит он слишком уж строго. Снимите выделение и выделите четыре вершины с тыльной стороны. Нажмите клавишу <S>, затем <Z> и немного растяните примитив по вертикали (рис. 2.53).

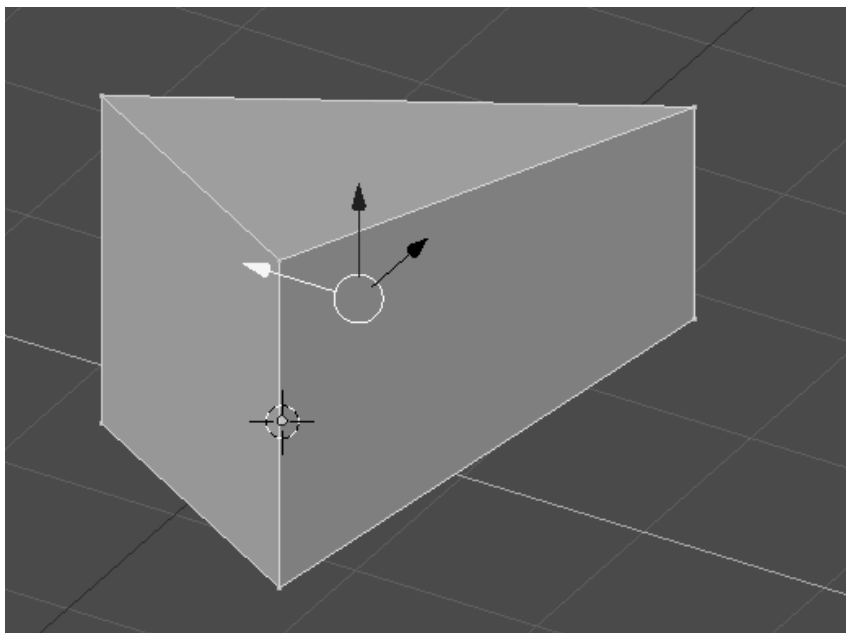


Рис. 2.53. Трехмерная заготовка для модели

Отлично, заготовка готова. Теперь поговорим о создании отверстий в нашем треугольнике.

Настоящий ячеистый сыр имеет большое количество неровных отверстий различного диаметра, что наводит на мысль использовать примитивы сфер для их создания. Разумеется, мы не будем выдавливать отверстия с помощью одной лишь сферы, а создадим массив таких объектов разного радиуса. Вот только воспользоваться ранее рассмотренным модификатором **Array** не удастся. К сожалению, он не умеет генерировать массивы объектов со случайными характеристиками. Придется сделать это вручную.

Перейдите в режим **Object Mode** и добавьте где-нибудь в сцене примитив **UV Sphere**. Задача простая, нужно размножить данный объект в хаотичном порядке и с различными радиусами на площади несколько превышающей заготовку модели. Это делается обычной командой дублирования **Duplicate Objects** на панели **Tool Shelf** или с помощью горячих клавиш <Shift>+<D>. Должно получиться нечто напоминающее рис. 2.54. Для удобства манипулирования их лучше объединить в один объект с помощью функции **Join** (<Ctrl>+<J>).

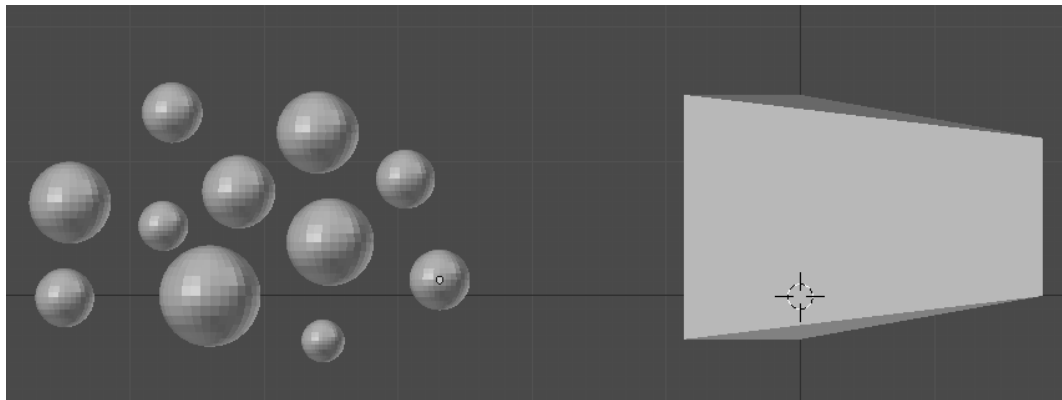


Рис. 2.54. Массив сфер для выдавливания

Вернемся к треугольной заготовке. Если предположить, что наш кусочек был вырезан из круглого цельного сыра, то основа его должна выглядеть несколько закругленной. Сделать это несложно, достаточно в режиме пропорционального редактирования немного вытянуть центр основы.

Выделите заготовку и перейдите в режим редактирования. Нам необходимо увеличить количество элементов структуры объекта. Нажмите клавишу <A> для выделения всего **Mesh**-объекта и выполните команду **Subdivide** (находится в меню, вызываемом горячей клавишей <W>). Такую операцию разбивки нужно выполнить четырежды.

Разверните объект тыльной стороной и выделите центральные точки по вертикали (см. рис. 2.55).

Перейдите для просмотра в вид **Top View** (<NumPad 7>). Включите режим пропорционального редактирования клавишей <O> и немного передвиньте (<G>) выделение в сторону. Степень округлости можно регулировать колесиком мыши (рис. 2.55)

Сейчас предстоит заняться созданием отверстий в модели с помощью булевых операций. Но давайте немного отвлечемся и поговорим вообще о работе модификатора **Boolean**.

Взгляните на рис. 2.56, где представлен результат работы **Boolean** на треугольной заготовке без дополнительной разбивки ее структуры.

Нарушение геометрии налицо. А если таких сфер было задействовано множество? Можно смело предположить, что такая модель окажется никуда негодной. Особенно искажения будут видны после применения стандартного сглаживания **Smooth**.

Можно несколько улучшить внешний вид модели после булевой операции, если первоначально увеличить разрешение структуры объекта. Это и понятно, ведь алгоритм **Boolean** будет работать только с близлежащими элементами. Но тут важно чувство меры. Чем больше разбита решетка объекта, тем медленнее и тяжелее работать с моделью. Кроме того, при излишней перегруженности может вообще произойти программный сбой.

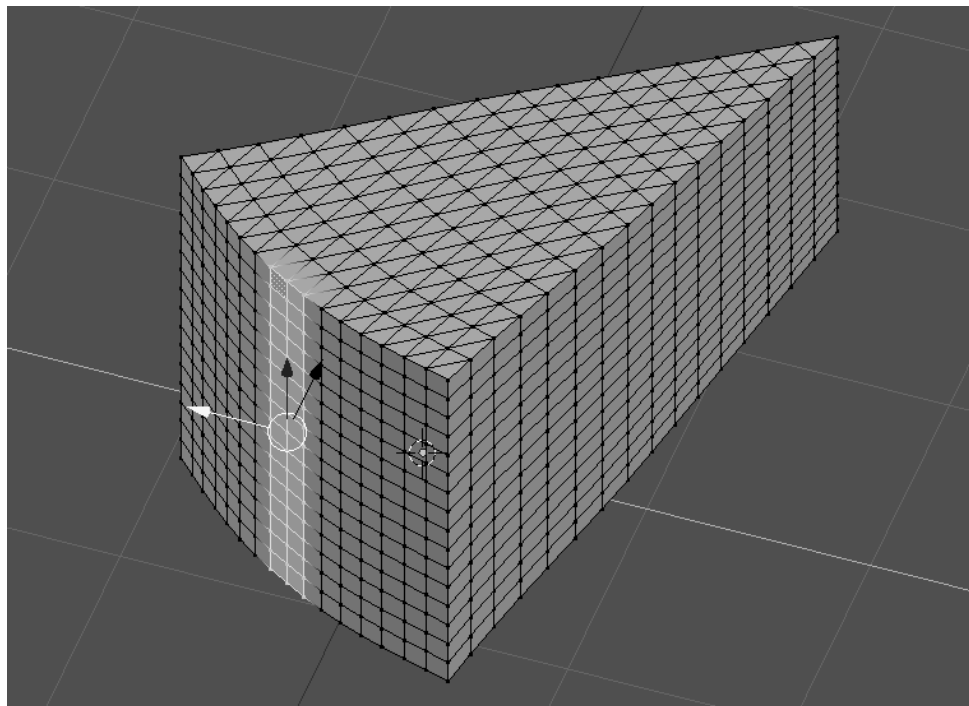


Рис. 2.55. Доработка тыльной стороны

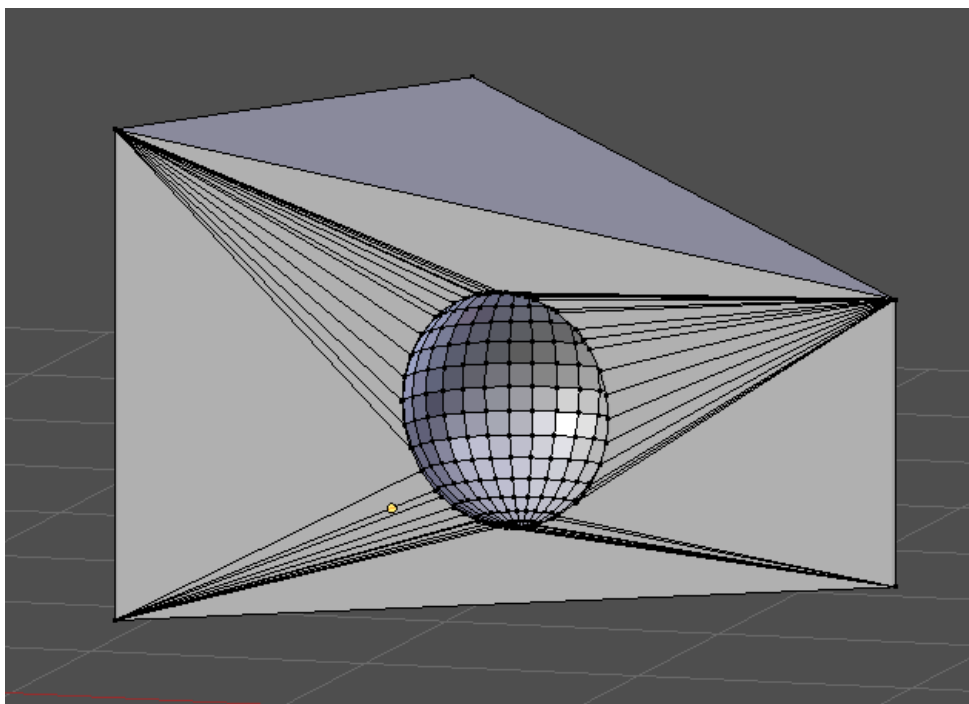


Рис. 2.56. Нарушения геометрии после применения **Boolean**

Есть несколько методов борьбы с этой напастью.

- ◆ *Ручное удаление лишних элементов.* Для относительно несложных объектов такой способ подойдет, но и только.
- ◆ *Воспользоваться модификатором **Decimate** (Упрощение).* Этот инструмент предназначен для регулирования количества элементов в объекте. Работа с ним осуществляется в режиме **Object Mode**. Имеется всего одно поле **Ratio** (Пропорция), где можно установить степень упрощения. Вот только если геометрия значительно нарушена, то модификатор выдаст сообщение об ошибке: "Non-manifold mesh as input". Это означает, что где-то в структуре **Mesh**-объекта произошла путаница среди элементов. В Blender есть замечательная функция нахождения и выделения таких испорченных сегментов. Находится она в окне **3D View** в меню **Select | Non-manifold**. После ее вызова проблемные места в решетке будут выделены. В зависимости от ситуации, лишние элементы удаляются или объединяются с помощью **Merge**. Вот только **Decimate** вряд ли подойдет для высокополигональных объектов.
- ◆ *Устранить нарушения в скульптурном режиме.* Оптимально для высокополигональных моделей. Именно этим способом мы в дальнейшем воспользуемся.

Теперь вернемся к нашему уроку. Воспользуемся приведенным ранее советом и еще увеличим разбивку структуры заготовки. Выделите весь объект (<A>) и сделайте **Subdivide**.

Займемся созданием отверстий. Совместите массив сфер и заготовку так, чтобы первые покрывали одну из сторон треугольника и наполовину входили в его поверхность (рис. 2.57).

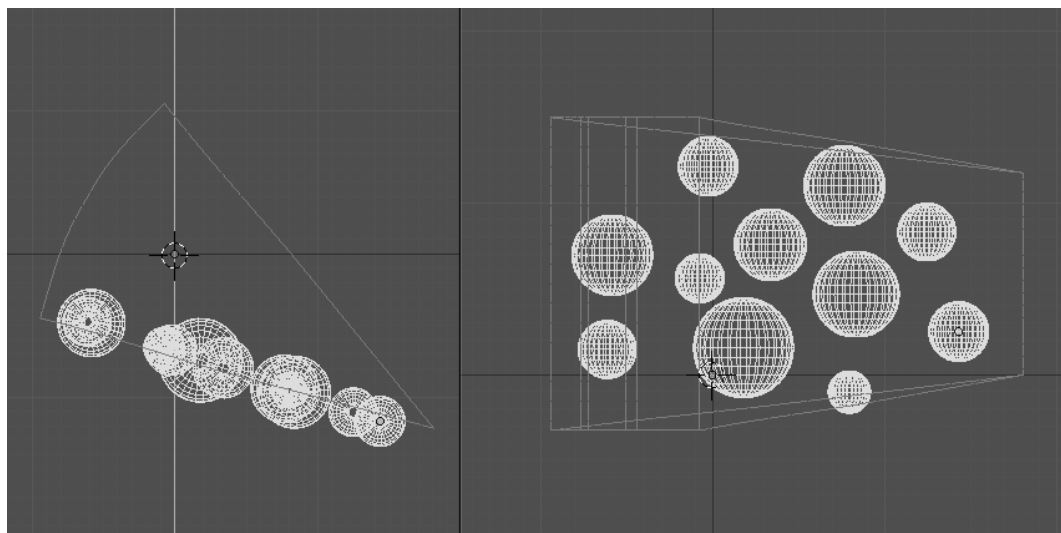


Рис. 2.57. Совмещение объектов для булевой операции

Добавьте модификатор **Boolean** для заготовки. В его настройках выберите метод **Difference**, а в поле **Object** укажите массив сфер. Возможно на некоторое время

компьютер "задумается", так что наберитесь терпения. После завершения просчета нажмите кнопку **Apply** на панели модификатора для его применения. Выполните эти действия для каждой стороны заготовки, кроме тыльной (рис. 2.58).

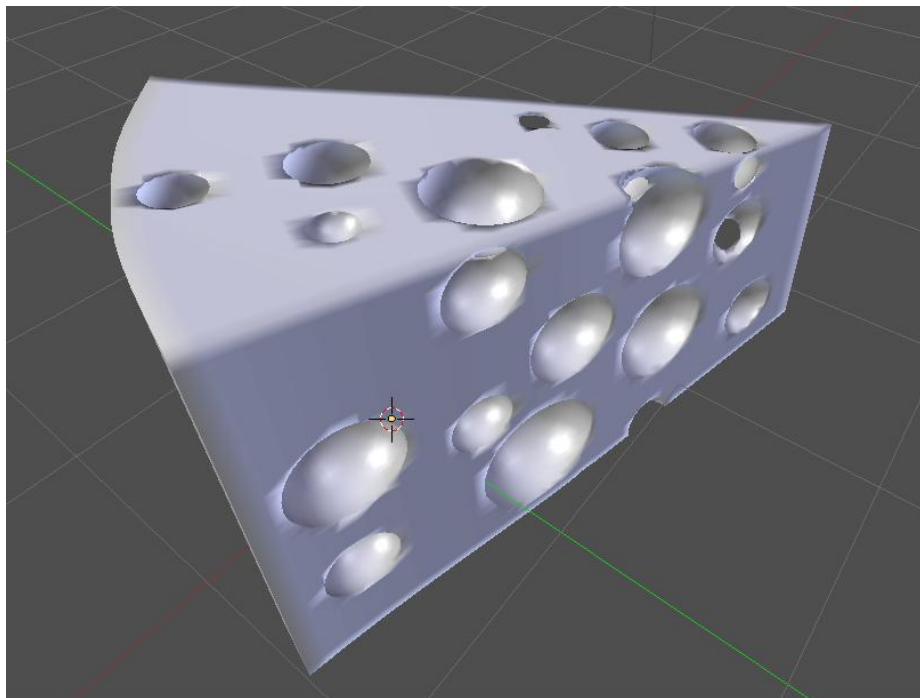


Рис. 2.58. Заготовка с проделанными отверстиями

Для модели на рис. 2.58 было применено общее сглаживание **Smooth**. Очень хорошо видно, что края отверстий выглядят рваными со странными темными пятнами. Это как раз и есть результат нарушения геометрии объекта.

Отверстий здесь много, поэтому способ скрупулезного расплетения запутанных вершин использовать нерентабельно. Зато эти самые огрехи можно подправить в режиме скульптурного моделирования. Выберите пункт **Sculpt Mode** в меню **Mode**.

Наша модель уже имеет достаточную разбивку своей структуры. Если есть желание, то можете добавить дополнительный уровень с помощью модификатора **Multires**.

При рассмотрении краев объекта в режиме редактирования отчетливо видно, что в этих местах к вершинам прикреплено слишком много ребер. Хотя визуально данные грани лежат на одной линии с плоскостью объекта. Поэтому оптимальным будет выбор кисти с действием стяжки. С помощью нее мы просто подтянем края отверстий к дефектным участкам и тем самым уберем видимую область искажений. Заодно этим действием немного деформируем формы наших слишком идеальных отверстий.

Такая кисть в наборе Blender имеется и называется **Pinch/Magnify** (Стяжка/Расширение). Выберите ее из списка кистей на панели **Tool Shelf**.

Разверните модель так, чтобы рабочая сторона находилась к вам лицом. Если при этом действии система тормозит, то установите галочку в пункте **Fast Navigate** (Быстрая навигация) в группе **Options** в настройках кисти.

Задача состоит в следующем. Сначала нужно подобрать оптимальный размер кисти (**Radius**), ведь обрабатывать придется каждое отверстие отдельно. Это вы сможете сделать только опытным путем. И все же советую установить силу нажатия (**Strength**) на максимальную — дело пойдет быстрее. Центр курсора совмещаете с краем отверстия и круговыми движениями зачищаете поверхность. При этом края отверстий могут немного деформироваться (рис. 2.59). Таким способом нужно обработать все видимые края.

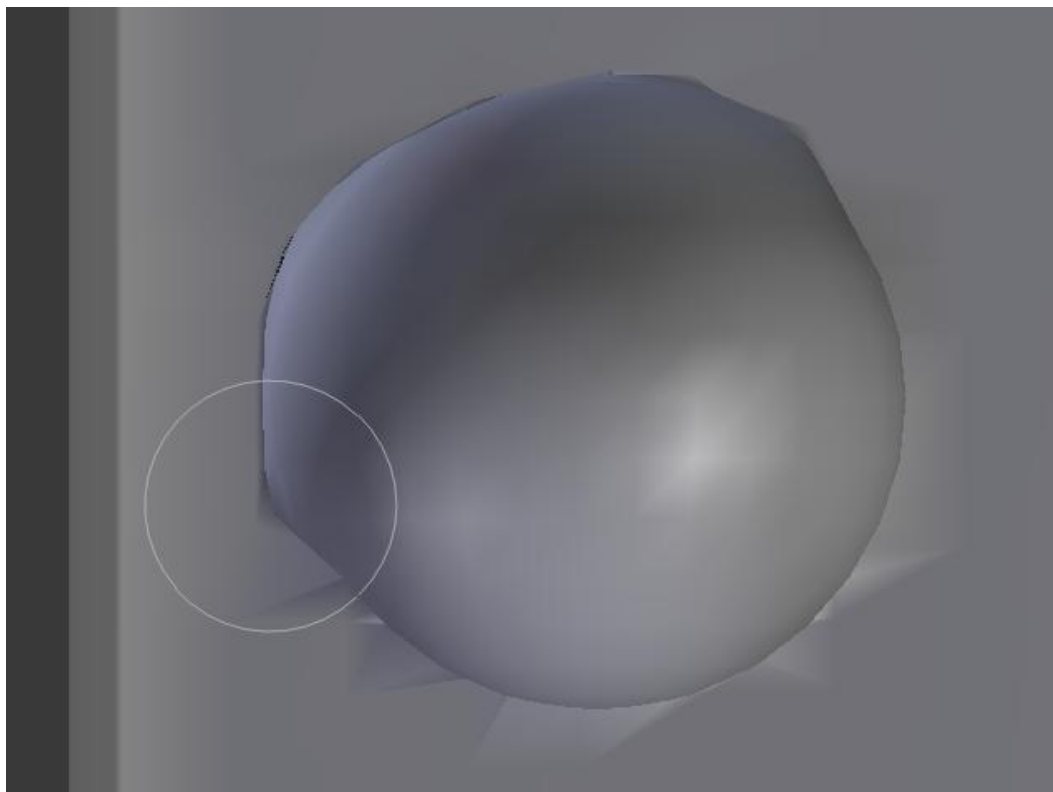


Рис. 2.59. Обработка отверстия кистью

Добавьте немного деформации для сторон модели с помощью кисти **Brush** большого диаметра. Их можно делать как в режиме **Add**, так и **Subtract**. Попробуйте также создать укус на острие модели. Для этой цели может подойти кисть **Blob** в режиме **Subtract**. Разверните объект острием к себе и выдавите курсором область. В общем, все остальное на ваше усмотрение и фантазию (рис. 2.60).

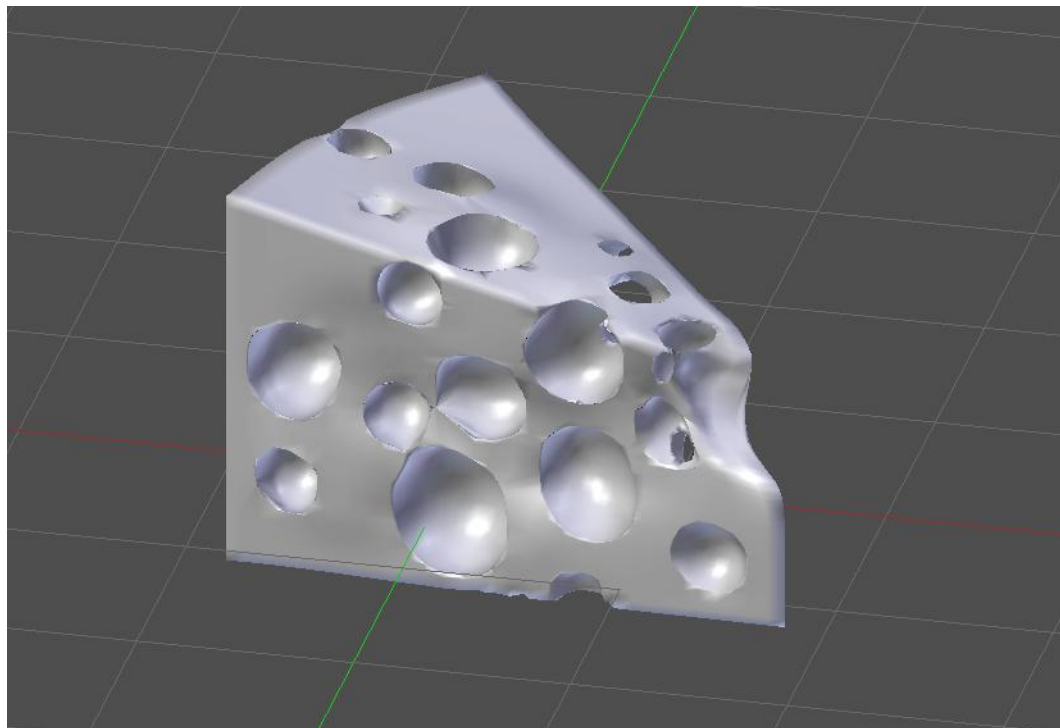
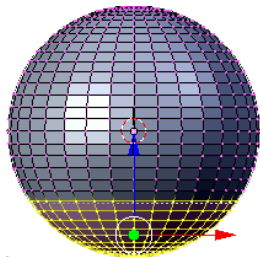


Рис. 2.60. Модель сыра

СОВЕТ

Вы можете обработать сцену с помощью горячей клавиши <F12>. Запомните, рендер всегда осуществляется с позиции камеры. Так что при необходимости манипулируйте камерой, а не объектами. Если вдруг захотелось сохранить результат обработки, то в окне рендера нажмите клавишу <F3>, и программа предложит сохранить его в графическом формате.

ГЛАВА 3



Кривые, поверхности *NURBS*

Работа над моделью начинается с примитива и от того, какой тип был выбран, зависит качество и скорость исполнения. Blender предлагает большой набор объектов **Mesh**, но в некоторых случаях использование их выглядит нерациональным. Допустим, нужно сделать модель изящной вазы. Путем наложения различных модификаторов, использования соответствующих инструментов можно добиться хороших результатов. Правда время на создание модели будет неоправданно большим. А ведь в мире имеется немало объектов с обтекаемыми, плавными формами: лодки, музыкальные инструменты, игрушки, кухонная утварь. Для таких моделей стоит использовать совершенно иной тип примитивов, а именно кривые.

В этой главе вы узнаете, что такое кривые, мы поговорим о создании объектов с их помощью, познакомимся с поверхностями **NURBS**.

3.1. Основные понятия

Кривые или, как их еще называют, *сплайны* — это примитивы, создаваемые программой на основе определенных математических функций.

Mesh-объект имеет жесткую структуру, состоящую из вершин, ребер и плоскостей, данные о которых хранятся в полном и неизменном состоянии. В свою очередь, для создания сплайна программа выполняет определенные вычисления между контрольными точками. В итоге получается, что для хранения кривых тратится существенно меньше памяти, нежели для **Mesh**. Но это не главное свойство сплайнов.

Кривые и примитивы, созданные на их основе, обладают одним неоспоримым плюсом, а именно гибким и эффективным редактированием. В принципе, с помощью этих примитивов можно создавать объекты любой формы. Некоторые даже используют только их, игнорируя **Mesh**. Но важно уметь работать с разными подходами, чтобы впоследствии правильно определиться с выбором начального "кирпичика".

Blender предлагает две группы примитивов, основанных на использовании математических функций: **Curve** (Кривая) и **Surface** (Поверхность). Если первые представляют собой простые двухмерные объекты, то вторые — это полноценные,

замкнутые фигуры. Кроме того, они различаются по типу расчетной функции: **Bezier** (Безье) и **NURBS** (Неоднородный рациональный B-сплайн). Не будем касаться подробностей по способу их вычислений, важно знать, что редактирование примитивов на основе их несколько различается.

Работая с **Mesh**, мы оперировали такими терминами, как вершины, ребра и грани. А вот редактирование сплайнов осуществляется только с помощью вершин или, как их еще называют, контрольных точек.

Контрольные точки — это узлы кривой или поверхности, с помощью которых изменяется форма примитива. Эти вершины подчиняются общим правилам манипулирования. Их можно перемещать, вращать и масштабировать. В свою очередь контрольные точки кривых Безье имеют рычаги (рис. 3.1).

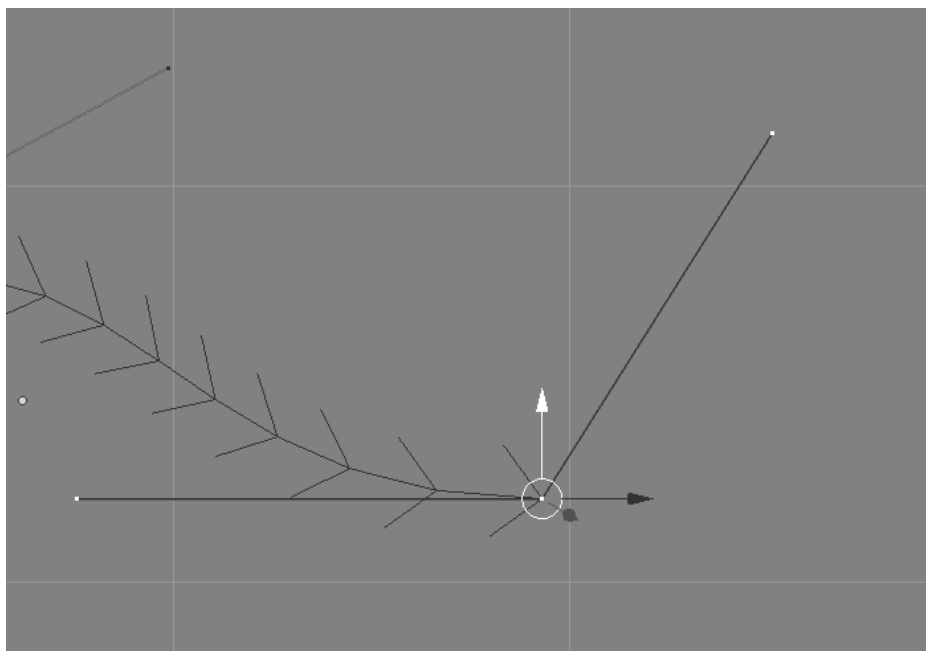


Рис. 3.1. Контрольная точка с рычагами

Рычаги — это элементы, позволяющие управлять изгибом кривой между контрольными точками. Каждая вершина снабжается двумя рычагами, которые ответственны за свой сегмент.

Таким образом, процесс редактирования сплайнов Безье заключается в размещении ключевых вершин в нужных местах и регулирования переходов между ними с помощью рычагов.

Blender предлагает несколько типов рычагов для разных нужд.

◆ **Automatic** (Автоматический). Используется по умолчанию. В этом режиме оба рычага взаимосвязаны и при изменении положения одного, равноценно меняются координаты второго.

- ♦ **Vector** (Векторный). Рычаг всегда смотрит в сторону следующей контрольной точки. Соответствующим образом изменяется и кривая. Этим способом можно добиться острых углов у сплайна.
- ♦ **Aligned** (Выровненный). В этом режиме рычаг устанавливается параллельно по отношению к соседнему рычагу.
- ♦ **Free** (Свободный). В отличие от остальных типов здесь рычаги полностью независимы друг от друга.

Заметьте, что применение типа возможно как для одного рычага в отдельности, так и к выделенной группе контрольных точек. Это позволяет по-разному конструировать форму кривой. К примеру, если с одной стороны от вершины нужен прямой сегмент, а с другой извилистый, то применяются типы **Vector** и **Free** к соответствующим рычагам.

Выбор типа рычага осуществляется в меню **Curve | Control Points | Set Handle Type** (Кривая | Контрольные точки | Установить тип рычага) или с помощью контекстного меню, вызываемого горячей клавишей <V>.

Добавить кривую или поверхность можно стандартным способом из меню **Add**. Blender предлагает пять примитивов для сплайнов и шесть — для **NURBS**.

Примитивы **Curve**:

- ♦ **Bezier** — простая кривая с двумя контрольными точками;
- ♦ **Circle** — замкнутая окружность с четырьмя управляющими вершинами;
- ♦ **Nurbs Curve** — простая кривая **NURBS** с 4 вершинами;
- ♦ **Nurbs Circle** — замкнутая окружность с восьмью контрольными точками;
- ♦ **Path** — вспомогательная кривая для создания траектории движения другого объекта.

Примитивы **Surface**:

- ♦ **NURBS Curve** (Кривая);
- ♦ **NURBS Circle** (Окружность);
- ♦ **NURBS Surface** (Поверхность);
- ♦ **NURBS Cylinder** (Цилиндр);
- ♦ **NURBS Sphere** (Сфера);
- ♦ **NURBS Torus** (Тор).

3.2. Простейшие операции со сплайнами

Если добавить в сцену кривую и попробовать обработать, то в итоге получите пустой экран. Дело в том, что сплайны Безье часто используются как вспомогательные объекты в анимации или при деформации **Mesh**. Для использования их в качестве зрительных объектов необходимо добавить глубину (**Depth**). Этот и иные параметры доступны на панели **Object Data** окна **Properties** (рис. 3.2).

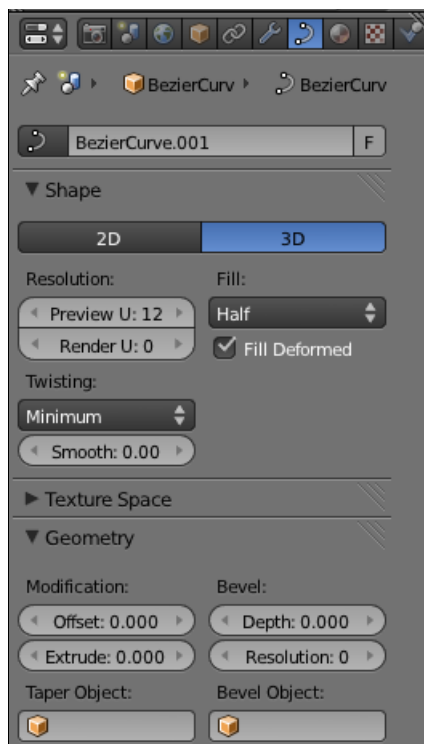


Рис. 3.2. Настройки кривой

Так как сплайн является объектом двухмерным, а мы все же работаем в 3D-пространстве, то разработчики озаботились возможностью переключения его из режима **2D** в **3D**. Переключение относится только к возможностям манипуляции. К примеру, если работа идет над созданием плоского логотипа, то удобнее включить режим **2D**. В этом случае не будет возможности перемещать вершины кривой по координате Z, что исключит случайное искажение формы. Причем это касается только редактирования. Перемещать по сцене кривую можно как заблагорассудится. Выбор режима осуществляется в настройках сплайна кнопками **2D** и **3D**.

ЭТО ВАЖНО!

Если, при редактировании кривой вершины объекта располагаются на разной высоте по Z, то переключение в режим **2D** выстроит все контрольные точки на одной плоскости.

Простейшая модель, которую можно сделать из кривой (конкретно имеется в виду примитив **Bezier**), — это волос, провод, труба и все остальное в таком роде. Blender предлагает два варианта создания объема у сплайнов: **Extrude** (Выдавливания) (рис. 3.3) и **Depth** (Глубина) (рис. 3.4). Обе опции находятся в закладке **Geometry** (Геометрия) окна **Properties** (см. рис. 3.2).

Если функция **Extrude** выдавливает объем по координате Z, то **Depth** производит это с "гранями" кривой. Слово "границы" не зря взято в кавычки, ведь по умолчанию у сплайнов их просто нет. Но если представить кривую как набор близко стоящих сфер, то можно понять, как **Depth** изменяет примитив.

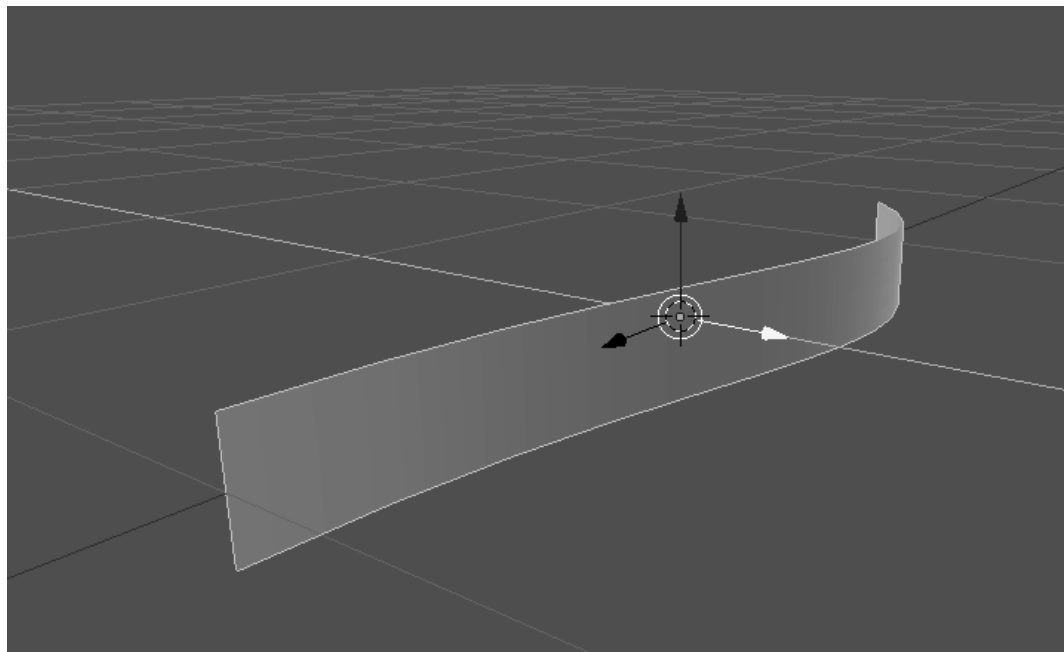


Рис. 3.3. Результат изменения Extrude

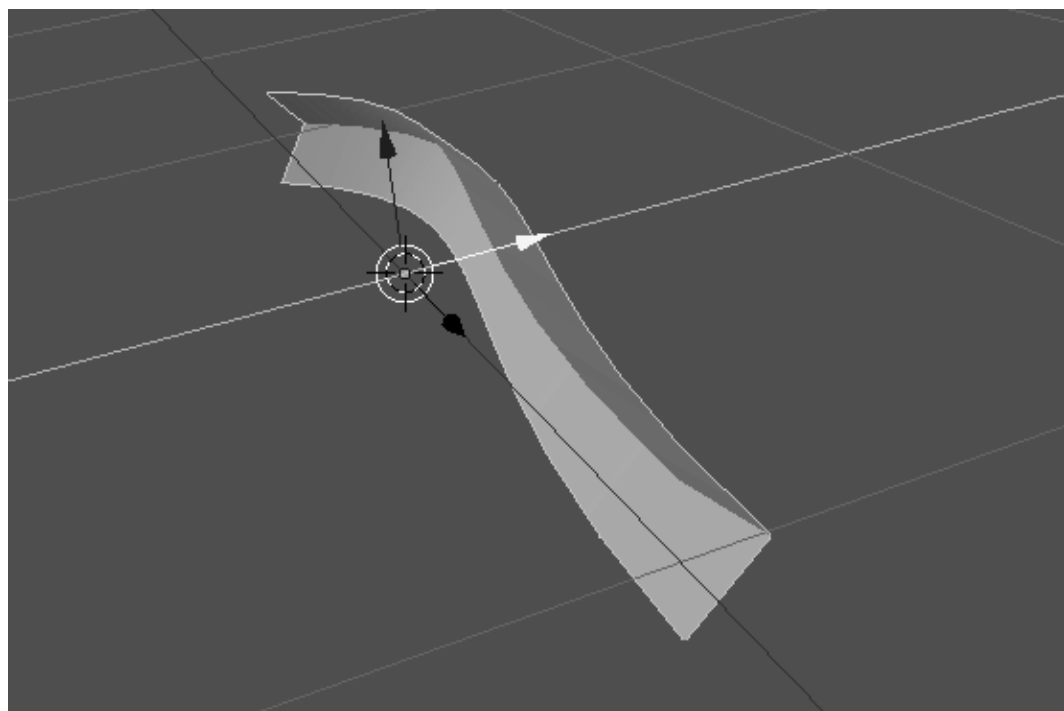


Рис. 3.4. Результат изменения Depth

Странный вид кривой после изменения **Depth** объясняется текущими атрибутами заливки. Возможные варианты находятся в группе **Fill** (см. рис. 3.2):

- ◆ **Half** (Половина);
- ◆ **Front** (Спереди);
- ◆ **Back** (Сзади);
- ◆ **Full** (Полностью);
- ◆ **Fill Deformed** (Заливка после деформации). Используется для корректировки после применения некоторых модификаторов или при анимации.

Для получения желаемого вида провода нужно выбрать в меню **Fill** пункт **Full** (рис. 3.5).

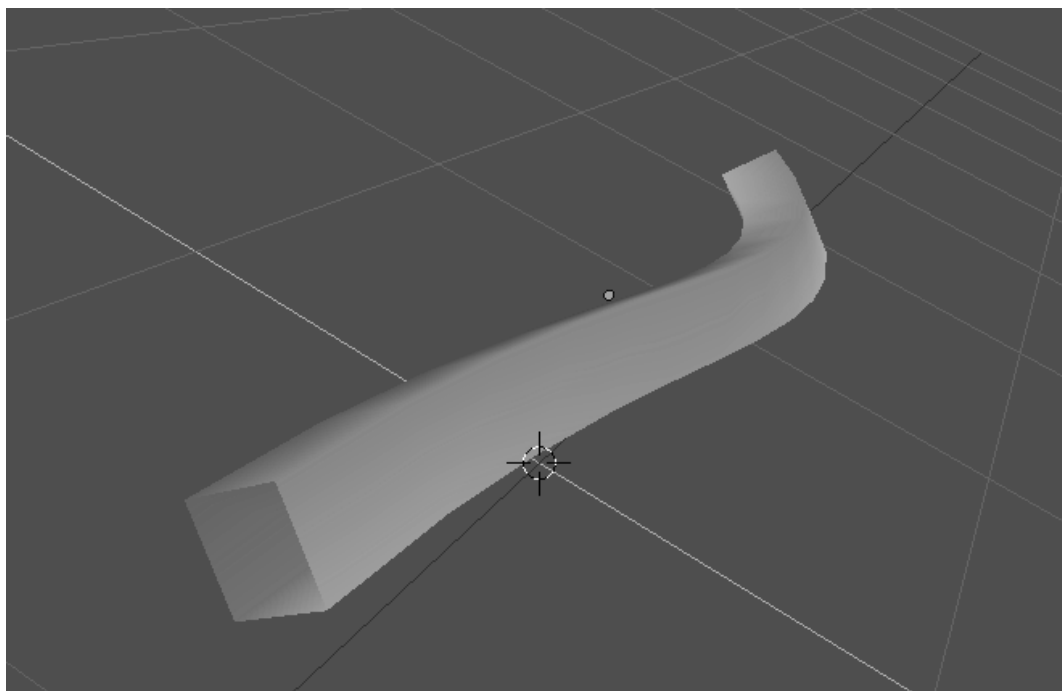


Рис. 3.5. Кривая после установки параметра **Fill**

Квадратность модели объясняется низким разрешением у кривой, которое стоит по умолчанию. Для изменения этого свойства имеется поле **Resolution** (Разрешение) в группе **Geometry**. Увеличение его приведет к нужному результату (рис. 3.6).

По сравнению с **Mesh** функции редактирования у сплайнов весьма небогатые, но их достаточно для воплощения любой задумки. Возьмем опять за образец примитив **Bezier**. По умолчанию он имеет всего две контрольные вершины. В некоторых случаях этого явно недостаточно. Самый простой способ добавления новой точки — это выполнить хорошо знакомую функцию **Subdivide**. В отличие от **Mesh**, желательно разбивать сегмент между двумя вершинами, а не по всему объекту. Делается

это просто: выделяются две смежные контрольные точки, между которыми нужна еще одна, нажимается клавиша <W> и выбирается пункт **Subdivide** (кнопка с таким же названием присутствует и в **Tool Shelf**). Удаление ненужной вершины выполняется с помощью стандартной команды **Delete** (<X>).

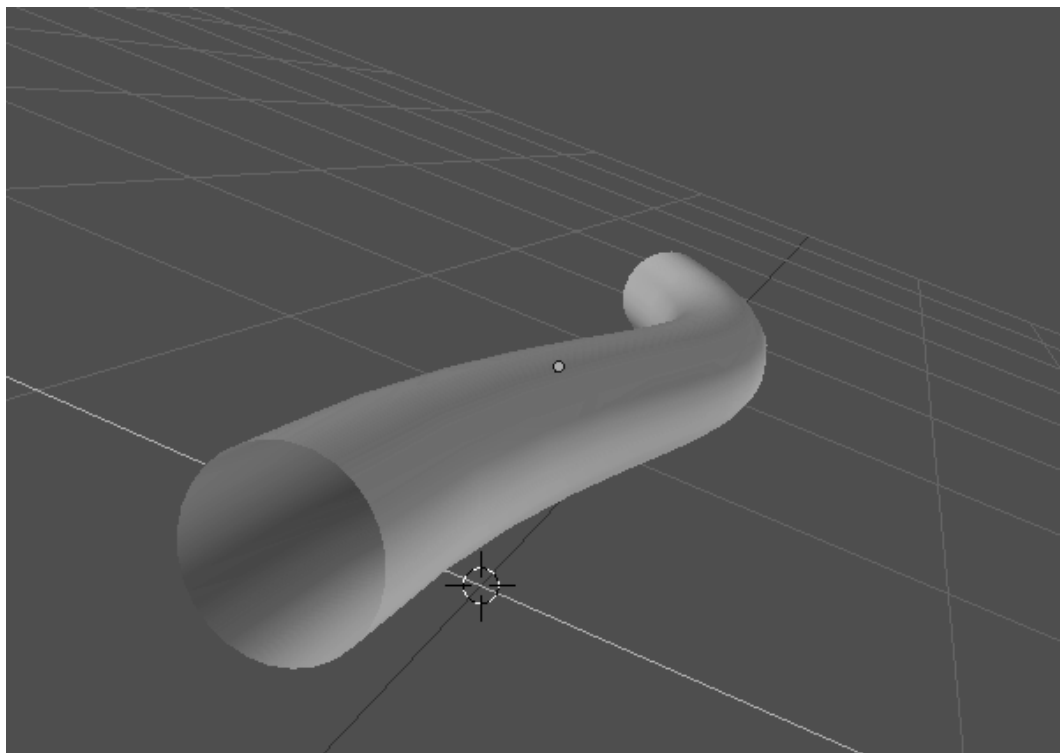


Рис. 3.6. Провод из сплайна

Можно также использовать команду **Extrude**. Применять ее следует к крайним вершинам. В этом случае будет создан новый сегмент с добавлением контрольной точки.

Иногда бывает нужно сделать кривую замкнутой. Такая функция в Blender имеется и называется **Toggle Cyclic** (Переключение зацикливания). Выполнить ее можно только в режиме редактирования. Смысл работы заключается в том, что между крайними вершинами прокладывается еще один сегмент и таким образом фигура замыкается. Причем интересно, что для ее выполнения необходимо выделить хотя бы одной контрольной точки, неважно какой. Вторичный вызов **Toggle Cyclic** вернет кривую в первоначальное состояние. Запомните еще одну особенность — результат работы этой команды различен для режимов **2D** и **3D**. В первом случае вы получите замкнутую кривую с заливкой, во втором — просто окружность (рис. 3.7).

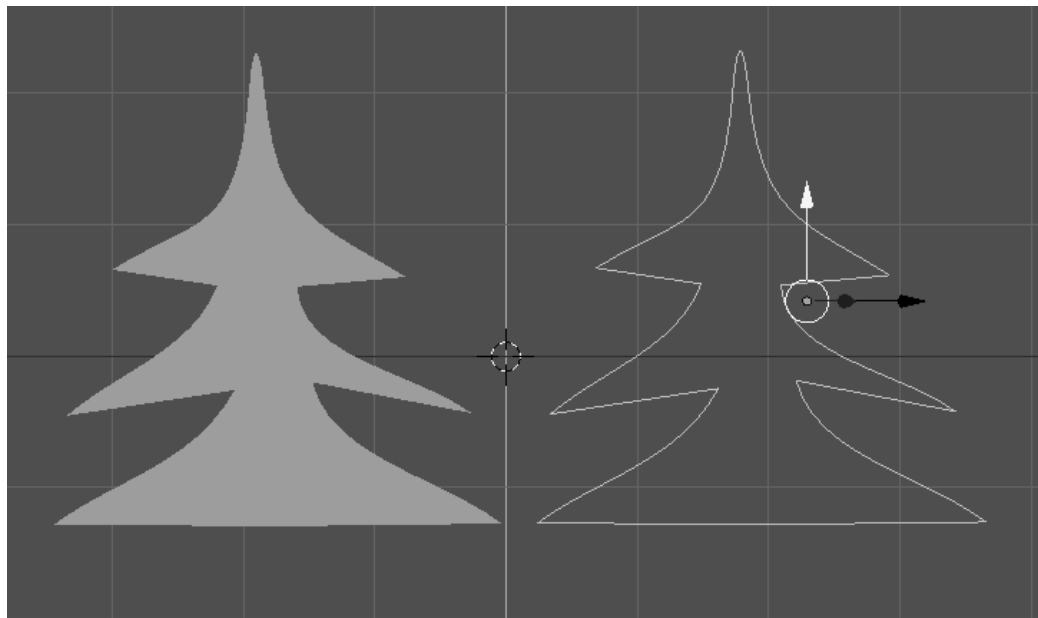


Рис. 3.7. Объект слева в 3D, справа — 2D

3.3. Деформация объектов с помощью кривой

Часто кривую используют как вспомогательный примитив для деформации иного объекта. Причем в качестве второго могут служить как сплайны, так и **Mesh**. С помощью этого способа можно с легкостью добиться поразительных успехов. Начнем сначала с деформации сплайнов.

Допустим, нужно деформировать примитив **Bezier Circle**. Как правило, в качестве вспомогательного объекта берут простую кривую **Bezier**, но ничто не мешает использовать любой сплайновый примитив.

На панели настроек **Bezier Circle** в группе **Geometry** есть два поля: **Taper Object** и **Bevel Object** (см. рис. 3.2). Именно там выбирается вспомогательный объект в сцене.

Как вы уже знаете, группа **Bevel** управляет созданием кромки. Если просто изменить параметр **Depth**, то окружность получит вид, как на рис. 3.8.

Теперь если создать в сцене примитив **Bezier** и выбрать его в поле **Bevel Object**, то окружность изменится в соответствии с формой вспомогательной кривой (рис. 3.9).

ЭТО ВАЖНО!

Если поле **Depth** у объекта имеет значение, отличное от нулевого, то выбор вспомогательной кривой в **Bevel Object** сбросит параметр **Depth**.

Управление деформацией объекта осуществляется путем манипулирования формы кривой в режиме редактирования. Кроме того, на конечный результат влияет и масштаб вспомогательного примитива.

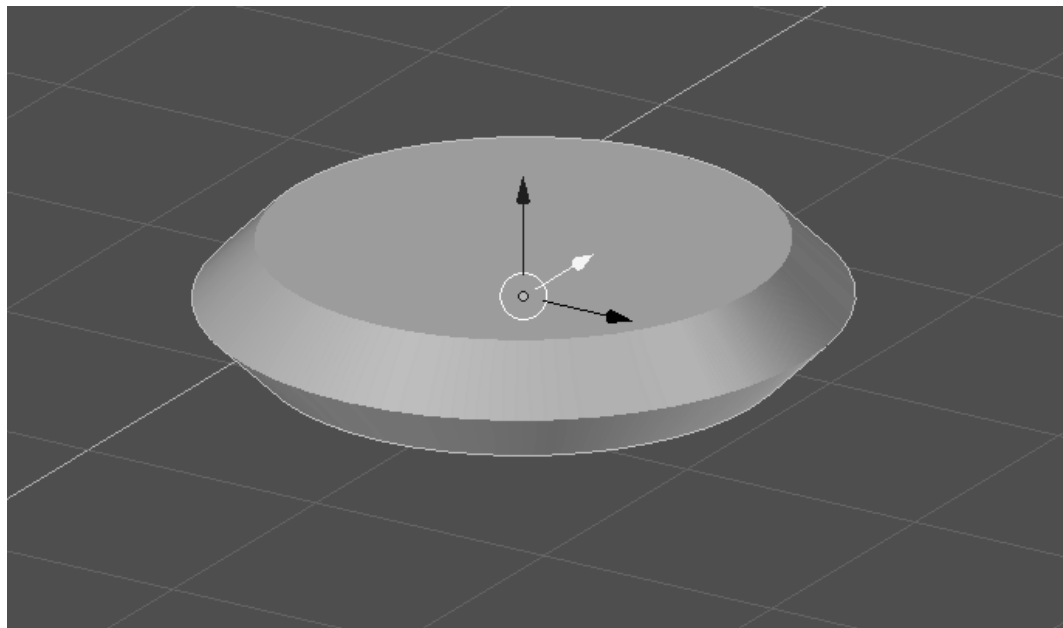


Рис. 3.8. Результат изменения **Depth** у примитива **Bezier Circle**

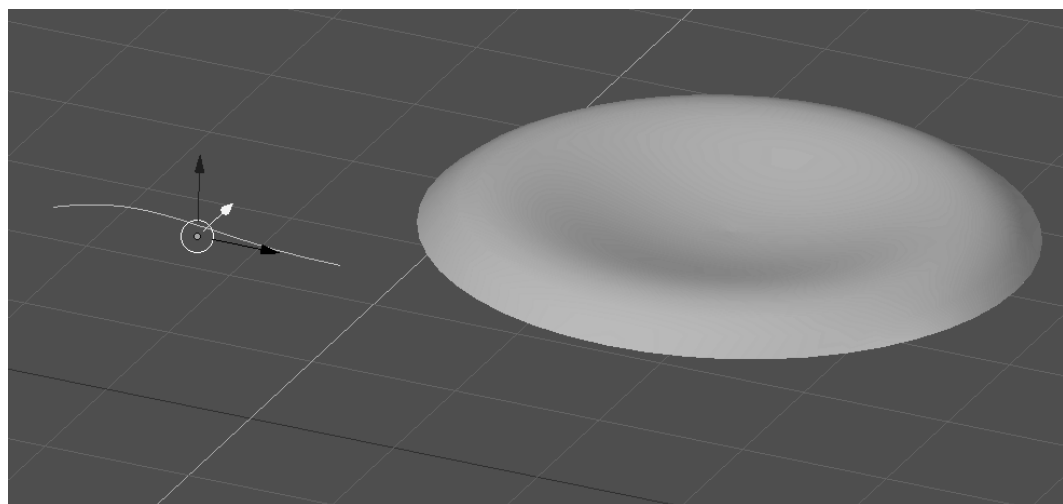


Рис. 3.9. **Bezier Circle** приняла форму кривой

Путем несложных манипуляций с вершинами кривой можно быстро получить приемлемый результат. Так, на рис. 3.10 изображена тарелка, созданная с помощью описанного способа. Причем основной объект **Bezier Circle** не подвергался никакому редактированию (рис. 3.10).

Слово "Taper" с английского можно перевести как "сужать". В принципе, это и происходит при активации опции **Taper Object**. Главное условие его использования — это наличие объема у исходного примитива. Возьмем, к примеру, две кри-

вые Безье, где одна будет вспомогательной, а другая основная. Установим для второго объекта параметр **Depth** равным 1, а **Extrude** — равным 0. Выравним обе кривые с помощью режима рычагов **Vector** и добавим в поле **Taper Object** вспомогательный примитив. В итоге получится ламповый абажур (рис. 3.11).

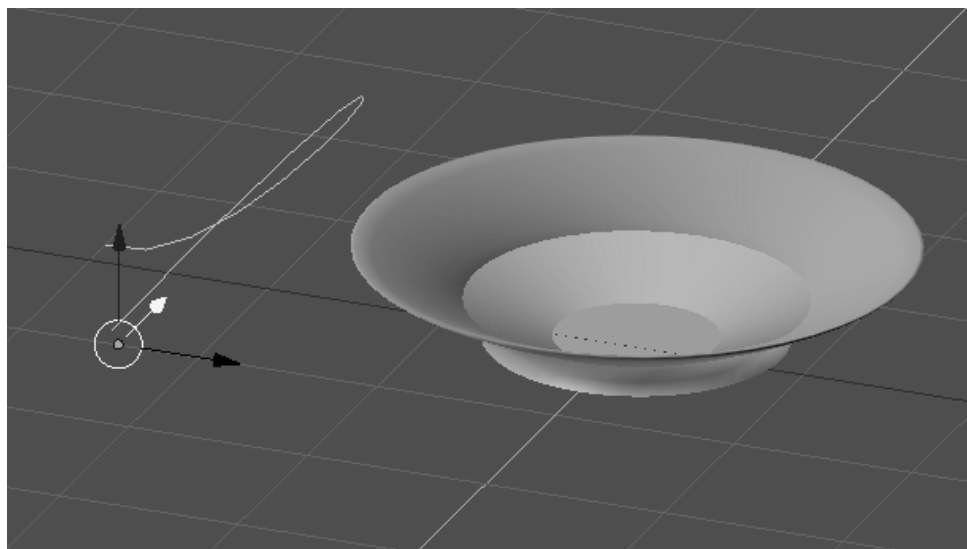


Рис. 3.10. Слева вспомогательная кривая, а справа деформированный примитив **Circle**

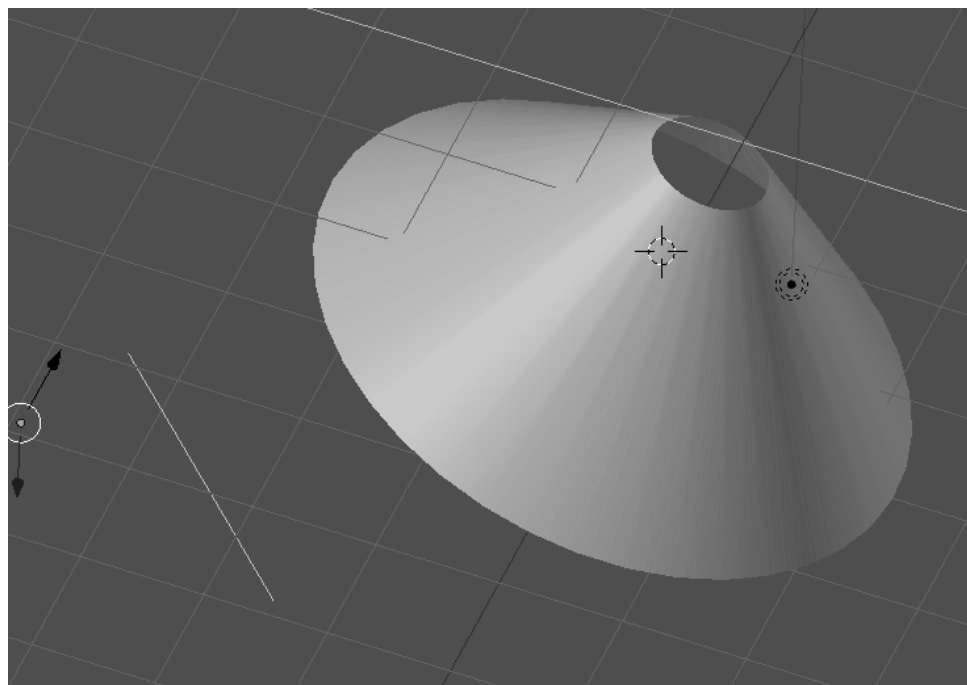


Рис. 3.11. Абжур из кривой Безье с помощью **Taper Object**

Ко всему сказанному нужно добавить, что кривые можно использовать для деформации объектов **Mesh**. Вот только для них существует специальный модификатор **Curve** (Кривая) (рис. 3.12).

Достаточно указать в поле **Object** нужную кривую и выбрать ось, вдоль которой будет происходить деформация. Принцип работы простой: изменяете форму кривой — меняется сам объект (рис. 3.13).

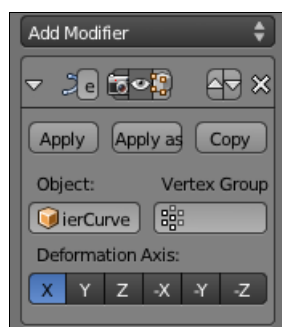


Рис. 3.12. Настройки модификатора **Curve**

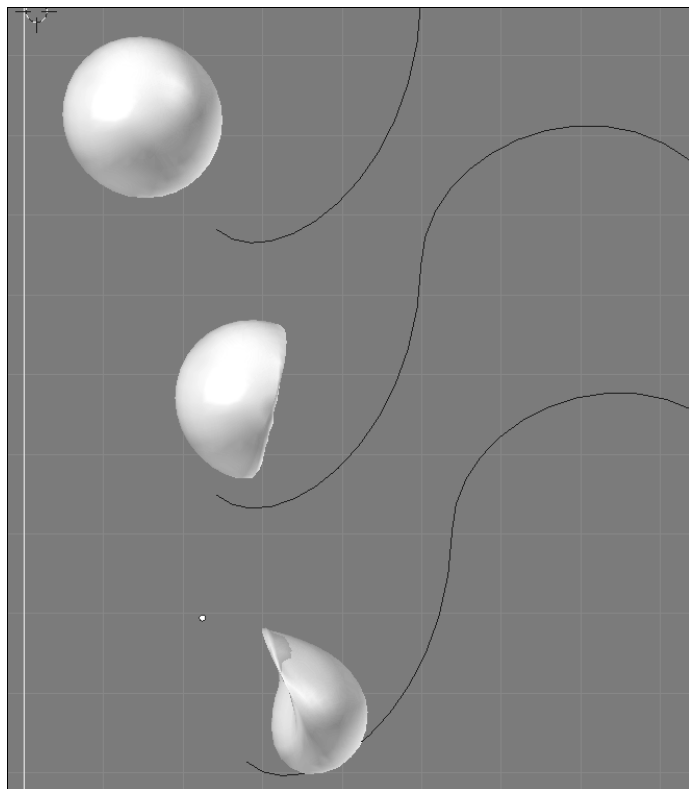


Рис. 3.13. Сверху вниз показаны фазы движения сферы вдоль кривой

СОВЕТ

Используйте механизм **Snap** (Привязка) для совмещения центров двух объектов. Иначе рискуете получить непонятное поведение контролируемого объекта.

3.4. Создание объемных моделей

Уже понятно, что сплайны удобно использовать для создания плоских моделей или объемных фигур с помощью дополнительной кривой. И все же возникает мысль о некоторой ущербности такого рода моделирования, по сравнению с возможностями **Mesh**. Так можно ли сделать что-либо серьезное с их помощью?

Оказывается, да. Сплайны — это мощнейший инструмент для создания обтекаемых, объемных моделей... все тех же **Mesh**. И не в качестве вспомогательных объектов, а как основа создания модели.

Во многих книгах по 3D-моделированию в различных программах рассказывается о способе создания таких моделей, как ваза, бокал, кувшин. И это вовсе не зря. Моделирование этих объектов — сложное и муторное дело, если использовать стандартные функции редактирования примитивов **Mesh**. То же самое относится и к Blender.

Допустим, нужно сделать трехмерный стакан. Первая идея — это воспользоваться объектом **Circle** из группы **Mesh**. Кажется, нет ничего сложного. Выдавливает из него форму по координате Z, если нужно, немного расширяем путем масштабирования верхнюю часть, улучшаем внешний вид с помощью **Multeres** и вуаля, объект готов!

А если понадобится смоделировать кувшин или обычную бутылку с их неравномерной формой? Данный способ явно не годится.

Примитивы **Mesh** имеют замечательную функцию **Spin** (Вращение). Суть ее заключается в том, что она создает копии объекта в процессе вращения на определенный градус. В итоге получается, что достаточно сделать один сегмент модели и повернуть его вокруг оси. Можно, конечно, данный образец сделать изначально в **Mesh**. Но сплайны имеют одно неоспоримое преимущество перед **Mesh**-объектами — они оптимально подходят для создания плавных изгибов фигуры. Рассмотрим на практике, как можно создать ту же самую бутылку.

Для начала найдите в качестве образца любую фотографию с бутылкой или возьмите готовую из папки `Scenes\glava3\bottle.jpg` из архива примеров к этой книге (см. приложение 2). Создайте новый проект и удалите куб из сцены.

Теперь установите картинку в качестве фона окна **3D View** на панели свойств (<N>). Переключитесь в режим вида **Front View** (<NumPad 1>) в ортогональной проекции (<NumPad 5>).

Добавьте в сцену примитив **Add | Curve | Bezier**. По умолчанию он располагается в плоскости XY. Поэтому нужно развернуть его "лицом" к себе. Установите на панели свойств в группе **Rotation** следующие значения:

- ◆ X = 90
- ◆ Y = -90
- ◆ Z = 0

Увеличьте масштаб кривой так, чтобы она по вертикали совпадала с образцом (рис. 3.14).

Настало время создать контур бутылки, причем не всей, а только одной из сторон, скажем, левой. По умолчанию сплайн имеет всего две контрольные точки, чего явно мало. Поэтому в режиме редактирования (<Tab>) выделите все клавишей <A> и нажмите <W>. В появившемся меню **Specials** выберите функцию **Subdivide**. Итак, уже стало три вершины. Теперь можно новую вершину подгонять под контур при-

мера: выделяете точку, переносите на нужное место, затем с помощью рычагов настраиваете степень кривизны. Точно так же происходит работа и для всей части бутылки (рис. 3.15).



Рис. 3.14. Образец картинки с подготовленной кривой

Если сейчас непосредственно перейти к операции **Spin**, то у нас получится бутылка из очень тонкого стекла. Особенно это будет заметно у горлышка. Поэтому нужно добавить толщину стеклу.

Сделать это просто. Достаточно продублировать в режиме редактирования кривую. Нажмите клавишу <A> для выделения всех вершин и кнопку **Duplicate Curve** (Дубликат кривой) на панели **Tool Shelf**. Теперь передвиньте ее на нужное расстояние. Так как дубликат был создан в режиме редактирования, то он является частью нашей кривой.

Обратите внимание на то, что между двумя кривыми начальные и конечные точки не соединены сегментами, т. е. сплайн не замкнут. Оставлять это ни в коем случае нельзя, ибо после операции **Spin** там будут отсутствовать полигоны. Нужно их объединить.

Выверните конечные точки так, чтобы они находились параллельно друг другу. Выделите вершины на нижней части и нажмите клавишу <F> (команда **Make Segment** (Создать сегмент) в меню **Curve**). Они должны объединиться новым сегментом. То же самое проделайте с верхней частью (рис. 3.16).

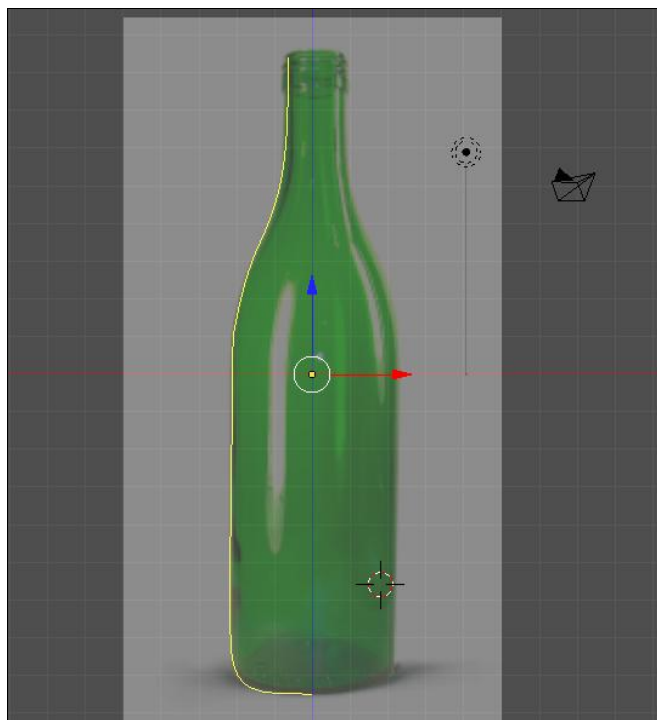


Рис. 3.15. Кривая обрисовывает одну из сторон бутылки

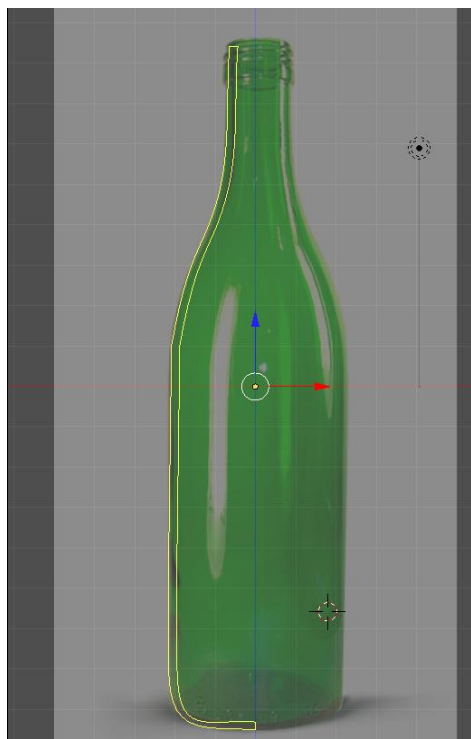


Рис. 3.16. Готовая кривая

Функция **Spin** доступна только для **Mesh**-объектов, поэтому выполним конвертирование сплайна в **Mesh**. Выделите в режиме **Object Mode** кривую и выберите пункт меню **Object | Convert to | Mesh from Curve/Surf/Text**.

Как уже было сказано, **Spin** выполняет поворот выделенных элементов **Mesh**-объекта на определенный градус вокруг центра **3D Cursor**. Соответственно нужно установить данный курсор с помощью команды **Object | Snap | Cursor to Selected**.

Войдите в режим редактирования, выделите весь объект клавишей <A> и перейдите в окно просмотра **Top View** (<NumPad 7>).

После нажатия кнопки **Spin** на панели **Tool Shelf** откроются настройки функции. Установите следующие значения:

◆ **Steps** (Шаги) = 30

◆ **Degrees** (Угол) = 360

Получившаяся модель бутылки изображена на рис. 3.17.

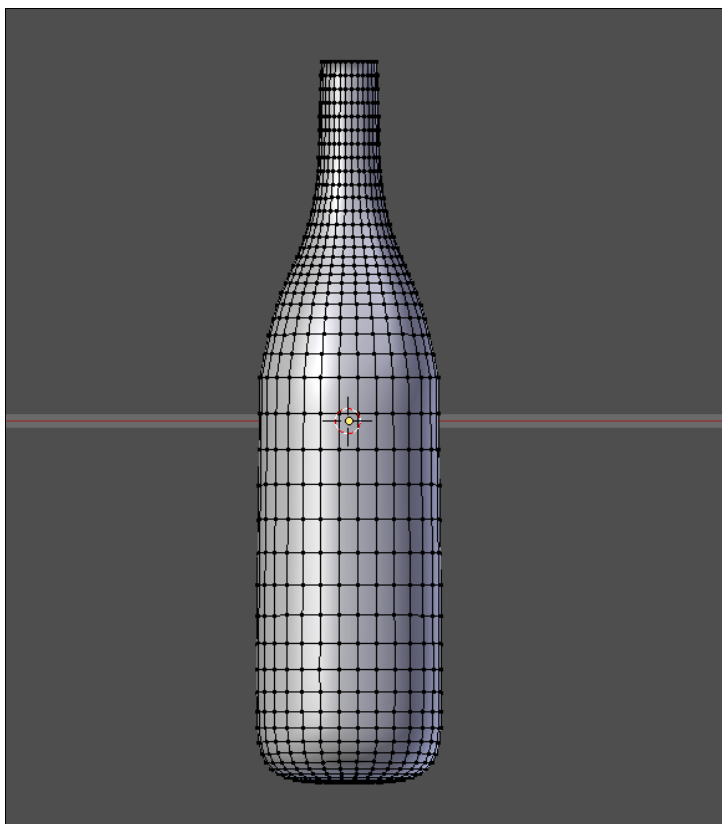


Рис. 3.17. Модель бутылки

3.5. Знакомимся с поверхностями *NURBS*

Новички при знакомстве с **NURBS** часто недоумевают, для чего столь непослушный и сложный инструмент может понадобиться. В действительности **NURBS** позволяет создавать сложные формы за считанные мгновения. Правда, для этого нужно обладать определенным пространственным воображением, уметь представлять будущую модель в разрезе, ведь технология моделирования с **NURBS** сильно отличается от уже привычного **Mesh**.

В отличие от рассмотренных кривых Безье, **NURBS** имеют совершенно иную структуру, способы редактирования. Это не значит, что функции редактирования и манипуляции у них в корне иные. Просто нужно учитывать определенные правила работы. Если кривые **NURBS** в работе особо не отличаются от **Bezier**, то с поверхностями совсем другая ситуация.

По сравнению с Безье, поверхности можно рассматривать как трехмерные объекты, но с очень большой натяжкой. С одной стороны, у них есть дополнительное измерение, с другой стороны, все поверхности, даже зацикленные, не имеют объема. По логике, трехмерными объектами могут называться только те, что имеют объем. И все же в сравнении с простыми кривыми **NURBS** являются трехмерными. Это важно уяснить для понимания работы с ними.

В режиме редактирования поверхности предлагают структуру, внешне напоминающую вспомогательный объект **Lattice** (рис. 3.18).

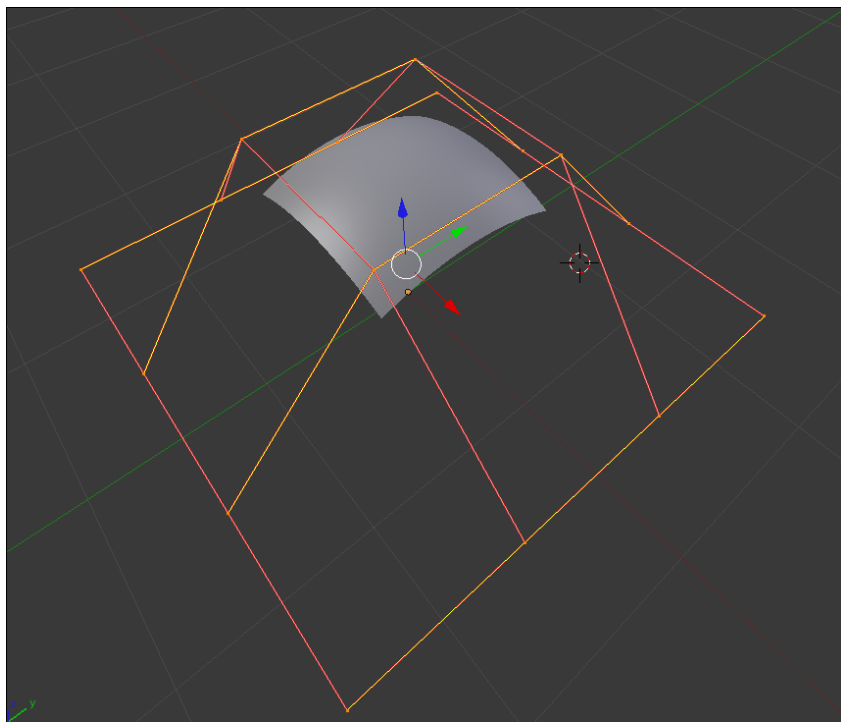


Рис. 3.18. Nurbs Surface в режиме редактирования

Да и принцип работы напоминает **Lattice**. Вы не можете напрямую изменять объект, но влияете на него с помощью вспомогательной решетки.

В отношении **NURBS** важно знать, что решетка имеет два измерения: U и V. В режиме редактирования ребра U окрашены желтым цветом, а V — розовым.

Для изменения формы примитива существуют контрольные точки, но в отличие от Безье тут не имеется рычагов, и они всегда управляются в пространстве 3D (а не 2D, как у кривых).

В отличие от сплайнов, изменять количество элементов структуры решетки возможно только целиком по одному из направлений. Скажем, в кривой Безье нет никаких сложностей с добавлением одной контрольной вершины, но в случае с поверхностями добавлять придется целую линию U или V. Это справедливо для всех остальных операций, таких как **Extrude**, **Delete**, **Subdivide**.

Рассмотрим основные способы редактирования поверхности.

- ◆ *Редактирование формы.* Как уже было сказано, здесь нет отличий от той же работы с кривыми Безье. Выделяется вершина или группа вершин и выполняется стандартное манипулирование: перемещение, масштабирование, ротация.
- ◆ *Добавление ребер.* Эту задачу можно выполнить, к примеру, при помощи команды **Extrude** (<E>) и только для крайних сторон решетки. Для центральных ребер выдавливание невозможно. Необходимо выделить все точки нужной стороны и нажать клавишу <E>.

СОВЕТ

Blender предлагает эффективный способ выделения всех точек по нужному направлению. Для этого достаточно отметить одну ключевую вершину и нажать <Shift>+<R>. Эта команда переключает выделение ребра, на котором лежит точка, с измерения U на V и обратно.

- ◆ *Удаление ребер.* Принцип все тот же. Выделяется с помощью <Shift>+<R> ребро нужного направления и выполняется команда **Delete** (<X>). В отличие от **Extrude** можно удалять любые ребра, а не только крайние.

NURBS имеют настройки, расположенные в окне **Properties** (рис. 3.19).

Группа **Resolution** устанавливает разрешение примитива для окна **3D View** и рендера. Причем это можно сделать для разных измерений. Опции **Preview** отвечают за окно **3D View**, а **Render** — за обработку. Причем установка значения 0 в полях **Preview** приведет к обработке примитива с максимальным качеством.

Закладка **Active Spline** отвечает за внешний вид примитива. Все поля являются уникальными для разных направлений:

- ◆ **Cyclic** (Зацикливание) — замыкание фигуры;
- ◆ **Bezier** (Безье) — включите эти опции, если хотите, чтобы фигура выглядела, как простая кривая без объема;
- ◆ **Endpoint** (Конечные точки) — включение этих опций заставит примитив растягиваться по всей решетке;

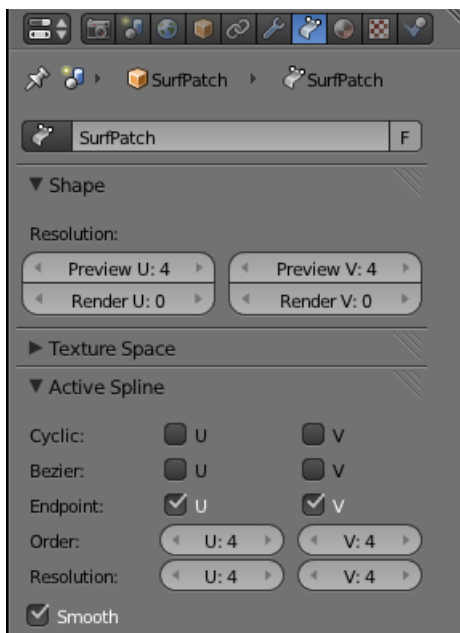


Рис. 3.19. Настройки NURBS

- ◆ **Resolution** (Разрешение) — качество примитива по координатам U и V;
- ◆ **Smooth** (Сглаживание) — стандартное сглаживание для объекта. Равноценно той же опции в **Tool Shelf**.

Особо нужно остановиться на опции **Order**. С ее помощью можно установить соответствие формы примитива с решеткой редактирования. Лучше всего принцип ее работы можно понять, взглянув на рис. 3.20.

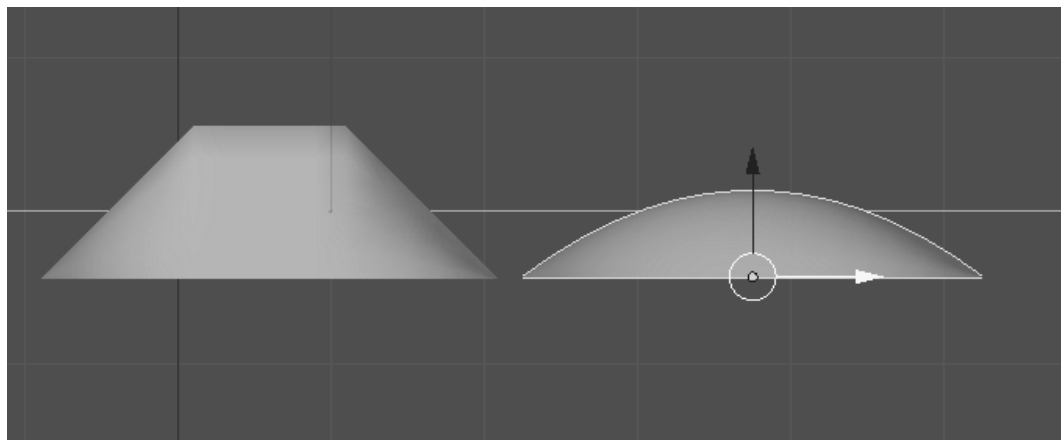


Рис. 3.20. Слева **Order** = 2, а справа **Order** = 4

Рассмотрим небольшой практический пример работы с **NURBS**. На рис. 3.10 показывался вариант создания модели тарелки с помощью двух кривых Безье. Так делать можно, вот только способ больно капризный и зависит от точности деформа-

ции вспомогательного сплайна. С помощью **NURBS** это можно сделать гораздо быстрее.

Прежде чем начать работу над моделью, нужно представить ее скелет, точнее, мысленно разбить объект на ключевые сегменты. В качестве основного примитива воспользуемся **NURBS Circle** (Окружность). Добавьте этот примитив в сцену из меню **Add | Surface | NURBS Circle**.

Задача очень простая — в окне просмотра **Front View** (<NumPad 1>) нужно сделать еще три копии примитива и изменить их масштаб. Принцип следующий:

1. В режиме **Object Mode** выделить **Circle** правой кнопкой мыши.
2. Нажать клавиши <Shift>+<D> для выполнения дублирования и переместить новый объект по координате Z.
3. Нажать клавишу <S> для изменения масштаба.

В итоге сцена должна выглядеть так, как на рис. 3.21.

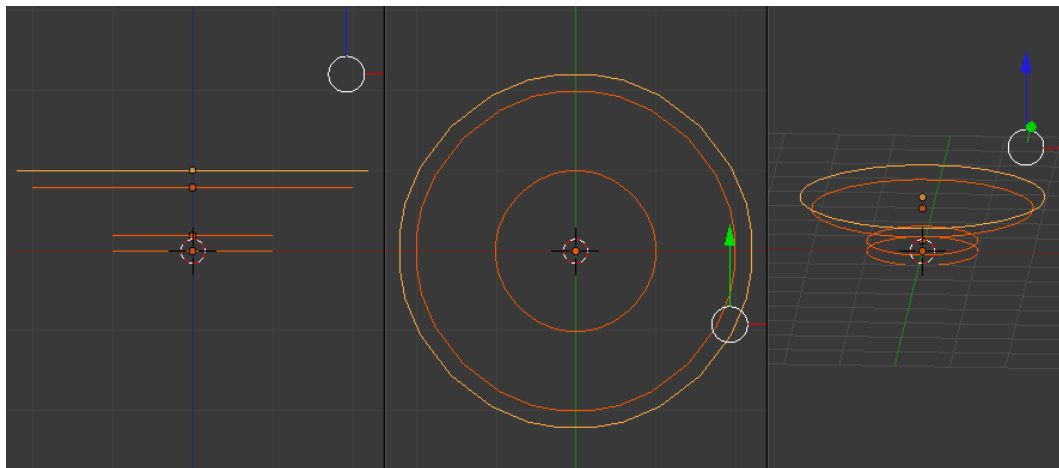


Рис. 3.21. Заготовки для модели тарелки в видах просмотра: **Front**, **Top** и **User**

Вот это и понималось под понятием "пространственное воображение". Как видите, схематически окружности обрисовывают контур будущей модели.

Сделайте еще одну копию самой нижней окружности, сдвиньте немного вниз и уменьшите масштаб. Это необходимо для заливки днища тарелки (рис. 3.22).

Теперь нужно объединить все примитивы в один объект. Выделите их и нажмите <Ctrl>+<J>. Войдите в режим редактирования и выполните заливку объекта с помощью клавиши <F> (рис. 3.23).

Уже нечто похожее на тарелку, вот только отверстие в днище...

На самом деле убрать его очень легко, путем масштабирования самой нижней окружности. Для этого перейдите в режим редактирования и выделите все вершины окружности. Уменьшите с помощью клавиши <S> диаметр отверстия до мини-

мального. Теперь перейдите в настройки поверхности и включите опцию **Endpoint** для координаты U в **Active Spline**. Отверстие должно исчезнуть. Поднимите выделенные вершины на один уровень с дном, чтобы убрать ненужную выпуклость.

Модель получилась весьма обтекаемой с неясными формами. Это выглядит неестественно. Поэтому уменьшите значение в поле **Order U** на одну единицу. Увеличьте для качества разрешение в полях **Resolution** до 6 единиц для обеих координат U и V. Модель тарелки готова (рис. 3.24).

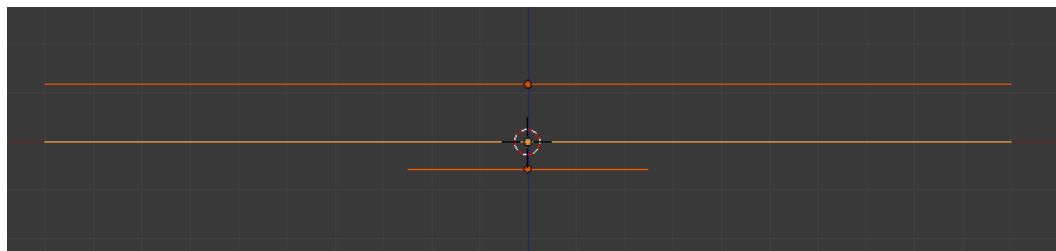


Рис. 3.22. Добавление примитива для дна

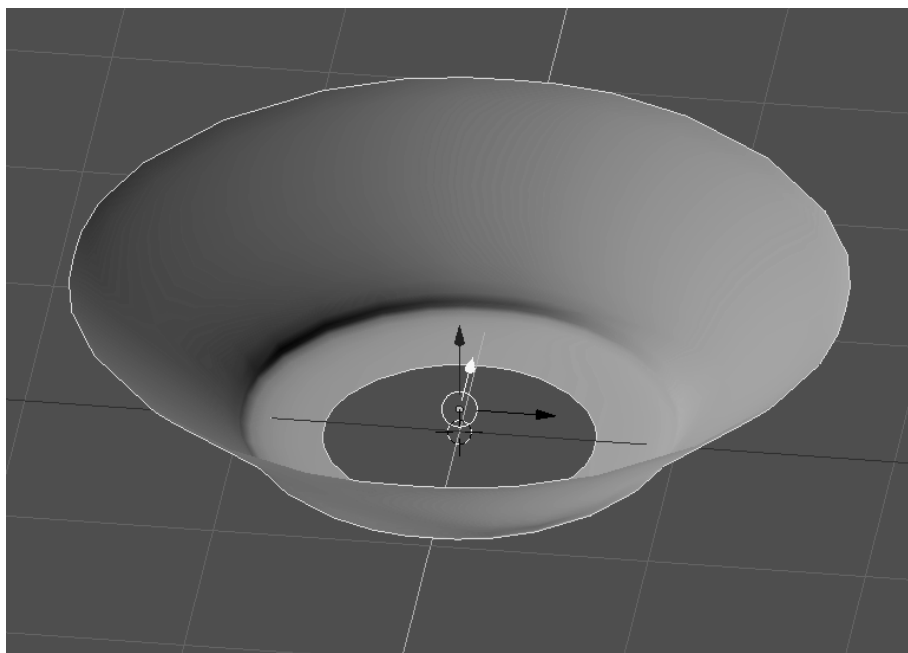


Рис. 3.23. Пробный образец модели

Как видите, с помощью **NURBS** можно быстро и достаточно легко моделировать подобные объекты. А теперь прикиньте объем работы, необходимый для создания тарелки с помощью того же **Mesh**. Перевес явно не в его сторону.

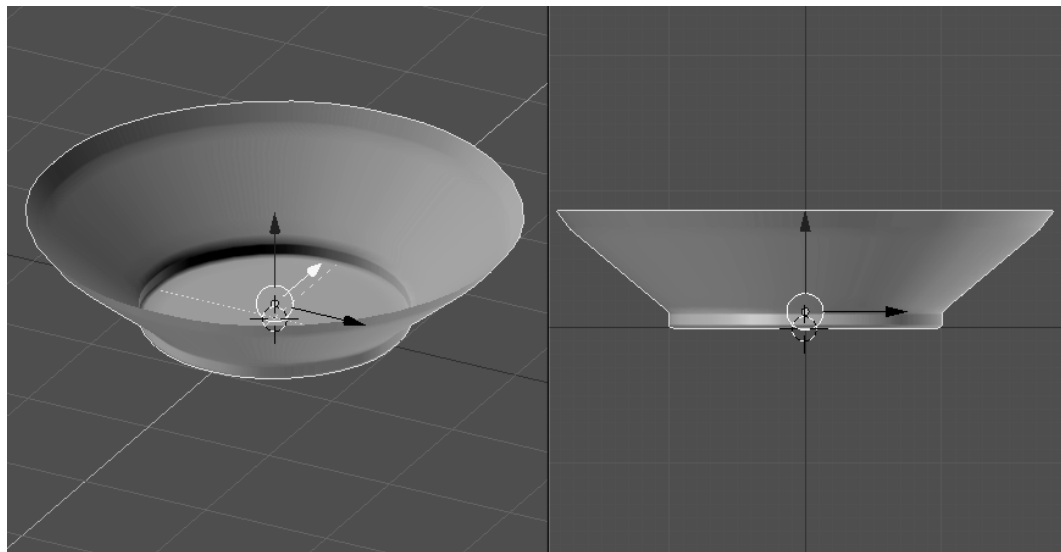


Рис. 3.24. Готовая модель тарелки

3.6. Работа с текстом

Вы уже познакомились с большим количеством инструментов и примитивов, которые позволяют создавать модели любой сложности. Наверное, уже не составит особого труда смоделировать пару букв приемлемого вида, но что делать, если нужно создать слово или целые фразы? Не трудно представить объем работы, требуемый для этой задачи. К счастью, Blender предоставляет специальный объект, который с легкостью позволяет работать с текстом. Собственно, он так и называется — **Text**.

В понимании Blender примитив **Text** является типичным объектом, который подчиняется основным правилам манипуляции. Его можно масштабировать, перемещать или вращать, как, к примеру, тот же **Mesh**. Но главная ценность **Text** в том, что он позволяет вводить, редактировать обычный текст прямо в окне **3D View**. Причем имеющихся настроек вполне хватает на небольшой текстовый редактор. Имеется возможность устанавливать выравнивание на странице, выбирать шрифт и его начертание, управлять межстрочным и буквенным интервалом. В то же время **Text** — типично трехмерный объект, к которому можно применять различные модификаторы или создавать анимацию.

Создание **Text** доступно из меню **Add | Text**. При этом на месте **3D Cursor** появится объект с недвусмысленным словом "Text". Попробуйте покрутить, подвигать примитив и убедитесь, что в управлении он ничем не отличается от **Mesh**. А вот при переходе в режим **Edit Mode** (клавиша <Tab>) появится характерный текстовый курсор, приглашающий к работе (рис. 3.25).

Все остальное напоминает использование типичного редактора. Просто набираете на клавиатуре любую фразу, а программа послушно переводит ее в трехмерный объект. Сделали опечатку? Не проблема, подогнали курсор к нужному месту и удалили с помощью клавиши <Delete>. Начало нового абзаца, как и положено, создается клавишей <Enter>.

Настройки объекта **Text** весьма обширны и располагаются в окне **Properties** (рис. 3.26).

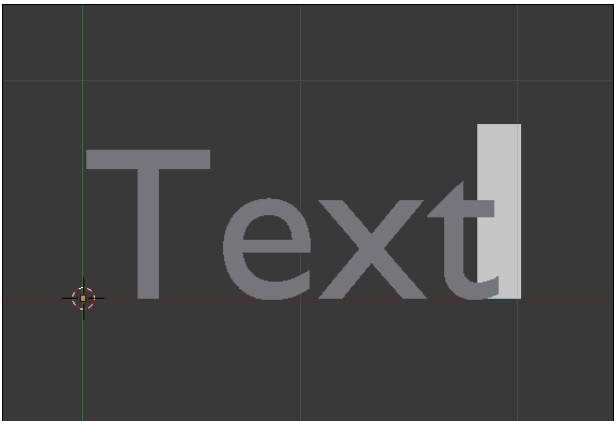


Рис. 3.25. Объект Text в режиме редактирования

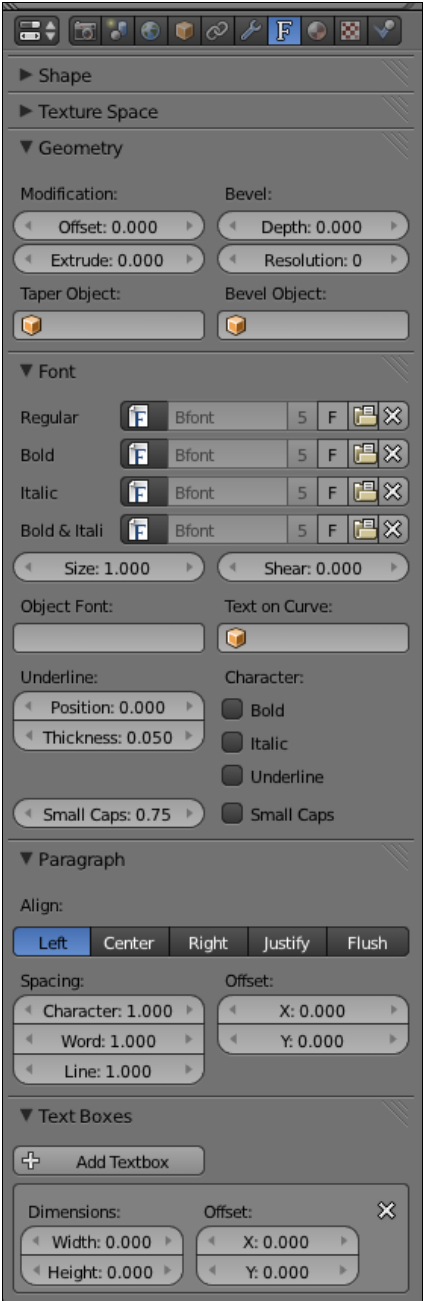


Рис. 3.26. Настройки объекта Text

Закладки **Shape** и **Geometry** вы уже знаете. Они рассматривались ранее. Первая отвечает за качество фигуры (заливка, разрешение), а вторая — за изменение очертаний (выдавливание, кромка). Кстати говоря, к объектам **Text** можно применять вспомогательные кривые для деформации в полях **Taper Object** и **Bevel Object**. Принцип работы с ними одинаков, как для кривых, так и для **Text**.

Blender предлагает несколько иной подход в работе со шрифтами, чем текстовые редакторы. Все нужные настройки сосредоточены на закладке **Font**. Программа умеет использовать разные типы шрифтов, в том числе популярный TrueType. Для выбора нужного существуют списки: **Regular**, **Bold**, **Italic**, **Bold & Italic**. Таким образом, Blender требует для каждого начертания индивидуальный файл шрифта.

По умолчанию в программе содержится всего один шрифт — Vfont. Собственно он используется для вывода начального слова "Text". Для присоединения нового шрифта имеются кнопочки с характерным рисунком желтой папки рядом со списком. После нажатия откроется окно файлового браузера, где и выбирается нужный файл. В дальнейшем загруженный шрифт можно будет найти в списке доступных шрифтов проекта. Именно выбором нужного шрифта решается проблема кириллицы, ведь по умолчанию Blender "не знает" русские буквы.

Размер букв устанавливается в поле **Size**, а наклон — в поле **Shear**.

СОВЕТ

В различных системах шрифты расположены в разных местах. Для семейства Windows это папка Windows\Fonts. В UNIX-системах стандартный путь: /usr/lib/X11/fonts. Если предполагается работать с проектом на разных компьютерах, то лучше файлы шрифтов скопировать в отдельную папку.

В одном блоке одновременно могут находиться буквы с разным начертанием, но одного типа шрифта. Выбор нужного осуществляется путем включения опций в группе **Character**. Важно запомнить, что начертание нужно выбирать до ввода символов, а вот изменение размера и наклона влияет на весь текст. Кстати, заметьте, что Blender для подчеркнутого текста позволяет устанавливать произвольное расположение линии и ее толщины. Эти настройки доступны в группе **Underline** (Подчеркивание), где **Position** — смещение подчеркивания по вертикали, а **Thickness** его толщина.

Редактор программы позволяет выполнять выделение текста. Это возможно при помощи клавиши <Shift>. Приведем краткий список горячих клавиш для выделения.

- ◆ <Shift>+<Left>, <Shift>+<Right>. Побуквенное выделение текста влево или вправо соответственно.
- ◆ <Shift>+<Up>, <Shift>+<Down>. Выделение по вертикали.
- ◆ <Shift>+<Pg Up>, <Shift>+<Pg Down>. Выделение строк.
- ◆ <Shift>+<Ctrl>+<Left> или <Right>. Выделение по словам.

Как видите, все эти сочетания хорошо знакомы и используются в большинстве текстовых редакторов.

Закономерно предположить, что Blender умеет работать с текстовым буфером. Да, это так, причем используются все те же привычные комбинации клавиш.

- ◆ <Ctrl>+<X>. Команда **Cut** (Вырезать).
- ◆ <Ctrl>+<C>. Команда **Copy** (Копировать).
- ◆ <Ctrl>+<V>. Команда **Paste** (Вставить).

Таким образом, проблем с быстрым набором и корректировкой текста не имеется. К сожалению, выделение символов нельзя использовать для изменения типа начертания.

Теперь поговорим о способах добавления текста. Первый самый простой — это ручной набор непосредственно в окне **3D View**. Он годится, если вводить требуется небольшое количество символов. Второй способ — использование встроенного текстового редактора.

Blender, среди прочих окон, имеет еще одно с характерным названием **Text Editor**. Это специализированный текстовый редактор для создания скриптов на языке Python, но его с успехом можно использовать для простого сопроводительного текста к проекту или как промежуточный инструмент для набора текста. Как и все окна, **Text Editor** выбирается при помощи меню **Editor Type** в начале заголовка любого окна (рис. 3.27).

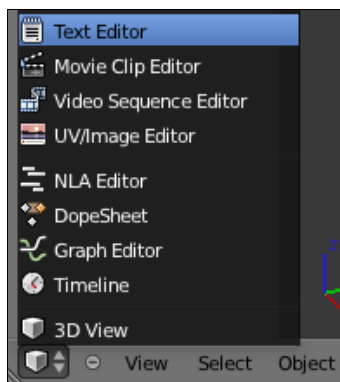


Рис. 3.27. Выбор **Text Editor** из меню **Editor Type**

Создание нового текста начинается с нажатия кнопки **New**. Программа выведет запрос "Create new block?", на который нужно ответить утвердительно. После этого можно вводить текст в окне редактора. Для перевода его в трехмерный объект имеется пункт меню **Edit | Text to 3D Object**. В нем содержатся дополнительные пункты:

- ◆ **One Object** (Один объект);
- ◆ **One Object Per Line** (По объекту на каждую линию).

Выбираете желаемое действие, и в сцене появляется новый текстовый объект.

Объект **Text** позволяет устанавливать выравнивание всего текста и установку значений промежутков между его элементами. Эти функции доступны на закладке

Paragraph (см. рис. 3.26). Кнопки от **Left** до **Flush** отвечают за тип выравнивания по отношению к центру объекта:

- ◆ **Left** — весь текст прижимается к левой условной рамке;
- ◆ **Right** — текст выравнивается по правой стороне;
- ◆ **Center** — выравнивание по центру;
- ◆ **Justify** — растягивание текста между левой и правой стороной, путем регулирования размера пространства между словами. Последняя строка не изменяется;
- ◆ **Flush** — растягивание текста по горизонтали. В отличие от **Justify**, регулировка осуществляется изменением размера пространства между буквами.

Закладка **Paragraph** позволяет устанавливать значения промежутков между словами, буквами и строками текста. Для этого имеется специальная группа **Spacing**:

- ◆ **Character** — межсимвольный промежуток;
- ◆ **Word** — расстояние между словами;
- ◆ **Line** — расстояние между строками.

А вот группа **Offset** управляет смещением всего текста по координатам X и Y.

В отличие от первых трех способов выравнивания, **Justify** и **Flush** функционируют только при создании так называемых текстовых блоков (или рамок).

Представьте объект **Text**, как рабочий стол системы. Обычные окна могут располагаться на нем в произвольном порядке, иметь разный размер, но только в его пределах. Такие окна называются в Blender *текстовыми блоками*.

Frames (Рамки) — это удобное средство позиционирования блоков текста в пределах одного объекта. С их помощью, например, можно создать многоколоночный текст.

Рассмотрим пример разбиения текста по двум колонкам. Создайте новый объект **Text** и заполните его каким-нибудь содержанием. Установите выравнивание по левому краю.

По умолчанию новый объект **Text** уже имеет созданную область. Настройки фреймов доступны на закладке **Text Boxes** (см. рис. 3.26). Каждая рамка имеет свою группу опций:

- ◆ **Dimensions** (Размеры) — параметры **Width** и **Height** позволяют управлять размером блока по ширине и высоте;
- ◆ **Offset** (Смещение) — смещение блока относительно центра объекта.

Визуально рамки видны только в режиме редактирования в виде пунктирных линий, но в данный момент вы их не обнаружите. Дело в том, что блок вполне может обходиться без ограничителей, по умолчанию так и есть. Размещением текста в этом случае управляет сам пользователь. Чтобы включить отображение рамок, достаточно изменить параметры **Dimensions**. Отрегулируйте размер на свой вкус.

Теперь добавьте новый блок, нажав кнопку **Add Textbox** (Добавить текстовый блок). Добавится еще одна область настроек. Используя опции **Offset**, сместите новый блок вправо (рис. 3.28).

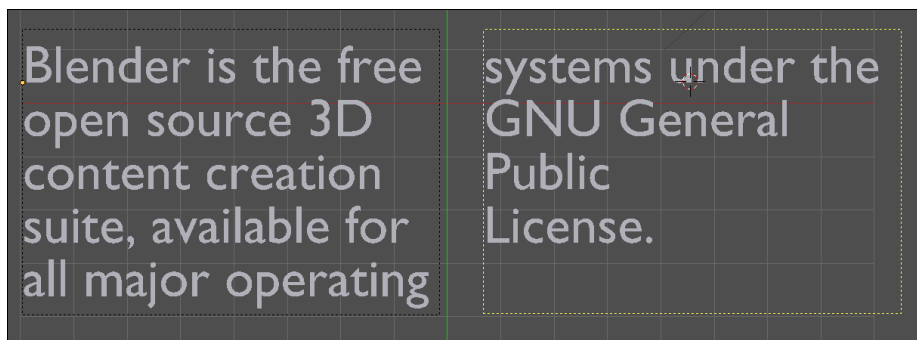


Рис. 3.28. Двухколоночный текст

К объекту **Text** можно добавлять модификатор **Curve** и тем самым управлять расположением текста в трехмерном пространстве. Рассмотрим способ создания объемных букв, изогнутых по окружности.

Создайте новый проект. Добавьте в сцену следующие примитивы: **Text** и **Curve Circle**. Измените стандартный текст на "Hello World!" и добавьте модификатор **Curve**. В настройке модификатора в поле **Object** выберите название окружности (рис. 3.29).

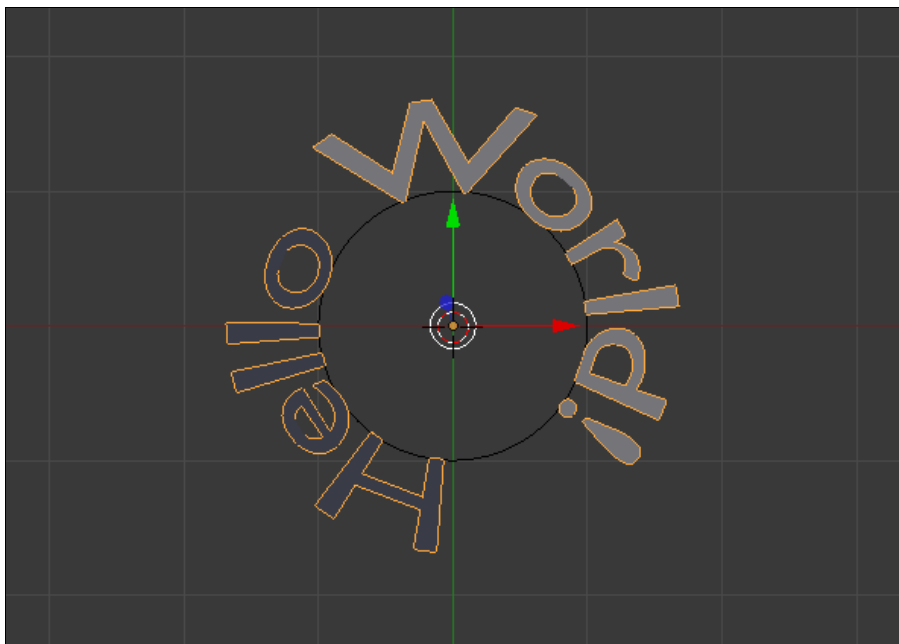


Рис. 3.29. Результат использования модификатора **Curve**

На рис. 3.29 видно, что текст изогнулся по кривой и лежит в одной плоскости с ней. Это произошло из-за выбора координаты X в настройках модификатора. Попробуйте сделать следующее:

1. В настройках модификатора в группе **Deformation Axes** нажмите кнопку **-X**.
2. Разверните текст по координате X на 180 градусов. Проще всего это сделать на панели свойств объекта в окне **3D View**. Нажмите клавишу <N> для ее вызова и установите значение 180 в поле **Rotation X**.
3. Увеличьте масштаб **Curve Circle** так, чтобы буквы не напоздали друг на друга.
4. Если вы выполнили все описанные шаги, то текст должен выглядеть так, как на рис. 3.30.

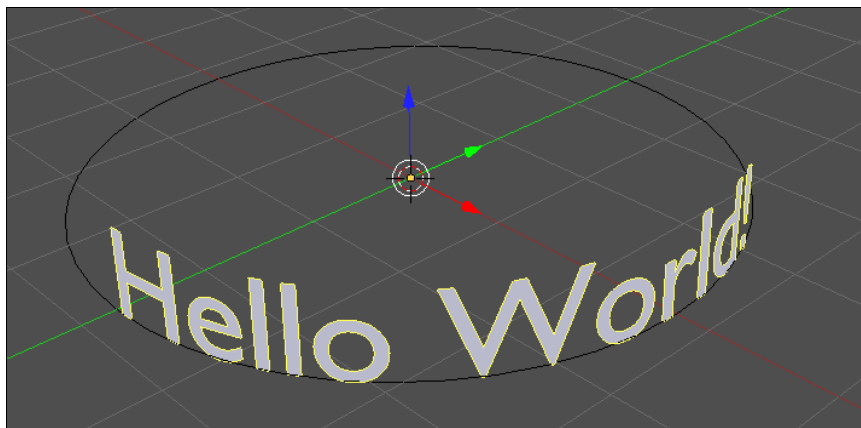


Рис. 3.30. Буквы выстроились по окружности

Для придания объема буквам нужно выбрать в окне **Properties** закладку **Object Data** с настройками объекта **Text** (см. рис. 3.26). В группе **Geometry** установите значения для следующих опций:

◆ **Extrude** = 0.05

◆ **Depth** = 0.02

Вот таким несложным способом можно получить симпатичные трехмерные буквы (рис. 3.31).



Рис. 3.31. Объемные буквы вдоль кривой

СОВЕТ

Вы можете конвертировать текстовый объект в **Mesh** или кривые. Нажмите <Alt>+<C> для вызова меню **Convert To**. Первый пункт этого меню **Curve from Mesh/Text** создает кривые, пункт **Mesh from Curve/Surf/Text** генерирует **Mesh**-объект.

3.7. Практика. Как сделать смайлик

Сплайны просто идеально подходят для создания двухмерных моделей, к примеру, логотипов. Обычно для этих целей используют векторные программы, такие как CorelDRAW, Inkscape. Собственно они тоже работают с примитивами, наподобие кривых Blender. Вот только зачем изучать другую программу, если то же самое можно сделать в Blender?

В этом уроке вы узнаете, как можно компоновать несколько кривых для получения отверстий в модели на примере создания двухмерного смайлика. Эту улыбающуюся рожицу вы сделаете буквально за пару минут.

Создайте новый проект и удалите куб из сцены. В качестве образца опять-таки можно воспользоваться готовой картинкой смайлика. Вы можете взять файл smile.jpg из папки Scenes\glava3, расположенной в электронном архиве к данной книге, или скачайте подходящую картинку из сети Интернет.

Перейдите в режим просмотра **Top View** ортогональной проекции (<NumPad 7>, <NumPad 5>). Нажмите клавишу <N> для вызова панели свойств окна **3D View**. Теперь вы можете на закладке **Background Images** (Фоновые изображения) загрузить нужный файл картинки (рис. 3.32).

Посмотрите на рис. 3.32. Смайлк выглядит очень просто. Основанием служит окружность, пара отверстий для глаз, да изгиб улыбки. Соответственно понадобится три примитива **Bezier Circle** и одна кривая.



Рис. 3.32. Нечто подобное должно у вас получиться

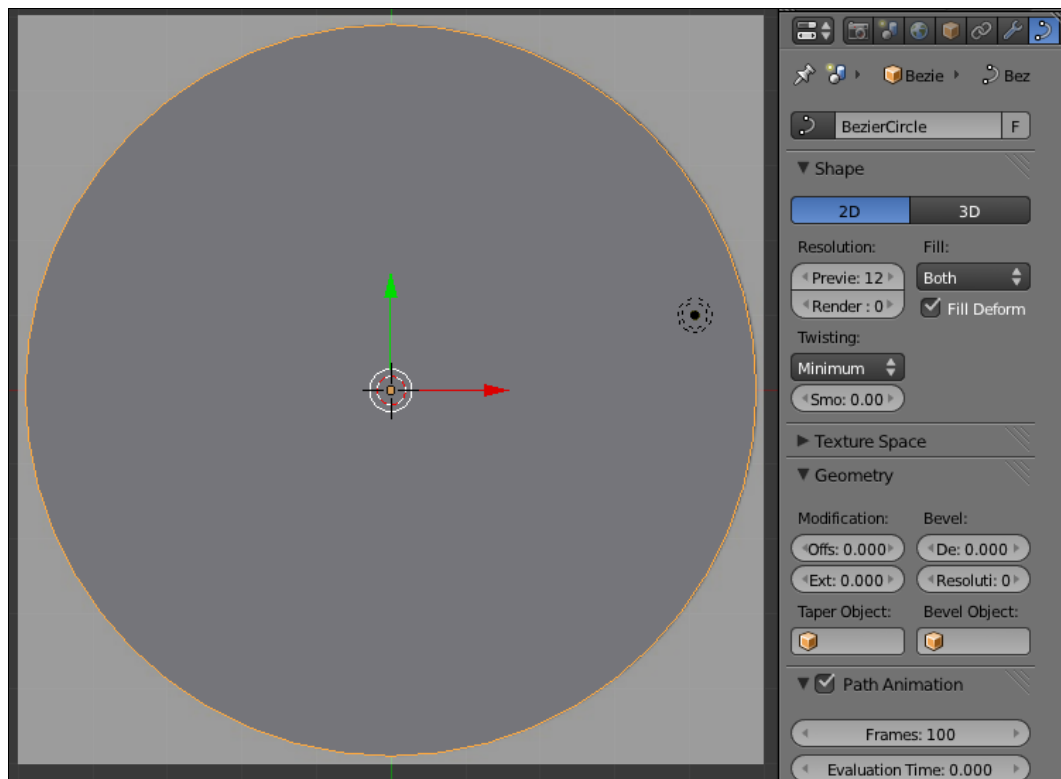


Рис. 3.33. Bezier Circle в режиме 2D приобретает заливку

Добавьте пока в сцену первую окружность (**Add | Curve | Circle**). Даже не переходя в режим редактирования, просто отмасштабируйте ее в соответствии с пропорциями нашего колобка. Теперь перейдите в настройки сплайна в окне **Properties** и установите на закладке **Shape** режим прорисовки **2D**. Примитив приобрел заливку (рис. 3.33).

Теперь займемся созданием глаз. Чтобы заливка не мешала просмотру картинki-образца, нажмите клавишу **<Z>** для перехода в режим отображения **Wireframe**.

Перейдите в режим редактирования и добавьте еще один примитив **Bezier Circle**. Не выходя из режима редактирования, нажмите **<G>** и переместите новую окружность в центр первого глаза. Выделите две горизонтальные вершины и немного сожмите их с помощью масштабирования (**<S>**). Если нужно подправить местоположение "глаза", то лучше воспользуйтесь рамкой выделения (клавиша ****). Клавиша **<A>** здесь не поможет, ведь вторая окружность стала единым целым с первой. Теперь можно продублировать готовый овал для второго глаза (**<Shift>+<D>**). Нажмите клавишу **<Z>** для выхода из режима **Wireframe** (рис. 3.34).

Итак, запоминаем. Чтобы вырезать отверстие в двумерной замкнутой фигуре, достаточно в режиме редактирования добавить к ней еще одну замкнутую кривую.

Теперь предстоит кропотливая работа по вырезке улыбки. Для этой цели воспользуйтесь кривой Безье (**Add | Curve | Bezier**). Перенесите одну из вершин новой

кривой в любой угол "улыбки". Для создания обводки вокруг рисунка можете воспользоваться следующим алгоритмом:

1. Выделите свободную контрольную точку и перенесите в нужное место.
2. С помощью рычагов создайте нужный изгиб.
3. Выделите последнюю вершину, нажмите клавишу <E> для создания новой контрольной точки. Перенесите в нужное место.

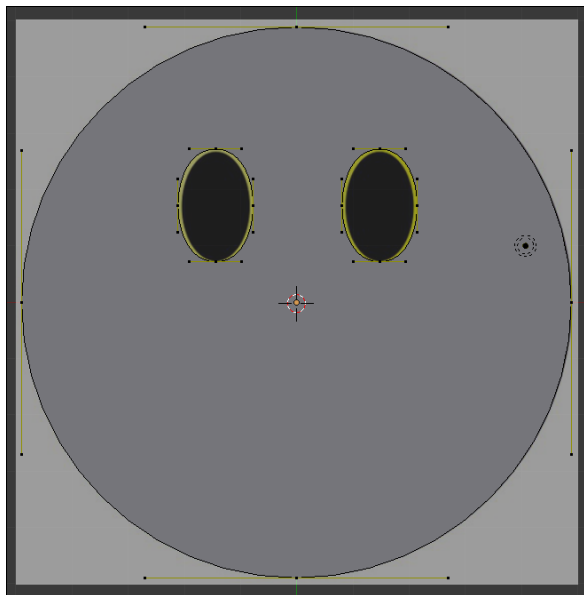


Рис. 3.34. У смайлика появились глазки



Рис. 3.35. Так может выглядеть петля

ЭТО ВАЖНО!

Будьте осторожны и не создавайте петли. Если все же при добавлении новой вершины она получилась, то раскрутите ее с помощью рычагов. Наличие петель в модели приведет к неправильной заливке (рис. 3.35).

Обведите все изгибы улыбки и дотяните кривую до первой точки (рис. 3.36).

Осталось замкнуть сплайн "улыбки". Выделите две крайние вершины и нажмите $\langle \text{Alt} \rangle + \langle \text{C} \rangle$. Появится новый сегмент, который можно будет отредактировать по форме рисунка. Если все выполнено верно, то нажатие клавиши $\langle \text{Z} \rangle$ проявит на экране улыбающуюся рожицу (рис. 3.37).

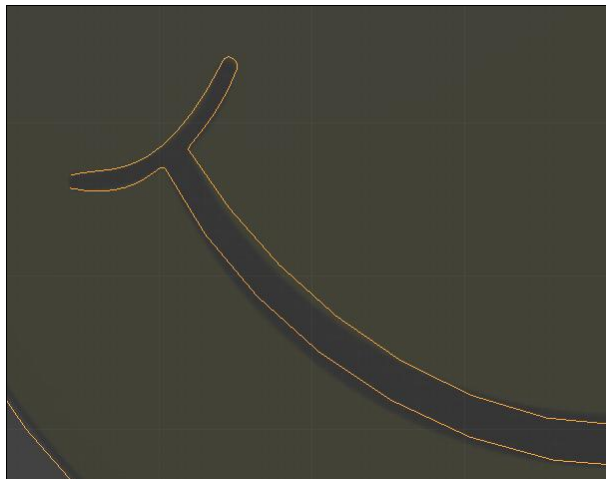


Рис. 3.36. Почти законченная улыбка



Рис. 3.37. Смайлик готов!

3.8. Практика. Модель лодки

Лодка имеет обтекаемую замкнутую форму, поэтому логично было бы использовать в качестве основы сплайны. Вот только на этот раз лучше воспользоваться **NURBS**. Правда, для доработки модель придется конвертировать в **Mesh**.

Несмотря на достаточно сложную фигуру, создать форму лодки с помощью **NURBS** очень просто. По сути, достаточно разработать один основной элемент и его раскопировать. В качестве основы возьмем примитив **NURBS Circle**.

Создайте новый проект и удалите куб из сцены. Добавьте нужный примитив из меню: **Add | Surface | Nurbs Circle**.

По умолчанию **Circle** "смотрит" по координате Z. Это не очень удобно и его следует развернуть. Нажмите клавишу <N> для вызова панели свойств. В группе **Rotation** установите значение 90 для координаты Y.

Для изменения формы окружности перейдите в просмотр **Right View** (<NumPad 3>). В режиме редактирования выделите верхнюю точку и опустите ее почти до уровня нижней (рис. 3.38).

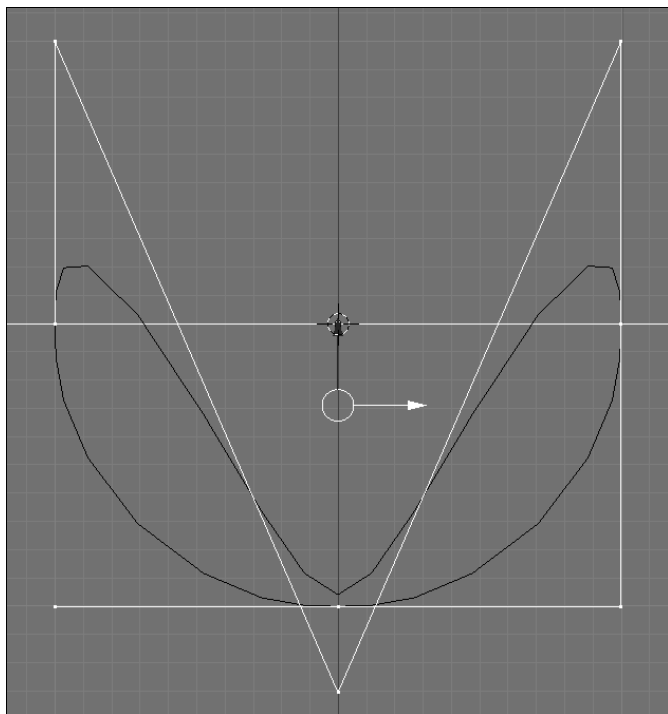


Рис. 3.38. Углубление в модели

Таким же способом с помощью остальных вершин добейтесь формы, как на рис. 3.39.

Перейдите в режим просмотра **Front View** (<NumPad 1>). Ваша задача путем дублирования примитива создать реберный каркас лодки. Нажмите <Shift>+<D> и пе-

ренесите влево копию объекта. Сделайте еще одну копию и сдвиньте ее также влево. В этой части будет находиться нос суденышка. Поэтому новый примитив нужно немного сжать с помощью масштаба (<S>) и переместить его по координате Z до совпадения верхней условной линии для всех объектов. Создайте его копию еще меньшего размера, который будет являться конечной точкой носа (рис. 3.40).

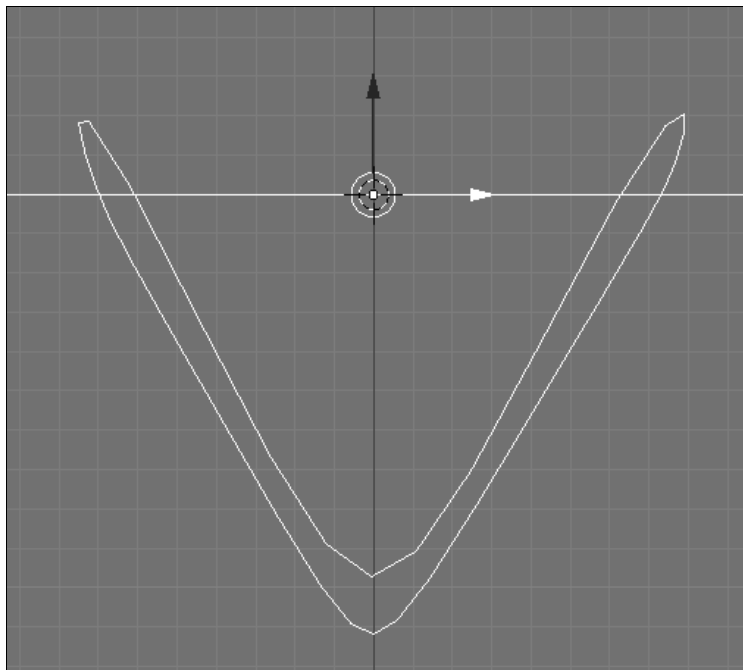


Рис. 3.39. Готовый примитив для каркаса лодки

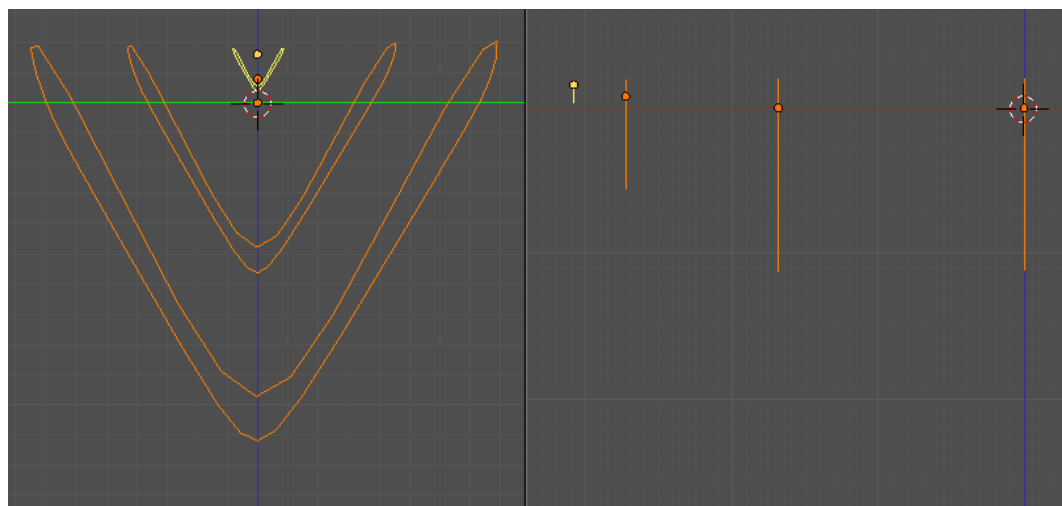


Рис. 3.40. Ребра от центра до носа

Таким же образом нужно будет размножить элементы и для создания кормы. Только учтите то, что у лодок корма является тупой, поэтому не переусердствуйте с масштабом (рис. 3.41).

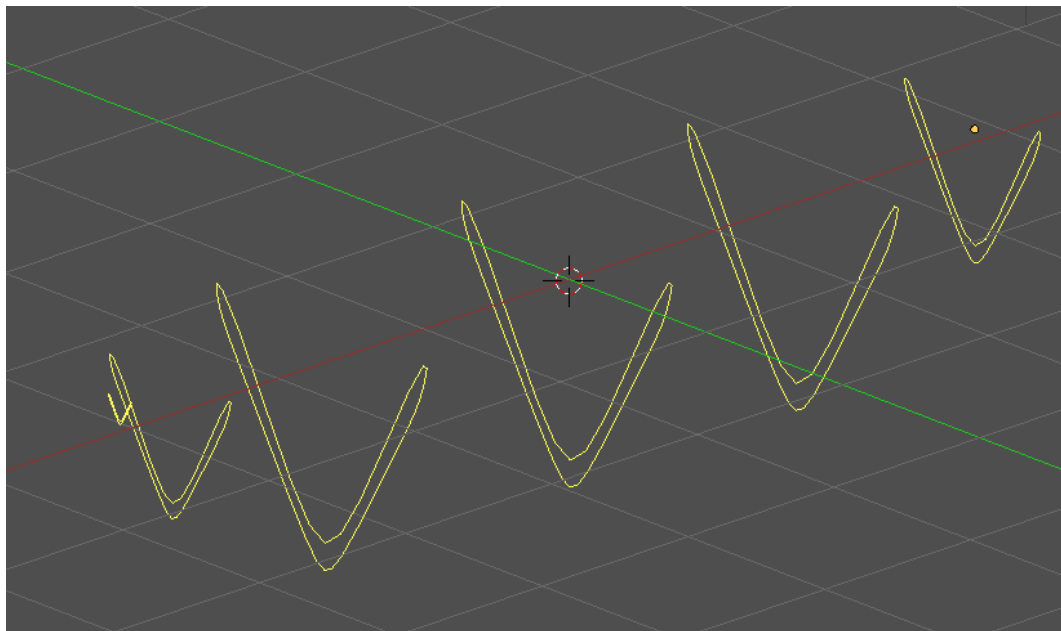


Рис. 3.41. Каркас лодки

Осталось выполнить заливку. Выделите все объекты лодки и нажмите <Ctrl>+<J> для объединения их в один объект. В режиме редактирования нажмите клавишу <F> и модель обретет объем (рис. 3.42).

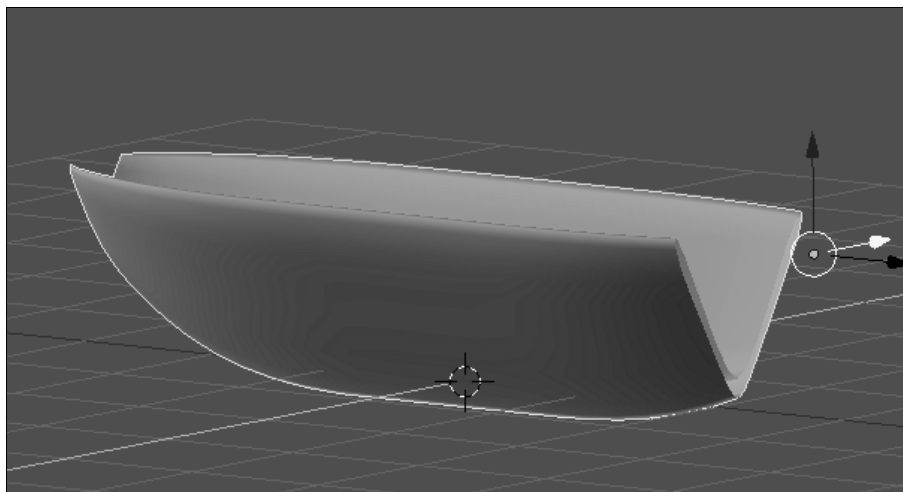


Рис. 3.42. Результат объединения и заливки

Получилось не особо похоже на лодку. Причина заключается в том, что первое и последнее ребра не участвовали в заливке. Однако достаточно установить галочку в свойстве объекта **Endpoint**, чтобы лодка обрела полные очертания.

Перейдите в режим редактирования на панели свойств объекта (окно **Properties**, закладка **Object Data**) и установите следующие галочки в группе **Active Spline**:

- ◆ **Cyclic V** — замкнули поверхность;
- ◆ **Endpoint U** — растянули до границ редактируемых кривых.

Осталось только закрыть корму. В просмотре **Front** выделите крайние вершины и нажмите клавишу <E> (рис. 3.43). Не меняя положения новой группы, уменьшите до минимума масштаб. Отрегулируйте по высоте и длине корму модели (рис. 3.44).

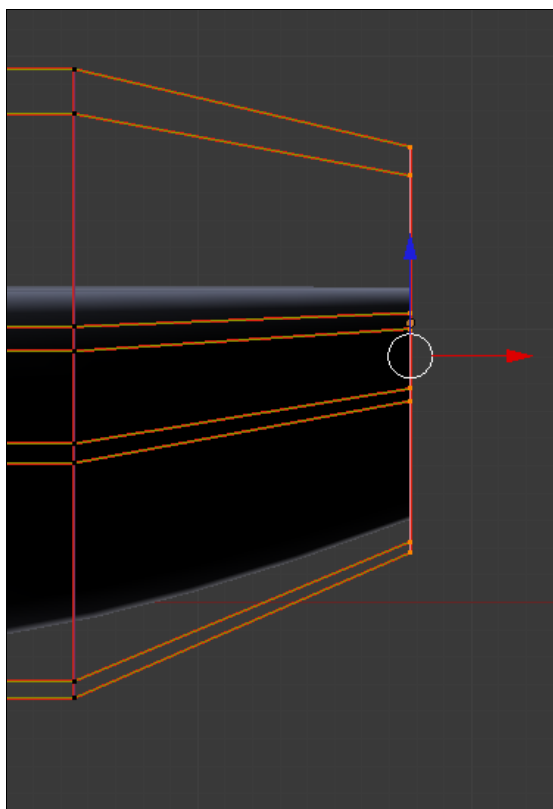


Рис. 3.43. Выделенные вершины для закрытия кормы

Если присмотреться внимательно к крайним частям модели, то можно увидеть про-
рези, оставленные из-за специфичного вида начальных ребер. Не беда, это очень
просто исправить после перевода модели в **Mesh**. Данное конвертирование также
понадобится для создания перекладин у лодки.

Для выполнения конвертирования нажмите <Alt>+<C> и выберите в появившемся
меню пункт **Mesh from Curve/Surf/Text**. Перейдите в режим редактирования и

убедитесь, что лодка стала **Mesh**-объектом (должна быть характерная ячеистая структура).

Закрывать разрывы на носу и корме поможет функция **Merge**, которая объединяет выделенные элементы. Отметьте все нужные вершины с одного конца лодки, нажмите клавишу <W> и в появившемся меню выберите пункт **Merge** (рис. 3.45).

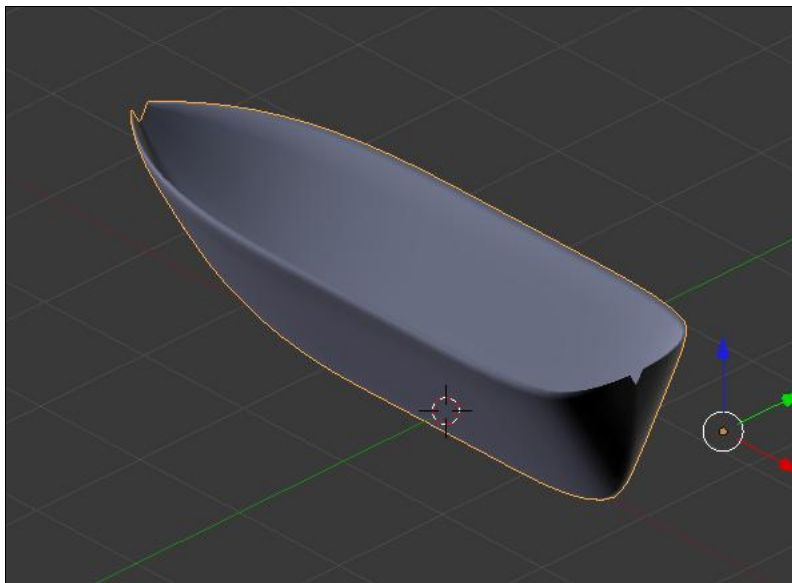


Рис. 3.44. Лодка почти готова

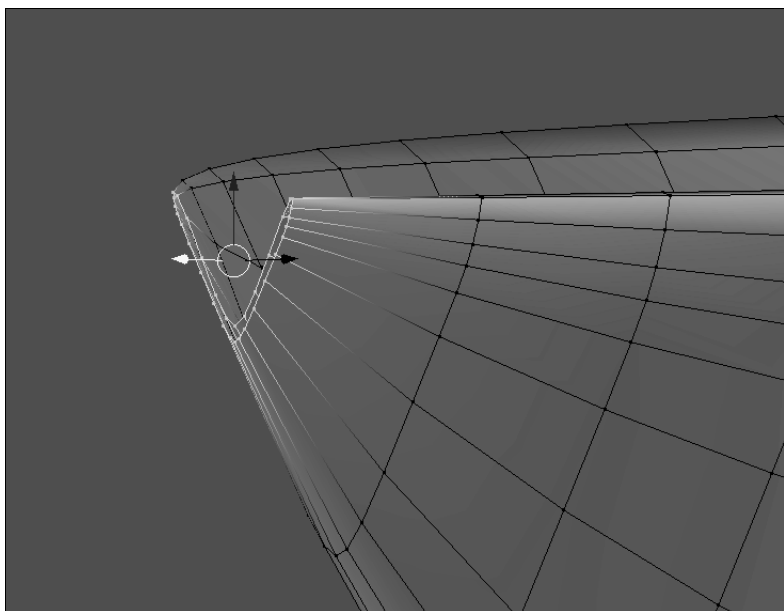


Рис. 3.45. Выделенные вершины на носу лодки

Здесь лучше подойдет смещение точек в центр (**At Center**). После объединения немного подправьте положение новой вершины.

Похожую операцию придется сделать и с другим концом модели.

Теперь займемся созданием перекладин для сидений. Тут стоит вспомнить о ранее рассказанном способе симметричного моделирования. В самом деле, если разделить лодку пополам от носа до кормы, то можно будет выдавить нужные перекладины лишь с одной стороны, а модификатор **Mirror** продублирует вторую.

В режиме редактирования выделите вершины одной из сторон, как на рис. 3.46, и удалите их. Удобнее будет, если вы перейдете в просмотр **Right View** (<NumPad 3>) ортогональной проекции. Разумеется, включите режим прорисовки **Wireframe** клавишей <Z>.

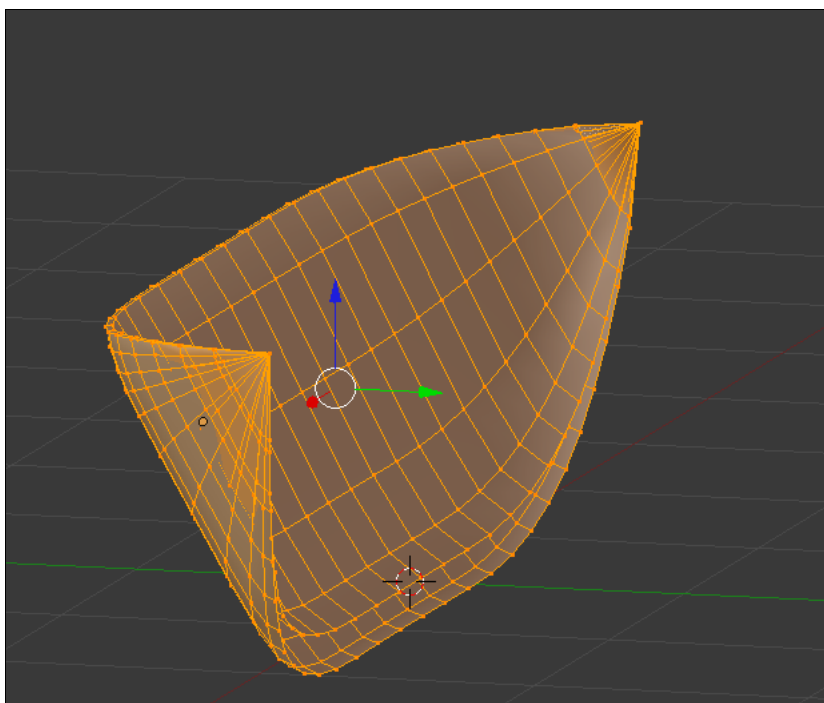


Рис. 3.46. Объект после удаления одной стороны

После удаления стороны добавьте к модели модификатор **Mirror**. Если зеркальная копия будет показана с неверной стороны, то помогут опции **Axis**. Пошелкайте по ним для достижения результата.

Теперь поговорим о создании перекладин. В принципе достаточно выделить верхний внутренний полигон лодки и выполнить **Extrude**. Беда в том, что подходящего размера граней у лодки просто нет, ведь она получилась достаточно низкополигональной. Придется добавить еще одно ребро по всему периметру лодки.

Разверните модель так, чтобы в режиме редактирования была видна рабочая сторона. Нажмите клавиши <Ctrl>+<R> для вызова инструмента **Loop Cut** (Кольцевой

разрез). Возможные варианты добавления ребра будут показаны фиолетовым цветом. Подвигайте мышью и выберите оптимальный вариант. Закрепите щелчком левой кнопки мыши ребро и немного передвиньте его для создания узкой полосы граней. Вторичный щелчок мышью окончательно закрепит новые элементы в структуре (рис. 3.47).

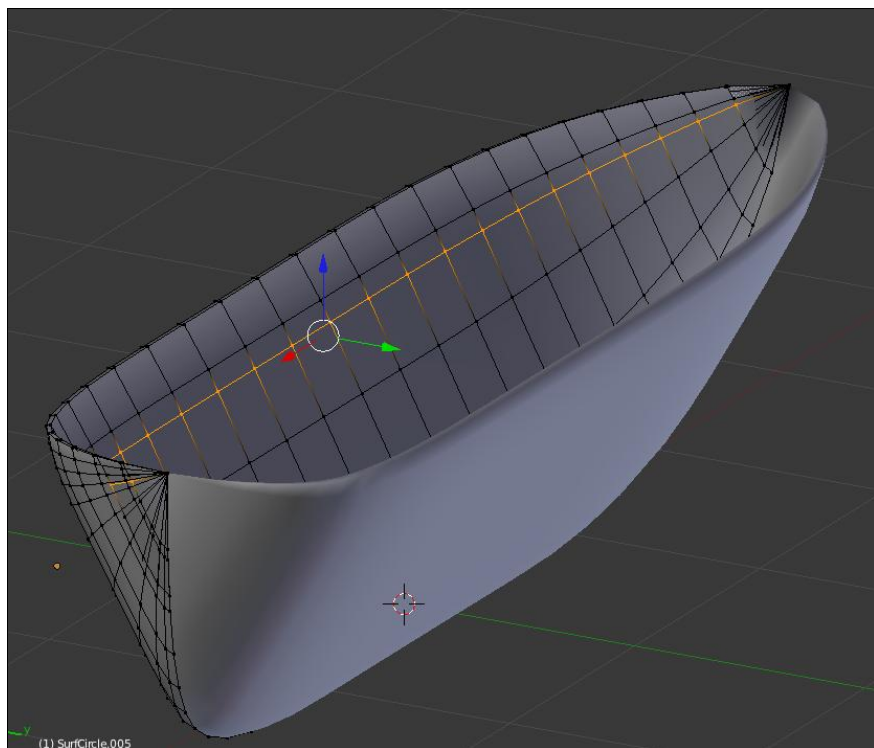


Рис. 3.47. Новое ребро модели

Дальше все просто:

1. Выделяете приглянувшуюся грань.
2. Переключаетесь в просмотр **Top View**.
3. Нажимаете <E> для выдавливания и закрепляете результат щелчком мыши.
4. Вытягиваете полигон по координате Y (клавиши <G> и <Y>).

Посмотрите на результат (рис. 3.48).

На рисунке отчетливо видно, что произошло пересечение граней оригинала и копии. Это недопустимо и может в дальнейшем привести к артефактам обработки (например, при анимации будет неприятное мерцание). На данном этапе исправить это можно, просто развернув (<R>) и передвинув грань (<G>). Конечно, точно совместить копию и оригинал "на глазок" просто невозможно.

Модификатор **Mirror** в настройках имеет две полезные опции: **Merge** (Объединение) и **Clipping** (Обрезка). Первая выполняет слияние близлежащих вершин копии

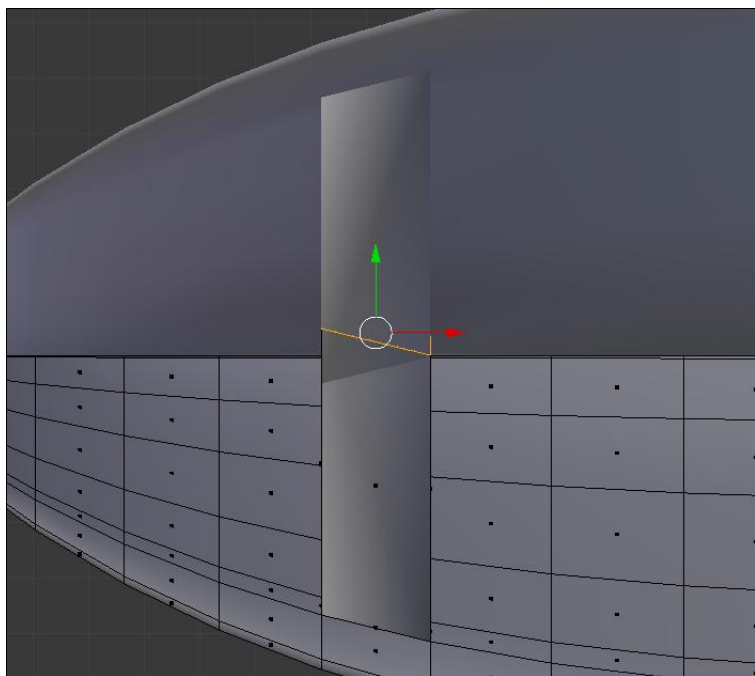


Рис. 3.48. Пересечение граней

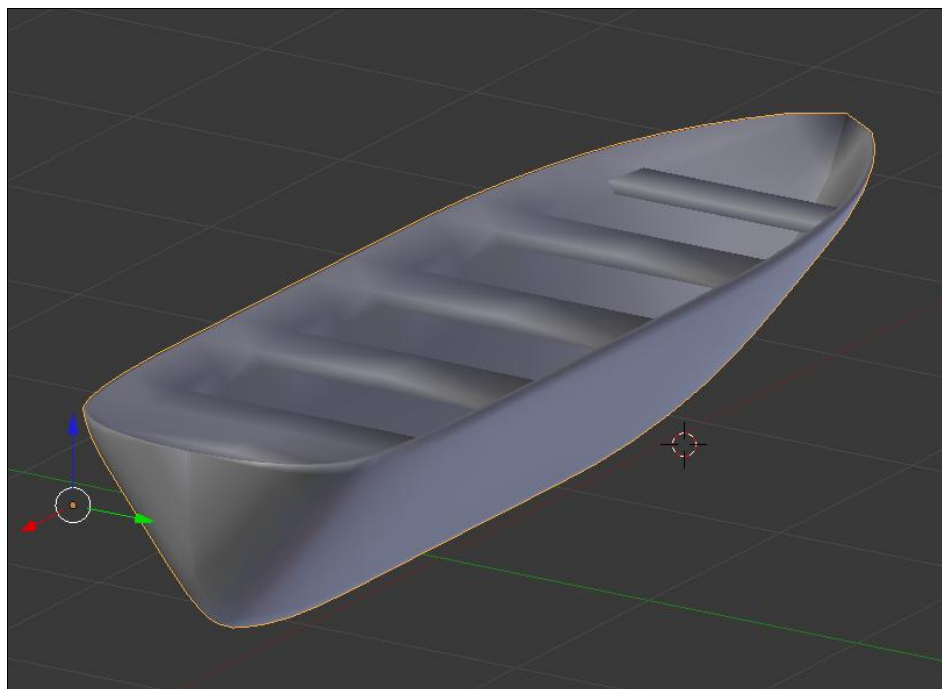


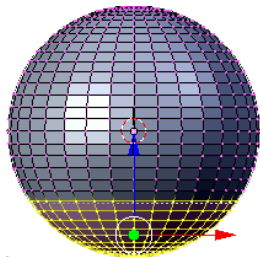
Рис. 3.49. Готовая модель лодки

и оригинала на расстоянии в зависимости от значения в поле **Merge Limit** (Ограничитель для объединения). Вторая опция отвечает за обрезку лишних элементов при пересечении. Включите обе опции.

Теперь при аккуратном перемещении грани в сторону копии нужно добиться идеального слияния — визуально это можно описать, как эффект "защелки". Если это не удастся сделать, то увеличьте параметр **Merge Limit**.

После создания всех перекладин внимательно осмотрите модель со всех сторон на наличие артефактов в шве слияния, если все нормально, то модификатор можно применить кнопкой **Apply** (рис. 3.49).

ГЛАВА 4



Материалы и текстуры

Вот вы и приступили к чтению, наверное, самой интересной главы книги. Не знаю, как вам, но мне уже приелись модели уныло серого цвета. С помощью материалов и текстур можно добиться превосходного визуального результата, раскрасить цветами виртуальный мир. Кроме того, умелое их использование позволяет скрыть огрехи моделирования и добавлять мелкие детали.

Допустим, нужно сделать модель апельсина. Внешне этот фрукт выглядит как практически идеальная круглая сфера, но вот кожа у него пупырчатая. Создание этих неровностей на этапе моделирования выглядит, мягко говоря, нецелесообразным, а вот использование рельефных карт подходит идеально.

Вы, наверное, уже встречались с термином "текстура". В пользовательском смысле — это простой файл с картинкой. В трехмерном моделировании часто используют текстуры, заранее созданные в других программах или отснятых фотокамерой. Разумеется, Blender умеет работать с такими файлами, но в некоторых случаях есть возможность обойтись внутренними ресурсами программы. Например, делается деревянный или металлический предмет. Зачем искать сторонние картинки, если можно воспользоваться процедурными текстурами, встроенными в саму программу! Blender предлагает и такой способ текстурирования модели.

А создание прозрачных или отражающих предметов? Оказываются и эти эффекты создаются с помощью материалов.

4.1. Что такое материал

Во вступлении к главе часто использовалось слово "материал". Обычно возникает ассоциация с чем-то, что покрывает какой-либо предмет, например краска на корпусе автомобиля или одежда на человеке. В трехмерном мире материал обозначает то же самое — покрытие начального каркаса. Отсюда главный вывод — его нельзя использовать без объекта.

Материал не стоит рассматривать, как простой цвет модели. Скорее всего — это контейнер, объединяющий сразу несколько технологий: цвет, шейдеры, слои тек-

стур, прозрачность и преломление, градиент. Ничто не мешает, скажем, создать реалистичный материал, имитирующий хрусталь или дерево, сохранить его в файл и использовать в своих проектах. Соответственно, второй вывод — один и тот же материал можно использовать для разных объектов. Причем к модели можно применить два, три, а то и больше материалов, что открывает практически необъятный простор для творчества.

Несмотря на многомерность, структура материала достаточно жесткая. Проще всего ее представить в виде многослойного пирога. Самый нижний ярус занимает *базовый цвет*. К примеру, куб, который есть в новой сцене, уже имеет присвоенный материал серого цвета. Нет ничего сложного в том, чтобы поменять цвет на любой другой, благо Blender предлагает целую палитру, но только он будет однородным.

Если присмотреться к этому кубу в сцене, то можно увидеть, что у него имеются темные и светлые области в зависимости от местонахождения источника света, а также блики. Такие световые эффекты создаются при помощи специальных *отражающих шейдеров*. По желанию их можно настраивать, добиваясь жесткого или мягкого отблеска. В иерархии материала шейдеры занимают самое верхнее место.

А вот текстуры располагаются между ними. Они могут быть любыми: обычными картинками или процедурными картами. Причем имеется возможность использовать сразу несколько текстур в одном материале, как бы своеобразной стопкой. Такая стопка называется *текстурными слоями* или каналами.

Все элементы структуры материала можно смешивать между собой разными способами: установкой прозрачности, функциями наложения, использованием специальных фильтров.

К примеру, рассмотрим материал того же апельсина. Базовым цветом у него будет оранжевый. Для создания неровностей на коже используется процедурная текстура, наложенная при помощи механизма **Normal Map**. Ну и мягкие, нерезкие отражающие шейдеры.

Так и получается, что материал состоит из базового цвета, блока текстур и отражающих шейдеров.

4.2. Создание и настройка материала

Материал имеет огромное количество настроек, и все они располагаются на специальной панели окна **Properties** (рис. 4.1).

Новый объект, добавленный в сцену, не имеет присоединенного материала. Сделать это можно двумя способами: либо создать новый материал, либо выбрать имеющийся из списка. В первом случае необходимо нажать кнопку на панели **New**. Обратите внимание на стилизованное изображение сферы рядом с кнопкой **New**. При нажатии на нее появится список имеющихся материалов в сцене. По умолчанию там уже находится один материал, который присвоен начальному кубу. Интересно, что если удалить объект из сцены, то его материалы и текстуры останутся в наличии.

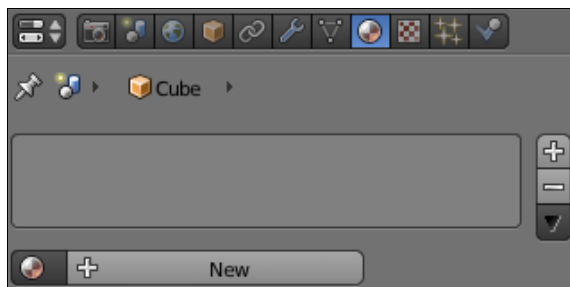


Рис. 4.1. Закладка **Material** окна **Properties**

Поиграем с материалами на практике. Создайте любой **Mesh**-объект в новой сцене и откройте панель материалов. Как и предполагалось, она пустая (см. рис. 4.1). Нажмите на сферу рядом с кнопкой **New** и выберите из списка материал. Вот теперь панель наполнилась большим количеством опций. Будем рассматривать их по блокам. Самый верхний уровень занимают глобальные настройки (рис. 4.2).

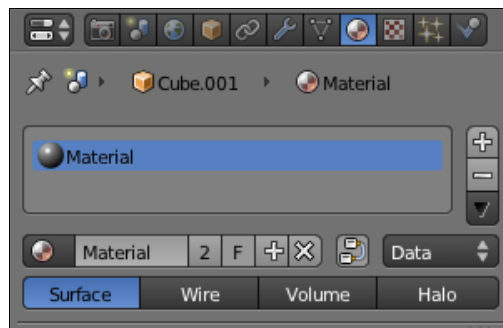


Рис. 4.2. Глобальные опции материалов

Этот блок позволяет управлять присоединением и отсоединением материалов к объекту. Здесь не зря слово "материалы" использовано во множественном числе. Ведь к одному объекту можно прикрепить разное их количество. Самое крупное окошко в данном блоке как раз и показывает материалы, прикрепленные к объекту. Рядом с этим списком находятся дополнительные кнопки для управления порядком множества материалов данного примитива. Но тема мультиматериалов достаточно обширная и работа с ними будет рассматриваться в отдельном разделе, а пока обратите внимание на строку, расположенную чуть ниже этого окошка.

Здесь позволяет изменить имя материала. Просто щелкните по названию и введите новое имя. Вообще возьмите за правило давать понятные названия своим материалам. В дальнейшем это позволит с легкостью находить нужный из списка.

Обратите внимание на цифру 2 рядом с названием материала. Так Blender показывает, сколько объектов использует данный материал. В вашем случае такими примитивами являются куб, который находится в сцене по умолчанию, и новый объект. Запомните, если вы меняете настройки такого общего материала, то все изменения отразятся на всех родительских объектах. Чтобы сделать присоединенный

материал уникальным для текущего объекта, нужно щелкнуть мышью по этой цифре. Сделайте это.

Теперь откройте список материалов. Как видите, в нем уже находятся два пункта (рис. 4.3).

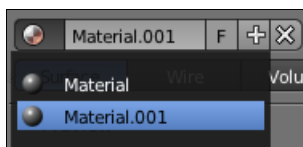


Рис. 4.3. Список выбора материалов

Если материал не имеет родительского объекта, т. е. ни к чему не прикреплен, то после закрытия программы он удаляется из сцены. Чтобы этого не произошло, нажмите кнопку **F**.

Остальные две кнопки вам уже знакомы. Первая со знаком плюс (+) создает новый материал, а вторая (×) удаляет его.

Следующие четыре кнопки переключают режим работы (см. рис. 4.2):

- ◆ **Surface** (Поверхность) — стандартный режим работы материала, который устанавливается по умолчанию. В этом случае объект визуализируется, как есть;
- ◆ **Wire** (Решетка) — в этом случае рендерится структура объекта, т. е. ребра с вершинами без полигонов. Внешне это похоже на режим **Wireframe** окна **3D View**;
- ◆ **Volume** (Объем) — интересный режим, который позволяет обрабатывать объекты с сохранением их формы, как будто они заполнены объемным туманом;
- ◆ **Halo** (Ореол) — структура объектов визуализируется в виде светящихся элементов.

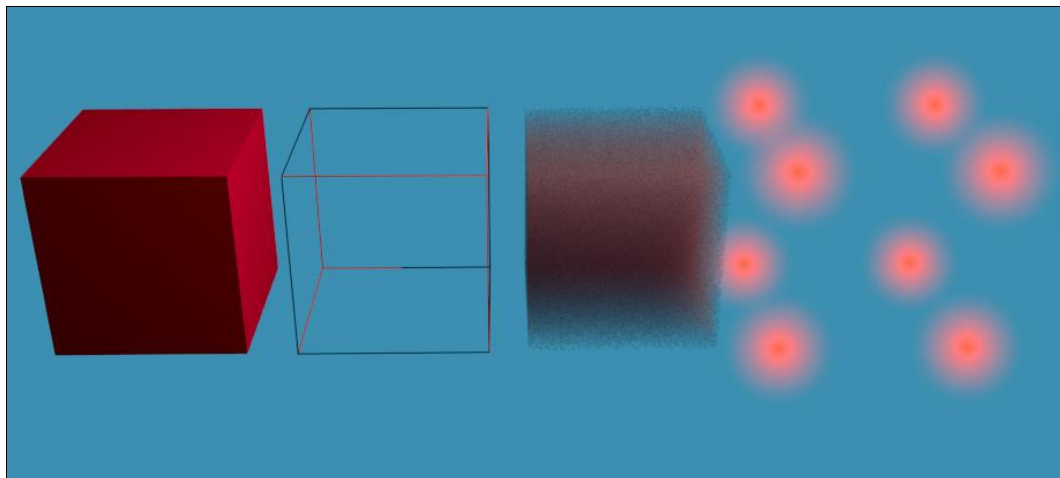


Рис. 4.4. Результат рендера материалов: **Surface**, **Wire**, **Volume**, **Halo**

Закладка **Preview** обеспечивает просмотр материала так, как он будет выглядеть после рендера. Дополнительные кнопки сбоку окна позволяют выбирать типы примитивов для отображения превью. Всего их имеется шесть штук (рис. 4.5): плоскость, сфера, куб, Monkey, частицы (волосы), сфера с цветной заливкой заднего фона.

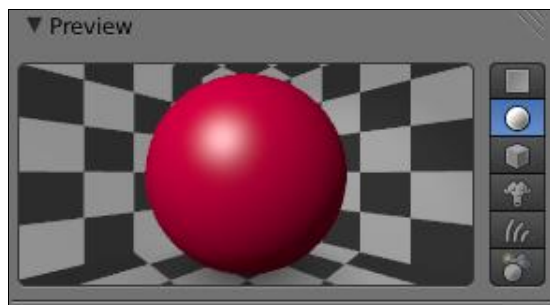


Рис. 4.5. Панель Preview материала

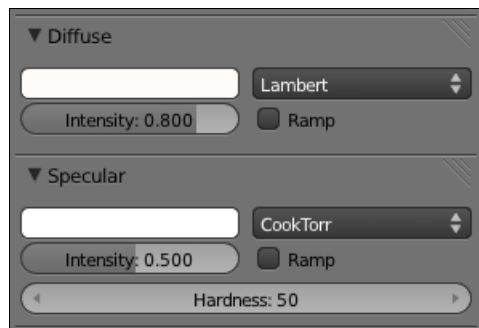


Рис. 4.6. Настройки цвета и отражения материала

Закладки **Diffuse** и **Specular** отвечают за настройку основного цвета и отражающих шейдеров (рис. 4.6).

Shading (Затемнение) содержит опции, отвечающие за реакцию материала на источник света (рис. 4.7).

- ◆ **Emit** (Свечение). Управление свечением объекта. С его помощью можно создать, например, материал светящейся лампы.
- ◆ **Ambient** (Окружение) и **Translucency** (Полупрозрачность). Эти параметры отвечают за поведение материала при включенных опциях глобального освещения, т. е. освещения сцены, вне зависимости от существующих в ней ламп.
- ◆ **Shadeless** (Однотонный). Включение этой опции заставляет материал не реагировать на источники света. Визуально он будет выглядеть однотонно, без полутеней.

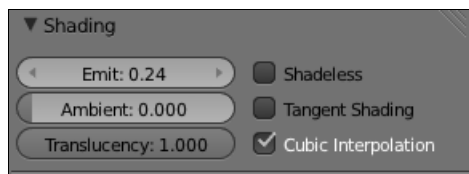


Рис. 4.7. Параметры Shading

- ◆ **Tangent Shading** (Оттенок под углом). Специальный режим работы отражающих шейдеров, который позволяет создавать не точечные блики, а под определенный угол. Например, у металлической ножки реального стола блик от лампы имеется на протяжении всей ножки. Нечто подобное создает данная опция.

- ♦ **Cubic Interpolation** (Кубическая интерполяция). Установка данной опции позволяет создавать более мягкие переходы полутонов. Особо это важно для работы некоторых типов шейдеров.

Параметры закладок **Transparency** (Прозрачность), **Mirror** (Отражение) и **Sub-surface Scattering** (Глубина рассеивания) отвечают за отражение и преломление лучей света. Они будут рассмотрены в соответствующем разделе.

Опции закладки **Strand** (Нить) обычно используются при работе с частицами. Позволяют более точно настраивать материал, характерный для таких объектов, как волосы.

А вот следующую закладку **Options** (Опции) рассмотрим более тщательно. Здесь располагаются основные, глобальные настройки материала, отвечающие за поведение материала по отношению к сцене (рис. 4.8).

- ♦ **Traceable** (Трассировка). Установка этого параметра позволяет материалу участвовать в трассировке лучей. Допустим, в сцене имеется зеркало, которое отражает рядом находящиеся объекты. Если выключить **Traceable**, то объект с данным материалом будет невидимым для этого зеркала. Такой материал не участвует в отражении и преломлении.

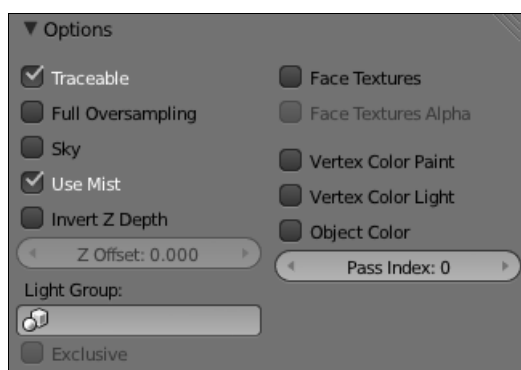


Рис. 4.8. Группа Options

- ♦ **Full Oversampling** (Полное сглаживание). В параметрах рендера сцены имеется возможность установки сглаживания (**Anti-Aliasing**). Если включить эту опцию, то материал будет обрабатываться в максимальном качестве. Учтите, в этом случае время обработки значительно увеличивается.
- ♦ **Sky** (Небо). В этом случае цвет материала меняется на цвет фона сцены (Sky).
- ♦ **Use Mist** (Использовать туман). Blender имеет возможность использовать в сцене туман (не путать с режимом **Volume** материала). Отключите ее, если не хотите, чтобы объект реагировал на туман.
- ♦ **Face Textures** (Текстурированные грани) и **Face Textures Alpha** (Текстурированные грани с прозрачностью). В этом случае основной цвет материала будет заменен на вывод текстуры в обычном режиме и с альфа-каналом (с прозрачностью).

- ◆ **Vertex Color Paint** (Окраска вершин). Blender позволяет раскрашивать в разные цвета вершины объекта. Установка этой опции заменит базовый цвет материала на раскраску вершин.
- ◆ **Vertex Color Light** (Освещение вершин). Дополнительное освещение от окрашенных вершин.
- ◆ **Light Group** (Группа света). Материал можно поместить в группу отдельного источника света.

Группа **Shadow** позволяет настраивать тени (рис. 4.9). По умолчанию объект с новым материалом умеет отбрасывать и принимать тени, в некоторых случаях это нежелательно. Если вы хотите, чтобы ваш материал не принимал чужие тени, то снимите галочку с пункта **Receive** (Принятие). Чуть ниже находится пункт **Receive Transparent** (Принять прозрачность). По умолчанию он отключен. Активация данной опции позволит объекту принимать тени от прозрачных материалов.

Бывает и обратная ситуация, когда нужна тень, но без самого объекта. Опция **Cast Only** (Только отбрасывать) отключит рендер объекта и оставит только тень.

Необычна по результату опция **Shadows Only** (Только тени). Смысл ее действия в том, что тень остается, а сам материал принимает цветность фона сцены. Имеется несколько вариантов, доступных в расположенном ниже меню **Shadow Only Type: Shadows and Shading** (Тени и полутени), **Shadow Only** (Только тень), **Shadow and Distance** (Тень и расстояние). Например, в первом случае объект будет выглядеть как обычно, только с цветами фона, а вот в последнем варианте он сольется с фоном сцены.

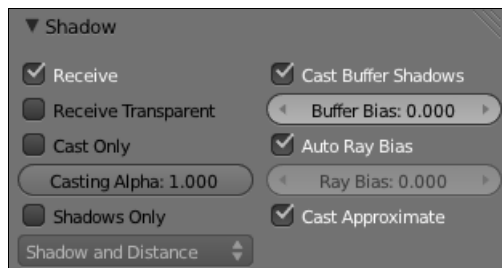


Рис. 4.9. Настройки тени

4.3. Базовый цвет и отражение

Цвет материала состоит из двух составляющих: основного (диффузного) и бликового (зеркального) шейдеров. Оба элемента имеют свои собственные настройки в соответствующих группах **Diffuse** (Рассеянный или диффузный) и **Specular** (Зеркальный).

И вот уже, в который раз, в тексте книги мелькает слово "шейдеры". Обычно возникает ассоциация с техническими характеристиками 3D-ускорителей. Действительно, шейдерами называют мини-программы, которые выполняются на уровне

видеокарт, что обеспечивает их высокое быстродействие. В мире Blender шейдерами являются специальные функции, ответственные за постобработку цвета материала. А вот как они выполняются, на уровне видеоплаты или центрального процессора — это несущественно. Правильная настройка шейдеров материала позволяет достичь нужной реалистичности.

Группа **Diffuse** (Диффузный) образует так называемый основной цвет материала (см. рис. 4.6). По умолчанию он серый. Чтобы выбрать иной, нужно щелкнуть левой кнопкой мыши по образцу. Откроется специальная панель (рис. 4.10). Выбранный в ней цвет сразу же отображается в окне **Preview** и на родительском объекте.

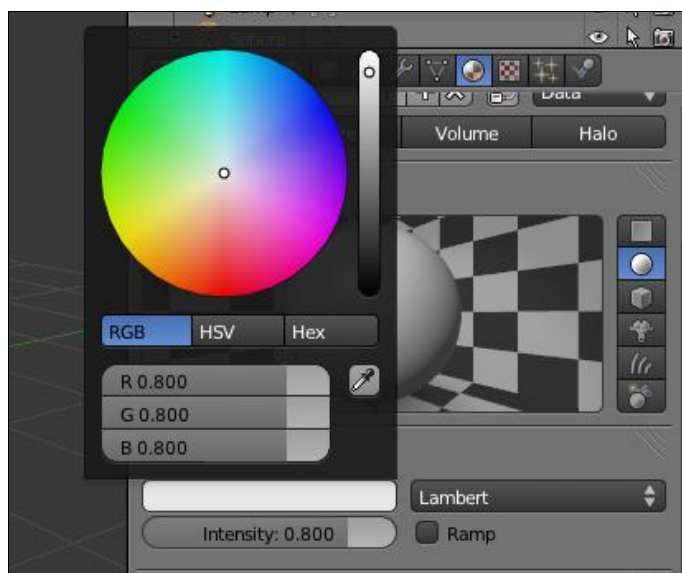


Рис. 4.10. Палитра цвета

Шкала **Intensity** позволяет установить интенсивность цвета (см. рис. 4.11). А вот выбор шейдерной модели осуществляется из меню **Diffuse Shader Model** (Диффузный шейдер).

Для диффузного цвета доступно пять шейдерных функций, названных по имени их создателей.

- ◆ **Lambert.** Этот шейдер установлен по умолчанию и обеспечивает равномерное отражение света от поверхности.
- ◆ **Oren-Nayar.** При выборе становится доступен параметр **Roughness** (Шероховатость). Если честно, визуально не особо отличается от **Lambert**.
- ◆ **Toon.** Очень интересный нереалистичный шейдер, создающий эффект мультипликации. Это происходит за счет достаточно резких переходов между освещенными и темными областями. Имеется два параметра: **Size** (размер освещенной области) и **Smooth** (степень сглаженности между областями). Для лучшего результата параметр **Smooth** должен иметь минимальное значение.

- ♦ **Minnaert.** Этот шейдер создает ощущение мягкой, бархатистой поверхности. Имеется всего один параметр **Darkness** (Затемнение). Оптимальный результат при нулевом значении этого параметра.
- ♦ **Fresnel.** Этот шейдер, обратный по результату **Lambert**. В отличие от последнего, места, куда падают лучи источника света, кажутся более темными, а вот затененные, наоборот, просветляются. Имеется два параметра: **Fresnel** (степень выделения областей) и **Factor** (резкость границ).

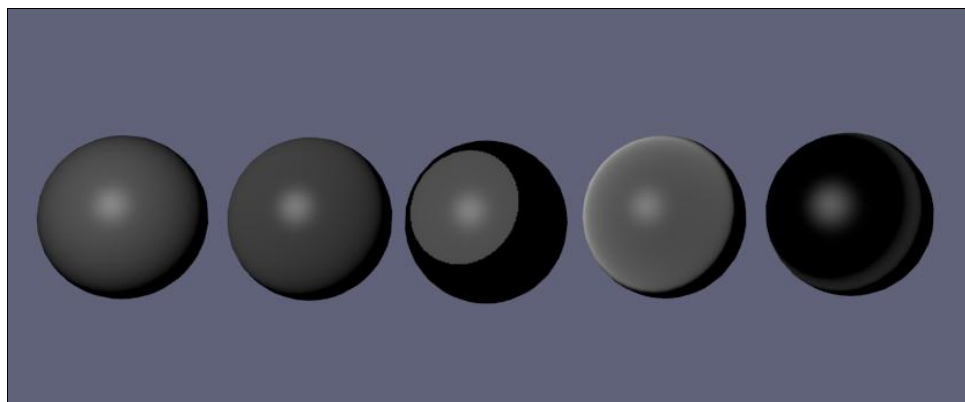


Рис. 4.11. Диффузные шейдеры: **Lambert**, **Oren-Nayar**, **Toon**, **Minnaert**, **Fresnel**

Шейдеры **Specular** или, как их еще называют, зеркальные шейдеры отвечают за создание бликов на объекте (рис. 4.12). В эту группу входят пять функций.

- ♦ **CookTorr.** Основной зеркальный шейдер, установленный по умолчанию. Дает размытые блики. Неплохо подходит для визуализации пластмассовых поверхностей. Имеется параметр **Hardness**, который отвечает за площадь излучения блика.
- ♦ **Phong.** Шейдер, наподобие **CookTorr**. Имеется такой же параметр **Hardness**. Разработчики Blender советуют использовать его для органических поверхностей.
- ♦ **Blinn.** В отличие от первых двух, этот шейдер имеет индекс преломления **IOR** (Index of Refraction), что является более качественным по физическим законам.
- ♦ **Toon.** Знакомый уже "мультяшный" шейдер, но только бликовый. Параметры те же самые, что и у диффузного собрата. Оптимально использовать шейдеры **Toon** в паре друг с другом.
- ♦ **Wardiso.** Уникальный шейдер, позволяющий создавать блики с резкими очертаниями. Такие блики характерны, скажем, для металлических поверхностей. Имеющийся параметр **Slope** как раз и позволяет устанавливать жесткость излучения.

Комбинируйте разные типы шейдеров для получения оптимального результата!

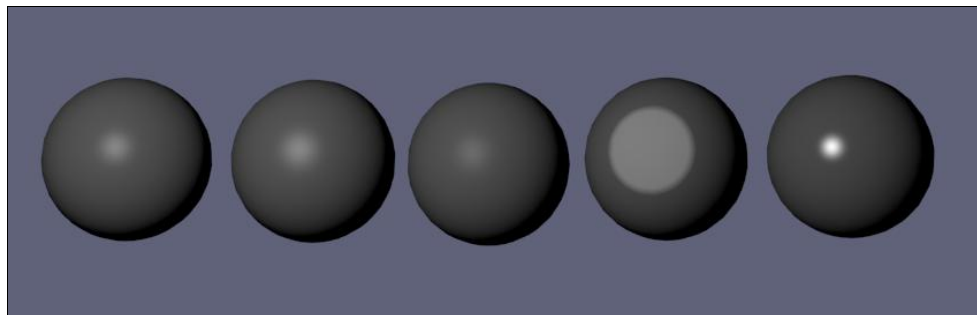


Рис. 4.12. Бликовые шейдеры: CookTorr, Phong, Blinn, Toon, Wardiso

4.4. Рамповые шейдеры

В реальном мире редко бывают предметы с полностью однотонной окраской. При-
смотрите к своей собственной коже. Легко заметить, что в различных местах она
имеет разные оттенки. Причем очень сильно влияет на восприятие расположение
источников света. Или, например, пламя костра, где отчетливо видна градация цве-
тов от светлого к темному.

Для корректировки основного цвета материала Blender предлагает специальный
инструмент — **Ramp** (рамповый шейдер). В основе его работы лежит смешивание
базового цвета с градиентной палитрой, которую пользователь устанавливает сам.

Шейдеры **Ramp** доступны как для диффузного цвета, так и для бликового. По
умолчанию данная функция отключена. Для ее активации достаточно установить
галочку в поле **Ramp** в нужной группе шейдеров (рис. 4.13)



Рис. 4.13. Опция включения Ramp



Рис. 4.14. Параметры Ramp

После включения этой опции становится доступной целая группа параметров
(рис. 4.14).

Построение градиента осуществляется на специальной шкале с помощью меток или
контрольных точек. Визуально метки выглядят, как вертикальные штрихи. По
умолчанию градиент уже имеет две метки от черного к белому цвету. Контрольные

точки можно перемещать по шкале, тем самым изменяя интенсивность и расположение цветов. Для этого служит поле **Pos** (рис. 4.15).

Однако до этого нужно выбрать саму метку, для которой будут производиться изменения. На панели **Ramp** имеется специальное поле, которое указывает номер активной метки. Посмотрите на рис. 4.16. По умолчанию там всегда стоит первая контрольная точка (счет меток в Blender начинается с нуля). Щелкая по угловым стрелкам данного поля, можно выбрать нужную метку.

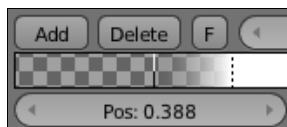


Рис. 4.15. Градиентная шкала и метки

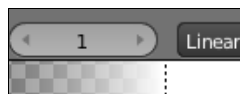


Рис. 4.16. Поле для активации нужной метки

Для изменения цвета конкретного сектора служит поле, расположенное рядом с параметром **Pos**. Щелчком левой кнопки мыши по нему открывается уже привычная палитра цветов (рис. 4.17).

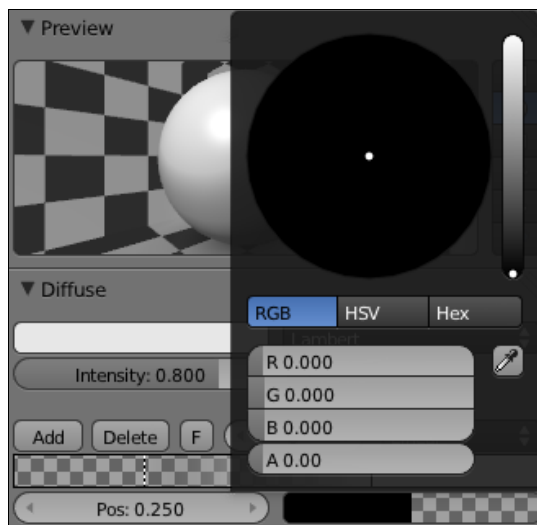


Рис. 4.17. Поле-образец и открытая палитра

Обратите внимание на то, что Blender позволяет устанавливать прозрачность (A — альфа-канал) вместе с выбором цвета. Таким образом, вы можете указать программе, как и в каких пропорциях будут смешиваться базовый цвет с цветом данного сектора. Установка параметра альфы в ноль означает полную прозрачность, т. е. в данной точке будет показываться исключительно основной цвет материала. Заметьте, что если цвет второй метки имеет меньшую прозрачность, то программа выполнит постепенную замену базового цвета цветом контрольной метки. Используя альфа-канал, вы можете либо смешивать цвета, либо полностью заменить его градиентом.

В некоторых случаях двух контрольных точек бывает мало. Blender позволяет добавлять ровно столько меток, сколько нужно. Это делается с помощью кнопки **Add**. После добавления новой точки ее можно переместить на нужную позицию с помощью параметра **Pos** и установить свой цвет. Ненужная метка удаляется с помощью кнопки **Delete**.

Рамповый шейдер имеет еще два важных параметра.

◆ **Input** (Вход). Здесь указывается точка входа для работы шейдера. Имеется четыре варианта:

- **Shader** (Шейдер) — этот параметр установлен по умолчанию. Результатом его действия становится изменение цвета материала под воздействием направления излучения света;
- **Energy** (Мощность) — в этом случае, вместе с направлением излучения учитывается и его мощность;
- **Normal** (Нормаль) — результат работы ориентируется по отношению к камере сцены. Визуально это проявляется, как усиление действия градиента по краям объекта;
- **Result** (Результат) — в отличие от первых трех вариантов, здесь работа шейдера начинается только по завершении всех остальных вычислений. Из-за этой особенности **Result** можно использовать совместно с текстурами.



Рис. 4.18. Меню смешивания **Blend**

◆ **Blend** (Смешивание). Этот параметр указывает, каким именно образом будет производиться смешивание градиента с основным цветом. Нужный вариант выбирается из специального меню (рис. 4.18). По умолчанию установлен режим смешивания **Mix**, который просто показывает слои в соответствии с альфа-каналом. Типы смешивания, представленные в меню, используются и для других инструментов Blender, например каналов текстур. Рассматривать все типы нет смысла, попробуйте их на практике.

Для всех вариантов смешивания имеется глобальный параметр **Factor** (Коэффициент). С его помощью вы можете установить необходимый уровень влияния рампового шейдера на материал.

СОВЕТ

Как и положено, результат работы рампового шейдера отображается в окошке **Preview** панели материала, но это не совсем удобно. Blender предлагает возможность полного просмотра материалов в окне **3D View**, правда с небольшим условием, у вас

должна быть достаточно мощная видеоплата. На панели свойств (<N>) окна **3D View** найдите закладку с названием **Display**. На ней находится список **Shading**. По умолчанию там установлен вывод **MultiTexture**. Замените его на **GLSL**. Найдите в заголовке окна **3D View** меню **Viewport Shading** и выберите пункт **Texture** (см. рис. 1.10). Теперь программа, в зависимости от мощности вашей видеокарты, будет стараться выводить объекты в окне с максимально качественным отображением материалов и текстур.

4.5. Эффекты *Halo*

Материалы Blender имеют особый режим **Halo**, который позволяет выводить вершины объекта в виде светящихся элементов. Это не значит, что материал начинает функционировать наподобие источников света. В действительности свечение **Halo** не оказывает влияния на рядом находящиеся объекты. С помощью этого режима можно создавать весьма любопытные эффекты, например свечение звезд или фары автомобиля. Часто **Halo** используют для визуализации так называемых линзовых эффектов (**Lens Flares**). Подобные блики можно в реальном мире увидеть при фото- или видеосъемке, расположившись под определенным углом к источнику света.

Для работы с **Halo** необходимо переключить материал, щелкнув по одноименной кнопке в настройках. При этом панель существенно образом изменится (рис. 4.19).

Как уже было сказано, **Halo** применяется только к вершинам объекта, причем использовать удастся лишь **Mesh**. После его активации примитив примет весьма необычный вид. От того же привычного куба в сцене останутся лишь восемь точек. Таким образом, Blender показывает, что данный объект визуализироваться полностью не будет.

Всего программа предлагает четыре эффекта **Halo**, не считая **Flare**, который располагается в другой группе (рис. 4.20):

- ◆ **Halo** (Ореол) — это основной тип, который всегда включен по умолчанию. Визуально выглядит, как светящаяся точка;
- ◆ **Rings** (Кольца) — окружности вокруг ореола. Количество колец можно регулировать в одноименном поле;
- ◆ **Lines** (Линии) — хаотично выходящие прямые из центра **Halo**. Имеется поле количества **Lines**;
- ◆ **Star tips** (Звездные лучи) — эффект наподобие **Lines**, но в отличие от него, лучи располагаются на одинаковом расстоянии друг от друга. Количество также можно регулировать параметром **Star Tips**.

Группа **Halo** имеет как общие опции для всех типов эффектов, так и индивидуальные. К примеру, **Rings** и **Lines**, кроме параметра количества, предлагают индивидуально изменять свой цвет. Это можно сделать, щелкнув по образцу соответствующего эффекта.

К общим параметрам относятся:

- ◆ **Alpha** (Прозрачность);
- ◆ **Size** (Размер);

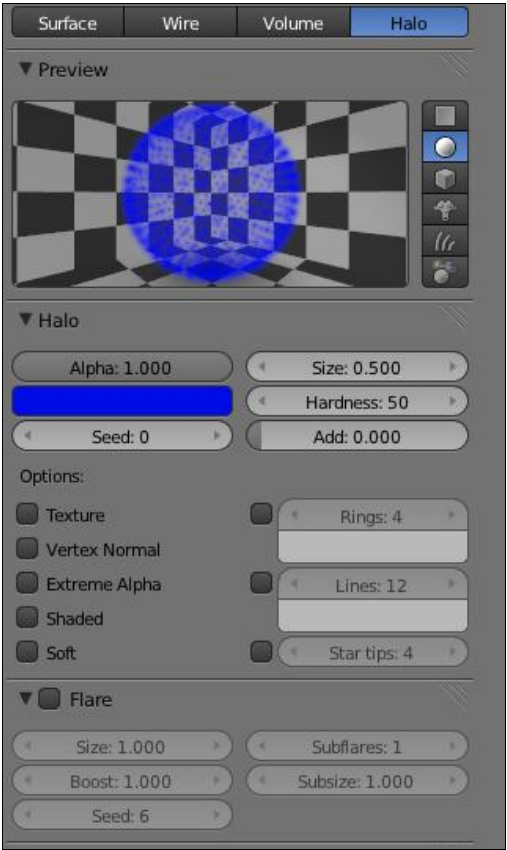


Рис. 4.19. Настройки Halo и Flare

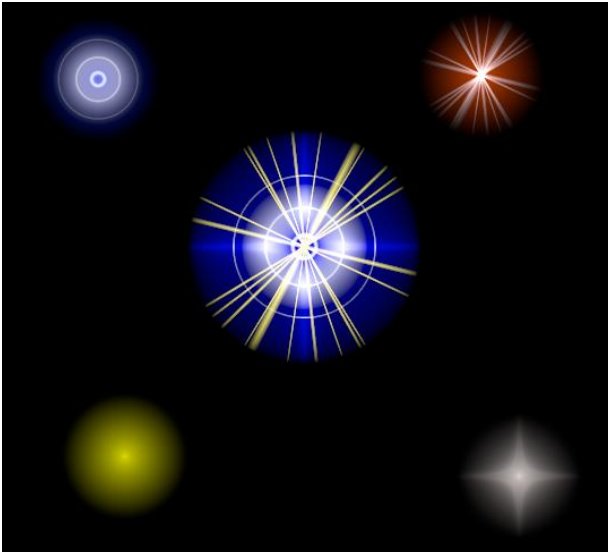


Рис. 4.20. Эффекты Halo

- ◆ **Hardness** (Жесткость) — чем меньше значение, тем интенсивнее излучение **Halo**. К другим эффектам это не относится;
- ◆ **Seed** (Разброс) — изменяя этот параметр, можно добиться различного расположения элементов у эффектов **Rings** и **Lines**;
- ◆ **Add** (Добавить) — степень воздействия **Halo** на материал для дополнительных опций:
 - **Texture** (Текстура) — если включено, то **Halo** взаимодействует с текстурами данного материала;
 - **Vertex Normal** (Нормаль вершины) — в стандартном варианте эффекты всегда направлены "лицом" по отношению к камере. Установка этой опции заставит **Halo** создавать свечение в сторону нормали вершин. Таким образом, яркость и размер эффектов зависит от угла поворота вершины;
 - **Extreme Alpha** (Сверхпрозрачность) — альфа-канал с более высоким качеством;
 - **Shaded** (Затененные) — включите эту опцию, чтобы эффекты **Halo** реагировали на источники света и тени. По умолчанию эта функция отключена;
 - **Soft** (Мягкость) — если пересечение **Halo** с геометрией других объектов вызывает артефакты, то эта опция позволит смягчить такие переходы.

Теперь займемся эффектом **Flare** и рассмотрим работу с ним на несложном примере.

В книгах по трехмерному моделированию очень часто приводят урок по созданию космической сцены с участием Земли и Солнца. Наверное, потому, что он несложен в разработке, но зрелищно эффектен. Правда, вы пока еще не дошли до раздела, описывающего работу с текстурами, но вот линзовый эффект с участием обычной сферы создать сумеете.

Эффект **Lens Flares** в Blender не следует всем физическим законам, а лишь призван визуально улучшать картинку. Тем не менее расположение колец, сила излучения и цветовая насыщенность зависят от многих факторов, в том числе от местоположения объектов. В этом примере, в сцене будут участвовать четыре объекта: камера, источник света, сфера и плоскость. Поэтому приготовьтесь манипулировать ими в точности с указаниями. Удобнее всего для этого будет пользоваться панелью свойств окна **3D View** (вызывается клавишей <N>).

Создайте новый проект и удалите куб за ненадобностью. Добавьте в сцену примитив сферы из меню **Add | Mesh | UV Sphere**. Откройте панель свойств объекта. В группе **Location** поменяйте параметры на следующие значения:

- ◆ X = 1.3
- ◆ Y = 1.3
- ◆ Z = -3

В группе **Scale** установите значение для всех координат, равное 3.8. Не мешает также добавить сглаживание к сфере. Это можно сделать, щелкнув кнопку **Smooth** на панели **Tool Shelf**.

Теперь нажмите клавишу <A> для снятия выделения со всех объектов и добавьте в сцену плоскость (**Add | Mesh | Plane**). Переместите примитив в любое место сцены, чтобы он не перекрывался сферой.

Плоскость будет играть роль Солнца. Точнее не вся целиком, а лишь одна из ее вершин. Дело в том, что в Blender не предусмотрена возможность добавления в сцену объекта с всего одной вершиной, и самым простым вариантом является использование плоскости с удаленными тремя точками.

Войдите в режим редактирования (<Tab>) и выделите любые три вершины (в этом случае удобно использовать рамку, вызываемую клавишей в просмотре **Top View**). Нажмите клавишу <X> для их удаления.

Итак, у нас от плоскости осталось всего одна точка. Вот только центр объекта лучше совместить с оставшейся вершиной. Выйдите из режима редактирования с помощью клавиши <Tab>. На заголовке окна **3D View** выберите из меню **Object** функцию **Transform | Origin to Geometry**.

Поместим будущее Солнце в нужном месте. На панели свойств в группе **Location** поменяйте параметры:

- ◆ X = -6
- ◆ Y = 15
- ◆ Z = -5.5

Найдите в окне **Outliner** объект **Lamp** и выделите его. Это источник света по умолчанию. Немного передвинув его, вы получите более правильную композицию. На панели свойств в группе **Location** установите следующие значения:

- ◆ X = 0.5
- ◆ Y = 7
- ◆ Z = 3

Перейдите в режим просмотра камерой с помощью клавиши <NumPad 0> и нажмите <F12> для быстрого рендеринга. Если у вас получилось то же самое, что на рис. 4.21, то подготовка сцены завершена.

Сейчас займемся настройкой **Lens Flares**. В окне **Outliner** выделите объект **Plane**. Откройте панель **Material** и добавьте новый материал с помощью кнопки **New**. Переключите его в режим **Halo**.

Основным цветом Солнца будет белый. По умолчанию в настройках **Halo** установлен светло-серый (рис. 4.22). Поэтому щелкните по образцу цвета и в параметрах RGB введите значение 1 (щелчок левой кнопки мыши по нужному полю позволит ввести в него цифру).

Остальные параметры группы **Halo** введите из следующего списка:

- ◆ **Seed** = 5
- ◆ **Size** = 1
- ◆ **Hardness** = 50

- ◆ **Add** = 0
- ◆ **Rings** = 4
- ◆ **Lines** = 12

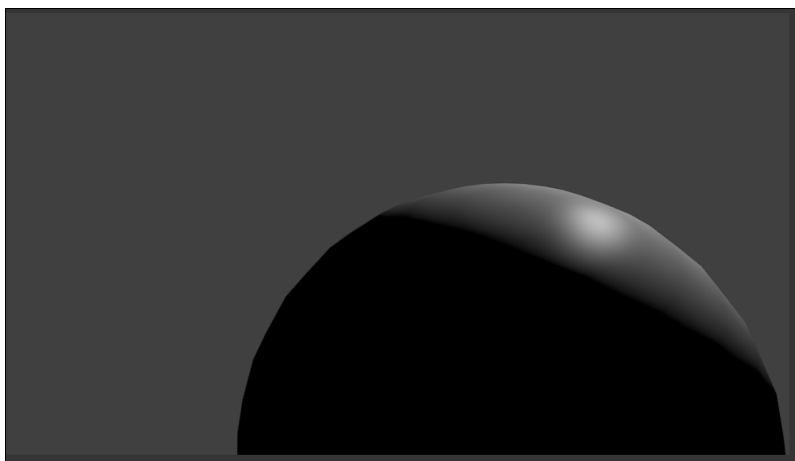


Рис. 4.21. Заготовка для Flare

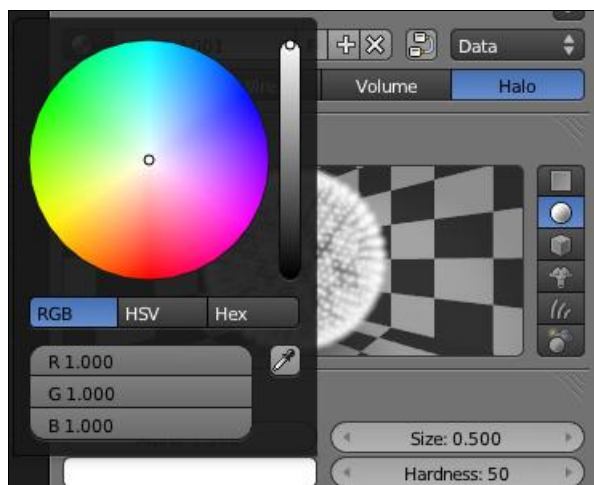


Рис. 4.22. Установка цвета для Halo

Включите опции **Rings** и **Lines**, установив галочки в соответствующих полях. В качестве цвета для обоих будем использовать светло-розовый с параметрами RGB:

- ◆ **R** = 1
- ◆ **G** = 0.75
- ◆ **B** = 0.63

Рис. 4.23. Настройки **Flare**

Эффект **Flare** активируется путем установки галочки у соответствующей надписи. После этого становится доступной целая группа опций (рис. 4.23).

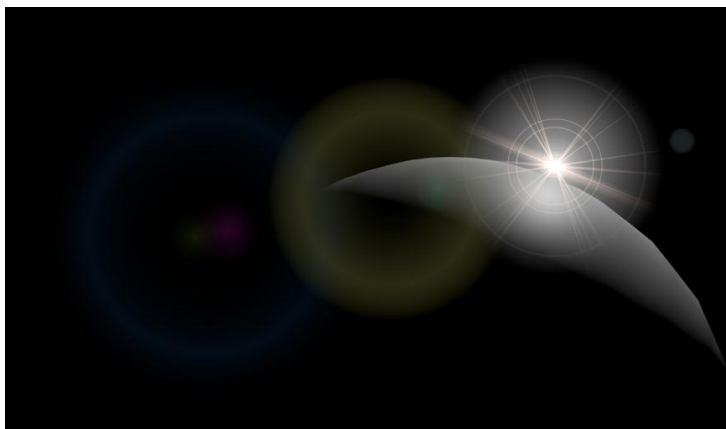
Рассмотрим параметры **Flare**:

- ◆ **Size** (Размер) — общий размер для колец;
- ◆ **Boost** (Усиление) — качество прорисовки. Большее значение дает более яркую картинку;
- ◆ **Seed** (Смешивание) — с помощью данного параметра можно перемешать результат работы функции **Flare**;
- ◆ **Subflares** (Дополнительные вспышки) — количество колец;
- ◆ **Subsize** (Размер вспышек) — максимальный размер самого первого кольца.

Как видите, параметры **Flare** простые, но путем правильной подборки значений можно добиться качественного результата.

Установите следующие значения для этих параметров и нажмите <F12> для обработки (рис. 4.24):

- ◆ **Size** = 5
- ◆ **Boost** = 0.2
- ◆ **Seed** = 6
- ◆ **Subflares** = 7
- ◆ **Subsize** = 3

Рис. 4.24. Эффект **Lens Flares** в действии

4.6. Мультиматериалы

В *разд. 4.4* рассказывалось о рамповых шейдерах, которые позволяют превращать однотонный материал в разноцветный градиент, но это действие распространяется только на весь объект. По сути дела, градиент — это возможность корректировки базового цвета. Blender предлагает мощный и удобный инструмент для присвоения отдельным частям модели собственных материалов.

Как и положено, работа с мультиматериалами осуществляется на панели **Material** окна **Properties** (рис. 4.25).

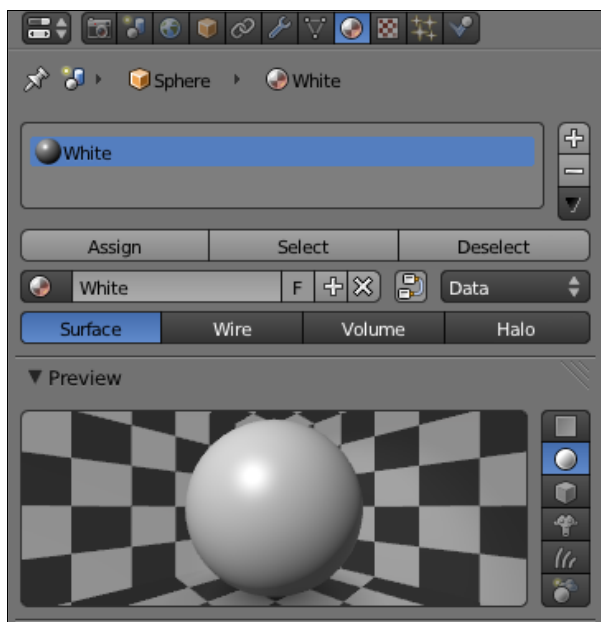


Рис. 4.25. Панель материалов

Работа с мультиматериалами возможна в режиме редактирования объекта. При его включении в основной группе настроек становятся доступными дополнительные кнопки: **Assign**, **Select**, **Deselect**.

Посмотрите на рис. 4.25. В большом окошке выделено слово "White". Это название материала, присвоенного к объекту. Если вы попытаетесь выбрать или создать новый материал, то "White" будет заменяться соответствующим именем. По умолчанию к объекту прикреплен только один материал, и действие его распространяется на всю структуру.

Для присоединения еще одного материала к объекту нужно сначала выделить в режиме редактирования область структуры примитива (рис. 4.26).

Нажатие кнопки с изображением плюса рядом со списком в окне свойств добавит новый пустой блок. Таким образом указывается, что к данному объекту прикреплено два материала. По умолчанию этот блок не содержит конкретного материала и

программа предлагает создать новый. Можно нажать кнопку **New**, а можно выбрать из списка уже имеющийся материал (см. рис. 4.26).

Но новый материал пока не задействован. Вот тут-то и пригодятся дополнительные кнопки: **Assign**, **Select**, **Deselect**.

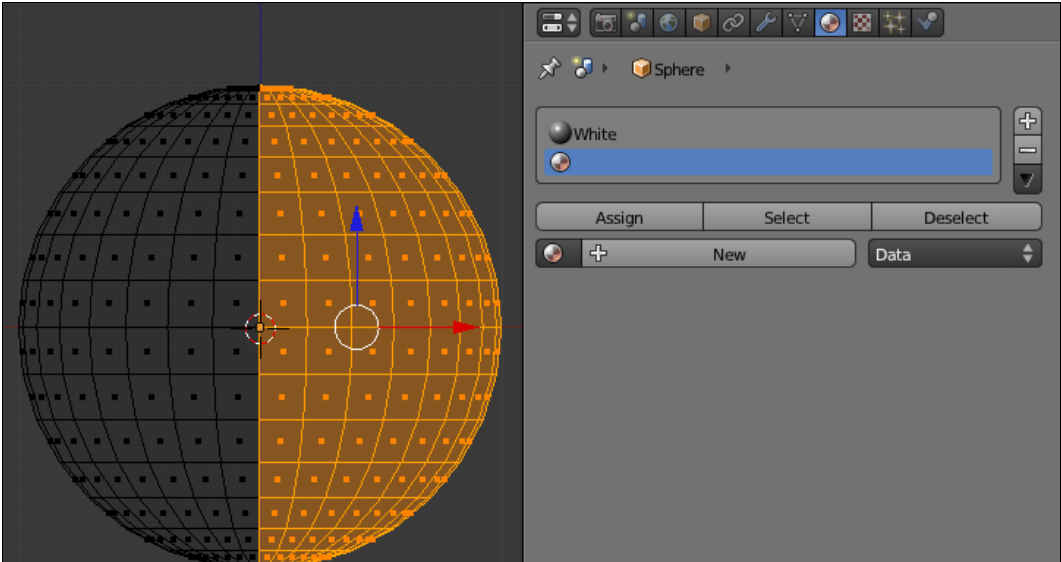


Рис. 4.26. К этому выделению прикреплен новый материал

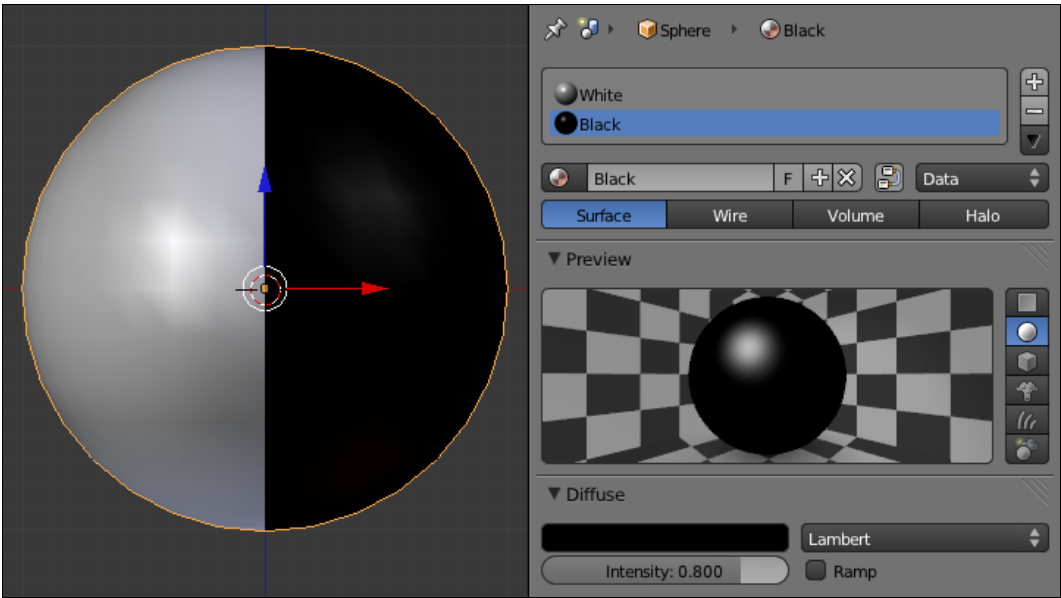


Рис. 4.27. Сфера с двумя материалами

Нажатие кнопки **Assign** (Присоединить) применит выбранный материал в списке к выделенной области объекта. Кнопки **Select** и **Deselect** служат для контролирования области воздействия материала. Нажатие кнопки **Select** заставит программу выделить в окне редактирования область вершин, к которой привязан материал, а **Deselect** соответственно снимает выделение (рис. 4.27).

Для удаления материала имеется кнопка "-", расположенная рядом со списком.

4.7. Отражение и преломление

Вы уже знаете, что цвет материала можно сделать прозрачным с помощью альфа-канала, а отражение бликов осуществляется зеркальными шейдерами. Но это малая часть того, что позволяет делать Blender с прозрачными и зеркальными объектами.

Существует специальная технология трассировки лучей (Ray tracing), которая предназначена для точного расчета преломления и отражения в соответствии с законами физики. Смысл ее заключается в том, что лучи света, попадающие на объекты, многократно отражаются от них, постепенно теряя насыщенность в соответствии с настройками преломления. Ray tracing — чрезвычайно ресурсоемкая задача, поэтому в Blender имеется возможность выбора более простых алгоритмов. Рассмотрим сначала создание прозрачного материала.

Все необходимые настройки находятся в группе **Transparency** панели **Material** (рис. 4.28).

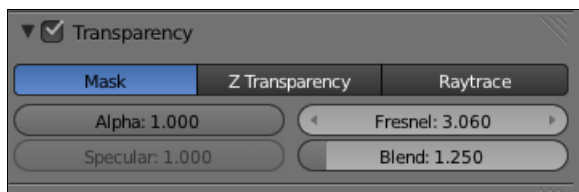


Рис. 4.28. Настройки прозрачности

Blender предлагает три режима установки прозрачности по степени качества и трудоемкости вычисления:

- ♦ **Mask** (Маска) — самый простой вариант. Объект становится прозрачным, но только для фона сцены;
- ♦ **Z Transparency** (Прозрачность по Z) — в этом случае, сквозь такой материал становятся видны другие объекты, но не учитывается преломление;
- ♦ **Raytrace** (Трассировка лучей) — полноценная прозрачность с расчетами преломления и размывания. Такой режим позволяет, например, точно моделировать преломление лучей света в жидкостной среде.

Выбор нужного режима осуществляется одноименными кнопками. Для каждого вида становятся доступными определенные группы опций.

Режим **Mask** (рис. 4.29) предлагает следующие настройки:

- ♦ **Alpha** (Альфа) — общая прозрачность материала;

- ♦ **Fresnel** — этот параметр отвечает за степень прозрачности, которая изменяется в зависимости от угла просмотра. Чем меньше угол между нормалью объекта и точкой просмотра, тем прозрачнее материал;
- ♦ **Blend** (Смешивание) — степень смешивания между прозрачными и непрозрачными областями. Большее значение — большая прозрачность.

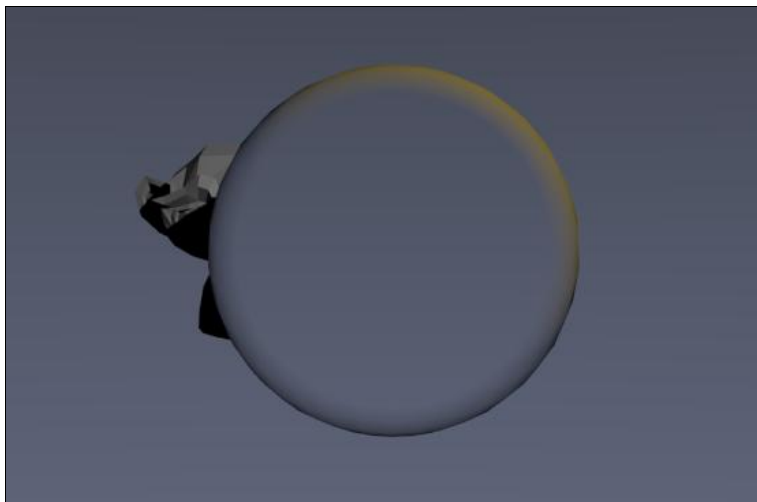


Рис. 4.29. Режим **Mask** позволяет видеть только фон сцены

Для **Z Transparency** к ранее рассмотренным опциям добавляется всего одна — **Specular** (Зеркальный). Это не что иное, как уже знакомый блик от зеркального шейдера. Увеличение его значения приводит к более яркому отблеску (рис. 4.30).

А вот **Raytrace** обогащается большим количеством настроек (рис. 4.31).



Рис. 4.30. Режим **Z Transparency** делает объекты прозрачными и с бликами

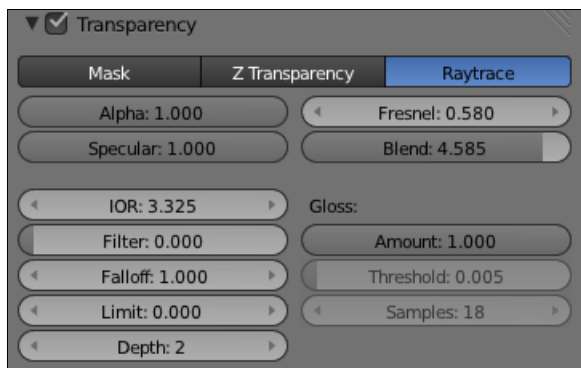


Рис. 4.31. Настройки Raytrace

Левая группа дополнительных опций отвечает за настройку преломления материала:

- ◆ **IOR** (Index of Refraction, индекс преломления) — основной параметр, отвечающий за индекс преломления. В реальном мире у каждого прозрачного предмета имеется свой индекс. На официальном сайте Blender есть замечательная таблица для многих материалов. Так обычное стекло будет иметь **IOR** равным 1.517;
- ◆ **Filter** (Фильтр) — вы можете использовать этот параметр для контролирования основного цвета прозрачного материала. Чем выше значение, тем больше базового цвета будет использоваться;
- ◆ **Falloff** (Спад) — ограничение поглощения луча материалом. Может использоваться для настройки толщины стекла;
- ◆ **Limit** (Лимит) — используется совместно с параметром **Filter**. Устанавливается порог, за которым начинает действовать базовый цвет;
- ◆ **Depth** (Глубина) — глубина просчета. Вы можете установить более высокое значение, если считаете, что качество трассировки неудовлетворительное. Однако это приводит к увеличению нагрузки на систему.

На рис. 4.32 показана прозрачная сфера с индексом преломления для обычного стекла. В качестве зеркального шейдера использовался **Wardiso**, дающий более жесткий отблеск, что характерно для стекла.

Режим **Raytrace** предлагает еще одну интересную возможность — создание матовых поверхностей (рис. 4.33). Эти настройки располагаются в группе **Gloss**:

- ◆ **Amount** (Количество) — при максимальном значении ясность преломления не изменяется и остальные опции отключены. Чем ниже значение, тем более матовое изображение получается;
- ◆ **Threshold** (Порог) — порог чувствительности. При определенном значении действие останавливается;
- ◆ **Samples** (Образцы) — качество зерна. Чем выше значение, тем более мелкое получается зерно. При этом пропорционально увеличивается время обработки.

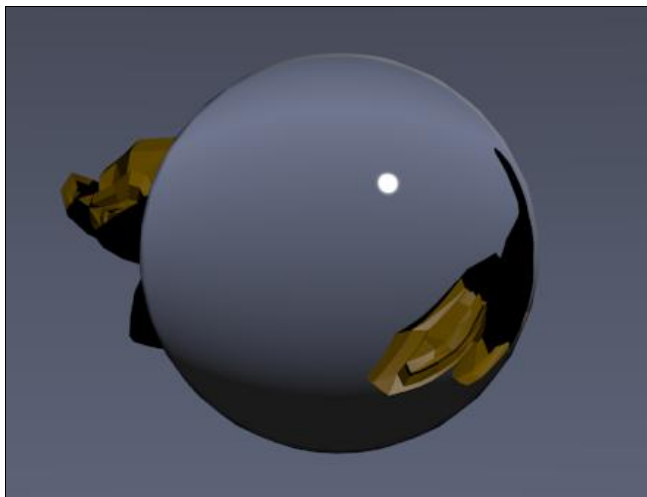


Рис. 4.32. Стекло́нный шар

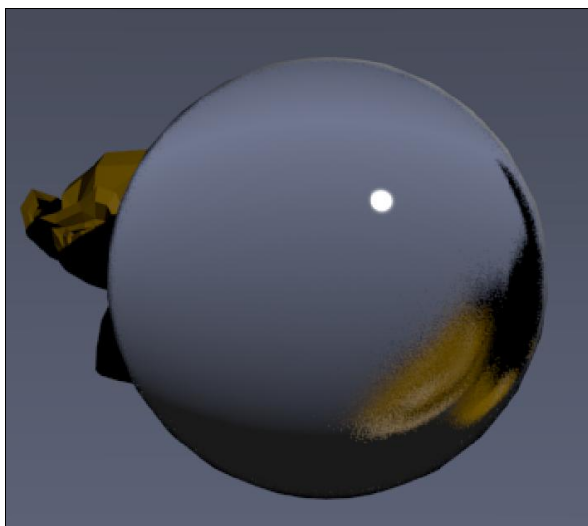


Рис. 4.33. Матовый шар

На этом закончим рассмотрение прозрачности и перейдем к созданию отражающих материалов.

Группа **Mirror** (Отражение) позволяет настроить материал так, что он начнет отражать в себе окружающие предметы (рис. 4.34). Эта функция также построена на технологии трассировки лучей, что обеспечивает высокое качество отображения. **Mirror** можно использовать одновременно с **Transparency**. Для включения этой возможности нужно установить галочку рядом с **Mirror** (рис. 4.35).

Рассмотрим предлагаемые программой настройки **Mirror**:

- ◆ **Reflectivity** (Коэффициент отражения) — качество отражения. При минимальном значении материал ничего не отражает;

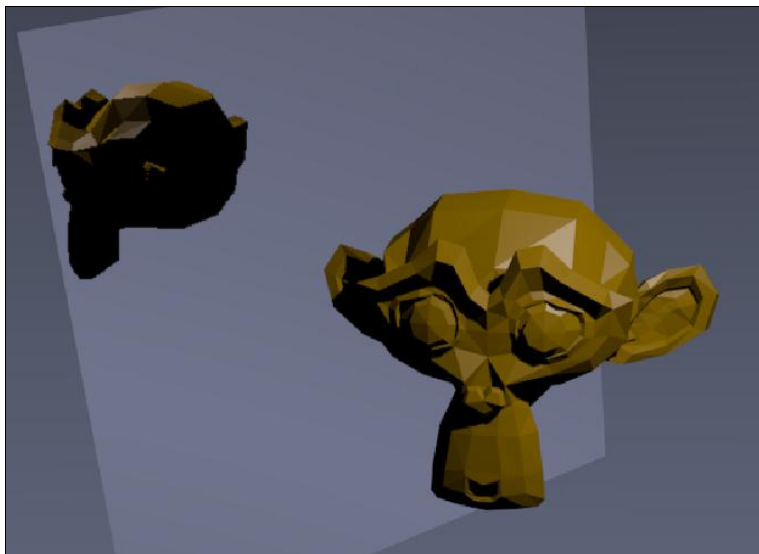


Рис. 4.34. Отражение обезьянки в плоскости

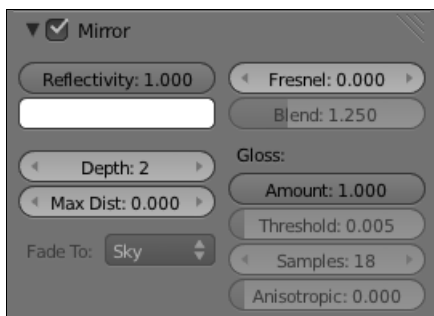


Рис. 4.35. Настройки Mirror

- ◆ **Depth** (Глубина) — количество возможных перекрестных отражений между зеркальными объектами;
- ◆ **Max Dist** (Максимальная дистанция) — установка максимального расстояния для отраженных объектов, после чего они перестают отображаться. По умолчанию эта опция отключена и содержит 0. Если изменить ее значение, то станет доступно дополнительное меню, которое позволяет выбрать метод растворения: **Sky** (фон сцены) или **Material** (цвет материала);
- ◆ **Fresnel** — наподобие такой же опции, как у **Transparency** (Прозрачность). Только в данном случае от нее зависит не прозрачность, а зеркальность;
- ◆ **Blend** (Смешивание) — смешение между отражающими и не отражающими областями.

Интересно, что имеется возможность установки собственного цвета отражающей поверхности. Это выполняется путем изменения образца внизу опции **Reflectivity**.

Настройки **Mirror** также позволяют управлять распылчатостью отражения (рис. 4.36). Данные параметры находятся в группе **Gloss**. Параметры ее ничем не

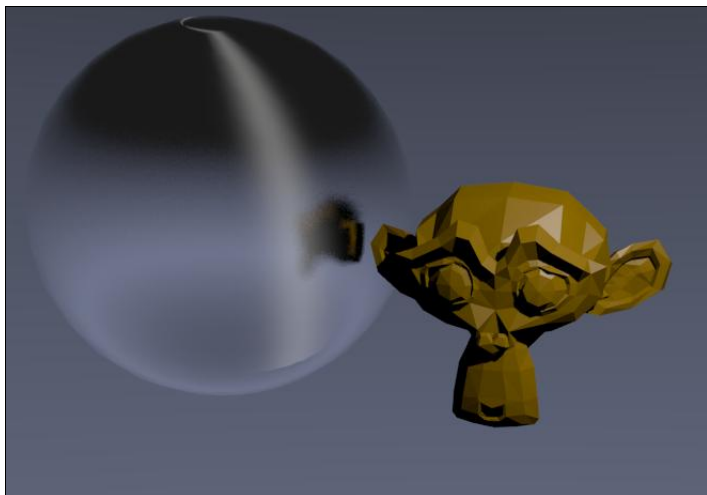


Рис. 4.36. Пример отражения с включенными опциями **Blend**, **Tangent Shading**

отличаются от одноименной для **Transparency**, но имеется одна дополнительная опция — **Anisotropic**. Эта опция позволяет контролировать качество изображения при использовании функции **Tangent Shading** в группе **Shading**.

4.8. Создание и настройка текстур

Под словом "текстура" в Blender понимается нечто большее, нежели простой графический файл. Текстурами могут быть графические и видеофайлы; процедурные функции; карты смещения, рельефа и окружения. Все их можно комбинировать, смешивать с помощью различных алгоритмов и тем самым добиваться поставленной задачи. Создание модели не заканчивается только разработкой каркаса — в нее нужно вдохнуть жизнь, сделать реалистичной и красивой. С помощью материалов, а точнее мультиматериалов, добиться можно немногого. В лучшем случае у вас получатся пластмассовые фигурки. Чтобы довести модель до совершенства, необходимо пользоваться текстурами.

Допустим, нужно сделать дерево для использования в игре. Низкополигональный каркас плюс готовая текстура нарисованных веток и уже получилось нечто смахивающее на дерево. Но современные игры требуют больше детализации предметов. А так как это нельзя делать только наращивая полигоны, то приходится выкручиваться за счет текстур и технологий с ними связанных. В ход уже идут рельефные текстуры, имитирующие кору, заранее просчитанные карты освещения и многое другое. В случае же создания реалистичной модели такие текстуры исчисляются десятками.

Вы уже знаете, что к объекту можно прикрепить несколько материалов. Но и к каждому материалу можно присвоить до 18 текстур! Причем каждая из них будет обладать уникальными параметрами смешивания со всеми остальными. Так в одной стопке могут свободно ужиться обычные текстуры и карты выдавливания, видеофайлы и процедурные текстуры. Если ко всему этому добавить базовый цвет,

рамповые шейдеры, блики, преломление и рефракцию, то приходишь в восторг от той мощи, что дарит Blender художнику.

Все манипуляции с текстурами осуществляются на панели **Textures** окна **Properties** (рис. 4.37).

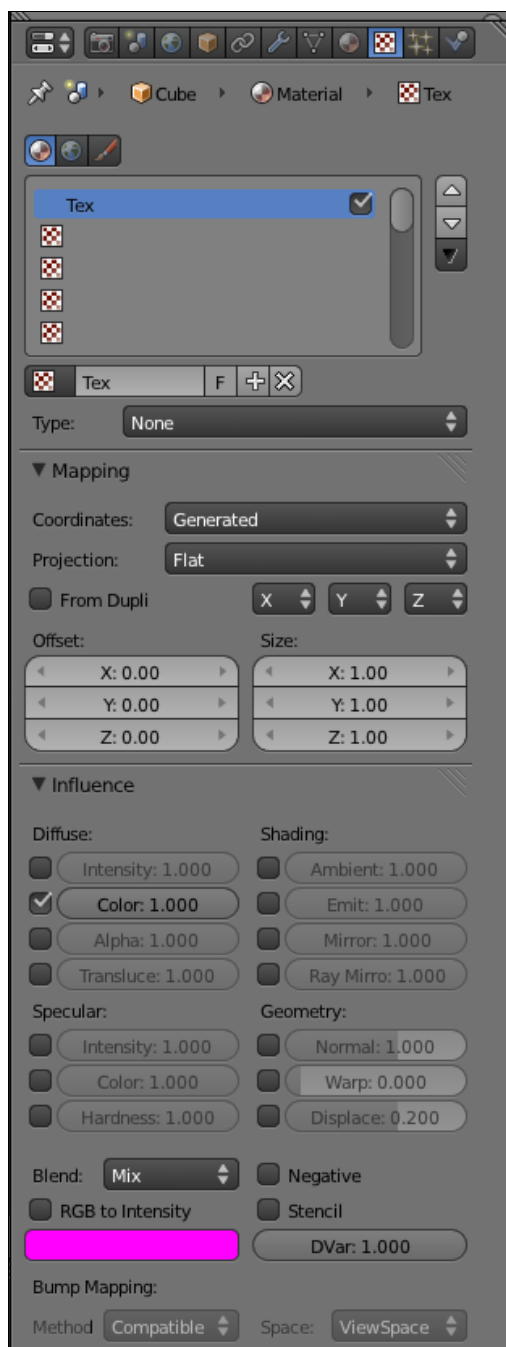


Рис. 4.37. Панель Textures

Текстуры нельзя присоединять к объекту без материала. Сначала нужно добавить материал, а лишь потом создавать сами текстуры. В случае если материал у объекта отсутствует, то Blender предложит создание текстуры только для инструментов кисти (например, скульптурного режима) и окружения. Посмотрите на рис. 4.38, где представлена область управления слотами.

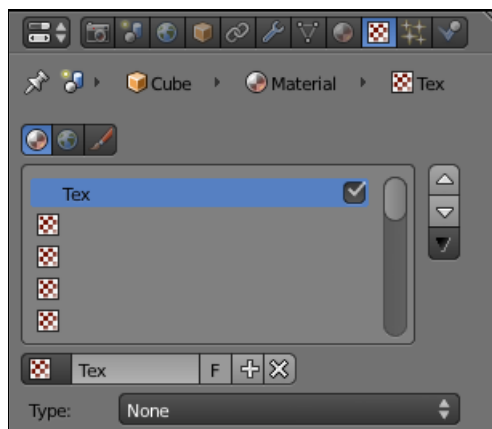


Рис. 4.38. Тектурные слоты

Верхние три кнопки как раз и переключают режим создания текстур: **Material** (Материал), **World** (Окружение) и **Brush** (Кисть). Blender сам определяет режим, и если выделен объект с материалом, то будет выбран **Material**.

Принцип управления слотами или текстурными каналами не особо отличается от работы с мультиматериалами. Единственное исключение — слотов имеется фиксированное количество, а именно 18 штук. Для создания или добавления имеющейся текстуры нужно вначале выбрать слот в окне каналов. Если выделенный слот пустой, то нажмите кнопку **New** для создания текстуры. При наличии уже какой-либо текстуры в слоте можно воспользоваться кнопкой "+" для создания новой или выбрать из списка уже имеющуюся.

Обратите внимание на то, что каналы в списке можно не удалять, а временно отключать, убирая галочку в соответствующем слоте. Кроме того, выделенный слот можно поднимать или опускать в списке соответствующими кнопками (см. рис. 4.38).

ЭТО ВАЖНО!

Не путайте понятия "тектурный слот" и "текстура". Слот — это отведенная ячейка в стопке конкретного материала, а текстура — независимый элемент, который может использоваться в любом слоте и с любым материалом.

Чуть ниже строки с именем текстуры имеется очень важное меню **Type** (Тип). Именно оно позволяет выбрать нужный тип текстуры (рис. 4.39).

Имеется три группы возможных текстур.

- ◆ *Процедурные.* Создание этих текстур осуществляется с помощью специальных математических функций. К ним относятся: **Blend**, **Clouds**, **Distorted Noise**,

Magic, Marble, Musgrave, Noise, Stucci, Voronoi, Wood. В большей степени все они являются результатом работы различных шумовых алгоритмов. Процедурные текстуры будут рассмотрены в соответствующем разделе.

- ♦ **Файловые.** К ним относятся сторонние графические или видеофайлы, загружаемые в программу. Blender воспринимает эти текстуры так, как они есть. Кроме отдельных файлов имеется возможность создания своих собственных текстур непосредственно в Blender. Такой режим называется **Texture Painting** и будет рассмотрен отдельно.
- ♦ **Окружение.** Этот тип представляет всего одна текстура **Environment Map**. Карта окружения дает возможность замены трудоемких по вычислению отражений в сцене готовой текстурой, которая создается особым образом самой программой.

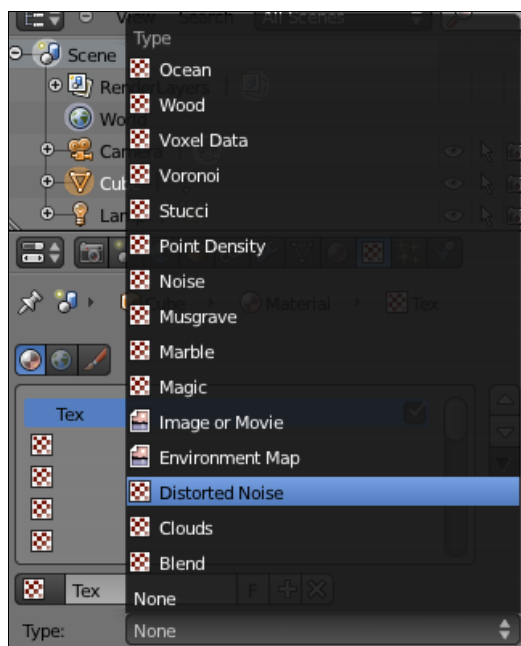


Рис. 4.39. Меню выбора типа текстуры



Рис. 4.40. Группа настроек **Mapping**

Следующая группа опций **Mapping** (Отображение) отвечает за расположение текстуры. Так как объекты в сцене трехмерные, а текстуры двухмерные, то существуют определенные способы совмещения их (рис. 4.40).

Сначала нужно определиться с выбором системы координат для текстуры. Это возможно в меню **Coordinates** (Координаты). По умолчанию там выбран пункт **Generated** (Генерируемый). В этом случае программа использует стандартные координаты без их корректировки. А вот способы наложения текстуры на объект предлагает меню **Projection** (Проекция). Всего там представлено четыре пункта по наиболее популярным формам: **Flat** (Плоский), **Cube** (Куб), **Tube** (Труба), **Sphere** (Сфера). В зависимости от формы модели нужно выбрать наиболее подходящий

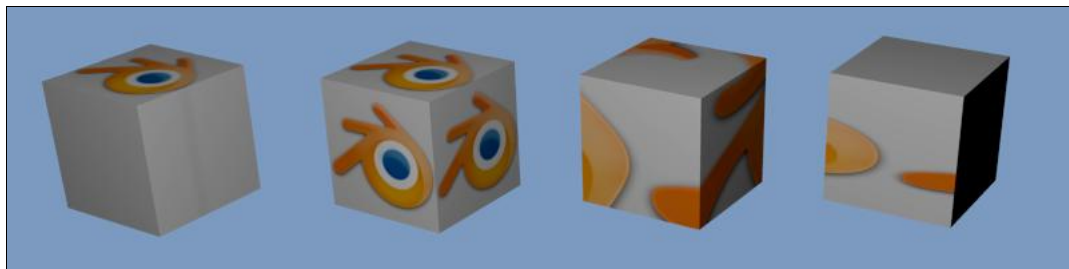


Рис. 4.41. Куб с текстурой в различных проекциях

тип проекции. На рис. 4.41 показан куб с наложенной текстурой по координатам **Generated** в проекциях: **Flat**, **Cube**, **Sphere**, **Tube**.

Помимо **Generated** меню **Coordinates** предлагает еще 10 вариантов:

- ◆ **UV** (Текстурные координаты) — самый оптимальный способ точного наложения текстуры на объект. Каждая вершина имеет двухмерные координаты **UV**, которые впоследствии могут быть перенесены на текстуру. Для этого можно использовать встроенный редактор **UV Editor**;
- ◆ **Sticky** (Липкие) — применяется при использовании так называемых "липких" координат. Нечто наподобие **UV**;
- ◆ **Object** (Объект) — для позиционирования текстуры используется вспомогательный объект, например **Empty**, который указывается в одноименной опции. Перемещение **Empty** вызовет перемещение текстуры;
- ◆ **Global** (Глобальные) — учитываются глобальные 3D-координаты сцены. В этом случае координаты текстуры не привязываются к объекту. При его движении изображение будет как бы плыть по поверхности объекта;
- ◆ **Stress** (Стресс) — если объект деформирован модификаторами, то можно попробовать данный способ;
- ◆ **Reflection** (Отражение) — учитываются координаты отражающего вектора. Этот вариант необходимо использовать для **Environment Map**;
- ◆ **Normal** (Нормаль) — в качестве координат используется направление вектора поверхности. Результат зависит от угла просмотра;
- ◆ **Tangent** (Тангенс) — если у материала включена опция **Tangent Shading** (см. разд. 4.2), то для правильной работы зеркального шейдера используйте наложение **Tangent**, которое также учитывает координаты **UV**;
- ◆ **Window** (Окно) — здесь используются координаты окна рендера;
- ◆ **Strand/Particle** (Частицы) — данный вариант подходит для текстурирования частиц.

Результат проекции можно подкорректировать настройками **Offset** (Смещение) и **Size** (Размер).

Группа опций **Influence** (Влияние) отвечает за взаимодействие конкретной текстуры с другими слотами и самим материалом. Именно здесь можно настроить сме-

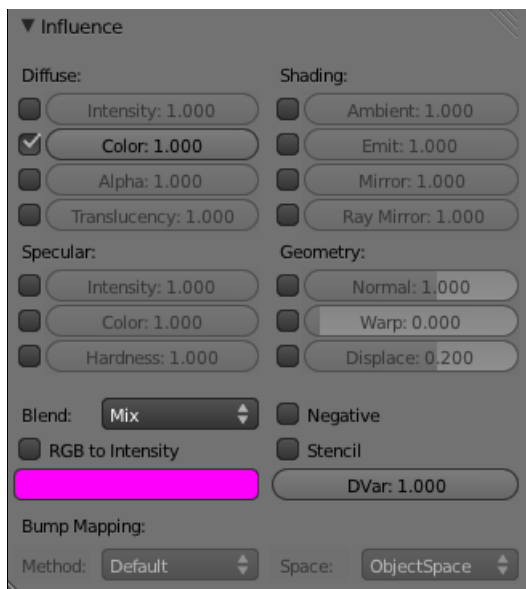


Рис. 4.42. Группа Influence

шивание, влияние шейдеров материала, работу алгоритмов рельефа и карт смещения, а также многое другое (рис. 4.42).

Приведем описание опций **Influence**.

- ◆ Группа **Diffuse** отвечает за взаимодействие текстуры с основным цветом материала:
 - **Intensity** (Интенсивность) — отражающая способность базового цвета;
 - **Color** (Цвет) — степень смешивания текстуры с основным цветом материала. При минимальном значении вывод текстуры отключен;
 - **Alpha** (Прозрачность) — настройка альфа-канала;
 - **Translucency** (Полупрозрачность).
- ◆ Группа **Specular** регулирует работу с бликами:
 - **Intensity** — влияние текстуры на коэффициент отражения зеркального шейдера;
 - **Color** — взаимодействие с цветом отражения;
 - **Hardness** (Жесткость) — жесткость отражения.
- ◆ Группа **Geometry** (Геометрия). Здесь настройки алгоритмов карт рельефа и смещения:
 - **Normal** — настройка рельефной карты;
 - **Warp** — деформация по текстурным координатам. Причем это действие распространяется на все каналы до тех пор, пока не встретится другое значение **Warp**;
 - **Displace** — настройка карты смещения.

- ◆ Меню **Blend** позволяет выбрать способ смешивания. По умолчанию установлен вариант **Mix**. Взаимодействие всегда происходит со слотами, расположенными ниже текущего.
- ◆ **RGB to Intensity**. Замещение выбранным цветом текстуры. Взаимодействует с опцией **Intensity** группы **Diffuse**.
- ◆ **Negative**. Создается негативное изображение текстуры.
- ◆ **Stencil**. При включении этой опции текущая текстура используется как маска для всех последующих.
- ◆ Группа **Bump Mapping** отвечает за более точную настройку функции рельефных карт.

И это еще не все параметры, которые возможны для настройки текстуры. Каждый тип текстуры добавляет на панель свои собственные опции. Рассмотрим работу с наиболее популярным типом **Image/Movie**.

Нетрудно догадаться, что данный тип позволяет использовать сторонние графические или видеофайлы, а также текстуры, созданные в Blender.

Откройте новый проект и увеличьте масштаб куба для удобства работы. По умолчанию этот примитив уже имеет материал и созданную нулевую текстуру (т. е. такую, в которой не выбран тип). Поменяйте базовый серый цвет материала на что-нибудь иное и откройте панель **Textures**.

Так как первый слот уже занят нулевой текстурой, то выберите из меню **Type** пункт **Image or Movie** (Картинка или видео). На панель добавятся целых пять новых групп: **Preview** (Предпросмотр), **Colors** (Цвета), **Image** (Картинка), **Image Sampling** (Выборка), **Image Mapping** (Отображение изображения).

Откройте группу **Image**. Здесь вы можете создать или выбрать текстурный файл соответствующими кнопками **New** и **Open**. Оставим пока в покое создание файла. Нажмите кнопку **Open** и в окне файлового браузера выберите текстуру `blender.png` (`Scenes\glava4` из архива примеров к этой книге, см. приложение 2). По желанию вы можете загрузить любую иную (рис. 4.43).

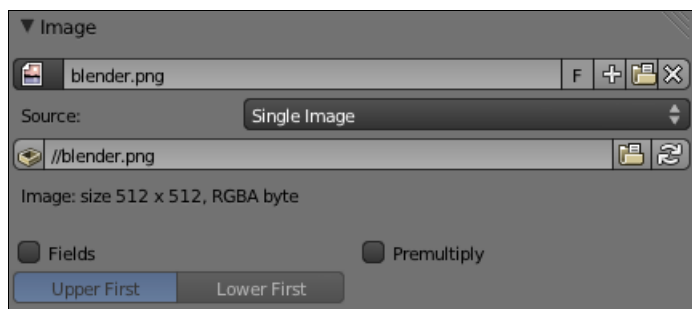


Рис. 4.43. Группа Image

Обратите внимание на меню **Source**. В нем можно выбрать, какой файл будет загружен в данный слот, причем у каждого типа имеются свои настройки.

- ◆ **Single Image** (Графический файл). Содержит две опции:
 - **Fields** (Поля) — поля имеются в видеофайлах, при этом различают нижнее поле (**Lower**) и верхнее (**Upper**). Использование **Fields** с обычными графическими файлами явно излишне;
 - **Premultiply** (Усиление альфы) — включите эту опцию, если вокруг изображения с альфа-каналом наблюдается "мусор".
- ◆ **Image Sequence** (Последовательность картинок). Blender умеет проигрывать анимацию из графических файлов с порядковыми числами (например: file1.jpg, file2.jpg,..., fileN.jpg). Загрузите картинку с наименьшим числом, а программа найдет все остальные. Файлы должны находиться в одной и той же папке. **Image Sequence** добавляет свои собственные опции к уже рассмотренным ранее:
 - **Frames** (Кадры) — данное поле указывает количество возможных кадров в анимации (в скобках), а также позволяет выбрать активный кадр;
 - **Start** (Старт) — номер кадра, с которого начнет проигрываться анимация;
 - **Offset** (Смещение) — сколько пропускать кадров при каждом шаге анимации. Можно использовать совместно с полями;
 - **Auto Refresh** (Автообновление) — если включить, то будет проигрываться анимационная последовательность, иначе обрабатывается только активный кадр, выбранный в поле **Frames**;
 - **Cyclic** (Циклично) — включите опцию, если хотите, чтобы анимация проигрывалась бесконечно.
- ◆ **Movie File** (Файл видео). Содержит те же настройки, что и **Image Sequence**.
- ◆ **Generated** (Генерируемые). Это всего лишь изображения, генерируемые программой, которые можно использовать для тестирования. Опции **X** и **Y** позволяют установить разрешение картинки. Blender предлагает три варианта текстуры:
 - **Blank** — чистая, черная текстура;
 - **UV Grid** — черно-белая решетка с поддержкой координат UV;
 - **Color Grid** — цветная решетка.

СОВЕТ

В России и странах СНГ используется телевизионный формат PAL/SECAM, который обеспечивает передачу видео со скоростью 25 кадров в секунду. При этом для создания картинки используются полуполя, что в совокупности составляет целый кадр. Различают нижнее (**Lower**) и верхнее (**Upper**) поля, одно из которых является первым. Если в настройках Blender перепутать очередность полей, то возможно появление эффекта дрожания. Поэтому при загрузке видео в слот текстуры необходимо установить именно то поле, которое характерно для текущего файла. Например, популярный формат PAL DV по стандарту первым показывает нижнее поле. Во избежание проблем желательно такой же порядок полей использовать в параметрах рендера.

Если вы обработаете сцену, то увидите, что картинка занимает только одну сторону куба. Для заполнения всех сторон необходимо поменять текущую проекцию **Flat** на **Cube**. Это можно сделать в группе **Mapping**.

Результат проекции текстуры на объект можно подкорректировать с помощью опций группы **Image Mapping**, например размножить изображение. Откройте указанную группу и найдите в ней меню **Extension** (Расширение). Выберите пункт **Repeat** (Повтор).

Откроются дополнительные параметры. Для полей **Repeat X** и **Y** замените значение 1 на 2 (рис. 4.44).

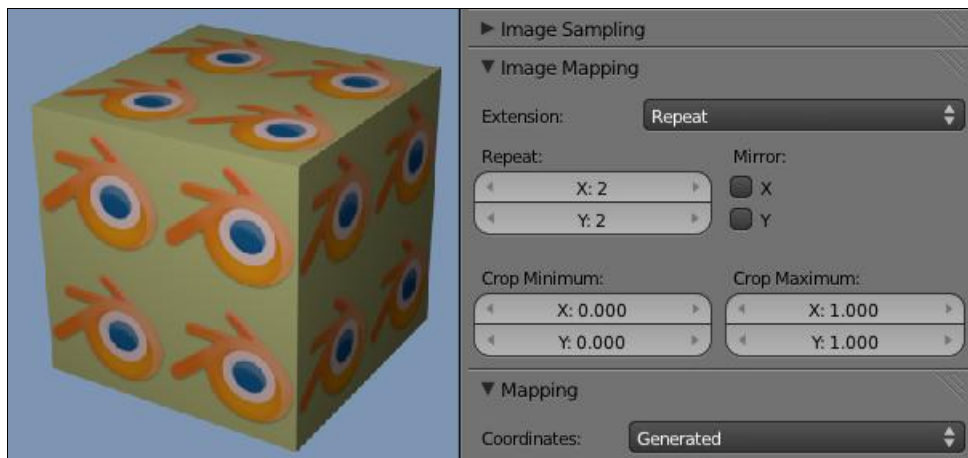


Рис. 4.44. Куб с настройками **Repeat**

Рассмотрим настройки **Image Mapping**. Переключение режимов осуществляется с помощью меню **Extension**.

- ◆ **Extend** (Расширение). За пределами изображения растягивается цвет крайних пикселей.
- ◆ **Clip** (Обрезка). Смысл этой функции в том, что она масштабирует текстуру между координатами **Crop Minimum** и **Crop Maximum**. Оставшаяся область на поверхности объекта становится прозрачной и повторения картинки не происходит.
- ◆ **Clip Cube** (Кубическая обрезка). Функция работает наподобие **Clip**, но только с трехмерными координатами.
- ◆ **Repeat** (Повтор). Обеспечивает размножение картинки по координатам **X** и **Y**. Имеется возможность зеркального отражения текстуры по обоим направлениям (**Mirror**).
- ◆ **Checker** (Шашки). Способ размножения картинки в шахматном порядке. Опции **Even/Odd** позволяют переключаться между четными и нечетными формированиями. Поле **Distance** служит для создания расстояния между "шашками". Для работы функции масштаб текстуры следует уменьшить либо с помощью настроек **Size** группы **Mapping**, либо в этой же группе с помощью **Crop** (рис. 4.45). Например, вы можете быстро создать шахматную доску, если в одном слоте будет находиться белая текстура с включенным **Even**, а в другом черная, но уже с опцией **Odd**.

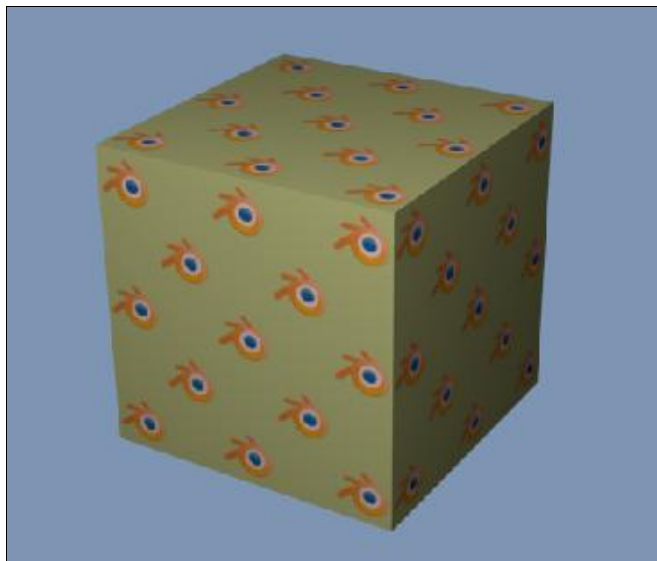


Рис. 4.45. Пример использования Checker

Группа **Preview**, которая появилась после выбора типа текстуры, так же как и в настройках материала, служит для предварительного просмотра текстуры, но имеет несколько дополнительных свойств.

Добавились кнопки для выбора типа вывода в окне **Preview**. Это может быть сама текстура, материал или оба вместе (рис. 4.46). Причем материал демонстрируется со всеми текстурами, включая и настраиваемую. Для удобства просмотра имеется опция **Show Alpha**, которая показывает текстуру с включенной прозрачностью, если она, конечно же, есть.

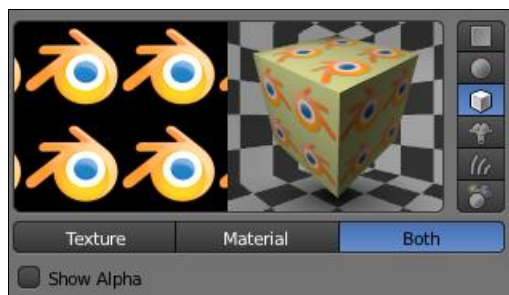


Рис. 4.46. Группа Preview

Опции **Image Sampling** предназначены для базовой настройки загруженной текстуры. Здесь можно выбрать способ определения альфа-канала (**Alpha**):

- ◆ **Use** (Использовать) — некоторые форматы текстурных файлов уже имеют информацию о прозрачных областях, к таким относятся, например, форматы PNG и Targa. Эта опция включает использование альфа-канала, если таковой имеется у картинки;

- ♦ **Calculate** (Высчитать) — второй вариант вычисления прозрачности при отсутствии альфа-канала по принципу: чем темнее пиксел, тем выше прозрачность;
- ♦ **Invert** (Инвертирование) — инвертирование картинки.

По желанию можно перевернуть текстуру, воспользовавшись опцией **Flip X/Y Axis**. Кроме того, в этой группе можно определить текстуру, как **Normal Map** для последующего создания рельефа.

Группа **Colors** позволяет корректировать цвет, яркость, контрастность текстуры (рис. 4.47).

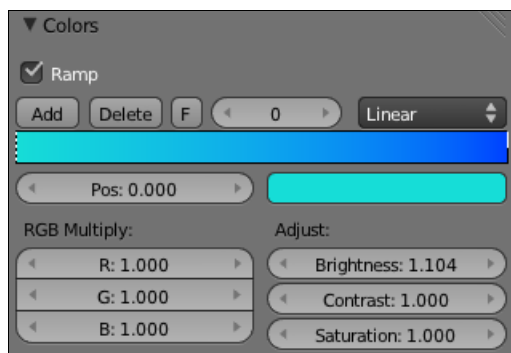


Рис. 4.47. Группа Colors

Здесь присутствует уже знакомый вам рамповый шейдер (**Ramp**). Рассматривать вторично его не будем. Нужно только отметить, что влияние градиента распространяется только на текущую текстуру.

Корректировка яркости, контрастности и насыщенности возможна в области параметров **Adjust**:

- ♦ **Brightness** (Яркость);
- ♦ **Contrast** (Контрастность);
- ♦ **Saturation** (Насыщенность).

Пара движений мышью и нет никакой необходимости открывать сторонний редактор. Кроме этого, Blender предлагает возможность "подкрутить" цветовой баланс изображения с помощью стандартной регулировки RGB.

4.9. Процедурные текстуры

Как ранее было сказано, процедурные текстуры создаются с помощью математических функций. В связи с этим они имеют некоторые особенности в сравнении с файловыми текстурами.

Процедурные текстуры являются бесшовными. Это означает, что в местах стыков не будет видно резкого скачка. По сути дела они не имеют ограничений по размеру, ведь при необходимости Blender продолжит генерировать изображение в соответствии с настройками текстуры.

Вторая особенность заключается в возможности широкого регулирования вида рисунка в соответствии с используемой функцией. Чаще всего в своей основе они имеют шумовые алгоритмы, которые позволяют настраивать шаг и генерацию шума. Причем заданные параметры всегда генерируют одинаковый рисунок.

В наличии имеется 10 процедурных текстур.

- ♦ **Wood** (Дерево). Уже по названию понятно, что данная функция подходит для создания текстуры дерева. С ее помощью можно с легкостью сделать как кольцевые разрезы, так и продольные завихрения. Текстура имеет большое количество настроек (рис. 4.48):
 - кнопки **Sine** (Синус), **Saw** (Пила), **Tri** (Треугольник) позволяют выбрать начальную форму волны для генерации полосок;

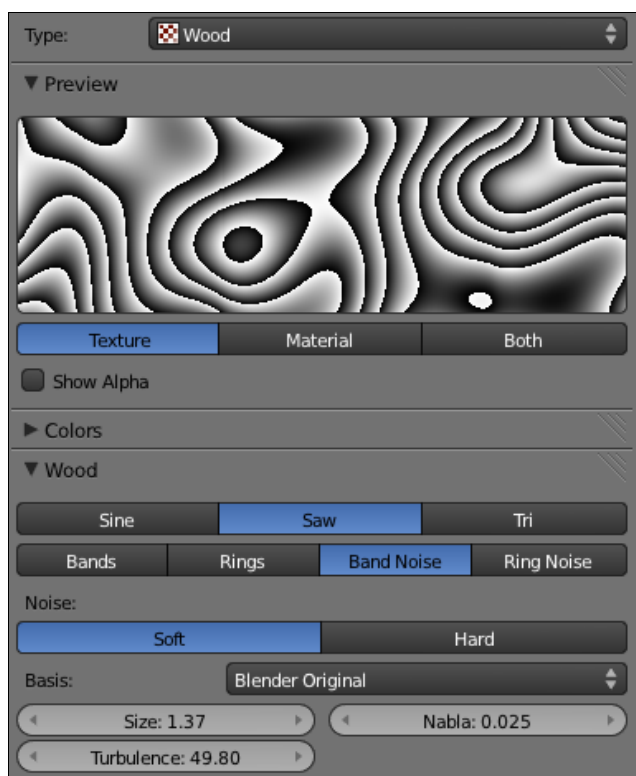


Рис. 4.48. Текстура **Wood** с настройками

- собственно рисунок текстуры выбирается с помощью кнопок **Bands** (Волокна), **Rings** (Кольца), **Band Noise** (Шум волокон), **Ring Noise** (Шум колец). **Bands** и **Band Noise** предназначены для создания продольного рисунка, а вот две оставшиеся кнопки генерируют характерные кольца. Особо обратите внимание на опции **Band Noise** и **Ring Noise**. Эти режимы имеют параметры дополнительной шумовой функции, позволяющие исказить начальный узор. При их включении становятся доступными кнопки **Soft** и **Hard**, которые по-

зволяют генерировать шумы с мягкими переходами и, соответственно, с резкими очертаниями. Важно знать, что данные опции работают только тогда, когда изменены параметры шумовой функции. К ним относятся: **Size** (Размер искажения) и **Turbulence** (Завихрение узора). Можно заменить тип дополнительной шумовой функции в меню **Basis**.

- ◆ **Blend** (Смешивание). Эта функция позволяет создавать различные типы градиентов. Ее можно использовать, например, для создания маскирующей текстуры (рис. 4.49). В ней немного параметров:
 - меню **Progression** (Прогрессия) позволяет выбрать способ заливки градиента от линейного до сферичного;
 - для некоторых пунктов **Progression** становятся доступными кнопки **Horizontal** и **Vertical**, которые переключают направление заливки.

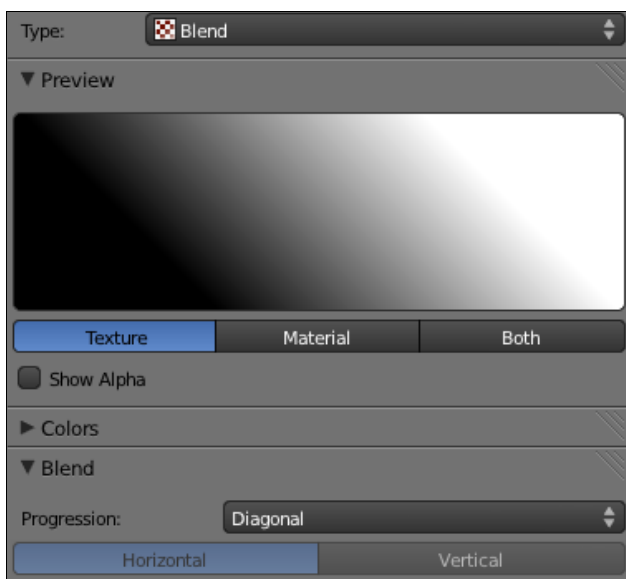
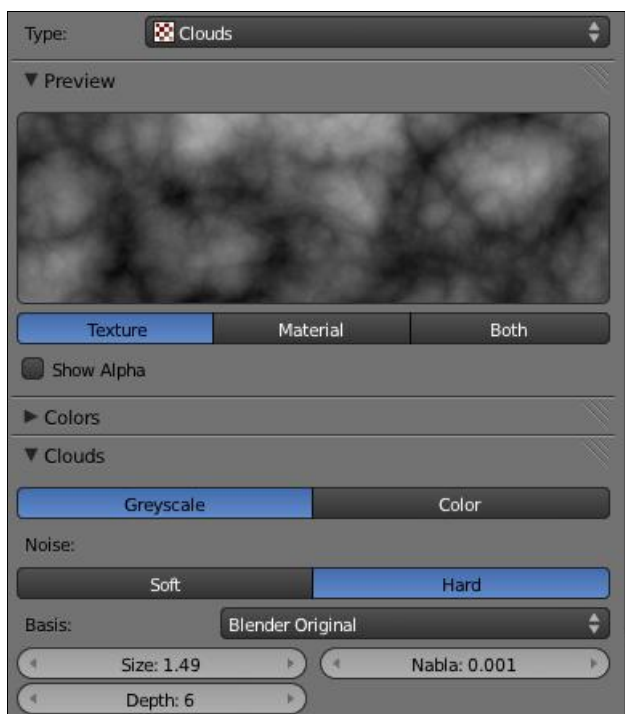
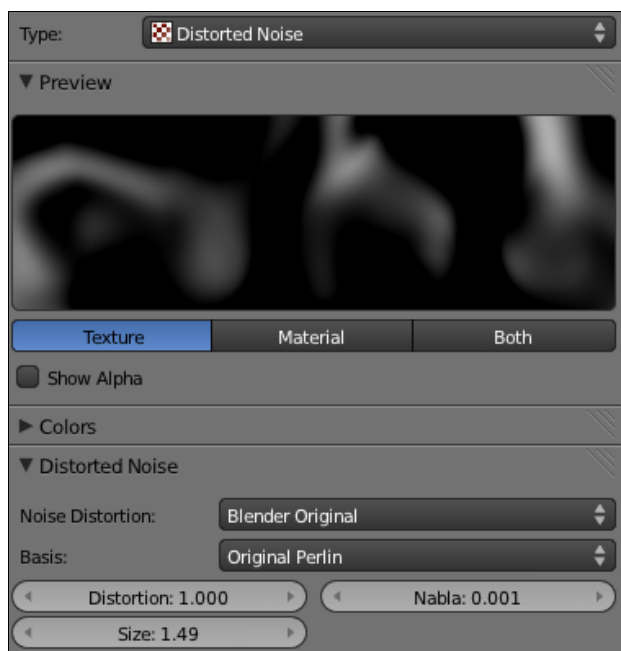


Рис. 4.49. Текстура **Blend** позволяет создавать заготовки для переходов

- ◆ **Clouds** (Облака). Очень интересная функция, которую можно использовать для генерации дыма или тех же облаков (рис. 4.50). Группа настройки шумов ничем не отличается от такой же, как в **Wood**. Зато **Clouds** позволяет выбирать черно-белый или цветной узор кнопками **Greyscale** и **Color**. Кроме того, эту функцию используют для создания мелкозернистого шума, который в дальнейшем может применяться как рельефная карта. Подобное годится для имитации мелких неровностей на объекте, например камня.
- ◆ **Distorted Noise** (Искаженный шум). Узоры с помощью этой функции могут получаться очень разнообразными (рис. 4.51). Особенность ее в том, что для создания картинки используется сразу два шумовых генератора. Первый, основной, выбирается из меню **Basis**, а второй из **Noise Distortion**.

Рис. 4.50. Процедурная текстура **Clouds**Рис. 4.51. Процедурная текстура **Distorted Noise**

Для последнего генератора имеются дополнительные опции:

- **Distortion** (Искажение);
 - **Size** (Размер).
- ♦ **Magic** (Волшебная). Эта функция умеет генерировать очень красивые узоры (рис. 4.52), но практическая ценность ее сомнительная. Только в очень специфических случаях удастся ее использовать, например, при создании разноцветной гирлянды. Имеется всего два параметра:
- **Depth** (Глубина просчета) — изменяя параметр, вы получаете новую палитру цветов;
 - **Turbulence** (Завихрение) — чем выше значение, тем большее искажение и меньше масштаб.

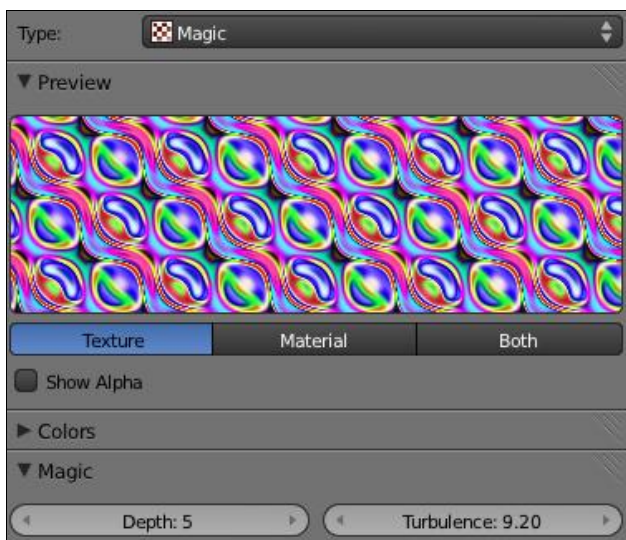
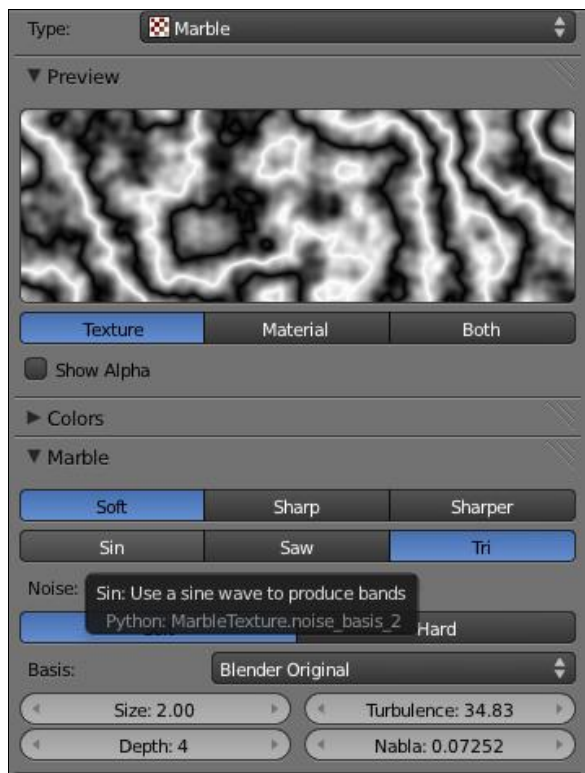


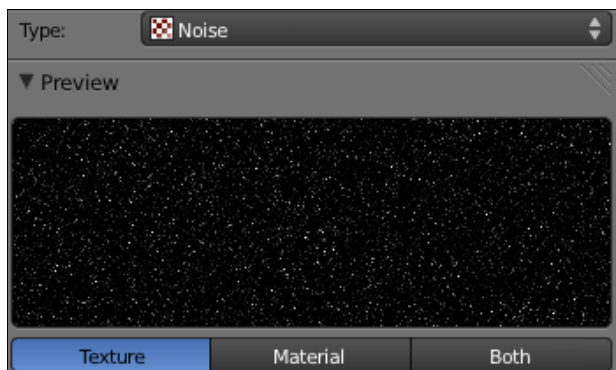
Рис. 4.52. Узоры "волшебной" текстуры

- ♦ **Marble** (Мрамор). Функция с большим количеством параметров и разнообразной генерацией узоров (рис. 4.53). Имеет широкое применение, начиная от текстурирования камней и заканчивая огнем. В этой текстуре, как и в **Wood**, также имеются кнопки для выбора формы волны: **Sin**, **Saw**, **Tri**. Тип генерации доступен в меню **Basis**, которое содержит уже привычный набор функций шумов. **Marble** предлагает возможность установки уровня мягкости переходов между темными и светлыми областями узоров. Для этого на панели имеются три кнопки:

- **Soft** (Мягкий);
- **Sharp** (Резкий);
- **Sharper** (Очень резкий).

Рис. 4.53. Процедурная текстура **Marble**

- ◆ **Noise** (Шум). Очень простая функция без каких-либо параметров. Производит генерацию "белого шума" (рис. 4.54). Подобный эффект можно создать с помощью **Clouds** и с ее обширными параметрами.

Рис. 4.54. "Белый шум" от **Noise**

- ◆ **Stucci** (Точечная). В основном эта функция используется для текстур, имитирующих шершавость разного размера. С ее помощью можно сделать кожуру апельсина, каменные стены и т. д. Чаще всего она применяется как рельефная

карта (рис. 4.55). Интересно, что имеются специальные опции, позволяющие приспособить **Stucci** к текстурованию внутренних и внешних частей модели:

- **Plastic** (Пластмасса) — генерируется узор с очень мягкими переходами между областями;
- **Wall in** (Внутри стены) — оптимально подходит для заливки отверстий и впадин;
- **Wall out** (Снаружи) — этот тип рекомендуется применять для наружных частей в режиме рельефной карты.

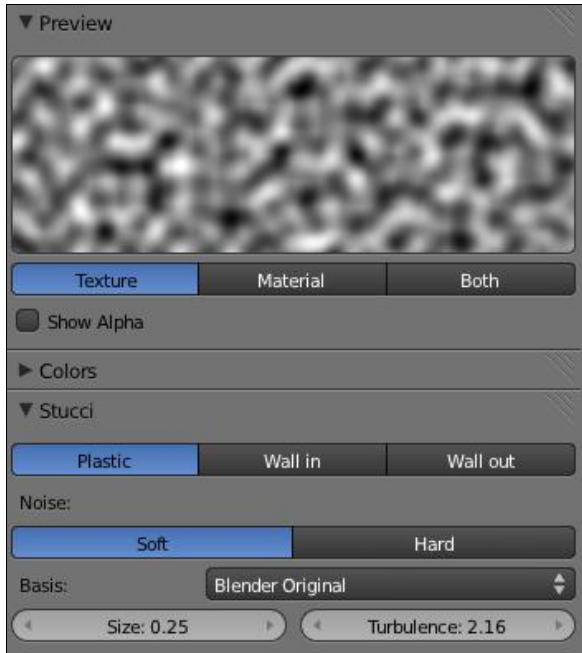


Рис. 4.55. Текстура **Stucci**

♦ **Voronoi**. Очень мощная функция для генерирования разнообразных узоров (рис. 4.56). С ее помощью можно создавать вполне законченные текстуры. Размах применения широк, от создания поверхности огненной лавы до многоцветной витражной мозаики. Узоры, генерируемые ею, строго структурированные, напоминающие детский пазл. Имеется возможность регулирования формы "пазла", расстояния между ними, настройка цветной заливки. Форма элементов выбирается в меню **Distance Metric** (Метрика дистанции): от строгих квадратов до причудливо разрезанных кусочков. Группа опций **Feature Weights** (Характеристика веса) отвечает за расстояние между клетками. С их помощью можно сделать так, что элементы начнут наползать друг на друга и это генерирует необычные визуальные эффекты. Опции меню **Coloring** позволяют выбрать тип заливки:

- **Position** (Позиция) — генерируется цветная мозаика с четкими переходами между элементами;

- **Intensity** (Интенсивность) — черно-белая текстура с мягкими переходами между областями. Можно использовать для создания лавы;
- **Position and Outline** (Позиция и окантовка) — цветная мозаика с черной окантовкой элементов. Напоминает стиль мультияшности;
- **Position, Outline and Intensity** (Позиция, окантовка и интенсивность) — смешивание всех трех вариантов.

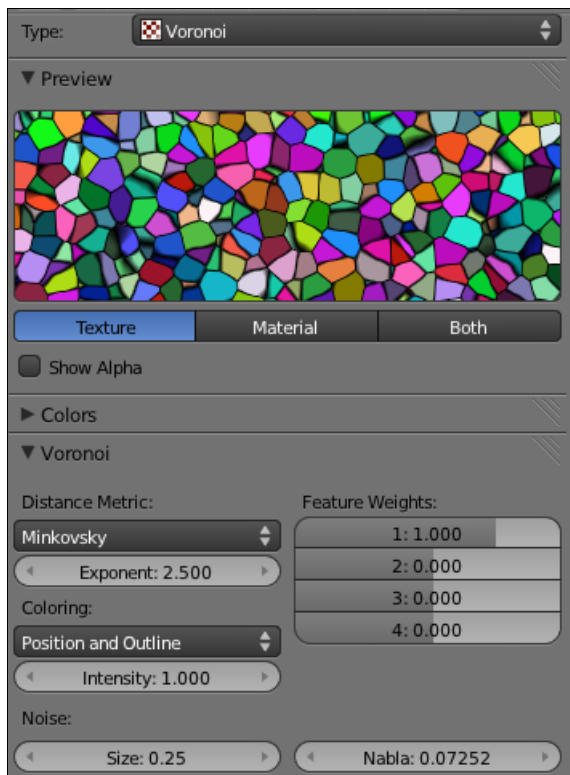


Рис. 4.56. Цветной витраж при помощи текстуры **Voronoi**

- ◆ **Musgrave**. Разработчики Blender советуют использовать эту функцию для создания органических текстур (рис. 4.57). Функция позволяет изменять: тип фрактала (меню **Type**), алгоритм шума (**Basis**), настройки фракталов. Результат работы ее чрезвычайно обширен: от переплетений четких линий до расплывчатых пятен.

Итак, вы познакомились с кратким описанием процедурных текстур Blender, а теперь поработаем с ними на практике.

Займемся созданием модели простого деревянного бруска, сделанного из цельного куска дерева. Соответственно, на нем должны быть характерные древесные разводы.

Создайте новый проект и отмасштабируйте имеющийся куб так, чтобы он напоминал толстую доску (рис. 4.58).

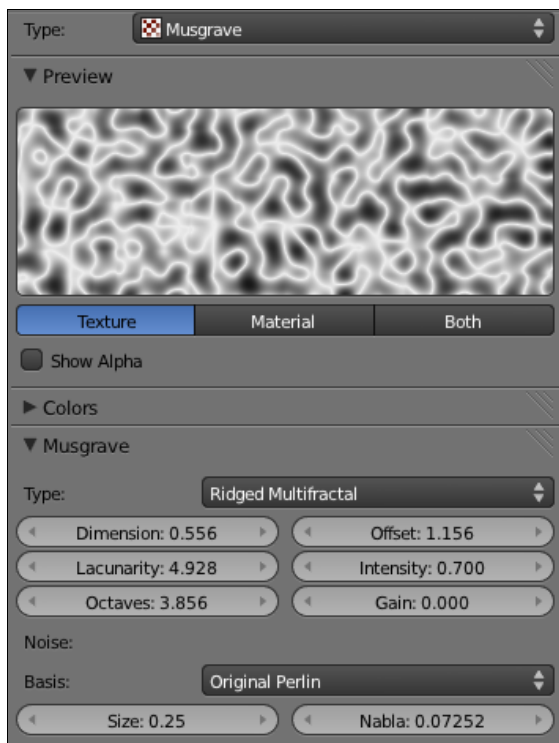


Рис. 4.57. Процедурная текстура **Musgrave**

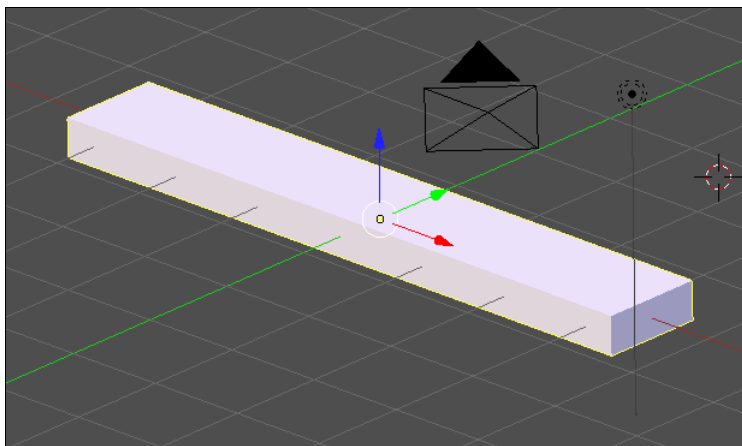


Рис. 4.58. Заготовка для бруска

Откройте настройки материала примитива. В качестве основного цвета возьмем светло-коричневый со следующими параметрами RGB:

♦ **R** = 0.800

♦ **G** = 0.370

♦ **B** = 0.117

Теперь займемся текстурой для создания на материале характерных древесных узоров. Откройте панель **Textures**. Так как текстура была создана по умолчанию вместе с материалом куба, то остается только выбрать ее тип. В этом случае оптимальным будет использование функции **Wood**. Выберите из меню **Type** нужную процедурную текстуру (рис. 4.59).

Для удобства просмотра включите кнопку **Both** в группе **Preview**. Теперь программа будет отображать саму текстуру и собственно материал. Прimitives для вывода в **Preview** лучше выбрать кубической формы. Нажмите кнопку с соответствующей фигуркой.

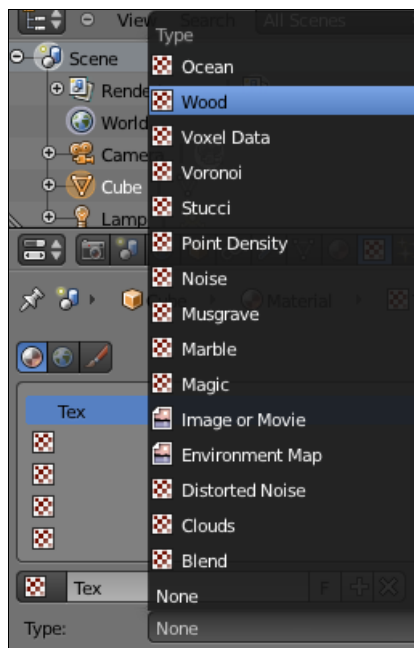


Рис. 4.59. Выбор текстуры **Wood**

Выполните настройку текстуры:

1. Нажмите кнопку **Brand Noise**.
2. Нажмите кнопку **Saw**. Изображение примет более четкие очертания.
3. В поле **Size** введите значение 2.0. Произошло увеличение узора.
4. **Turbulence** = 41. Линии узора закрутились.
5. Откройте группу **Mapping** и измените проекцию **Flat** на **Cube** (меню **Projection**) для более правильного отображения текстуры на модели.
6. Нажмите клавишу <F12> и сравните свой результат с рис. 4.60.

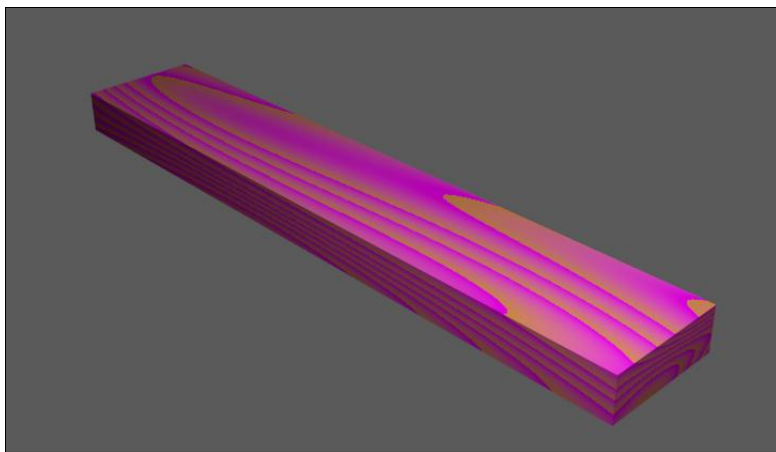


Рис. 4.60. У этой текстуры неправильный цвет

У вас должна получиться текстура с вполне симпатичным узором, но почему-то фиолетового цвета. Все очень просто. По умолчанию текстуры создаются действительно фиолетового цвета. Заменить его можно в группе **Influence**, щелкнув по образцу цвета. Установите для текстуры следующие значения RGB:

♦ **R** = 1.000

♦ **G** = 0.600

♦ **B** = 0.218

Вот теперь получилась доска с нормальными цветами (рис. 4.61).

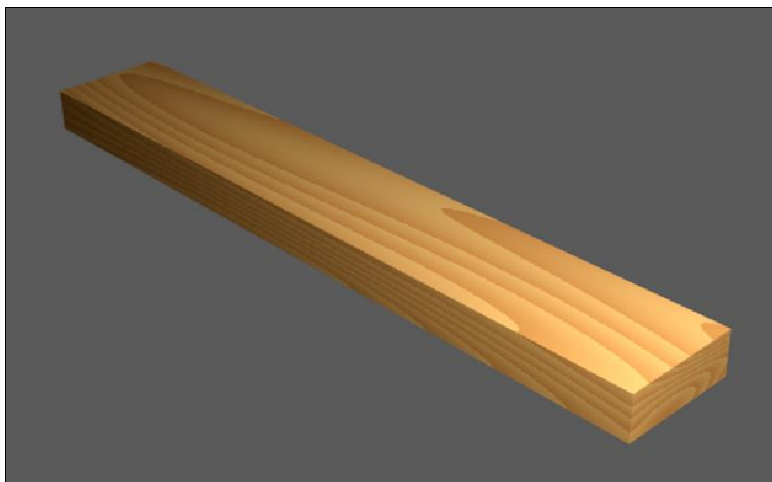


Рис. 4.61. Готовая деревянная доска

4.10. Карты *Normal* и *Displacement*

Существует золотое правило: если желаете, чтобы модель была максимально реалистичной, то будьте готовы уделить внимание самым мелким деталям. Но высокодетализированную модель Blender может обрабатывать многие часы, а если ее нужно использовать в программах реального времени?

Современные компьютерные игры радуют своих почитателей очень красочной и насыщенной картинкой. Трехмерные герои и окружение выглядят реалистично. Невольно задаешься мыслью, как игровой движок умудряется работать с такой сложной графикой да еще в реальном времени, если тот же Blender тратит часы на обработку похожей картинki.

В действительности в ракурсе игровой камеры одновременно находятся несколько десятков тысяч полигонов сцены. А такой объем современные графические платы с легкостью "прокрутят".

Допустим, в игровой сцене должна быть дорога, мощенная булыжником. Если все изгибы, впадины между камнями, да и сами булыжники прорисовывать с помощью

полигонов, то визуализация одной лишь дороги может занять все ресурсы графического процессора. Проще сделать низкополигональную модель грубой формы с текстурой булыжника. Вот только такая дорога будет выглядеть плоско и невзрачно. Но стоит к ней добавить особую текстуру, обрабатываемую программой с помощью специального алгоритма, и получится качественная, детальная картинка.

В настоящее время существует большое количество алгоритмов, способных на основе плоской текстуры демонстрировать объемные детали. В действительности, при этом структура модели не подвергается какому-либо изменению, а рельеф создается за счет игры светотеней.

Возьмем для примера модель деревянной доски, сделанной ранее. Она получилась гладенькой, как будто ее отшлифовали (см. рис. 4.61). Предположим, что поставлена задача сделать характерные неровности на торцах, которые остаются после распила пилой. Мелкие впадины и выпуклости создавать на этапе моделирования просто нерационально. Поэтому воспользуемся одной из процедурных текстур и технологией бампинга.

Bump Mapping — алгоритм, реализующий рельеф на объекте с помощью черно-белой текстуры. При этом учитывается интенсивность цвета пиксела. Чем темнее пиксел, тем ниже он опускается, и наоборот.

Откройте ваш проект с моделью доски или загрузите готовый из папки Scenes\glava4\glava4_wood.blend из архива примеров к этой книге (см. приложение 2).

Сейчас в нем имеется всего один материал и текстура. Для текстурирования торцов необходим будет новый материал, ведь редактирование имеющегося приведет к изменению внешнего вида модели целиком.

Откройте настройки **Material** и добавьте еще один слот в окне мультиматериалов (нажмите на плюсик, расположенный справа). Не спешите создавать новый материал, ведь при этом придется его заново настраивать в соответствии с общим материалом. Лучше выбрать последний из списка с помощью кнопки меню, расположенной слева от кнопки **New** (рис. 4.62).

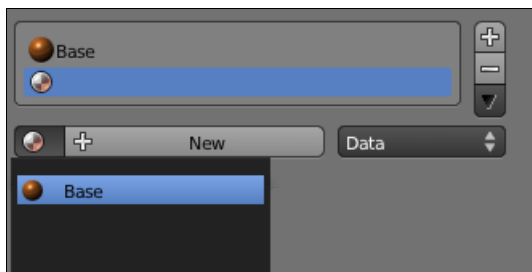


Рис. 4.62. Присвоение старого материала новому слоту

После этого действия основной материал будет присвоен обоим слотам модели (рис. 4.63).

Выделите нижний слот и нажмите кнопку с цифрой 2 рядом с названием материала. Теперь он стал уникальным, но с теми же настройками, как и основной. Желательно его переименовать во избежание путаницы, например как "Noise".

Второй материал должен быть присвоен не ко всей модели целиком, а только к полигонам торцов. Нажмите клавишу <Tab> для режима редактирования и выделите с помощью <Shift> грани с обоих концов модели. Для присвоения выделения к материалу щелкните по кнопке **Assign** на панели **Material**.

Теперь нужно добавить новую текстуру для рельефа. Откройте панель **Textures** и выделите свободный слот в списке. Нажмите кнопку **New** для создания текстуры. Последнюю переименуйте также в "Noise" и включите ее (рис. 4.64).

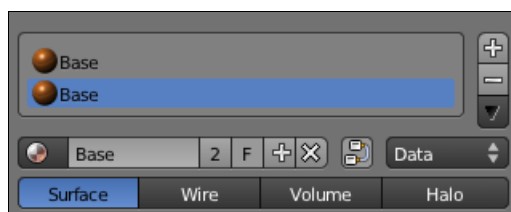


Рис. 4.63. Материалы объекта

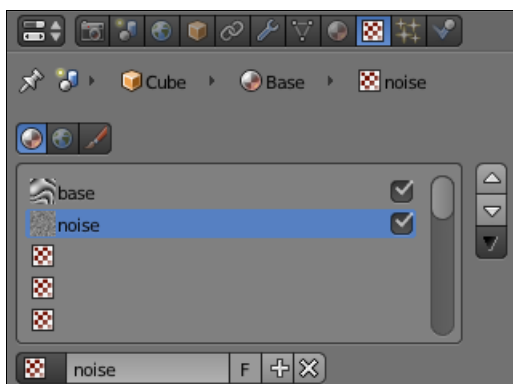


Рис. 4.64. Новая текстура Noise

В качестве типа текстуры выберите из меню **Type** пункт **Stucci**. Измените ее параметры в соответствии со следующими:

1. В группе **Noise** нажмите кнопку **Hard**.
2. Значение **Size** установите в 0.0001.
3. Параметр **Turbulence** сделайте равным 5.

Сейчас текстура отображается как обычный плоский рисунок. Для использования ее в качестве рельефной нужно настроить некоторые опции на закладке **Influence**:

1. Снимите галочку с опции **Color** в группе **Diffuse**. Это выключит отображение текстуры.
2. Включите опцию **Normal** в группе **Geometry** и установите в ней значение, равное 0.600. Таким образом был активирован режим **Normal Map**, т. е. алгоритм обработки рельефной карты. Значение этого параметра влияет на глубину рельефа.
3. В группе **Bump Mapping** имеется меню **Method** (Метод). Выберите в нем пункт **Best Quality** (Наилучшее качество).

После этих манипуляций наша доска выглядит более естественно (рис. 4.65).

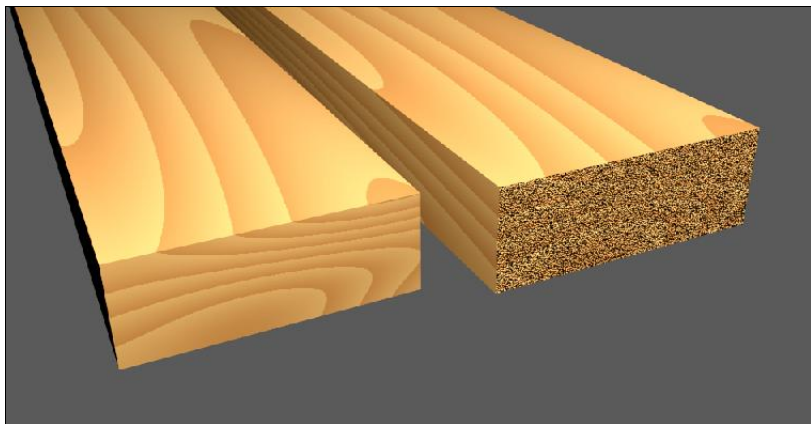


Рис. 4.65. Слева гладкая доска, а справа обрезная

В качестве **Bump Map** (Карта выдавливания) может быть не только процедурная текстура, но и обычная файловая. Такую текстуру можно создать в любом двухмерном редакторе, наподобие Gimp или Photoshop. Важно учитывать, что она должна быть черно-белой в большом разрешении.

Карты выдавливания легко сделать и настроить, но у них есть один существенный недостаток — они малоэффективны в создании высокоточного, сложного рельефа. Их лучше использовать для генерации небольших неровностей, таких как: шероховатость доски, мелкие выбоины на камнях. Для более сложных и выпуклых рельефов **Bump Mapping** не годится — их просто не будет видно.

Blender предлагает еще одну технологию создания рельефа — *Normal Mapping*, которая позволяет более точно моделировать неровности. Как и в бампинге, здесь используется интенсивность цвета пикселей текстуры, но уже не черно-белой, а полноцветной. Обе технологии работают с нормальными векторами объекта.

Нормаль — это вектор, располагающийся перпендикулярно поверхности и смотрящий всегда в одном направлении. Нормали играют важную роль в различных сферах работы с программой. Например, при моделировании стока воды в сосуд с помощью физики Blender важно правильно установить нормали объекта (в сторону жидкости). В противном случае, вода будет спокойно проходить сквозь стенки модели.

Blender позволяет увидеть нормали объекта и изменить их расположение. На панели свойств примитива (<N>) в режиме редактирования появляется новая закладка **Mesh Display** (Экран) (рис. 4.66). В ней имеется группа **Normals** (Нормали) с несколькими опциями:

- ◆ **Face** (Грань) — вывод нормали граней;
- ◆ **Vertex** (Вершина) — вывод нормали вершин;
- ◆ **Normal Size** (Размер нормали) — размер нормалей для визуализации в окне.

Для изменения направления вектора можно воспользоваться кнопками на панели **Tool Shelf** группы **Normals**:

- ◆ **Recalculate** (Пересчет) — изменение направления всех нормалей объекта внутрь (**Inside**) или наружу (**Outside**) вне зависимости от начального состояния векторов;
- ◆ **Flip Direction** (Изменить направление) — простое инвертирование направления.

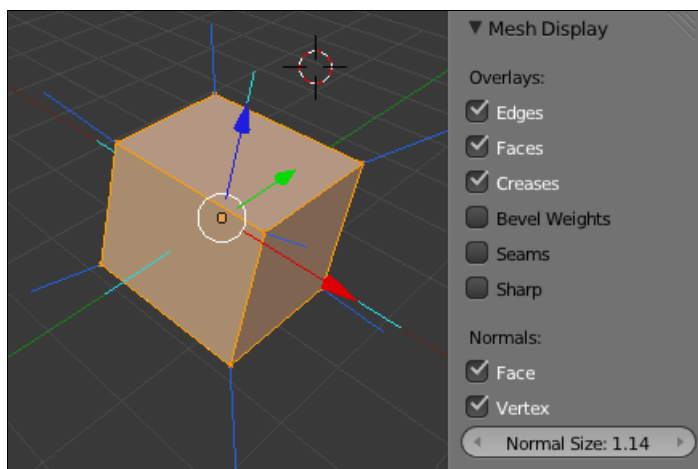


Рис. 4.66. Нормали куба

Карты нормалей хранят информацию о направлении векторов в каналах RGB. Это дает возможность учитывать расположение пиксела в трехмерном пространстве, что обеспечивает большую точность визуализации рельефа. Однако **Normal Map** не выгодно отличается от своего собрата тем, что вручную такую текстуру не нарисуешь.

Существует много сторонних программ, позволяющих генерировать **Normal Map** на основе имеющейся готовой текстуры, например редактор GIMP с плагином Normal. Но такие автоматически созданные карты не всегда могут передать нужный рельеф. Наиболее точно получить **Normal Map** можно из реальной трехмерной модели. Blender имеет возможность генерации **Normal Map**.

Создание рельефной карты можно разбить на несколько этапов:

1. Разработка модели с минимально необходимой детализацией.
2. Доработка деталей с помощью инструментов высокополигонального моделирования, таких как модификатор **Multires**, скульптурная лепка. В принципе можно создать копию основной модели и уже ее изменять.
3. Генерация **Normal Map** для высокополигонального объекта.
4. Наложение готовой карты на базовую модель.

Рассмотрим все эти этапы на примере создания модели кусочка дороги, вымощенной камнями.

В качестве основного примитива используем плоскость (**Plane**). Добавьте ее в сцену и немного вытяните по координате X (рис. 4.67). Сразу установите сглаживание для нее, нажав кнопку **Smooth** на панели **Tool Shelf**.

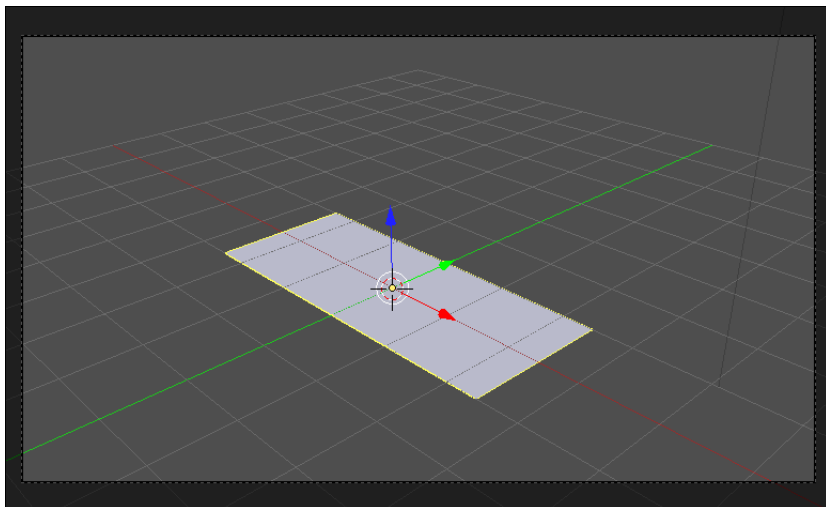


Рис. 4.67. Плоскость для дороги

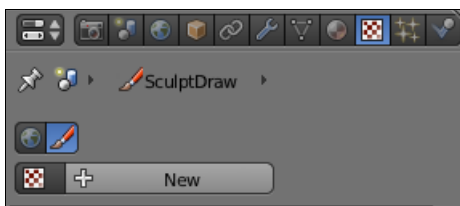
По сути, плоскость и будет являться той низкополигональной моделью. Сделайте ее копию с помощью **<Shift>+<D>** и нажмите клавишу **<M>** для переноса дубликата в другой слой. На последнем этапе именно эта копия будет играть роль основной модели.

Выделите оставшуюся плоскость, нажмите клавишу **<Tab>** для перехода в режим редактирования. Увеличьте количество элементов структуры в 4 раза (функция **Subdivide**) и перейдите в режим **Sculpt Mode**.

Хотя примитив уже имеет некоторую детализацию, но ее недостаточно для работы в режиме скульптуры. Откройте панель модификаторов в окне **Properties** и добавьте **Multires**. Установите режим **Simple**. Для той задачи, что нужно будет выполнить, минимально комфортным разрешением **Multires** является уровень 6. Нажмите кнопку **Subdivide** в настройках модификатора шесть раз (поля **Sculpt** и **Render** должны показывать цифру 6). В идеале нужно еще больше детализации, но это ляжет тяжким бременем на систему при вычислении карты.

Займемся созданием булыжников на мостовой. Разумеется, это будет делаться не вручную, а с помощью процедурной текстуры.

Откройте панель **Textures** и нажмите кнопку **New**. Так как материал к объекту еще не прикреплен, то текстура будет создана для использования в **World** (Окружение) и для **Brush** (Кисть) (рис. 4.68).

Рис. 4.68. Нажмите кнопку с изображением карандаша для текстуры **Brush**

В качестве типа текстуры подойдет **Voronoi**. Измените его настройки в соответствии с указанными:

1. В меню **Distance Metric** выберите **Actual Distance**.
2. В меню **Coloring** пункт **Position and Outline**.
3. Интенсивность (**Intensity**) установите в 1.
4. Параметр **Size** сделайте равным 0.10.
5. Установите значение **Nabula** равным 0.025.

Текстура сделана. В окне **3D View** включите режим просмотра **Top View** (<NumPad 7>) для удобства рисования.

Давайте убедимся, что текстура создана для использования в качестве кисти и доступна на панели инструментов **Brush** (панели **Tool Shelf**). Найдите на ней закладку **Texture** и откройте ее. Если все сделано правильно, то в окне **Preview** будет показываться используемая текстура (рис. 4.69).

Установите для кисти максимальные значения для параметров силы нажатия и радиуса:

◆ **Strength** = 1

◆ **Radius** = 200

Поместите кисть над примитивом так, чтобы она находилась над центром и полностью его охватывала (рис. 4.70). Вы можете колесиком мыши подгонять проекцию окна под размер кисти.

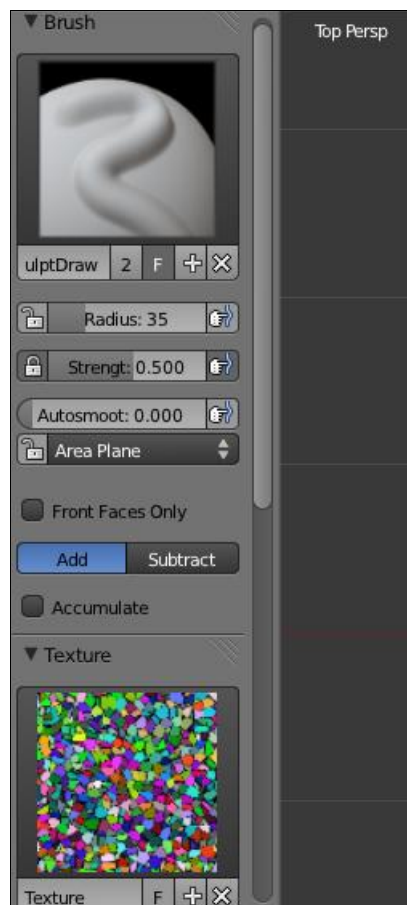


Рис. 4.69. Панель **Brush** и активная текстура

Однократно нажмите левую кнопку мыши, чтобы выдавить рисунок (рис. 4.71).

Присмотритесь к результату. На рисунке отчетливо видно, что сила нажатия кисти явно не одинаковая, ближе к центру сильнее, а к краям ослабевает. Нужно сделать так, чтобы кисть выдавливала равномерно по всей области. Нажмите клавиши <Ctrl>+<Z> для отката изменений.

Поведение кисти регулируется на панели **Brush** в группе **Curve**. По умолчанию там установлена плавная кривая, которая как раз обеспечивает неравномерное выдав-

ливание. Внизу окна с кривой есть ряд кнопок, предлагающих заготовки формы сплайна. Нажмите самую последнюю кнопку (рис. 4.72).

Выполните еще раз выдавливание рисунка (рис. 4.73).

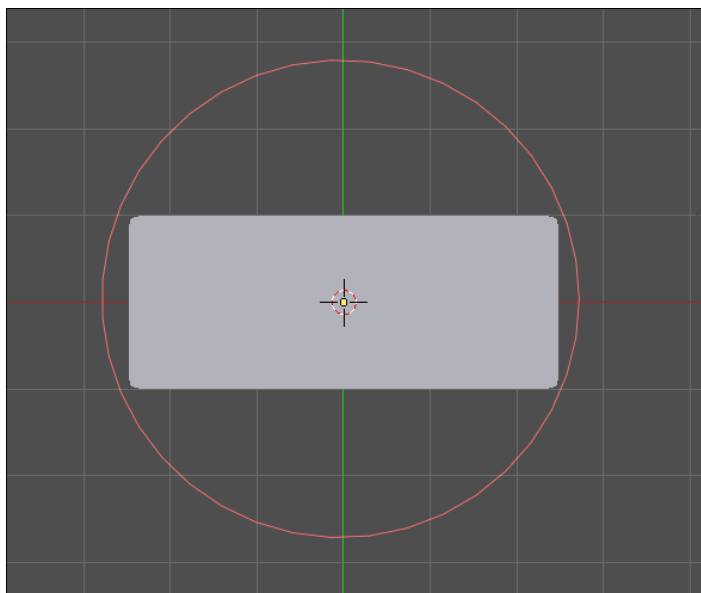


Рис. 4.70. Размещение кисти над плоскостью

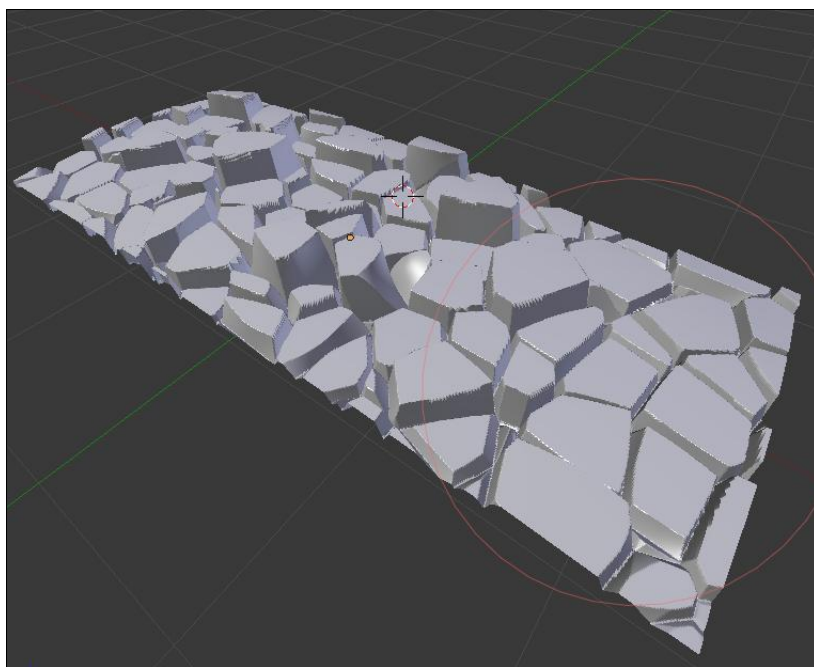


Рис. 4.71. Высокополигональная модель с нарушениями

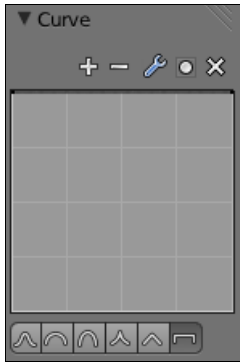


Рис. 4.72. Настройка поведения кисти

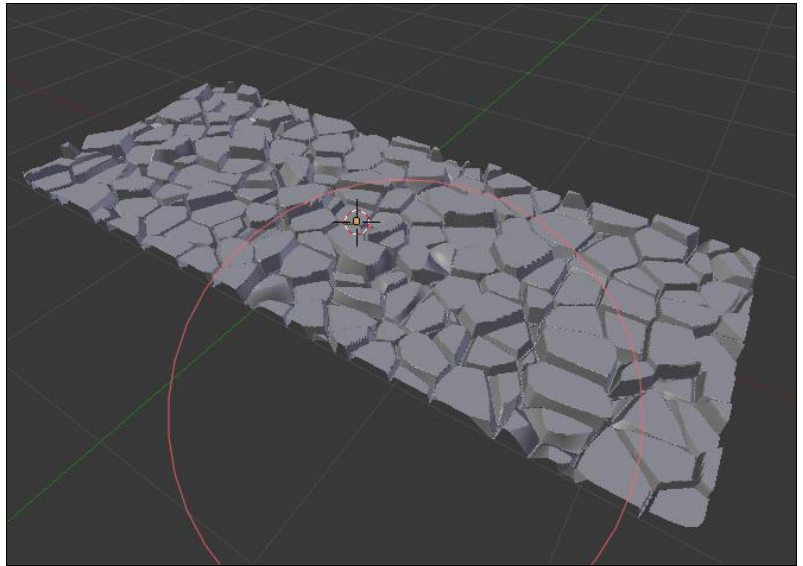


Рис. 4.73. Видно, что высота булыжников относительно равная

Теперь нужно подготовить модель для создания текстуры **Normal Map**. Для этого будет выполнена развертка UV-координат и сопоставление их с текстурой в редакторе **UV Editor**. Работа с ним подробно будет описана в следующем разделе, а пока просто выполните указанные далее предписания.

В верхней части программы выберите из меню **Screen Layout** пункт **UV Editing** (рис. 4.74).

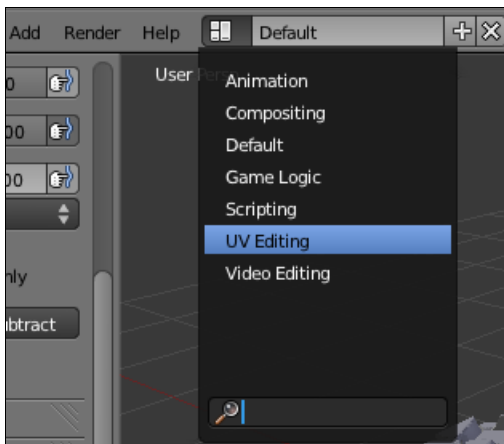


Рис. 4.74. Переключение вида окна программы

Экран программы разделится на две половинки. Слева появится окно **UV/Image Editor**, а справа стандартное **3D View**.

В правом окне перейдите из режима **Sculpt Mode** в **Edit mode**. Нажмите клавишу <A>, чтобы выделить всю структуру. Клавиша <U> вызовет меню **UV Mapping** для

настройки UV-координат объекта. Выберите там первый пункт **Unwrap**. Таким образом была выполнена развертка модели.

ВНИМАНИЕ!

В режиме редактирования модель выглядит как обычная плоскость безо всяких деформаций. Это нормально, ведь модификатор **Multires**, с помощью которого выполнялось скульптурное моделирование, не закреплен.

Результат создания карты нормалей должен куда-то сохраняться. Поэтому создадим для нее новую текстуру в окне **UV Editor**.

Найдите и нажмите кнопку **New** на заголовке окна редактора. Появится всплывающее окно **New Image**. Ничего не меняйте, просто нажмите в нем кнопку **OK**. По умолчанию Blender создаст черную текстуру, а так как до этого была выполнена развертка UV-координат, то **UV Editor** наложит эту сетку на новую текстуру. Выберите в **Screen Layout** пункт **Default** для восстановления прежнего вида программы.

Сейчас настало время заняться генерацией **Normal Map**. Этот процесс в Blender называется "выпечка" и выполняется он рендером программы. Для этого нужно открыть панель настроек **Render** в окне **Properties** (рис. 4.75).

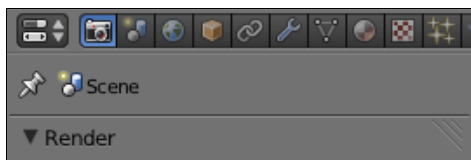


Рис. 4.75. Кнопка выбора панели **Render**

Внизу панели найдите закладку **Bake** (Выпечка). Работа с рендером программы будет рассмотрена в соответствующем разделе, а пока выберите из меню **Bake Mode** пункт **Normals**.

Чуть ниже находится еще одно меню **Normal Space**, которое позволяет выбрать способ создания карты:

- ◆ **Tangent** (Тангенс) — наиболее совершенный способ обработки нормалей объекта вне зависимости от деформации и изменения объекта с помощью манипуляторов;
- ◆ **Object** (Объект) — учитываются координаты объекта. Результат зависит от его деформации, но остаются в стороне манипуляции;
- ◆ **World** (Окружение) — расчет нормалей в соответствии с глобальными координатами. Зависимость и от деформации, и от манипуляций;
- ◆ **Camera** (Камера) — способ проецирования карты с точки зрения камеры. Установите камеру так, как хотели бы видеть рельеф.

Выберите в этом меню пункт **Object** (рис. 4.76).

Теперь выделите плоскость в сцене и нажмите кнопку **Bake**. Такая высокополигональная модель может обрабатываться некоторое время. Процесс работы можно наблюдать вверху программы в виде процентной полоски (рис. 4.77).

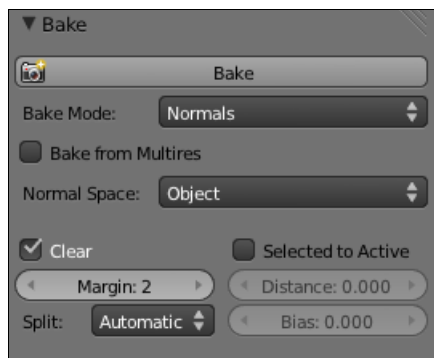


Рис. 4.76. Настройки рендера **Normal Map**



Рис. 4.77. Здесь Blender выводит информацию о текущей работе



Рис. 4.78. Временное отключение модификатора

Давайте проверим, как программа выполнила задачу. Во-первых, отключите работу модификатора **Multires** для окна **3D View** и рендера (рис. 4.78). Кнопка с изображением фотоаппарата управляет выводом рендера, а "глаз" — для окна **3D View**.

Выберите в заголовке окна **3D View** в меню **Viewport Shading** пункт **Texture** для показа текстуры в окне программы (рис. 4.79).

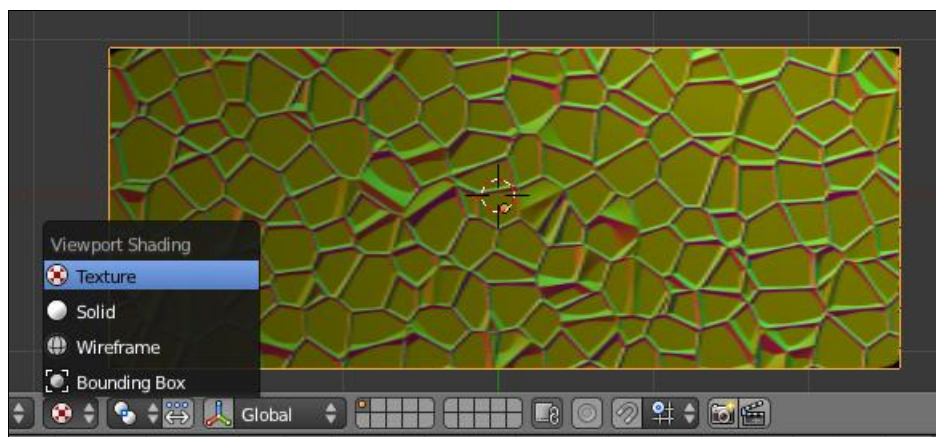


Рис. 4.79. Так выглядит текстура **Normal Map**

Сейчас свежесделанная карта используется как обычная текстура, в качестве использования ее как **Normal Map** нужно выполнить несколько настроек.

Создайте новый материал и добавьте к нему текстуру. Для последней установите тип **Image and Movie**. В группе **Image** нажмите на значок с картинкой и выберите из меню свою текстуру.

Откройте закладку **Image Sampling** и включите опцию **Normal Map**. Внизу под опцией имеется меню, где нужно выбрать ранее использованное пространство при выпечке карты. Установите там значение **Object**.

Так как **Normal Map** была рассчитана с помощью UV-координат, то выберите соответствующий пункт в меню **Coordinates** закладки **Mapping**.

Теперь включим ее как **Normal Map**. Откройте закладку **Influence**:

1. Отключите опцию **Color** в группе **Diffuse**. В этом случае текстура показываться не будет.
2. Включите опцию **Normal** в группе **Geometry**. Используйте значение для **Normal**, равное 0.5.

СОВЕТ

Скорее всего, вы столкнулись с тем, что, несмотря на изменения настроек текстуры, окно **3D View** продолжает упорно демонстрировать тот вариант, который выбран в **UV Editor**. Вы можете включить режим **GLSL** для окна программы в свойствах **Display** (<N>) или отсоединить уже ненужную текстуру в окне **UV Editor** (нажмите на крестик рядом с именем).

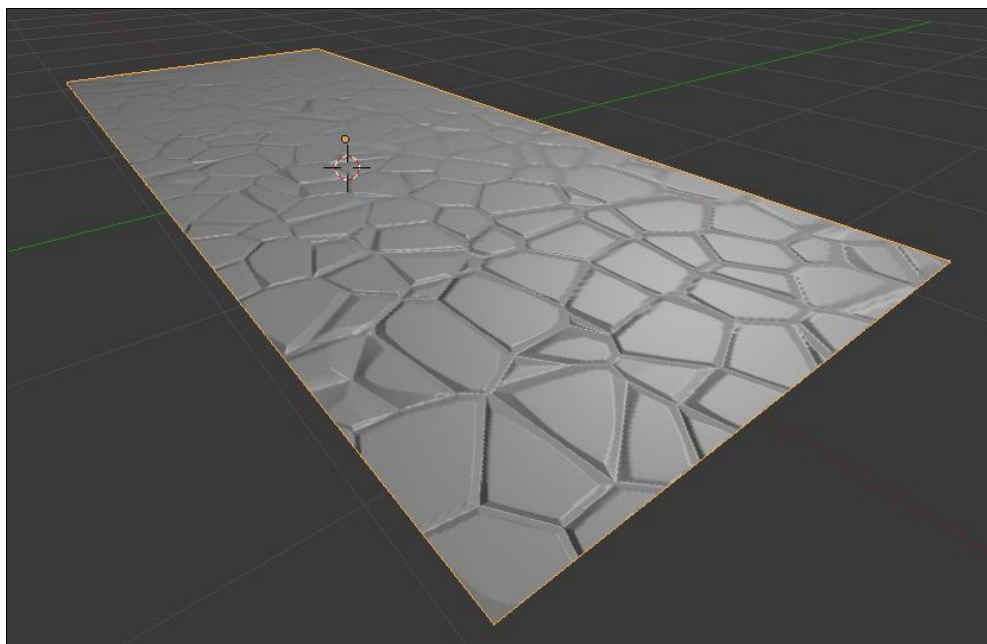


Рис. 4.80. Готовый низкополигональный объект с использованием технологии **Normal Mapping**

В начале работы над моделью вы сохранили копию плоскости без увеличения элементов структуры. Попробуйте к ней присвоить созданный материал и убедитесь, что всего 4 вершины с одной гранью способны выдавать столь детальную картинку.

ВНИМАНИЕ!

После закрытия программы потеряется созданная текстура **Normal Map**. Ее нужно либо сохранить, либо упаковать в файл с проектом. Для сохранения откройте окно **UV Editor** и нажмите клавишу <F3>. Для упаковки выберите пункт меню: **File | External Data | Pack into blend file**.

Вы, наверное, уже заметили, что результат работы технологий **Bump Mapping** и **Normal Mapping** сильно зависит от угла просмотра модели. Так при развороте **Plane** ребром к себе эффект рельефа пропадает, что и неудивительно, ведь это всего лишь игра светотеней.

Blender имеет и еще один способ создания рельефа с помощью текстуры, но в отличие от ранее рассмотренных, он фактически изменяет геометрию модели. Конечно, этот алгоритм уже не из разряда реального времени (Realtime).

Displacement Map (Карта выдавливания) — способ изменения геометрии модели на основе обычной текстуры. С его помощью можно создавать, например, ландшафты. В отличие от карт рельефа, полученная модель является полностью трехмерной и подчиняется всем законам 3D (используются тени и освещение).

В качестве основы для **Displacement Map** могут быть как процедурные текстуры, так и файловые, как цветные, так и черно-белые.

Имеется два способа работы с такими картами:

- ◆ *стандартный* — с помощью настройки опций **Influence**;
- ◆ *расширенный* — с помощью модификатора **Displace**.

Работа с первым вариантом:

1. Добавить текстуру к материалу и выбрать ее тип.
2. На закладке **Influence** выключить опцию **Diffuse Color**.
3. Там же включить опцию **Displace** и отрегулировать степень воздействия текстуры на **Mesh**-объект.

Нужно заметить, что работа по первому способу несколько неудобна из-за невозможности просмотреть результат в окне **3D View**. Дело в том, что карты выдавливания фактически изменяют **Mesh** только при рендере, а в окне по-прежнему будет оставаться стандартный примитив.

Этого неудобства лишена работа с модификатором **Displace** (рис. 4.81).

Рассмотрим его настройки.

- ◆ Группа **Texture** служит для создания или выбора имеющейся текстуры из списка, наподобие такой же, как на панели **Textures**.

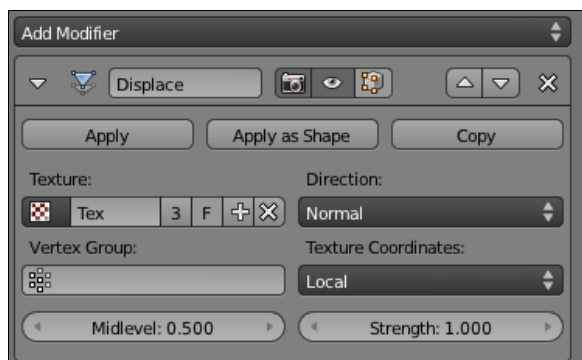


Рис. 4.81. Модификатор **Displace**

- ◆ Меню **Direction** (Направление) позволяет установить способ выдавливания **Mesh**-объекта. Имеются варианты:
 - **Normal** — использование нормалей вершин;
 - **X, Y, Z** — движение вершин по указанным координатам;
 - **RGB to XYZ** — для цветных текстур, где каждый цветовой канал присваивается к конкретной координате.
- ◆ **Vertex Group**. Работа не с объектом целиком, а только с установленной группой вершин.
- ◆ **Texture Coordinates**. Установка текстурных координат: **UV**, **Local**, **Global**, **Object**.
- ◆ **Midlevel** (Средний уровень). Установка высоты для всех вершин выделенной группы.
- ◆ **Strength** (Сила воздействия). Способ регулирования силы выдавливания.

Работать с модификатором очень просто. Создается нужная текстура, можно прямо с материалом самого объекта. Чтобы текстура эта не визуализировалась, нужно отключить опцию **Diffuse Color** в ее настройках. В самом модификаторе выбираете созданную текстуру и настраиваете ее воздействие указанными ранее опциями (рис. 4.82).

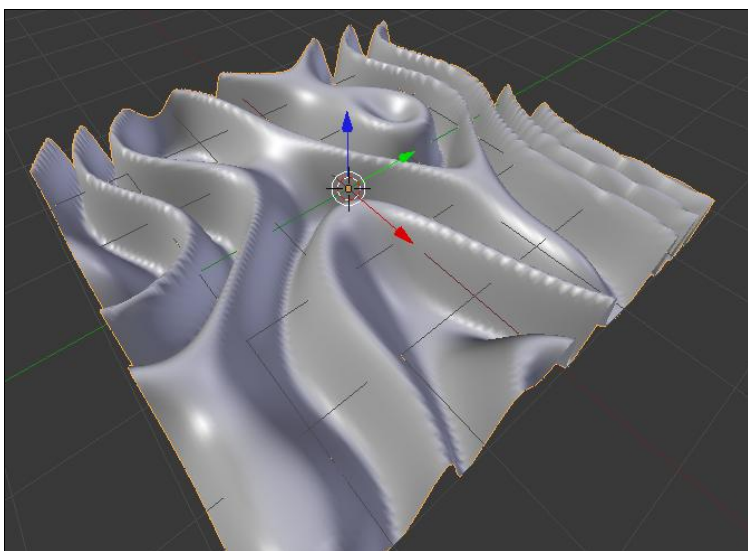


Рис. 4.82. Результат использования **Displacement Map**

Учтите только один важный момент в работе с картами выдавливания — для качественной работы они требуют высокодетализированный объект. Чем больше элементов, тем качественнее и глаже будет картинка. Кроме того, вы можете спокойно редактировать исходный **Mesh**-объект. Алгоритм **Displacement** учтет изменения и перестроит конечную картинку.

4.11. Наложение текстуры по развертке UV

Объекты Blender используют трехмерные координаты XYZ для позиционирования в пределах сцены, но у них еще есть особая сетка координат, которая предназначена для работы с текстурами.

UV — это двумерные координаты, которые используются программой для привязки каждой вершины к определенному месту на текстуре, соответственно, *разверткой UV* называется процесс распределения этих координат.

Blender предлагает два способа создания развертки UV: автоматический и ручной. В первом случае имеются несколько заготовок, которые более или менее могут правильно распределить структуру, зато второй вариант позволяет точно определить место для каждой вершины на текстуре.

Для управления разверткой программа предлагает использовать специальный редактор **UV/Image Editor**. Кроме того, уже имеется заготовка расположения окон, оптимально подходящая для работы с текстурами (рис. 4.83). Выбрать ее можно из меню **Screen Layout** на заголовке программы (пункт **UV Editing**).

Как видите, в этом режиме имеется всего два окна: **UV Editor** и **3D View**. Такая конфигурация окон действительно самая удобная, но вы можете создать личную под свою задачу.

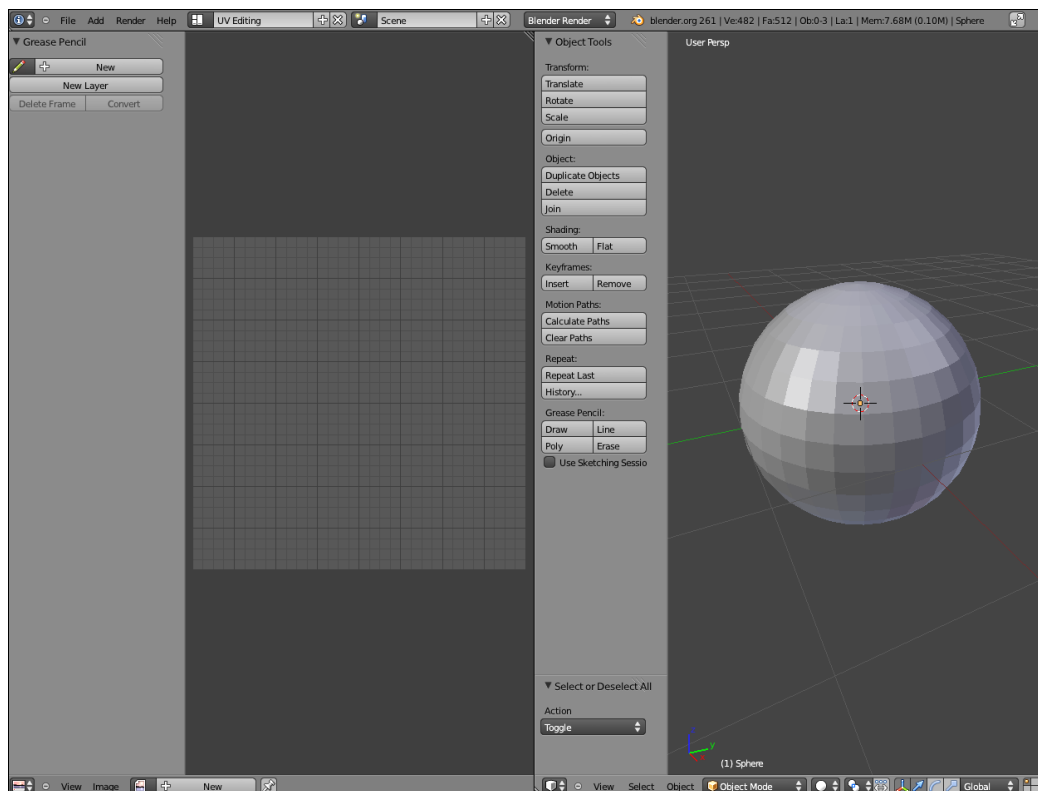


Рис. 4.83. Раскладка окон UV Editing

Работа с UV-координатами возможна только в режиме редактирования объекта. Причем можно изменять развертку как для всего объекта целиком, так и для выделенной части структуры.

Имеется основное меню **UV Mapping** для управления разверткой, вызываемое клавишей <U> в окне **3D View**. Оно содержит 10 пунктов.

- ◆ **Unwrap** (Развертка). Наиболее быстрый способ создания развертки. При этом полученная сетка будет оптимально покрывать всю текстуру.
- ◆ **Cube Projection** (Кубическая проекция). Оптимально подходит для моделей кубической формы. Это могут быть модели: шкафа, телевизора, стола.
- ◆ **Cylinder Projection** (Цилиндровая проекция). Подходит для моделей цилиндрической формы: карандаши, стаканы.
- ◆ **Sphere Projection** (Сферическая проекция). Подходит для моделей сферической формы (рис. 4.84).
- ◆ **Project from View** (Экранная проекция). Полезный режим, когда развертка создается по принципу: "как есть при просмотре". Вы можете развернуть модель или сцену в нужном ракурсе, а программа выполнит своего рода снимок экрана.

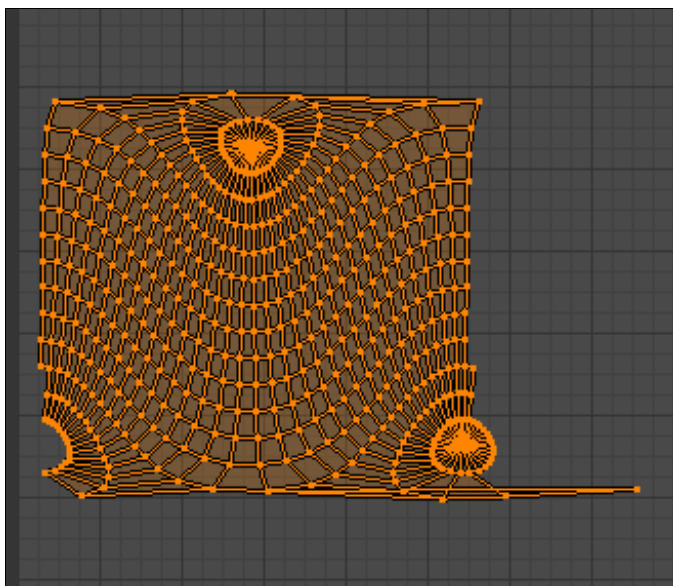


Рис. 4.84. Пример раскладки сферы с помощью **Sphere Projection**

- ◆ **Project form View (Bounds)** (Экранная проекция с заполнением). То же самое, что и выше рассмотренная опция, но в этом случае развертка будет занимать всю текстуру.
- ◆ **Reset** (Сброс). Сброс координат.
- ◆ **Smart UV Project** (Умная развертка). Инструмент для разбиения развертки на несколько независимых зон.

- ◆ **Lightmap Pack** (Развертка для карт освещенности). Создание специальной развертки для использования с картами **Lightmap** (освещение).
- ◆ **Follow Active Quads** (Следовать за активными квадратами). Работает с выделенными гранями и старается выполнить развертку, пока цепочка полигонов не заканчивается. Работает не всегда оптимально.

Результат работы развертки объекта будет отображаться в окне **UV Editor**. Не беда, если она не подходит форме рабочей модели. Всегда можно выбрать в меню **UV Mapping** пункт **Reset** и начать все сначала.

Теперь самое время загрузить текстуру и совместить развертку с картинкой. Для управления текстурами на заголовке окна **UV Editor** имеется стандартная область для выбора или создания новой текстуры (рис. 4.85).



Рис. 4.85. Управление текстурами

Попробуем поиграть с **UV Editor** на практике. Включите раскладку окон **UV Editing**. Выделите имеющийся куб и перейдите в режим редактирования. Нажмите клавишу <U> и выберите пункт меню **Cube Projection**.

Теперь создадим для работы новую текстуру. Нажмите кнопку **New** в заголовке окна **UV Editor**. Появится окно с настройками текстуры (рис. 4.86):

- ◆ **Name** (Имя);
- ◆ **Width** (Ширина) — указывается в пикселах;
- ◆ **Height** (Высота);
- ◆ **Color** (Цвет) — область выбора цвета для заливки;
- ◆ опция **Alpha** (Альфа-канал) — если включено, то текстура будет содержать дополнительный канал прозрачности;
- ◆ **UV Test Grid** (Тестовая решетка) — генерируемая текстура;
- ◆ **32 bit Float** (Глубина 32 бит) — глубина цвета.

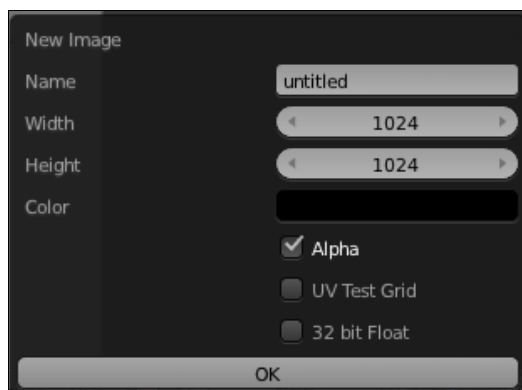


Рис. 4.86. Настройки новой текстуры

Выберите пункт **UV Test Grid** и нажмите кнопку **OK**. Под имеющейся разверткой куба должна появиться текстура. Для удобства просмотра включите в окне **3D View** режим **Texture** в меню **Viewport Shading** (рис. 4.87).

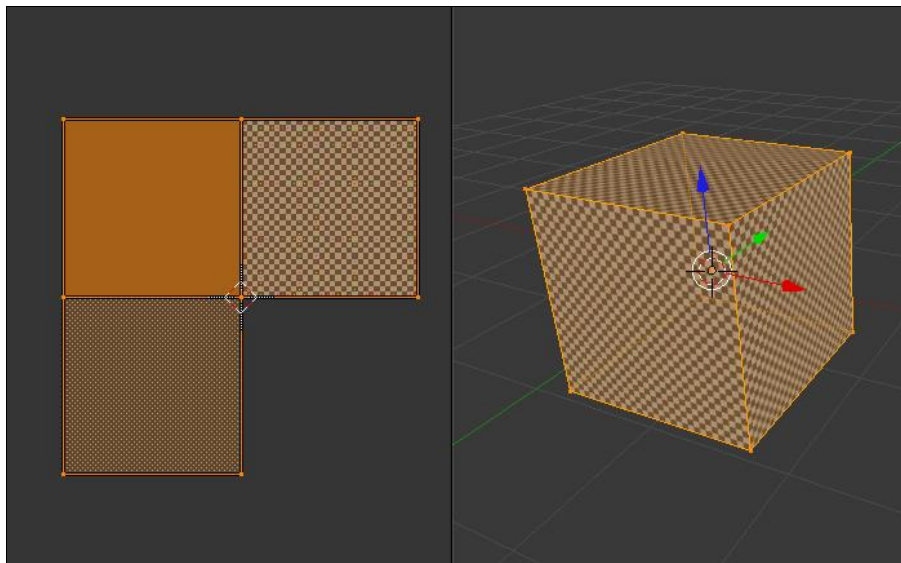


Рис. 4.87. Развертка с текстурой и сам куб

UV Editor позволяет управлять размещением как всей сетки целиком, так и отдельных ее элементов. Здесь действуют стандартные возможности окна **3D View** по выделению, перемещению и масштабированию элементов.

Попробуем увеличить изображение текстуры на кубе. Выделите всю сетку в редакторе клавишей <A> (по умолчанию, после создания развертки она уже выделена) и нажмите <S> для включения масштабирования. Сожмите выделение (рис. 4.88). Как видите, ничего сложного в редактировании сетки нет.

Иногда бывает нужно определить расположение конкретного элемента структуры **Mesh**-объекта на текстуре. В этом случае достаточно выделить нужный элемент в окне **3D View**. **UV Editor** после этого покажет только те вершины сетки, которые относятся к данному элементу.

По такому же принципу происходит создание развертки для выделенных полигонов объекта. Попробуйте выделить две смежные грани и выбрать пункт **Unwrap** в меню **UV Mapping**. С этого мгновения эта сетка становится отдельным элементом развертки куба. Вы можете свободно манипулировать ее, как заблагорассудится. Такой элемент еще называется "островком". Чтобы увидеть полностью всю сетку, достаточно выделить весь объект.

UV Editor имеет свой собственный набор кнопок, переключающий режим выделения, наподобие такого же, как в окне **3D View** (рис. 4.89).

Если рассматривать их слева направо, то первая кнопка отвечает за вершины, вторая за ребра, а третья выделяет грани. Особенно полезна последняя четвертая кноп-

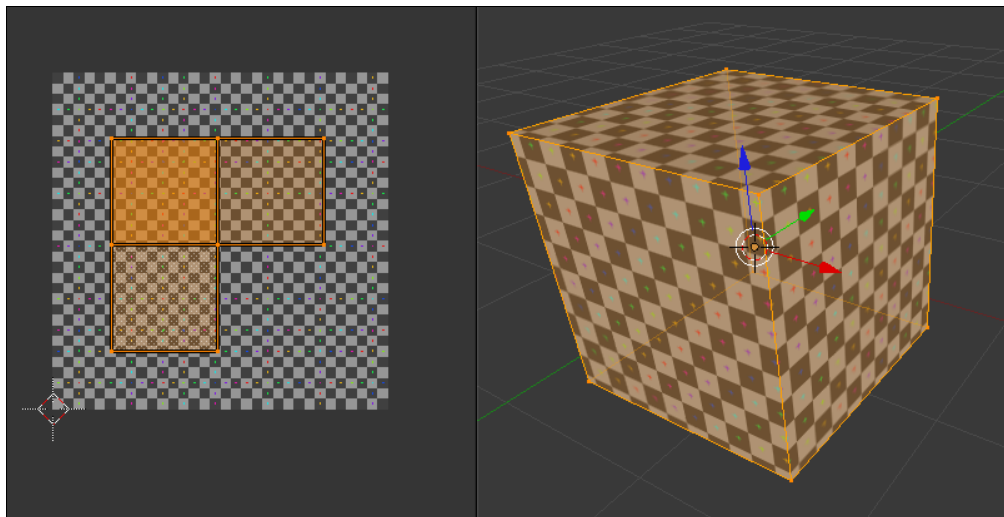


Рис. 4.88. Результат изменения масштаба развертки



Рис. 4.89. Кнопки управления выделением

ка. Ее назначение — в выделении островков. С этим набором переключателей можно не бояться заблудиться в дебрях развертки.

Куб, который вы сейчас текстурируете, — очень простой объект. Но бывают модели, состоящие из сотен, а то и тысяч полигонов. Для таких объектов развертка может быть очень сложной. Конечно, можно использовать систему "островков" и вручную выбирать полигоны для их создания, а можно поручить это дело программе.

Seam (Шов) — инструмент, позволяющий с легкостью "разрезать" модель на куски, пригодные для создания развертки. Естественно, никакого реального разделения объекта не происходит, просто таким образом указываются границы, по которым Blender сможет выполнить развертку.

Работа эта выполняется в окне **3D View**. Принцип действия заключается в следующем:

1. *Выбор области.* Выделять лучше всего ребра, т. к. именно по ним будет проходить шов. Выделение должно представлять собой замкнутую область. Удобно использовать инструмент **Edge Loop**, который старается выделить замкнутые ребра (используйте мышь совместно с нажатой клавишей <Ctrl>).
2. *Создание шва.* Для этого используйте меню: **Mesh | Edges | Mark Seem**. Отмеченные ребра окрасятся в красный цвет (рис. 4.90).

А дальше все просто. После создания нужных областей открываете меню **UV Mapping** (<U>) и выбираете пункт **Unwrap**. Если полученные островки не устраи-

вают, то можно удалить ненужный шов и создать новый. Для удаления шва выделите ребра и выполните команду **Clear Seam** из меню **Mesh | Edges**.

Бывает ситуация, когда из нескольких островков развертки какие-то получились удачными. Чтобы случайно не нарушить нужные сетки, такие островки можно временно "заморозить". Нужно выделить островок и нажать клавишу <P> в окне **UF Editor** (функция **Pin** меню **UVs**). Для снятия защиты служит комбинация клавиш <Alt>+<P>.

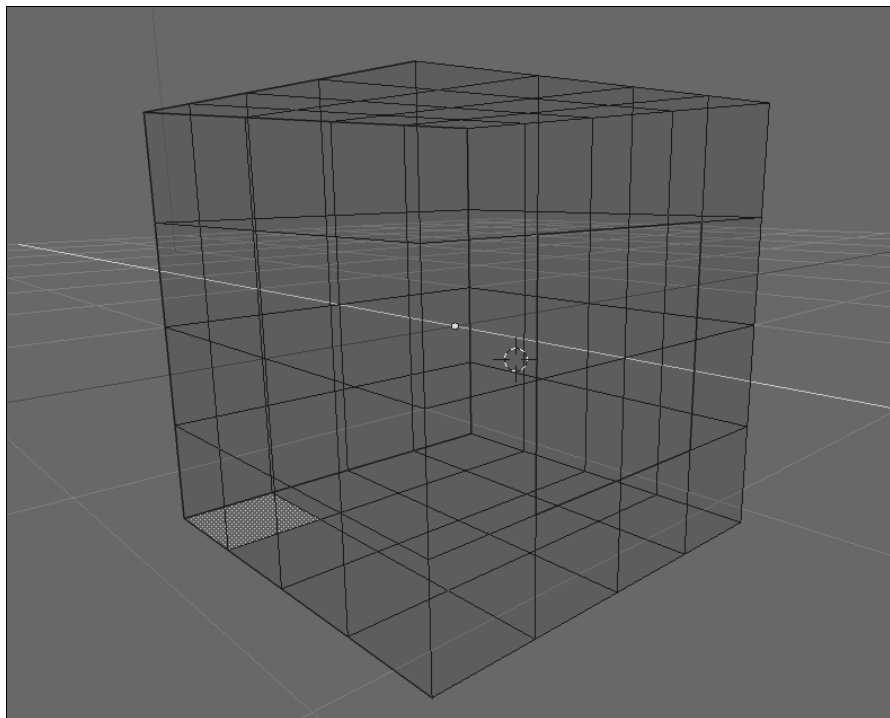


Рис. 4.90. Пример выделения **Seam**

Допустим, нужная развертка выполнена. Хорошо, если уже имеется готовая текстура, и сетка подогнана под нее, но бывает обратная ситуация, когда под сетку рисуется изображение. Blender имеет возможность сохранения развертки в виде обычного графического файла. Такой файл можно загрузить в любой двухмерный редактор и создать контент по имеющейся схеме. Чтобы сохранить развертку как картинку, используйте функцию **UVs | Export UV Layout**.

4.12. Ручная окраска текстуры и вершин

Blender предлагает возможность раскрашивания текстур, не выходя из приложения. Конечно, надобность в сторонней программе для редактирования текстуры не отпадает, но некоторые несложные вещи можно с успехом сделать прямо на месте.

Раскраска текстуры может выполняться в окне **UV/Image Editor**, что является очень удобным: создали текстуру, затем развертку и тут же окрасили нужными цветами.

Для включения режима раскраски на заголовке окна редактора имеется кнопка с характерным рисунком карандаша (рис. 4.91).



Рис. 4.91. Кнопки управления раскраской

После ее включения панель слева окна дополняется несколькими закладками. В принципе они практически ничем не отличаются от такой же панели, которая используется в **Sculpt Mode**.

Панель **Paint** (рис. 4.92) позволяет создать или выбрать кисть, а также установить ее цвет. Рассмотрим основные кисти:

- ◆ **Brush** — стандартная кисть для рисования;
- ◆ **Clone** — позволяет микшировать две текстуры, где одна базовая, а вторая выбирается в поле **Image**. Эффект при рисовании получается, как будто кисть стирает базовый слой, а сквозь него проступает вторая текстура;
- ◆ **Smear** — эта кисть выполняет растирание, размазывание рисунка по поверхности текстуры;
- ◆ **Soften** — в отличие от **Smear**, данная кисть выполняет простое размывание, не нарушая форму рисунка.

Помимо рисования простыми цветами палитры, **UV Editor** позволяет использовать отдельные текстуры. Эта функция доступна в группе **Textures**. Чтобы выбрать текстуру, нужно щелкнуть мышью по образцу на панели.

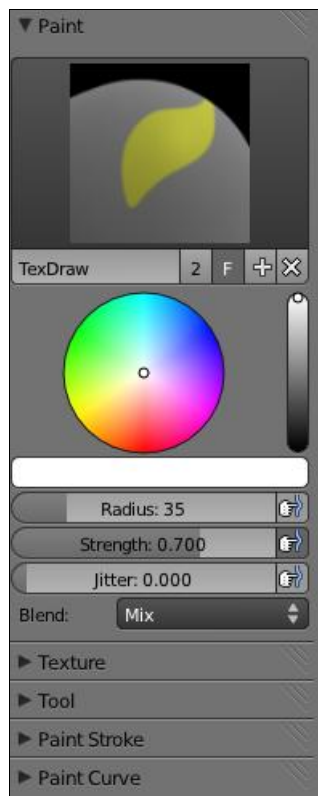


Рис. 4.92. Панель Paint

Рисование текстурой может быть сплошным (параметр по умолчанию) и фиксированным. В первом случае при движении кисти получается смазанная полоса, во втором кисть как бы открывает части сторонней текстуры. Включение и отключение этих режимов возможно с помощью опции **Fixed Texture** (Фиксация текстуры).

Остальные возможности режима рисования ничем не отличаются от таких же в **Sculpt Mode**. Это группы: **Tool** (Инструменты), **Paint Stroke** (Настройка кисти), **Paint Curve** (Поведение кисти).

Подобная раскраска текстуры возможна и в окне **3D View**, непосредственно на самой модели. Для этого нужно выбрать режим **Texture** в меню **Viewport Shading**, создать развертку модели и сопоставить ей текстуру.

Включение режима рисования выполняется с помощью выбора пункта **Texture Paint** в меню **Mode** окна **3D View** (рис. 4.93). Все инструменты панели **Tool Shelf** соответствуют таким же в окне **UV/Image Editor**.

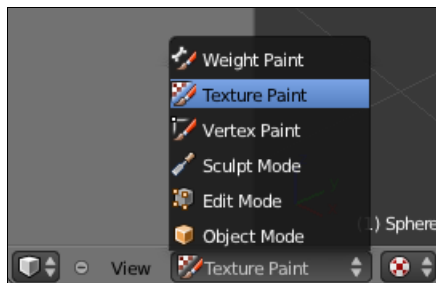


Рис. 4.93. Меню Mode

Помимо раскраски текстуры Blender предлагает окраску вершин. Причем выполняется это действие непосредственно на модели в окне **3D View**.

Vertex Paint — это возможность ручной окраски вершин объекта любыми цветами палитры и совмещение их с текстурами по необходимости. Нужно знать, что одновременная работа базового цвета **Diffuse** объекта и окрашенных вершин невозможна.

Для включения режима рисования нужно выбрать **Vertex Paint** в меню **Mode** (см. рис. 4.93).

В этой версии Blender разработчики постарались облегчить изучение программы и максимально унифицировать похожие функции. Поэтому нет ничего удивительного в том, что режимы **Sculpt Mode**, **Texture Mode**, **Vertex Mode** имеют одинаковый набор панелей и инструментов. Рассмотрим только предлагаемые кисти для окраски вершин:

- ◆ **Add** (Добавление) — смешивание цветов вершин путем прибавления нового цвета к старому;
- ◆ **Blur** (Размывание) — выполнение размывания цвета. При этом палитра не используется;
- ◆ **Brush** (Кисть) — основная кисть для рисования;
- ◆ **Darken** (Затемнение);
- ◆ **Lighten** (Осветление);
- ◆ **Mix** (Смешивание) — оба цвета смешиваются в зависимости от установленного коэффициента;
- ◆ **Multiply** (Умножение) — умножение двух цветов. Получаются более мягкие переходы;

- ♦ **Pinch/Magnify** (Стянуть/Увеличить) — постепенное изменение цвета путем стягивания;
- ♦ **Subtract** (Вычитание) — значение RGB нового цвета вычитается из имеющегося. Эта кисть может пригодиться для создания резких, контрастных переходов.

По умолчанию обработка окрашенных вершин в материале выключена, хотя в окне **3D View** цвета будут видны. Для управления этой возможностью есть опция **Vertex Color Paint** (Окраска вершин), расположенная в свойствах материала группы **Options**. Включите ее, если хотите обрабатывать окрашенные вершины, вместо базового цвета материала.

4.13. Практика. Замшелый камень

Вроде, что такое камень! Как говорится, раз-два и готова модель. Вот только использовать в работе придется большое количество инструментов, если, конечно, очень хочется добиться приемлемого результата.

Вы уже прочитали половину книги и, поверьте, способны на многое — создать качественную модель и снабдить ее не менее качественным материалом. По сути, этот урок подытоживает почти весь прочтенный материал. В работе будут задействованы инструменты скульптурного моделирования, придется вспомнить особенности использования объекта **Lattice**, а материал снабдить многочисленными текстурами с разным смешиванием. Скупиться на этот раз не придется, конечная модель будет обладать большим количеством полигонов и достаточно медленно просчитываться. Визуально камень должен выглядеть мокрым, покрытым водорослями, как давно находившийся в воде.

В качестве основы будет использоваться примитив **UV Sphere**. Добавьте его в сцену, а куб удалите. Сфере нужно придать овальную форму. Проще всего выделить примитив и растянуть с помощью масштабирования по оси X (нажмите клавишу <S>, а затем <X>).

Овальная форма не совсем подходит для камня, значит, нужно ее немного изменить. Можно это сделать в режиме редактирования, путем смещения групп вершин, а можно использовать вспомогательный объект **Lattice**. Последний вариант наиболее предпочтительней и проще в исполнении.

Добавьте в сцену объект **Lattice (Add | Lattice)**. Нажмите <S> и растяните его так, чтобы сфера оказалась внутри него. По умолчанию **Lattice** имеет всего 8 вершин, что недостаточно для деформации примитива. Откройте панель **Object Data** в окне **Properties** и добавьте дополнительные ребра в параметрах **UVW** (рис. 4.94).

Для использования **Lattice** к сфере нужно прикрепить модификатор с одноименным названием. Не забудьте указать в настройках модификатора сам **Lattice** (опция **Object**).

А вот дальше все в ваших руках. Используйте вершины **Lattice** для изменения формы объекта, как заблагорассудится. Можете выделять сразу несколько точек с помощью рамки (). У меня лично получилось вот так (рис. 4.95).

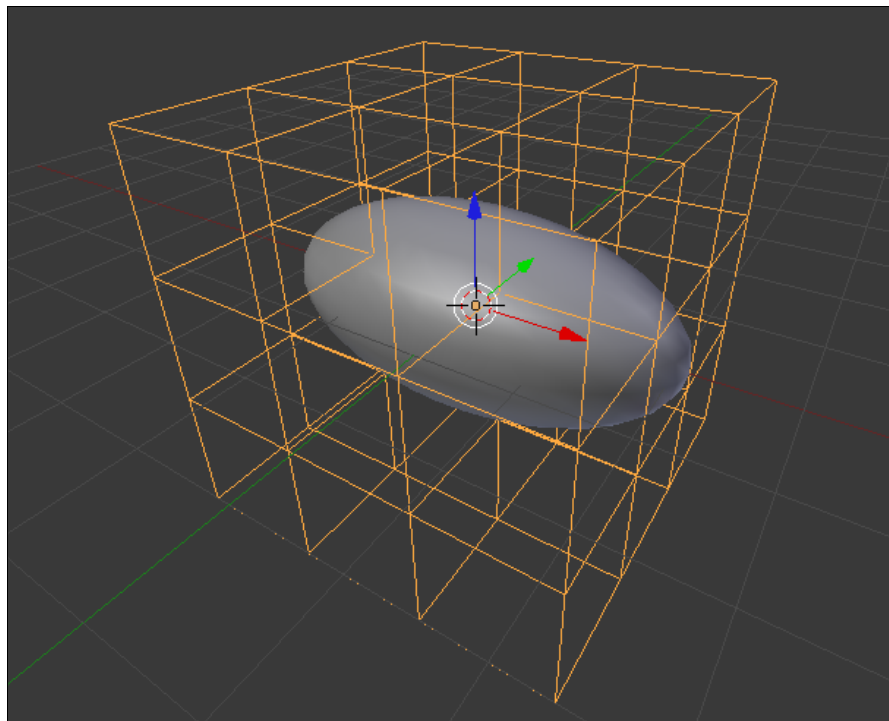


Рис. 4.94. Все готово к изменению формы примитива

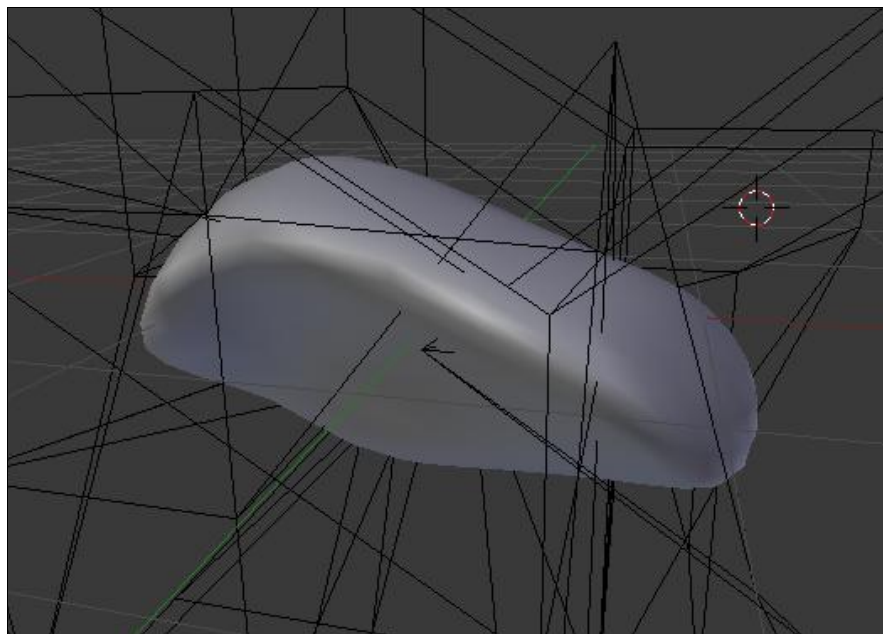


Рис. 4.95. Общая форма будущего камня

Это всего лишь общая заготовка будущего камня. Нужно добавить к ней более мелкие неровности. Разумеется, вручную делать это не будем, а воспользуемся процедурными текстурами в режиме скульптуры.

Сначала нужно убрать из сцены отработавший свое объект **Lattice**. Лучше не удалять его, а просто скрыть. Выделите **Lattice** и нажмите клавишу <H> (функция из меню **Object | Show/Hide | Hide Selected**). Эта команда временно убирает выделенный объект из сцены. Он продолжает функционировать, но его не видно. Если понадобится вернуть на место скрытые таким способом объекты, то используйте клавиши <Alt>+<H>.

Подготовим все необходимое для работы в скульптурном режиме. Сначала добавьте объекту новый модификатор **Multires**, ведь исходной структуры явно недостаточно. Добавьте пару уровней разрешения (кнопка **Subdivide**). Перейдите в режим скульптурного моделирования (**Mode | Sculpt Mode**) и откройте панель **Textures** в окне **Properties**. Добавьте новую текстуру, которая будет использоваться кистью для моделирования. В качестве типа вполне годится **Musgrave** с параметрами по умолчанию.

И вот настал момент рисования. Кисть остается по умолчанию, а вот поведение кривой нужно изменить так, чтобы она равномерно деформировала по всей рабочей области. Откройте на панели **Tool Shelf** закладку **Curve** и нажмите крайнюю кнопку (см. рис. 4.72).

Прежде чем начать лепку, проверьте на закладке **Texture** для кисти наличие созданной текстуры **Musgrave**. Если окошко **Preview** пустое, то щелкните по нему и из предложенного окна выберите нужную текстуру.

Процесс деформации — дело сугубо личное, тут сложно дать какие-либо рекомендации. Крутите сцену, используйте колесико мыши для масштаба и создавайте неровности на свой вкус (рис. 4.73).

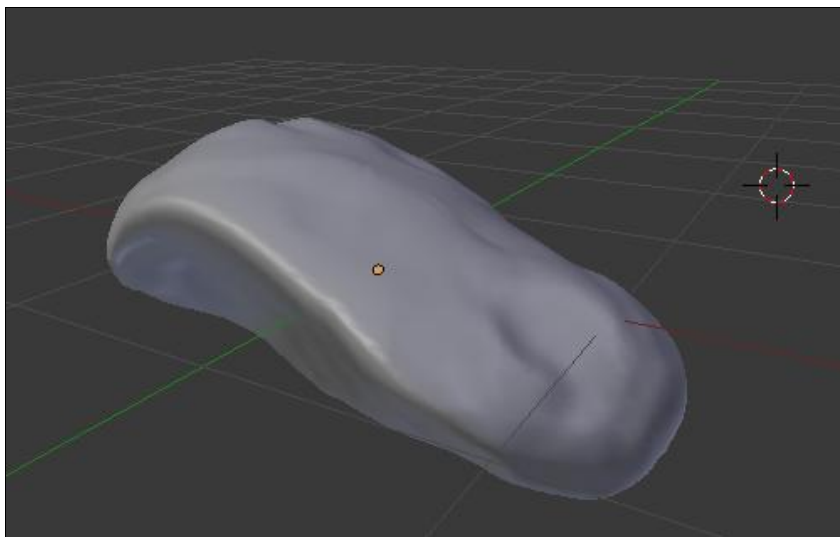


Рис. 4.96. Модель после обработки в **Sculpt Mode**

СОВЕТ

Деформация поверхности объекта в режиме скульптуры может сопровождаться появлением неприятных стяжек. Вы можете избавиться от них с помощью кисти **Flatten/Contrast**.

Итак, создание и грубая обработка формы будущего камня на этом завершена. Настало время заняться его раскраской.

На панели **Material** нажмите кнопку **New** для создания и привязки материала к объекту.

Цвет для **Diffuse** будет светло-коричневый. Установите для базового цвета следующие значения RGB:

◆ **R** = 0.152

◆ **G** = 0.126

◆ **B** = 0.103

Поверх основного цвета нужно расположить зеленый, который будет играть роль водорослей. Для этого добавьте новую текстуру к материалу и назовите ее "Base". В качестве типа оптимально подходит **Clouds**. Настройте ее параметры с помощью указанных значений:

◆ в группе **Noise** нажмите кнопку **Soft**;

◆ параметр **Size** установите равным 1.30;

◆ параметр **Depth** сделайте 2.

Теперь нужно указать текстуре, как она будет смешиваться с базовым материалом. На закладке **Influence** включите опцию **Diffuse Color**. Установите для ее цвета следующие значения:

◆ **R** = 0.232

◆ **G** = 0.120

◆ **B** = 0.198

А вот режим смешивания в меню **Blend** нужно изменить с привычного **Mix** на **Subtract**. В результате этих действий обработка сцены (<F12>) должна выдать картинку, внешне похожую на рис. 4.97.

Первый результат обработки выдает блеклую и совсем нереалистичную картинку. Здесь явно не хватает мелких деталей. Добавим еще один текстурный уровень, который немного разнообразит цвет камня.

Щелкните мышью в пустой слот **Textures** и создайте новую текстуру. Назовите ее "Noise". Задача состоит в дополнении несложными узорами материала камня, чтобы он не выглядел столь пустынно. Здесь нельзя использовать текстуры типа **Clouds** — они выдают слишком насыщенную шумами структуру. Можно попробовать воспользоваться одной из органических функций, например **Distorted Noise**.

Установите этот тип текстуры и измените ее базовые параметры:

◆ **Distortion** = 7.440

◆ **Size** = 0.70

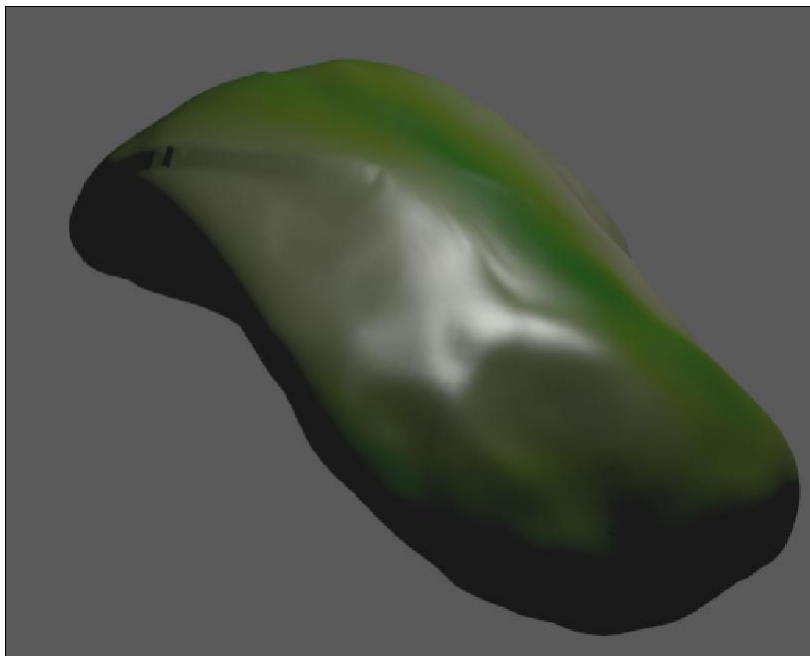


Рис. 4.97. Базовый цвет + текстура **Base**

В качестве шумовых функций пусть остается без изменений алгоритм Blender **Original**.

В параметрах закладки **Influence** нужно изменить настройки двух опций:

◆ Цвет текстуры:

- **R** = 1.000
- **G** = 0.940
- **B** = 0.967

◆ Включите опцию **Color** группы **Diffuse** и измените ее значение на 0.135.

Попробуйте обработать картинку. Примерно получится, как на рис. 4.98.

Камень выглядит интереснее, но он все еще слишком гладкий. Добавление мелких неровностей придаст материалу шероховатость и эффект влажности. На этот раз новая текстура будет использоваться как рельефная карта.

Добавьте новую текстуру в пустой слот и измените ее тип на **Clouds**. По умолчанию узор ее выглядит слишком крупным. Откройте параметры закладки **Clouds** и подкорректируйте их:

- ◆ в группе **Noise** нажмите кнопку **Hard**;
- ◆ шумовая функция остается **Blender Original**;
- ◆ параметр **Size** сделайте равным 0.20;
- ◆ глубина (**Depth**) пусть будет равна 6.

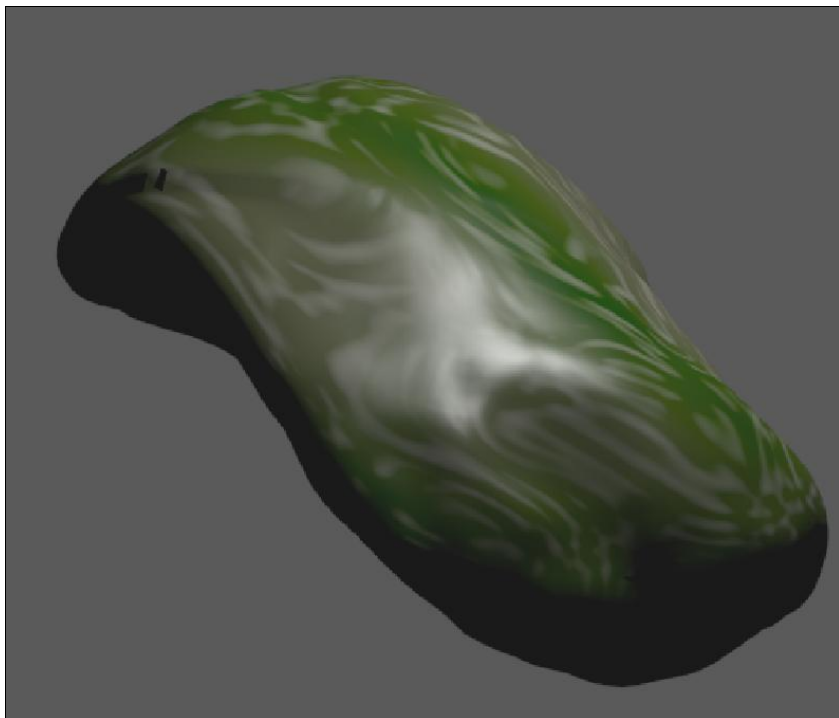


Рис. 4.98. Результат обработки с двумя текстурами

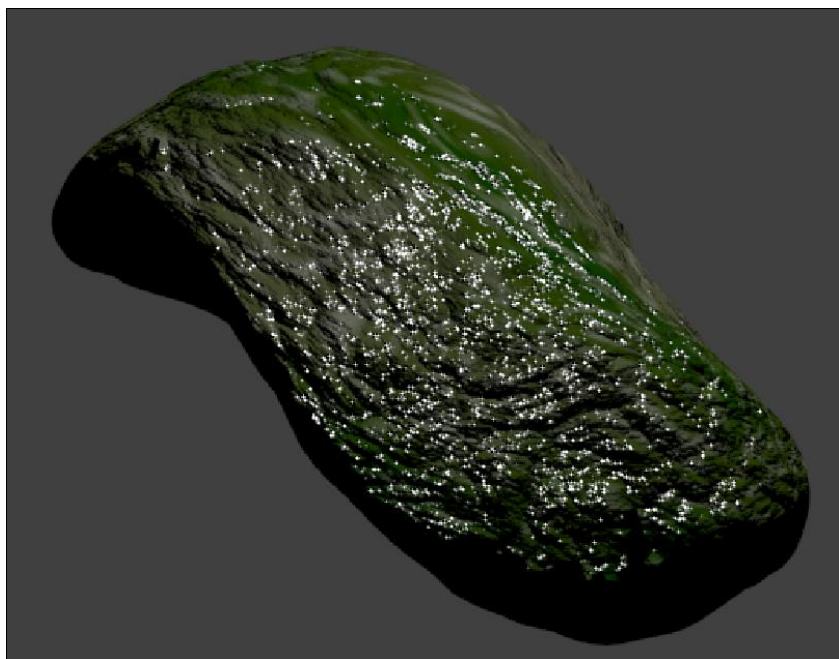


Рис. 4.99. Реалистичная модель камня

В настройках закладки **Influence** отключите опцию **Color**, т. к. цвет текстуры тут не нужен. Включите опцию **Normal** в группе **Geometry** и установите ее значение в 1.

Работа с текстурами на этом завершена. Осталось только немного подкорректировать зеркальный шейдер материала. По условию задачи камень должен выглядеть как только что вынутый из воды. Поэтому шейдер **Specular** должен выдавать резкий, четкий отблеск. С этой задачей справится функция **Wardiso**. Установите этот шейдер на закладке **Specular** и нажмите клавишу <F12> для обработки (рис. 4.99). Модель готова!

4.14. Практика. Сочное яблоко

Работая над книгой, я меланхолично грыз яблоко и раздумывал, какой предмет использовать для моделирования в этом уроке. Рассматривая предметы в комнате, я остановил взгляд на яблоке, которое продолжал держать в руке. А почему бы и нет? Сочный, крупный плод так и просился запечатлеться на экране монитора. Светло-желтое яблоко с небольшими вкраплениями областей зеленого цвета выглядело эффектно.

Итак, в этом уроке вы узнаете, как создать модель реалистичного яблока. Будут использоваться рамповые шейдеры, минимум текстур и очень простая заготовка **Mesh**.

Создайте новый проект. Имеющийся в центре куб не нужен, поэтому удалите его. Основой модели послужит **UV Sphere**. Добавьте ее в проект.

Структура примитива для этой задачи содержит слишком мало элементов. Перейдите в режим редактирования (<Tab>), нажмите клавишу <W> и выберите пункт **Subdivide**.

Форма яблока очень простая. Та же сфера, но немного сплюснутая с полюсов, причем неоднородно. Сверху, где находится черенок, вдавливание больше, а снизу меньше.

Выделите центральную вершину в верхней части примитива. Если просто попытаться переместить ее по координате Z, то она уйдет внутрь объекта, не изменив его формы. Здесь нужно использовать режим пропорционального редактирования. Нажмите клавишу <O> для его включения. Теперь можно смело изменять форму.

Включите манипулятор **Grab** (<G>), нажмите клавишу <Z> для движения по оси Z и переместите вершину немного вниз. Масштаб охвата появившейся круговой рамки управляется с помощью колесика мыши. Манипулируя масштабом окружности и движением вершины, добейтесь формы примитива, как на рис. 4.100.

То же самое действие нужно проделать и с нижней частью объекта, только с меньшим углублением.

Займемся созданием черенка. Включите просмотр сцены **Top View** в ортогональной проекции (<NumPad 7>). Нужно выделить часть полигонов для формирования черенка. Боксовая рамка здесь не подходит, но Blender имеет еще один вариант — круглую рамку (быстрая клавиша <C>). Переключите на заголовке окна **3D View** режим выделения граней и отметьте полигоны, как на рис. 4.101.

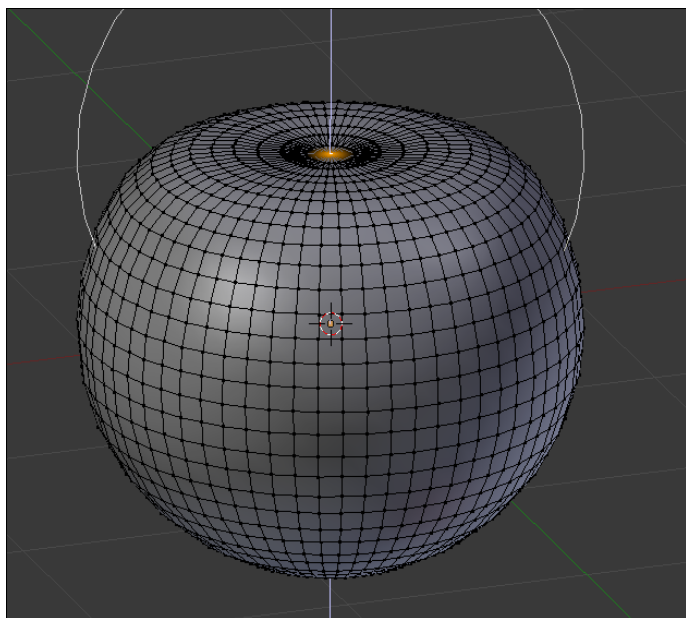


Рис. 4.100. Создание углубления под черенок

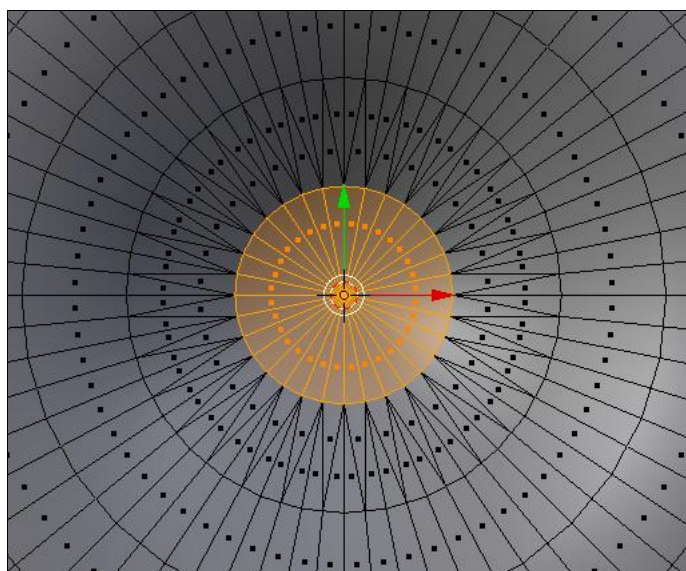


Рис. 4.101. Выделение области для черенка

Эту часть модели нужно выдавить, немного передвинуть и сжать под размер черенка. Сначала выключите режим пропорционального редактирования, чтобы не мешался. Нажмите клавишу <E> для вызова функции **Extrude** и немного передвиньте новые полигоны вверх. Используя инструмент масштабирования (<S>), сожмите выделение до подходящего размера.

Создать черенок просто. Используйте **Extrude** для создания очередной порции полигонов и немного вытягивайте вверх. После каждого этапа выдавливания можно немного наклонить выделение, чтобы черенок принял изогнутый вид. Лучше всего переключиться в просмотр **Front View** (<NumPad 1>), где проще контролировать длину и наклон черенка. Для наклона используйте клавишу <R> совместно с клавишей <Y>, чтобы ограничить ротацию координатой Y. В конце черенок можно немного увеличить при помощи масштабирования (рис. 4.102).

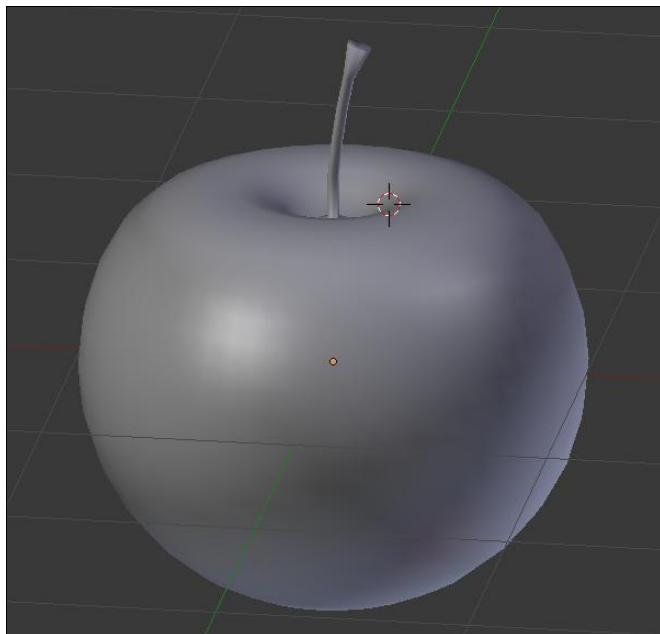


Рис. 4.102. Готовая форма яблока

На этом этап моделирования закончен, настало время заняться раскраской. Так как цвета самого яблока и черенка разные, то использовать будем возможности мультиматериалов.

Выделите черенок и основание примитива, как на рис. 4.103. Удобнее будет включить режим выделения граней.

Создайте новый материал, нажмите кнопку **Assign** для привязки к выделению и назовите его "Slip".

Нажмите кнопку "+" рядом со списком материалов для создания нового слота. Создайте еще один материал и назовите его "Base". Чтобы не мучиться со вторичным выделением уже самого яблока без черенка, можно воспользоваться командой инвертирования. Выберите пункт **Inverse** из меню **Select** или нажмите клавиши <Ctrl>+<I>. Теперь можно прикрепить инвертированное выделение к новому материалу.

Сначала займемся самым сложным — настройкой материала яблока. Сложным по сравнению с черенком, а не по действию — и в этом вы сейчас убедитесь.

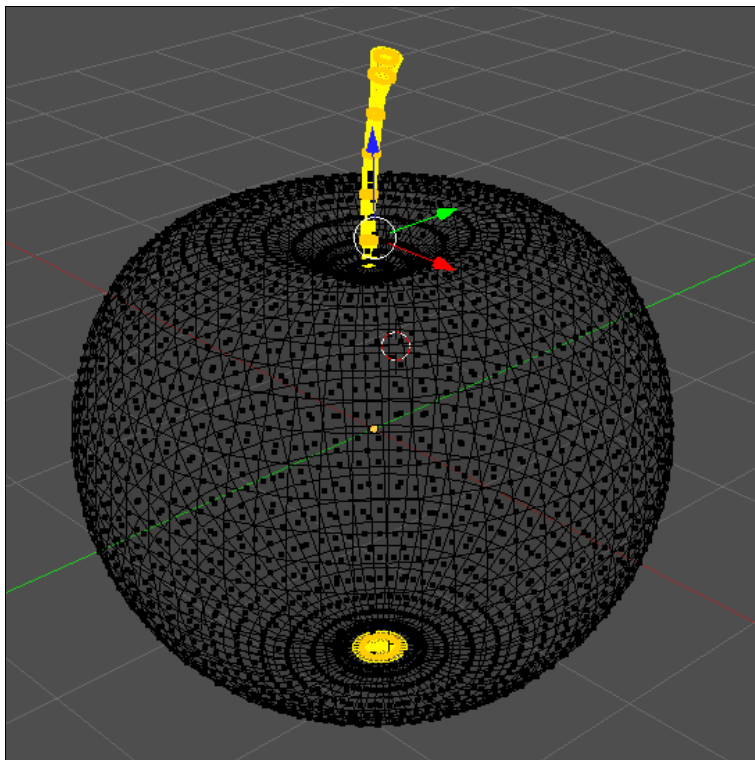


Рис. 4.103. Выделение элементов для первого материала

В качестве основного цвета установите цвет **Diffuse** как зеленый:

- ◆ **R** = 0.133
- ◆ **G** = 1.000
- ◆ **B** = 0

Это весьма ядовитый цвет, но при помощи рампового шейдера он значительно смягчится и примет нужные оттенки.

Включите опцию **Ramp** на закладке **Diffuse**. Как вы, наверное, помните, по умолчанию Blender уже имеет созданные два ключа. В принципе этого будет достаточно.

Выберите первый ключ (с точки зрения Blender он будет нулевым) и измените его цвет:

- ◆ **R** = 0.145
- ◆ **G** = 1.497
- ◆ **B** = 0
- ◆ **A** = 1

Для второго ключа будет использоваться желтый цвет со следующими параметрами RGB:

- ◆ $R = 1$
- ◆ $G = 0.984$
- ◆ $B = 0$
- ◆ $A = 1$

В качестве входа (**Input**) нужно оставить **Shader**, а наложение будет выполняться с помощью стандартного **Mix** (рис. 4.104)

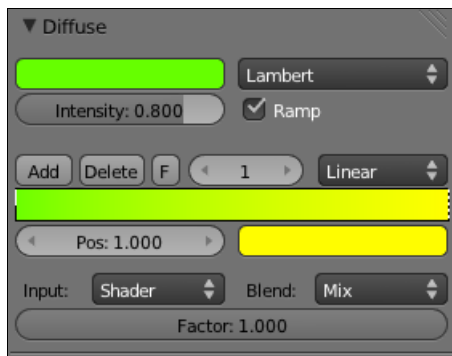


Рис. 4.104. Выделение элементов для первого материала

Теперь нужно немного настроить блики. Лучше всего для яблока подойдет шейдер **Phong**. Установите его в группе **Specular**.

Осталось настроить материал черенка. Выделите материал **Slip** в списке мультиматериалов. Установите следующие цвета для **Diffuse**:

- ◆ $R = 0.119$
- ◆ $G = 0.07$
- ◆ $B = 0.03$

Для черенка лучше всего практически выключить блики. Поэтому в группе **Specular** установите для параметра **Intensity** значение 0.

И в качестве последнего рывка поиграем немного со светом в сцене. Выделите лампу (объект **Lamp** в окне **Outliner**), нажмите клавишу <N> для вывода свойств объекта и введите следующие координаты для группы **Location**:

- ◆ $X = 7.900$
- ◆ $Y = -2.500$
- ◆ $Z = 7.000$

Одной лампы в сцене явно мало, поэтому нажмите <Shift>+<D> для создания копии объекта **Lamp**. Выделите новую лампу и измените для нее координаты:

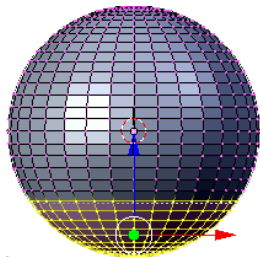
- ◆ $X = 0.400$
- ◆ $Y = -13.600$
- ◆ $Z = -1.000$

Вот теперь все готово, нажмите <F12> и наслаждайтесь результатом (рис. 4.105).



Рис. 4.105. Готовая модель яблока

ГЛАВА 5



Анимация

На протяжении четырех глав вы постигали основы и премудрости моделирования, но вдохнуть жизнь в свое детище поможет именно эта часть книги.

Вспомните современные фильмы, где трехмерная графика органично переплетается с живыми актерами и реальными декорациями. Это и нашумевшая сага о Вампирах, и ставшие бестселлерами серии Матрицы, и... список можно продолжать до бесконечности. 3D буквально пронизывает все современное кинематографическое искусство. Конечно, многое зависит от качества моделей, но если они в сцене двигаются неестественно, то на реалистичности можно поставить крест.

Создание анимации в трехмерной программе заслуженно считается самой сложной темой для изучения начинающими пользователями. Обилие инструментов в Blender позволяет воплотить в реальность любые смелые идеи, вот только выбрать нужные — задача не из легких.

В этой главе вы узнаете о базовых принципах анимации объектов, научитесь использовать деформацию в движении и, конечно же, разберетесь со скелетной анимацией, которая применяется для сложных моделей.

5.1. Основы анимации в Blender

Начнем с того, что в Blender можно анимировать практически все, начиная от самих объектов и заканчивая текстурами. В большинстве случаев для этого используются одни и те же инструменты, что сильно облегчает работу. Всего имеется четыре вида анимации.

- ◆ *Анимация объектов в сцене.* Под этим понимается, что любые манипуляции с объектами могут быть анимированы, такие как: масштабирование, перемещение, ротация.
- ◆ *Внутренняя анимация.* Уникальная возможность запечатлеть в движении изменения внутренней структуры объекта.

- ◆ **Скелетная анимация.** Использование вспомогательных объектов ("костей") для анимации частей сложных моделей.
- ◆ **Физика.** Анимация выполняется с учетом законов физики (см. главу 6).

Все типы анимации могут быть применены к одному и тому же объекту.

Краеугольным камнем любой анимации является использование "ключевого кадра". *Ключ* (key) — это способ фиксации текущих параметров объекта в определенный момент времени. Таким образом, для создания простейшей анимации в сцене достаточно установить начальный и конечный ключевые кадры, а программа дополнит недостающие фазы движения. Любые зафиксированные параметры объекта можно изменять, добавлять или удалять. В свою очередь и ключи поддаются редактированию — перемещению их по временной шкале.

Blender предлагает несколько встроенных редакторов для корректировки ключей, их значений и анимации в целом.

- ◆ **Timeline** (Временная шкала). Удобное средство для управления просмотром анимации в целом. Представляет собой временную шкалу, отображающую ключи выделенного объекта и снабженную интерфейсными элементами для проигрывания анимации. Здесь нет возможности редактирования параметров ключей.
- ◆ **Graph Editor** (Редактор кривых). Используется для просмотра и редактирования ключей выделенных объектов. Каждый зафиксированный параметр представлен в окне редактора в виде кривой. С помощью кривых можно точно настраивать поведение объекта.
- ◆ **Dope Sheet** (Таблица ключей). Редактор, предлагающий несколько режимов для различных видов анимации, но с одинаковым интерфейсом и управлением:
 - **Dope Sheet** — режим управления ключами с параметрами всех объектов, участвующих в анимации. Используется для точной подгонки времени выполнения анимации конкретного объекта по отношению к сцене. Это уже редактор более высокого уровня и по сравнению с **Graph Editor** не имеет возможности корректировки параметров ключа;
 - **Action Editor** (Редактор действий) — аналогичный режим, но только для выделенного объекта. Позволяет создавать так называемые *действия* (**Action**);
 - **Shape Key Editor** (Редактор ключей формы) — режим для управления внутренней анимацией объекта.
- ◆ **NLA Editor** (Редактор NLA). Высокоуровневый редактор для окончательного монтажа сценической анимации. Вся анимация представлена в виде элементов (действий), которые можно перемещать (сдвигать во времени), изменять скорость, тиражировать.

5.2. Простое управление с *Timeline*

При запуске программы окно **Timeline** (рис. 5.1) отображается в нижней части экрана (по умолчанию). Сделано это вовсе не зря, т. к. оно позволяет с легкостью управлять просмотром всей анимации сцены и переходом по ключам конкретного

объекта. Но прежде чем осуществлять проигрывание анимации, ее нужно собственно создать.

Создание анимационных кадров осуществляется с помощью окон **3D View** и **Timeline**. Возможно как автоматическое, так и ручное добавление ключей.

Ключ — это обобщающее название для зафиксированных в определенный момент параметров объекта. В одном ключевом кадре могут одновременно находиться координаты объекта, его размер, ротация. Их можно устанавливать и редактировать независимо друг от друга. В анимации Blender такие фиксированные параметры называются *ключевыми каналами* (key channels).



Рис. 5.1. Окно **Timeline**

Рассмотрим сначала вариант с автоматическим добавлением ключей. Смысл его заключается в том, что программа улавливает любые манипуляции, проводимые с объектом в окне **3D View**, и создает ключ с нужным каналом для активного кадра.

Посмотрите на рис. 5.1. Большую часть окна **Timeline** занимает шкала, разбитая на области, подписанные цифрами. В данном случае цифры обозначают кадры, но могут выводиться значения в секундах. На заголовке окна имеются поля **Start** и **End**, которые позволяют устанавливать начало и конец для всей анимации сцены. Именно этот кусочек будет проигрываться и обрабатываться рендером программы. А вот соседнее с ними поле **Current Frame** (по умолчанию там стоит единица) указывает на активный кадр сцены. Таким образом, для установки нового ключа необходимо выбрать сначала активный кадр. Это можно сделать, изменив значение в поле **Current Frame** либо щелкнув левой кнопкой мыши по шкале. В любом случае **Timeline** визуально отобразит в указанном месте вертикальную зеленую полосу — курсор. Перемещая последний, можно просматривать анимацию в **3D View**.

После выбора активного кадра нужно включить режим записи анимации. Для этого необходимо нажать кнопку с красным кружком на заголовке окна (рис. 5.2). Теперь при перемещении, ротации или масштабировании объекта будет установлен или обновлен ключевой кадр.



Рис. 5.2. Кнопка автоматической записи и ее настройки

Допустим, нужно сдвинуть объект от точки А до точки Б. Так как автоматическая запись работает только при каком-либо действии, то придется временно сдвинуть, а затем вернуть объект в прежнее положение — это первое, что может прийти в голову. На самом деле в этом нет надобности. Необходимо только воспользоваться

ручным способом установки ключа. Достаточно установить объект в нужном месте, нажать клавишу <I> в окне **3D View** и выбрать из появившегося меню **Insert Keyframe** (Вставить ключевой кадр) пункт **Location** (Позиция). Эта команда создаст для активного кадра ключ с каналом позиции объекта. А вот дальше можно смело использовать автоматическую запись.

Итак, стандартный алгоритм создания анимации:

1. Подготовить объект для запоминания первой позиции.
2. Нажать клавишу <I> в активном окне **3D View** и выбрать из меню нужный канал. Наиболее используемые: **Location** (Перемещение), **Rotation** (Ротация), **Scalling** (Масштабирование). Остальные представляют собой возможные комбинации из этих трех.
3. Включить кнопку записи.
4. Выбрать другой активный кадр с помощью поля **Current Frame** или просто передвинуть зеленый курсор в окне **Timeline**.
5. Передвинуть объект на новое место.
6. Отключить кнопку записи.
7. В принципе автоматическую запись использовать не нужно, если необходимо проставить только два ключа, но в случае сложной, многключевой анимации этот режим выручает.

ЭТО ВАЖНО!

В **Timeline** ключевой кадр отображается на шкале в виде вертикальной полоски, причем она может быть разного цвета. Каждый цвет обозначает определенную группу каналов: желтый — ключи манипуляции объекта; оранжевый — ключи, связанные с материалом; голубой — ключи скелетной анимации.

Рассмотрим другой пример. Допустим, имеется сложная анимация движения объекта. К ключевым кадрам нужно добавить еще изменение ротации. Если аккуратно устанавливать ротацию объекта от ключа к ключу, то проблем не возникнет. Но вполне вероятна ситуация, когда вместе с ротацией выполнялась ненужная манипуляция, например случайное перемещение. Естественно, Blender перезапишет канал и тем самым испортит анимацию.

Чтобы этого не случилось, в окне **Timeline** имеется возможность установки типа канала для автоматической записи. В итоге, какие бы манипуляции с объектом не выполнялись в ключевом кадре, перезаписанным будет только выбранный ранее канал.

Для использования этой функции необходимо из списка **Keying Set** выбрать нужный канал и включить кнопку маскирования (она расположена рядом с кнопкой записи). На рис. 5.3 как раз изображена настройка канала применительно к рассматриваемому примеру.

Взгляните опять на рис. 5.3. Рядом с меню **Keying Set** имеются еще две кнопки. Нажатие на первую установит ключ в соответствии с выбранным каналом в меню

Keying Set, а вот вторая с изображением перечеркнутого ключа, наоборот, удалит из кадра ключ выбранного канала. Итак, эта часть панели **Timeline** позволит вам максимально осторожно работать с ключами анимации.

Осталось только рассмотреть элементы управления анимацией в **Timeline** (рис. 5.4):

1. Перемотка в начало всей анимации в соответствии со значением в поле **Start**.
2. Переход на предыдущий ключевой кадр выделенного объекта.
3. Проигрывание инверсной анимации.
4. Проигрывание анимации вперед.
5. Переход на следующий кадр анимации.
6. Перемотка в конец анимации (параметр **End**).
7. Меню **Sync Mode**.

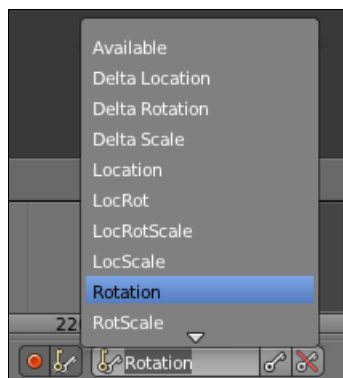


Рис. 5.3. Меню выбора активного канала

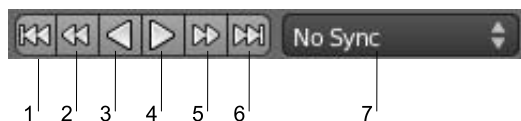


Рис. 5.4. Кнопки управления анимацией

Особо стоит остановиться на меню **Sync Mode**, которое позволяет выбрать способ воспроизведения анимации:

- ◆ **No Sync** (Без синхронизации) — стандартный режим, когда анимация в окне воспроизводится без звука и не в режиме реального времени;
- ◆ **AV-Sync** (Синхронизация с видео) — если какой-то объект в сцене отображает видео со звуком, то эта опция позволяет просматривать анимацию и прослушивать звук;
- ◆ **Frame Dropping** (Пропуск кадров) — этот вариант пригодится для просмотра в реальном времени анимации сложной сцены, когда стандартный **No Sync** не способен обеспечить нужную скорость. В этом режиме Blender будет пропускать некоторое количество кадров, которые позволят обеспечить нужную скорость проигрывания. Количество кадров для реального времени берется из настроек рендера программы.

5.3. Точная настройка анимации с *Graph Editor*

Graph Editor — это вспомогательный редактор для точной настройки анимации конкретного объекта с помощью кривых, которые визуальнo отображают изменение каналов во времени и пространстве. Достаточно непростой инструмент, поэтому работу с ним лучше рассматривать на небольших практических примерах.

Казалось, что сложного в создании простейшей анимации движения объекта? Тем не менее результат ее проигрывания нередко ставит в тупик начинающего пользователя. Чтобы понять, о чем идет речь, создадим это самое перемещение.

Откройте Blender и выберите раскладку окон **Animation** из главного меню программы. Переведите окно **3D View** в режим просмотра **Front View** ортогональной проекции.

Выделите куб, затем нажмите клавишу <I> и выберите пункт **Location** для фиксации текущей позиции объекта. Установите значение 50 в окошке активного кадра. Теперь переместите куб по координате X в любом направлении и опять создайте ключ **Location**.

Попробуйте проиграть анимацию, нажав соответствующую кнопку на панели **Timeline** или горячие клавиши <Alt>+<A>. Обратите внимание на то, что скорость движения объекта неравномерная. Маленькая при старте, она достигает максимального значения к середине анимации и так же плавно ослабевает к финишу. С одной стороны, это логично, с другой стороны, такое не всегда подходит для задуманного.

А вот теперь настало время открыть окно **Graph Editor**. Точнее, оно уже имеется и расположено в нижней левой части экрана (рис. 5.5).

Окно **Graph Editor** поделено на две зоны, в левой отображаются каналы выделенного объекта, а в правой — кривые этих самых каналов. Наверное, вы уже обра-

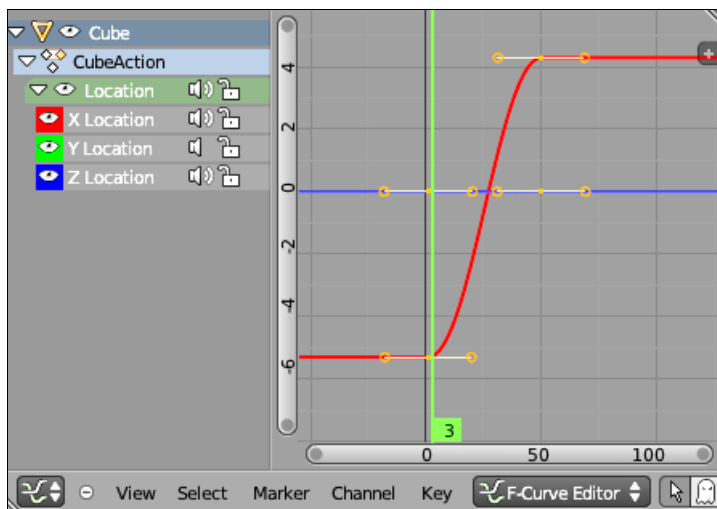


Рис. 5.5. Окно редактора кривых **Graph Editor**

тили внимание, что одна кривая располагается несколько иначе, нежели другие. Если учесть, что каждый сплайн графически изображает поведение отдельной координаты куба, то необычная кривая соответствует изменению значения *X* во времени и пространстве, ведь объект как раз и перемещается по этой координате. Область кривых снабжена двумя шкалами, где горизонтальная указывает кадры анимации, а вертикальная — значение переменной в данный момент времени. Кроме того, имеется уже привычный горизонтальный зеленый курсор, позволяющий прокручивать анимацию, не выходя из окна **Graph Editor**.

Присмотритесь к самой кривой. На ней можно заметить две вершины. Как раз они и являются установленными ключами анимации. А вот форма сплайна вовсе не прямая. Этим и объясняется неравномерная скорость движения объекта. Логично предположить, что выпрямление кривой канала *X* заставит объект двигаться с одинаковой скоростью в любой части анимации.

Работа со сплайнами **Graph Editor** ничем не отличается от манипулирования кривыми в окне **3D View**. Вы можете выделить нужную кривую с помощью правой кнопки мыши, нажать клавишу <Tab> для перехода в режим редактирования и, управляя рычагами, исправить форму сплайна. Точно так же доступны для манипулирования вершины кривой. Перемещение их по горизонтали изменит привязку ключей во времени, по вертикали — поведение объекта.

СОВЕТ

Если редактирование кривых в окне **Graph Editor** затруднено, то вы можете использовать колесико мыши для масштабирования, а вот нажатие и удерживание его же приводит к панорамированию содержимого.

Действительно, изменить форму кривой, перемещая рычаги, не составит особого труда, но зачем делать это вручную, если можно поручить программе?

Переведите курсор мыши в окно **Graph Editor** и нажмите клавишу <N>. Оказывается и у этого окна есть панель свойств **Properties**, правда с непривычным набором опций (рис. 5.6).

Выделите вершину кривой *X* для начального ключа анимации. Откройте вкладку **Active Keyframe** (Активный ключ) на панели свойств. Здесь имеется меню **Interpolation** (Интерполяция). Выберите в нем пункт **Linear** (Линейный). После этого действия кривая активного канала станет абсолютно прямой от точки до точки. Таким образом, скорость анимации будет неизменной. Можете проверить, нажав клавиши <Alt>+<A>.

Теперь давайте рассмотрим левую часть окна **Graph Editor**. Здесь указываются все имеющиеся анимационные каналы выделенного объекта (рис. 5.7).

Так как в анимации куба используется только перемещение, то и каналы соответствуют группе **Location**. Обратите внимание на кнопки, находящиеся рядом с названиями. Они предназначены для управления видимостью и работой конкретного канала:

- ◆ кнопка с изображением глаза отвечает за показ кривой или группы кривых в окне **Graph Editor**;

- ◆ кнопка с изображением динамика — это вовсе не управление звуком. Выключите ее, если хотите, чтобы данный канал не принимал участие в анимации;
- ◆ кнопка с изображением замка — защита канала от редактирования.

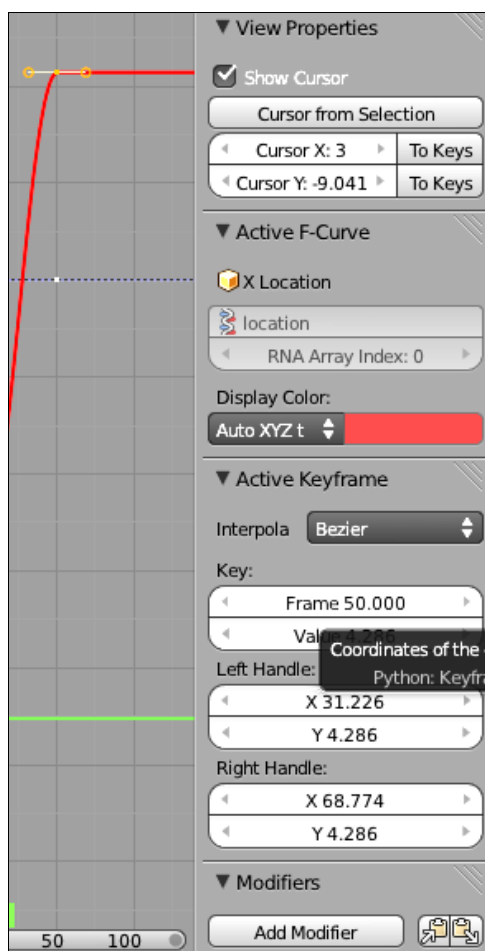


Рис. 5.6. Панель **Properties** окна **Graph Editor**



Рис. 5.7. Анимационные каналы объекта

Рассмотрим второй пример использования **Graph Editor**. Допустим, нужно сделать постоянное вращение объекта вокруг центра, не ограниченное во времени и с одинаковой скоростью. Такое может понадобиться для анимации модели планеты.

Можно создать первый ключ, затем второй со смещением ротации на 360 градусов, а потом размножить их в необходимом количестве. Однако такой вариант чреват ошибками, да и просто нуден в исполнении. Blender предлагает иной способ решения этой задачи, очень простой и эффективный.

Добавьте в сцену новый примитив **UV Sphere**, который будет играть роль планеты. Переключите активный кадр в позицию 1 и установите первый ключ для сферы (нажать клавишу <I>, затем выбрать из меню пункт **Rotation**).

Сделайте активным 50-й кадр на линейке **Timeline**. Немного разверните сферу по оси Z (<R>, <Z>) и установите второй ключ.

Итак, проигрывание этой анимации выглядит как оборот объекта вокруг оси и остановка движения после пересечения конечного ключевого кадра, что, собственно, следовало ожидать.

А вот теперь обещанное маленькое чудо. Выделите в окне **Graph Editor** кривую, ответственную за координату Z (она должна быть синего цвета). Это можно сделать, щелкнув мышью непосредственно по кривой или просто выделив канал **Z Euler Rotation** в левой части окна. Переведите курсор мыши в область с кривыми и нажмите клавишу <A> для отметки всех вершин активного сплайна.

Волшебная функция находится в меню **Channel** на заголовке окна **Graph Editor** (рис. 5.8). Откройте его и выполните функцию: **Channel | Extrapolation Mode | Linear Extrapolation** (Канал | Режим экстраполяции | Линейная экстраполяция). Кривая после этого превратится в прямую, уходящую в бесконечность. Попробуйте проиграть анимацию и убедитесь, что сфера вращается с одинаковой скоростью вне зависимости от реально установленных ключей.

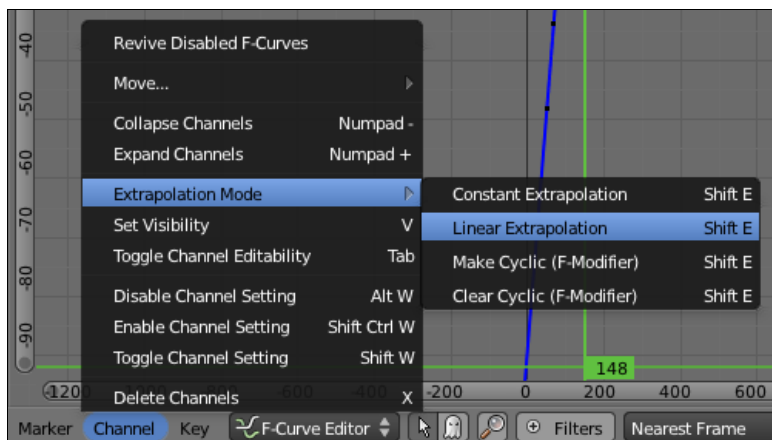


Рис. 5.8. Меню Extrapolation

Рассмотрим еще один пример анимации, но уже с цикличным повторением. Таким движением, например, обладает маятник часов.

Сначала создадим модель. Она будет очень простая. Понадобится всего два примитива: **Tube** — для создания стержня; **UV Sphere** — в качестве груза.

СОВЕТ

Переведите курсор мыши в окно **3D View** и нажмите клавиши <Ctrl>+<Up> для расширения окна на весь экран. После завершения создания модели используйте ту же комбинацию клавиш для возвращения стандартного размера окна.

Добавьте в сцену оба примитива и расположите их так, чтобы сфера оказалась на конце стержня (рис. 5.9).

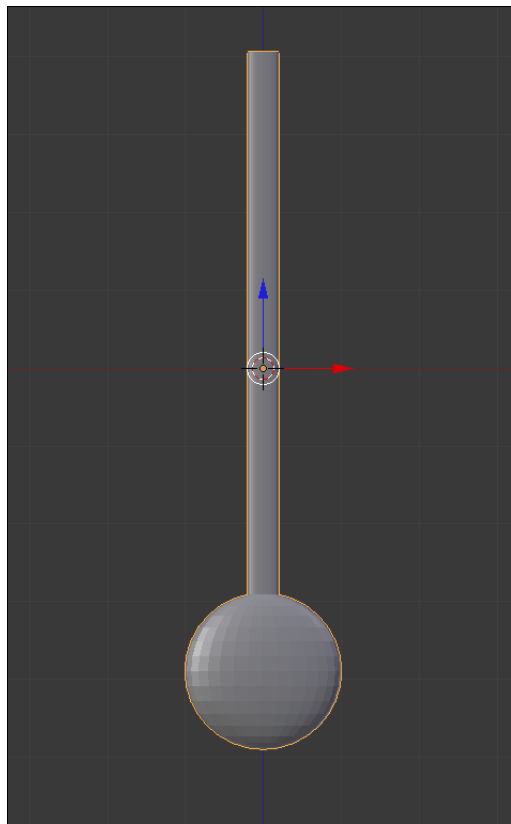


Рис. 5.9. Заготовка для маятника

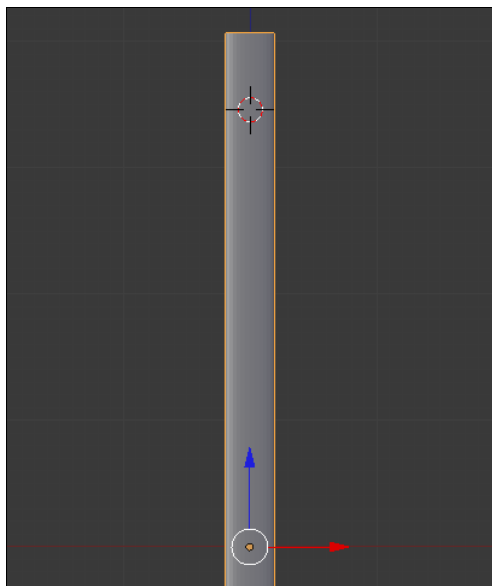


Рис. 5.10. Новый центр для объекта указывает 3D Cursor

Выделите оба примитива и объедините их в один командой **Join** (<Ctrl>+<J>). На рис. 5.9 видно, что центр объекта расположен посередине. Необходимо перенести его в верхнюю часть стержня. Приблизьте вид окна колесиком мыши и щелкните левой кнопкой мыши для переноса **3D Cursor** в новую позицию (рис. 5.10).

В окне **3D View** выберите пункт меню **Object | Transform | Origin to 3D Cursor** (Объект | Трансформация | Оригинал к 3D Cursor). Эта команда создаст новый центр объекта в месте, указанном курсором программы.

Для создания циклического движения необходимо подготовить образец анимации. Выполните следующие шаги:

1. Перейдите в окно **Front View** ортогографической проекции.
2. Нажмите клавишу <N> для вызова панели свойств в окне **3D View**.
3. Введите значение 10 для поля **Y** группы **Rotation**. Маятник должен будет отклониться немного влево.
4. Установите на панели **Timeline** первый кадр в качестве активного.
5. Нажмите клавишу <I> в окне **3D View** и выберите пункт **Rotation**. **Timeline** отобразит для первого кадра новый ключ.

6. Установите в окне **Timeline** значение 25 для активного кадра.
7. Измените в поле **Rotation Y** панели свойств значение на -10 . Маятник качнется в правую сторону.
8. Установите ключ **Rotation** для этого кадра.
9. Финальный ключ будет располагаться для кадра 50. Измените активный кадр.
10. Введите значение 10 для координаты **Y** группы **Rotation**. Маятник переместится обратно влево.
11. Установите ключ и для этой позиции.

Попробуйте проиграть анимацию, нажав комбинацию клавиш $\langle \text{Alt} \rangle + \langle \text{A} \rangle$. Если маятник совершит одиночное качание из стороны в сторону, то заготовку вы сделали правильно (рис. 5.11).

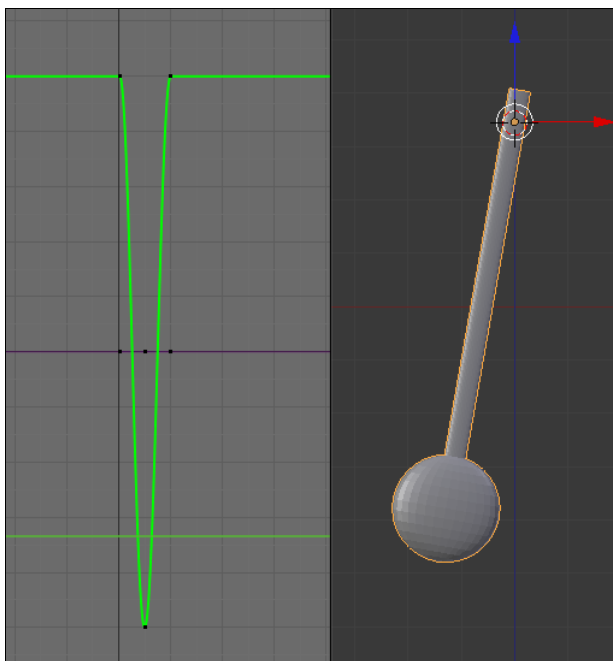


Рис. 5.11. Форма кривой канала **Y** для анимации-образца

Зациклить анимацию очень просто. Выберите в **Graph Editor** кривую **Y Euler Rotation**, затем функцию в меню **Channel | Extrapolation Mode | Make Cyclic** (Канал | Режим экстраполяции | Создать цикл). Это действие создаст копии образца и размножит его до бесконечности. Теперь маятник будет спокойно качаться из стороны в сторону неограниченное время (рис. 5.12).

Допустим, нужно сделать так, чтобы маятник начинал и заканчивал свое движение в определенных кадрах. Это можно сделать в настройках созданного цикла.

В действительности, в результате применения функции **Make Cyclic** к выделенной кривой был присоединен модификатор **Cycles** (Циклы). В отличие от уже привыч-

ных модификаторов объектов, создание и настройка анимационных модификаторов выполняется на панели свойств окна **Graph Editor**. Нажмите клавишу <N> для вызова этой панели.

Если кривая **Y Euler Rotation** в данный момент выделена, то на панели свойств на закладке **Modifiers** будут находиться настройки **Cycles** (рис. 5.13).

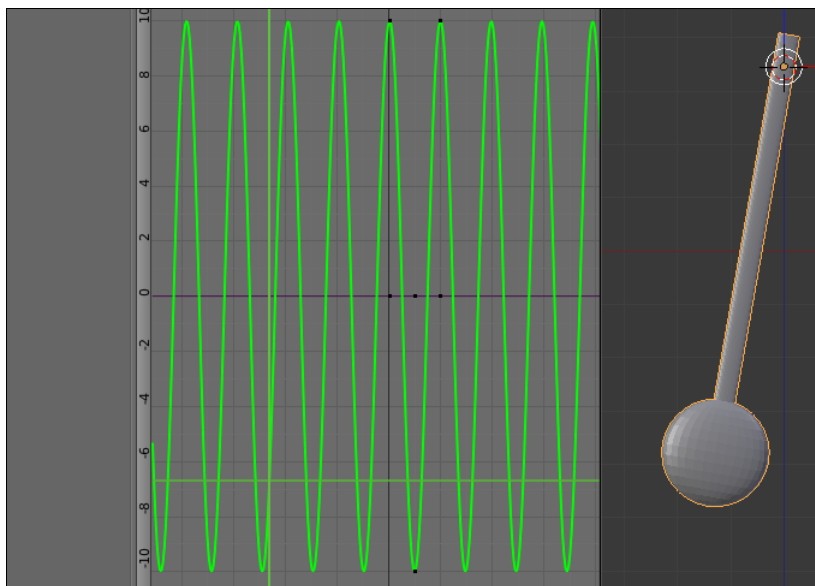


Рис. 5.12. Кривая Y после применения функции Make Cyclic

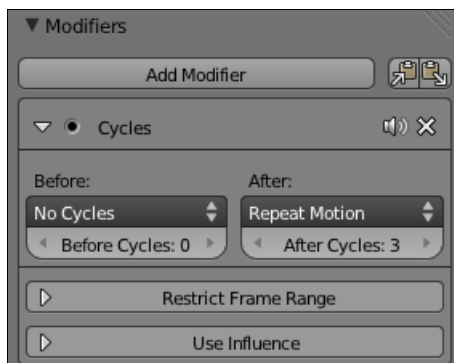


Рис. 5.13. Настройки модификатора Cycles

Группы **Before** (До) и **After** (После) отвечают за создание цикла первого и последнего ключей соответственно. Для каждого варианта имеется одинаковое меню выбора режима работы и поле установки количества циклов. Меню содержит четыре пункта:

- ◆ **No Cycles** (Без цикла);
- ◆ **Repeat Motion** (Повтор движения);

- ◆ **Repeat with Offset** (Повтор со смещением);
- ◆ **Repeat Mirrored** (Зеркальный повтор). Это значение по умолчанию.

Пусть движение маятника начинается с нулевого кадра, а завершается в 200.

Сначала в группе **Before** нужно выбрать из меню пункт **No Cycles**. В окне редактора тут же уберутся созданные повторы до нулевого кадра.

А вот для остановки маятника до 200 придется немного потрудиться. Проблема в том, что поле **After Cycles** предназначено для ввода количества колебаний, а не кадров. Проще поступить следующим образом: установить анимационный курсор в нужном месте и опытным путем подобрать значение для этой опции. Таковым является 3 (рис. 5.14). Кстати, установка этих полей в 0 означает отключение любых ограничений, что и было по умолчанию.

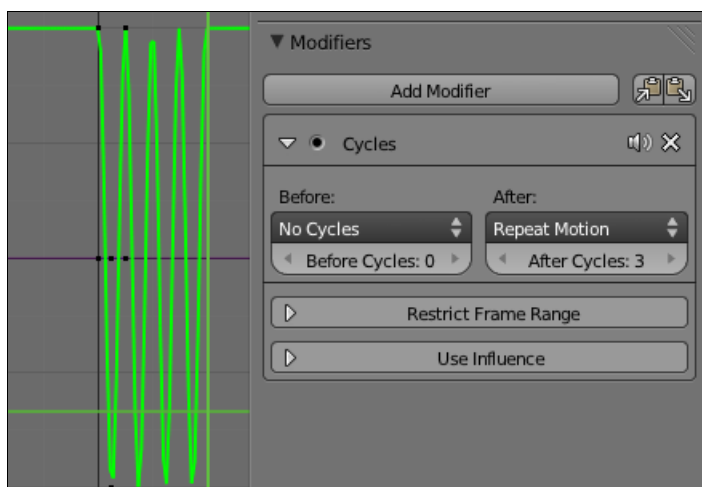


Рис. 5.14. Результат работы ограничителей цикла

5.4. Движение объекта по кривой

Использование кривых — чрезвычайно популярное средство для прокладки траектории движения объекта. Это объясняется удобством манипулирования сплайном в трехмерном пространстве и простотой настройки анимации.

Рассмотрим несложный пример анимации объекта с помощью кривой. В качестве подопытного создадим обычную стрелку, чтобы можно было отследить ориентацию объекта в соответствии с траекторией.

Создайте новый проект, нажмите клавишу <NumPad 7>, затем <NumPad 5> для перехода в просмотр **Тор** ортогональной проекции.

Стрелка будет создаваться на основе имеющегося куба в сцене. Чтобы не выполнять лишней работы, воспользуемся принципом симметричного моделирования.

Нажмите клавишу <Z> для включения режима **Wireframe**. Затем с помощью рамки () выделите крайние правые вершины и удалите их.

Откройте панель модификаторов в окне **Properties**. Добавьте из меню модификатор **Mirror**.

Нужно немного сжать объект по координате Z. Для этого переключите сцену в проекцию **Front View** (<NumPad 1>), нажмите клавишу <A>, чтобы выделить все вершины. Используя масштабирование (<S>), немного сожмите примитив по координате Z (<Z>). В целом у вас должна получиться заготовка, как на рис. 5.15.

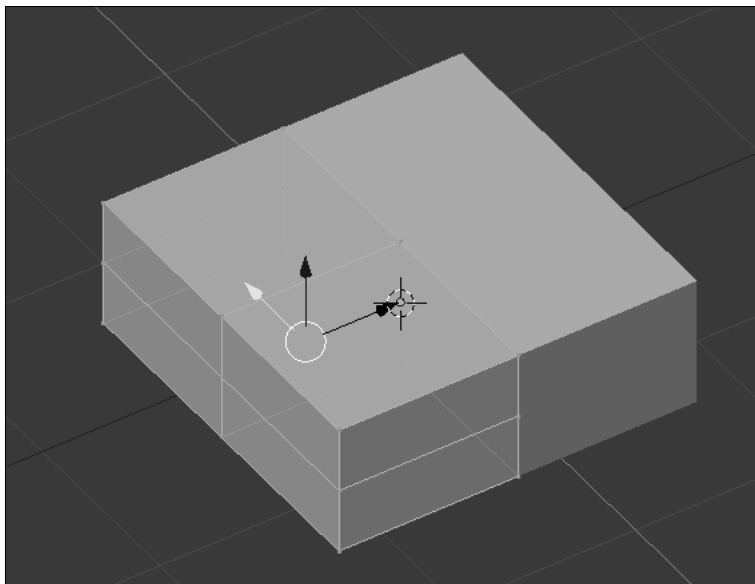


Рис. 5.15. Подготовка куба для создания стрелки

Изменить форму куба для получения стрелки очень просто. Перейдите снова в просмотр **Top**. Сначала необходимо объединить верхние вершины в одно целое. Для этого выделите их, нажмите клавишу <W>, а в появившемся меню выберите пункт **Merge (At Center)**. После этой операции у объекта получится два острия, ведь вершины были объединены по центру. Выделите боксовой рамкой новую вершину и передвиньте ее вправо до максимального сближения с зеркальной копией. Теперь нужно ее немного вытянуть вверх (рис. 5.16).

Осталось доделать хвостовую часть. Снимите выделение со всех вершин той же клавишей <A>. Выделите нижний угол и передвиньте его к центру объекта (рис. 5.17).

Теперь отметьте все вершины нижней части и переместите вверх для создания треугольника. Хвостовая часть будет сделана с помощью операции **Extrude**. Нажмите клавишу <E> и выдавите полигоны вниз. В итоге у вас получится примерно так, как на рис. 5.18. Можете применить модификатор — он больше не понадобится (кнопка **Apply** на панели настроек).

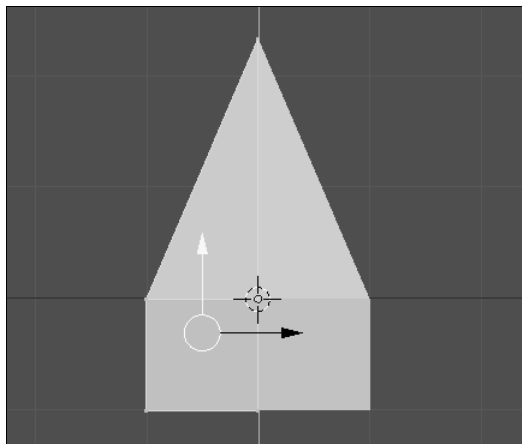


Рис. 5.16. Наконечник для стрелки

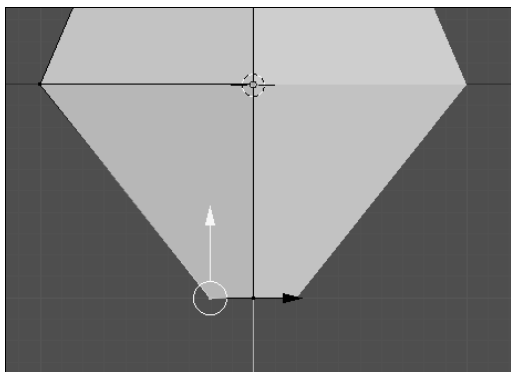


Рис. 5.17. Подготовка для основания

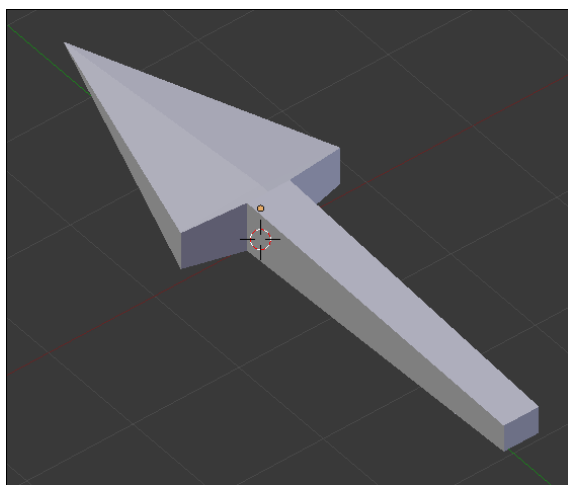


Рис. 5.18. Готовая стрелка

В качестве траектории может служить любой тип кривой, но лучше использовать специальный примитив **Path**. Добавьте его в сцену с помощью меню **Add | Curve | Path**. Примитив **Path** представляет собой обычную кривую **NURBS**, поэтому проблем с манипуляцией им возникнуть не должно. Если имеются вопросы, то обратитесь к *главе 3*.

Пусть стрелка будет огибать в сцене сферу. Создайте объект и расположите кривую для этого действия (рис. 5.19).

Настало время познакомиться с еще одним понятием в Blender — это **Constraint** (Ограничитель). Подобно модификаторам, **Constraint** может быть присоединен к объекту в любой момент. Только в отличие от последних, ограничители обычно применяются для анимации. С их помощью можно управлять поведением объекта при движении или вращении. **Constraint** имеет свою собственную закладку в окне **Properties** (рис. 5.20).

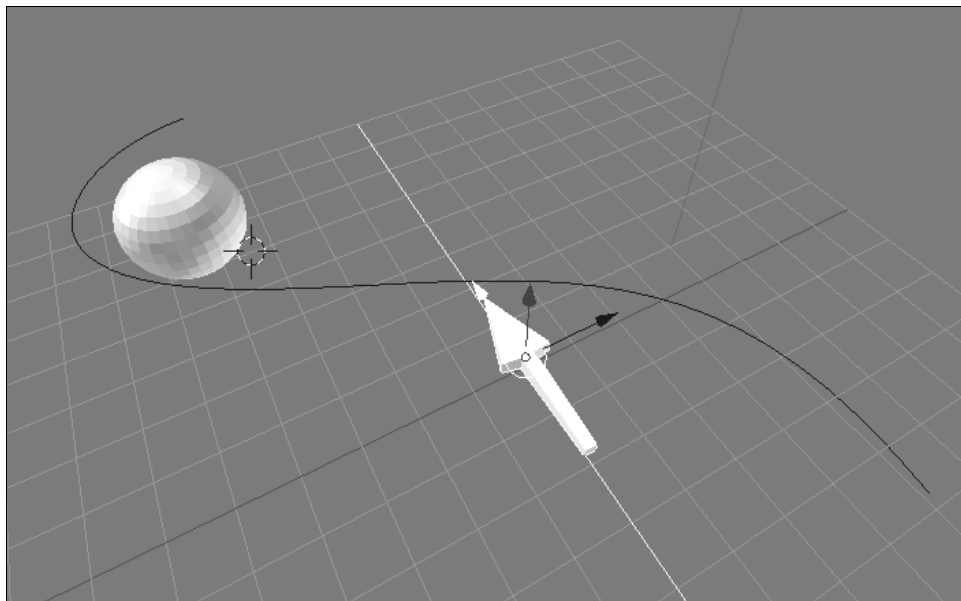


Рис. 5.19. Траектория для объекта

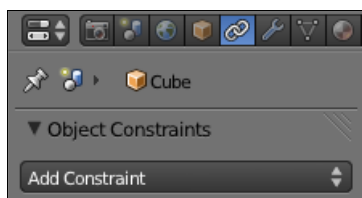


Рис. 5.20. Панель Constraint

В данном случае для присоединения траектории к стрелке понадобится ограничитель **Follow Path** (Следование по траектории). Откройте закладку **Object Constraints** (см. рис. 5.20) и выберите из меню на панели нужный ограничитель (рис. 5.21).

Выберите в поле **Target** созданный ранее объект **Path**. Теперь посмотрите в окно **3D View** — стрелка переместилась в начало кривой. Но анимация работать пока не будет, т. к. даже здесь понадобятся анимационные ключи.

Настройка ключей производится несколько непривычно. Они должны быть установлены не для объекта, который движется, а для кривой. Выделите примитив **Path**, затем откройте панель **Object Data** в окне **Properties** (рис. 5.22).

Среди них имеется закладка с названием **Path Animation** (по умолчанию она должна быть включена). Для создания анимации понадобится задействовать всего две опции:

- ◆ **Frames** (Кадры) — количество кадров для выполнения всей анимации;
- ◆ **Evaluation Time** (Вычисляемое время) — длина кривой, которую должен пройти объект в течение времени, указанного в поле **Frames**.

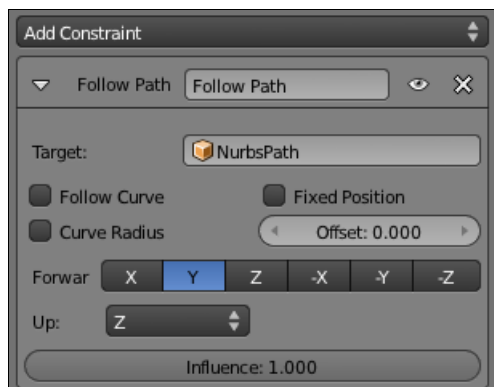


Рис. 5.21. Параметры Follow Path

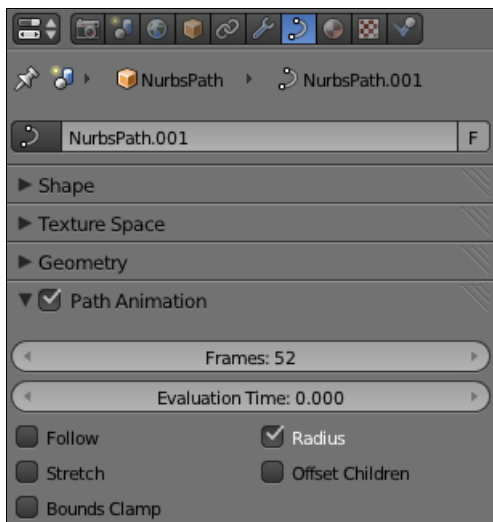


Рис. 5.22. Настройки Path

Сделано немного мудрено, поэтому рассмотрим настройку по шагам. Допустим, что стрелка должна пройти не весь путь, а лишь половину от длины кривой. При чем для всей анимации отводится 100 кадров:

1. Выделите кривую и откройте закладку **Path Animation** на панели **Object Data** (см. рис. 5.22).
2. Установите первый кадр активным в окне **Timeline**.
3. Наведите курсор мыши на поле **Evaluation Time** и нажмите правую кнопку мыши. Появится контекстное меню, где нужно выбрать пункт **Insert Keyframe** (Вставить ключ).
4. Установите в поле активного кадра значение 100.
5. Теперь нужно выбрать место на кривой, где остановится стрелка. Это выполняется с помощью поля **Evaluation Time**. При изменении его значения, объект будет двигаться по траектории. Выберите нужное место.
6. Щелкните правой кнопкой мыши по опции **Evaluation Time** и добавьте еще один ключ.
7. Установите значение 100 в поле **Frames**.

Теперь можно проверить анимацию. Вот только ориентация стрелки при движении остается без изменений (рис. 5.23).

СОВЕТ

При необходимости ключи кривой можно корректировать в окне **Graph Editor**. Другой способ — удалить или перезаписать неправильный ключевой кадр. Для этого установите его в качестве активного, щелкните правой кнопкой мыши по полю **Evaluation Time** и выберите пункт **Delete Keyframe** (Удалить ключ) или **Replace Keyframe** (Заменить ключ).

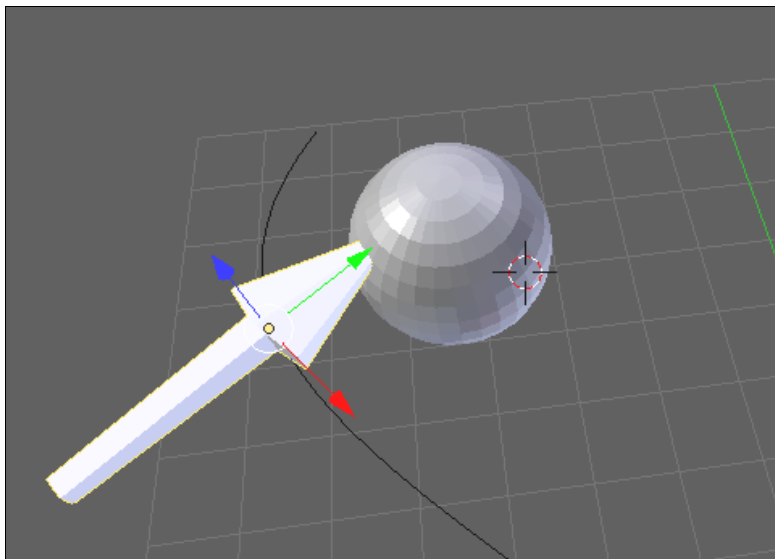


Рис. 5.23. Здесь острие стрелки "не следует" за траекторией

Откройте ограничитель **Follow Path**, прикрепленный к объекту, и активируйте опцию **Follow Curve** (Следовать за кривой). Эта функция заставляет объект изменять свою ориентацию в соответствии с изгибами траектории. Опции группы **Forward** (Вперед) позволяют развернуть объект так, как нужно. Теперь анимация стрелки соответствует задуманному (рис. 5.24).

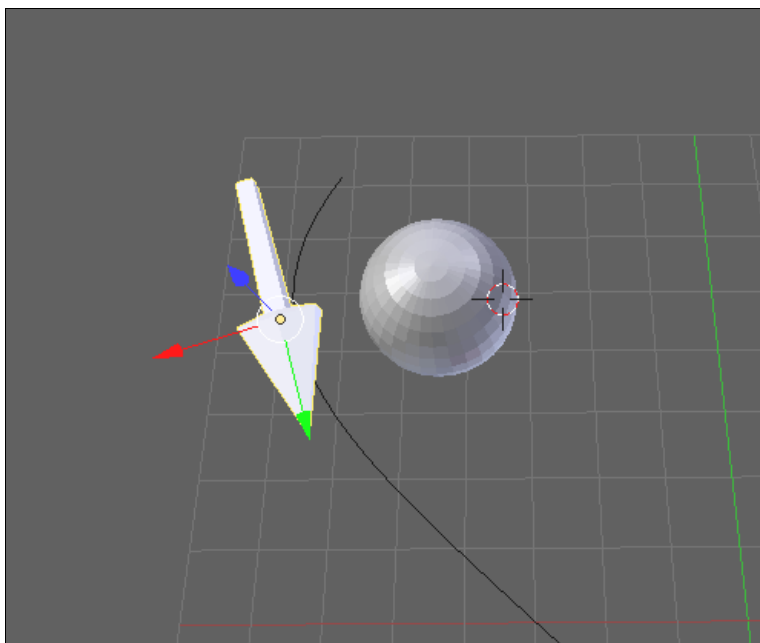


Рис. 5.24. Стрелка изменяет направление в соответствии с кривой

5.5. Анимация и деформация

В теплых морях обитает одна интересная рыбка, которая стала прообразом героев множества популярных мультфильмов и компьютерных игр. Речь идет о так называемых рыбах-ежах. В минуту опасности эта отнюдь небезобидная рыбка мгновенно раздувается и ошетиливается колючками, отчего напоминает свернутого в клубок ежа. Конечно, моделировать всю рыбу целиком мы не будем, но анимацию колючек сделаем. Тем более, этот пример как нельзя лучше подходит для данной темы. Разговор пойдет об особом виде анимации, когда происходит не движение объекта в пределах сцены, а трансформация на уровне его элементов.

Пусть прототипом рыбки будет служить примитив **Icosphere**. Добавьте его в сцену из меню **Add | Mesh**. Особенностью этого объекта является его структура, где элементы представляют собой треугольники одинакового размера. Однако для создания колючек стандартного разрешения сферы недостаточно. На панели **Tool Shelf**, в нижней части при создании примитива появляется вкладка с настройками **Icosphere**, где параметр **Subdivisions** отвечает за качество структуры. Поменяйте его текущее значение на 5 (рис. 5.25).

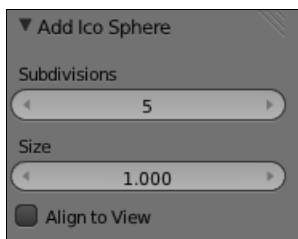


Рис. 5.25. Настройки **Icosphere** при создании

Работа с внутриобъектной анимацией, как и везде, базируется на использовании ключей. Только в данном случае называются они **Shape Keys** (ключи формы). Создание и настройка их разительно отличается от уже привычных ключевых кадров и осуществляется в одноименной группе **Shape Keys** панели **Object Data**.

Ключи формы устанавливаются в режиме редактирования объекта и привязываются к выделенным элементам. Важно знать, что каждый ключ **Shape** может иметь различное состояние: от исходной позиции элементов и до конечной. Кроме того, возможно использовать промежуточное значение. Собственно, **Shape Keys** хранят только изменения структуры, а вот сама анимация осуществляется с помощью привычных ключевых кадров. Но не будем забегать вперед.

Откройте панель **Object Data** редактируемого объекта. По умолчанию группа **Shape Keys** не имеет ключей (рис. 5.26).

Как видите, управление ключами формы не отличается особой сложностью. Кнопка с плюсом создает ключ, а с минусом — удаляет его из списка.

Запомните, управление ключами (создание, удаление, настройка) возможно только в режиме **Object Mode**, а вот присвоение действия осуществляется в режиме редактирования объекта.

Нажмите на кнопку "+". Программа создаст первый ключ с названием "Basis". Это базовый ключ, который несет информацию о начальной форме объекта.

Нажмите на эту же кнопку еще раз. В списке появится новый ключ "Key 1". Если название не нравится, вы всегда его можете изменить в поле, расположенном внизу списка (рис. 5.27).

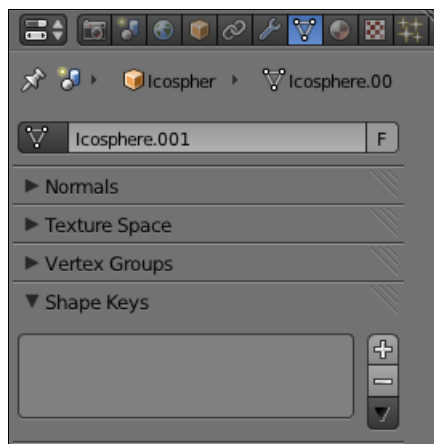


Рис. 5.26. Группа Shape Keys

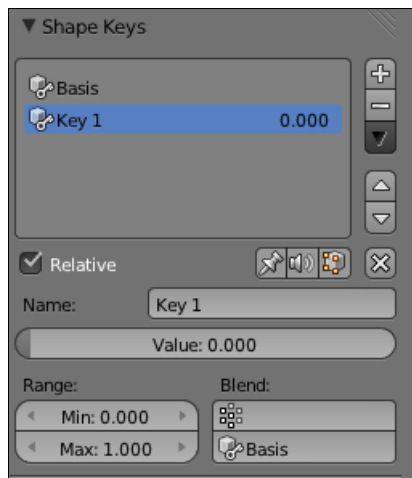


Рис. 5.27. Настройки рабочего ключа

Выделите первый ключ в списке. Пришло время заняться созданием игл у шара.

Логично предположить, что если выделить вершины объекта через одну и применить масштабирование, то получатся необходимые колючки. Вот только вручную делать это совсем не нужно, мы прибегнем к одной небольшой хитрости.

Войдите в режим редактирования (<Tab>) и выделите всю структуру (<A>), если она, конечно, еще не отмечена. Работа будет происходить на уровне полигонов, поэтому включите соответствующую кнопку на заголовке окна **3D View**.

В Blender имеется уникальный инструмент, позволяющий изменять текущее выделение в соответствии с желаемым количеством элементов, которые будут пропускаться. Откройте меню **Select** и выберите пункт **Every N Number of Verts** (Каждое число N вершин). На панели **Tool Shelf** появятся настройки этой функции, позволяющие вводить нужное количество пропущенных элементов. По умолчанию в поле **N Selection** значится цифра 2, пусть она и остается. Результат работы этой функции вы можете увидеть на рис. 5.28.

Следующая задача — это выдавливание игл. Стандартная функция **Extrude** перемещает элементы в одном направлении, нам же нужно индивидуальное выдавливание. Такая функция в Blender имеется и называется **Extrude Individual**. Вы можете выбрать ее из меню **Mesh** или нажать клавиши <Alt>+<E> и щелкнуть в появившемся меню по пункту **Individual Faces**. В любом случае программа создаст дополнительные полигоны. Используйте мышь для вытягивания новых элементов на небольшое расстояние от объекта, ведь в данном случае мы моделируем колючки "рыбы" в спокойном состоянии.

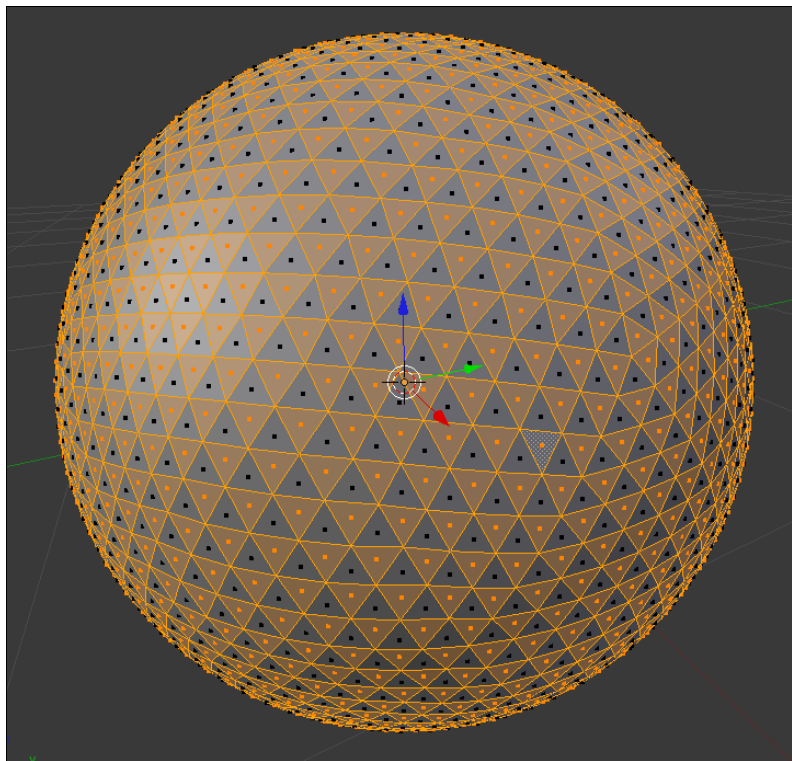


Рис. 5.28. Здесь полигоны выделены через один

Итак, была сделана модификация базовой формы модели. Теперь выберите второй ключ **Shape**. Нажмите клавишу <S> и вытяните иглы еще больше. Осталось только сжать крайние грани для придания остроты иглам. Это можно сделать, если использовать индивидуальное масштабирование (<Alt>+<S>), но есть вариант поизящнее. Нажмите клавишу <W>, выберите функцию **Merge**, а в появившемся меню — пункт **Collapse** (Стягивать).

Настройка ключей формы завершена и самое время заняться их анимацией. Нажмите клавишу <Tab> для выхода из режима редактирования.

Результат наверняка вас обескуражит. Выход из **Edit Mode** заставил объект мгновенно сжаться и принять форму, которая была сделана для ключа **Basis** (рис. 5.29).

Разгадка кроется в значении поля **Value** для второго ключа (см. рис. 5.27). Здесь регулируется степень деформации от базового **Basis** до **Key 1**, в данном случае. Так при значении 0 результат деформации будет отсутствовать. Попробуйте подвигать слайдер для просмотра перехода, но если вы нажмете <Alt>+<A>, то движения в сцене не будет.

Собственно, анимация выполняется в специальном режиме **ShapeKey Editor** в редакторе **DopeSheet**.

Откройте оконную раскладку **Animation**. Окно **DopeSheet** по умолчанию располагается в левой верхней части программы.

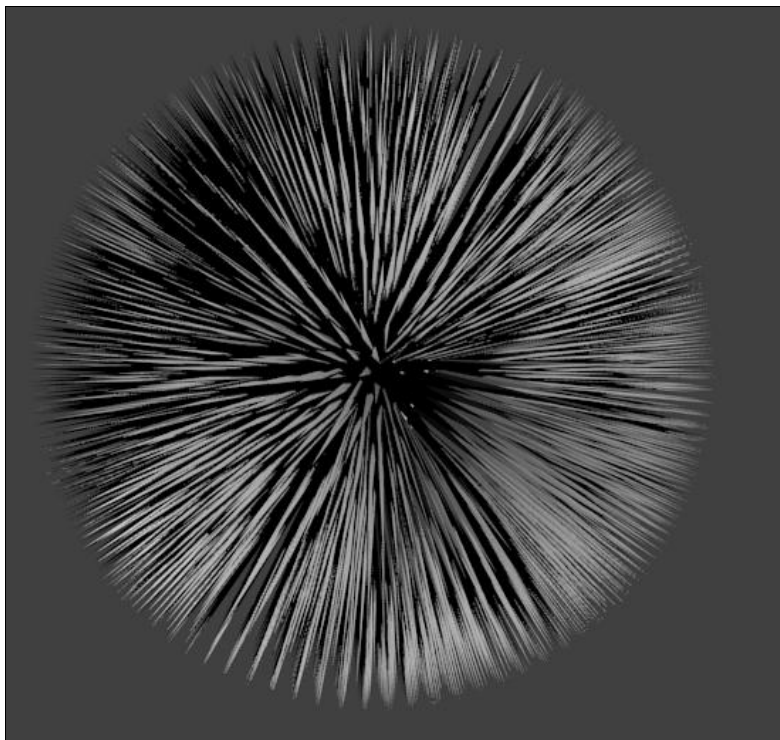


Рис. 5.29. Результат рендера сферы с иглами

Это уже более высокоуровневый редактор, нежели **Graph Editor**. Здесь можно управлять анимацией костей (скелета), всеми ключами сцены и **Shape**. В нашем случае нужно выбрать из меню **Mode** в этом окне пункт **ShapeKey Editor** (рис. 5.30).

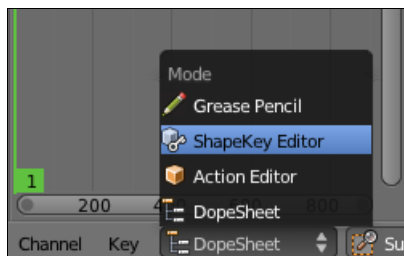


Рис. 5.30. Меню **Mode** окна **DopeSheet**



Рис. 5.31. Здесь указываются ключи объекта

После этого в левой части **DopeSheet** появятся все ключи **Shape** выделенного объекта. В нашем случае это будет **Key 1** (рис. 5.31).

Создавать анимацию с помощью этого редактора очень просто. Попробуйте подвигать слайдер в окне (см. рис. 5.31). При изменении значения **Value** происходит автоматическая установка ключа в текущем кадре сцены.

Соответственно, анимация ключей **Shape** осуществляется по алгоритму:

1. Установить начальный кадр в **Timeline**.
2. Выбрать желаемое значение ключа **Shape** в окне **DopeSheet**.
3. Установить конечный кадр.
4. Выбрать иное значение параметра **Value**.
5. В этом случае будут созданы ключи, которые появятся в самом **DopeSheet**, **Graph Editor** и на **Timeline** (рис. 5.32).



Рис. 5.32. Ключи в окне **DopeSheet**

Хотя в режиме **ShapeKey Editor** позволяет создавать ключи **Shape**, все же главное назначение редактора **DopeSheet** в управлении анимацией.

Допустим, имеются несколько ключей **Shape**, которые создают различную мимику лица. В окне редактора будут одновременно показываться все анимации **Shape** с ключами. Путем перемещения ключевых кадров в правой части редактора можно управлять временем старта и завершения каждой анимации **Shape**. Для этого нужно выделить правой кнопкой мыши ключ и переместить на новое место (<G>). Можно и удалить ненужные ключевые кадры **Shape** при помощи функции **Delete** (<X>).

5.6. Основы анимации персонажа

Создание анимации для персонажа закономерно считается трудным действием, ввиду сложности самого объекта. Возьмем, к примеру, модель человека. Движение рук, ног, пальцев, вращение головы — все это практически нереально выполнить с помощью рассмотренных ранее инструментов. Конечно, если модель состоит из отдельных, связанных между собой объектов, то можно ограничиться простым использованием анимационных ключей, но не стоит, есть вариант более удобный.

Скелетная анимация — это способ анимации модели с помощью вспомогательных объектов. Свое название она получила из-за аналогии со строением живого организма, где каркасом является скелет, состоящий из костей, связанных между собой суставами. С помощью специальных инструментов каркас объединяется с моделью и при движении его элементов осуществляется анимация частей объекта. Скелет остается невидимым для рендера программы.

Система костей в Blender носит название **Armature** (Арматура) и доступна для создания из меню **Add | Armature** (Добавить | Арматура) главного окна программы. По умолчанию это меню содержит всего один пункт **Single Bone** (Одиночная кость).

Bone (Кость) — это элемент арматуры. С помощью костей выстраивается скелет для модели. Вращение или перемещение костей позволяет выполнить анимацию нужной части объекта (рис. 5.33).

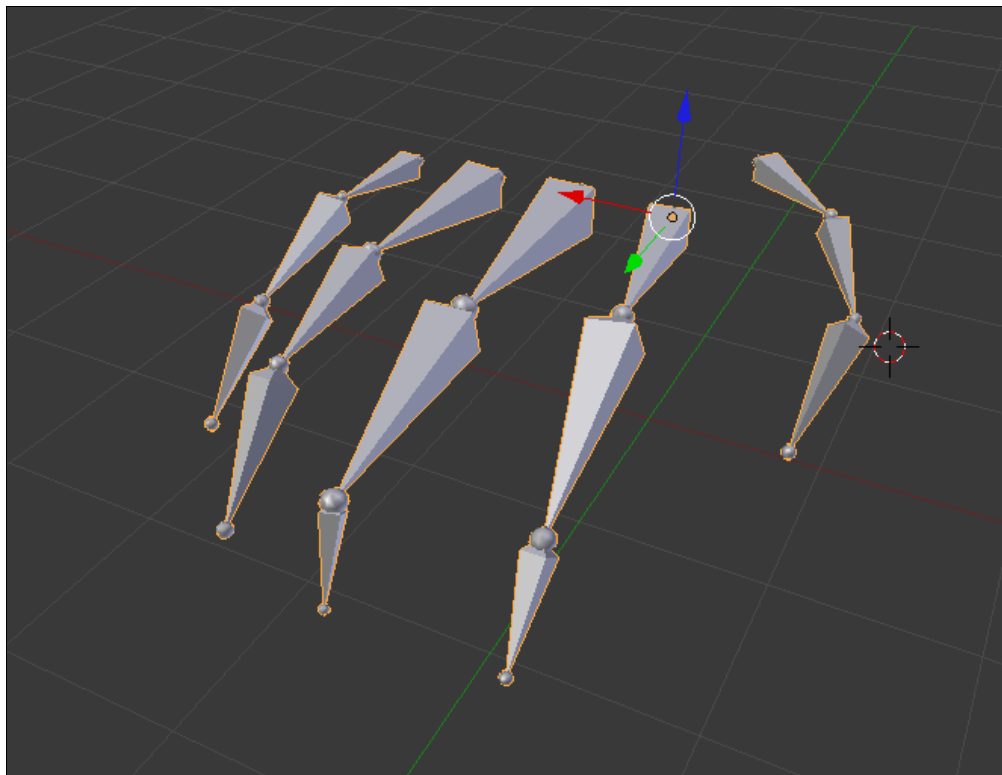


Рис. 5.33. Арматура кисти

В свою очередь **Bone** состоит из трех элементов: основания (**Root**), тела (**Body**) и наконечника (**Tip**). Так основание и наконечник служат для создания других костей, а также являются своеобразными шарнирами (рис. 5.34). К костям могут быть присоединены разнообразные ограничители (**Constraint**), с помощью которых устанавливаются определенные правила анимации. К примеру, голова человека не может вращаться на 360 градусов вокруг своей оси. Для ограничения поворота можно использовать специальный **Constraint**.

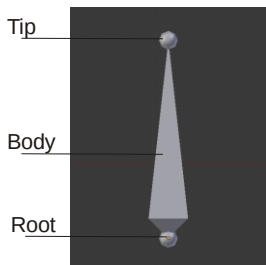


Рис. 5.34. Строение кости

После создания арматуры и настройки движения костей следует этап присоединения скелета к модели. Такое действие в Blender носит название **Skinning** (Скининг). Задача моделилера заключается в присвоении частей объекта определенным костям.

С помощью специальных инструментов имеется возможность установки области влияния каждой кости или шарнира. Например, при движении руки у живого человека изменяются формы грудных мышц, плечевого пояса и спины. Если бы этой возможности в Blender не было, получился бы эффект сломанной конечности. Для упрощения настройки **Skinning** программа предлагает автоматическое распределение областей влияния костей скелета. В некоторых случаях эта функция работает некорректно, поэтому имеется возможность ручной подгонки результата.

Третий, заключительный этап — создание анимации. Для этого программа предлагает специальный режим работы с арматурой, который называется **Pose Mode** (Режим позы). Работа заключается в создании анимационных ключей для каждого движения. Эти последовательности носят название **Action** (Действие) и могут комбинироваться в специальном редакторе.

Таким образом, для работы с арматурой программа имеет целых три режима:

- ◆ **Object Mode** (Режим объекта) — манипуляция всем скелетом в пределах сцены;
- ◆ **Edit Mode** (Режим редактирования) — редактирование костей и настройка связей;
- ◆ **Pose Mode** (Режим позы) — создание анимации.

Все эти режимы становятся доступными в меню **Mode** окна **3D View** при выделении объекта **Armature**.

Конечно же, использовать арматуру можно не только для движения персонажей. Рассматривайте скелетную анимацию лишь как еще один очень эффективный способ анимации объекта.

5.7. Создание и редактирование скелета

Создание скелета, правильная настройка костей — очень важный этап, от которого будет зависеть дальнейшая анимация модели. Тема сложная и многогранная, поэтому рассматривать работу с арматурой будем с помощью небольших практических уроков.

Как вы уже знаете, базовой частью скелета является **Bone**, или кость. Именно ее предлагает создать Blender в меню **Add | Armature**. Арматура является стандартным объектом программы, который подчиняется основным правилам манипулирования в сцене. Кость как отдельный элемент **Armature** доступна для редактирования в режиме **Edit Mode**.

Прежде чем начинать работу со скелетом, советую включить режим просмотра **Front View**, который является наиболее удобным. Итак, добавьте в сцену объект **Single Bone** и нажмите клавишу <NumPad 1>.

В режиме редактирования у кости можно выделять и манипулировать всеми элементами (см. рис. 5.34). Правда, в зависимости от выделения будут доступны разные режимы манипулирования:

- ◆ **Root** и **Tip** позволяют изменять размер и ориентацию кости с помощью манипулятора **Grab** (<G>);

♦ выделение **Body** отмечает целиком всю кость, поэтому становятся доступными все три вида манипуляции: перемещение, ротация, масштабирование.

Создавать дополнительные кости можно различными способами, но удобнее использовать стандартную функцию **Extrude**. Выделите верхнюю сферу кости (**Tip**) и нажмите клавишу <E>. Сдвиньте мышь в любом направлении (рис. 5.35).

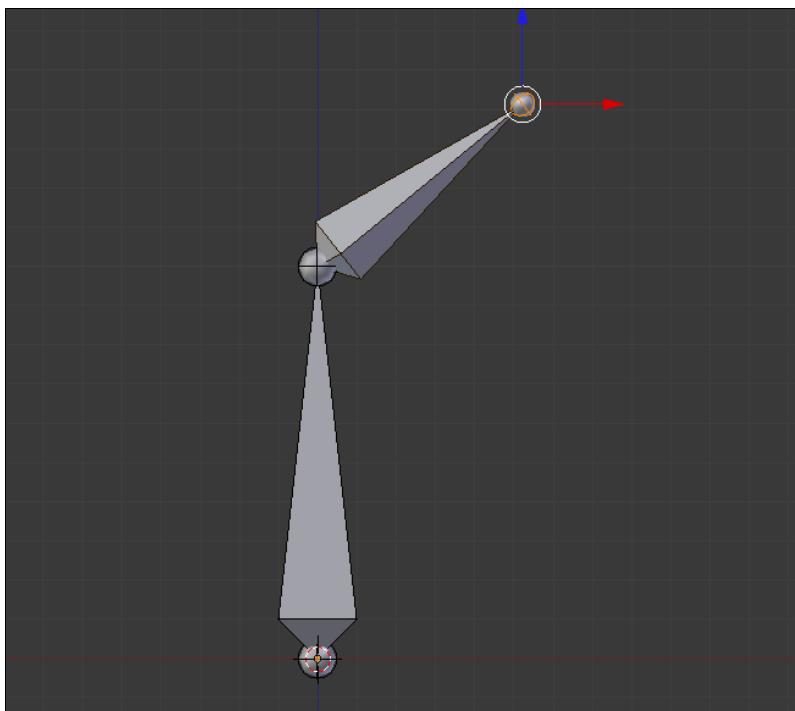


Рис. 5.35. Результат **Extrude** для **Tip**

Теперь проделайте ту же операцию, но со сферой **Root** первой кости. И в этот раз создастся новый элемент, но между этими, казалось, одинаковыми результатами имеется серьезное отличие. В первом случае была создана неразрывная цепочка, где начальный объект являлся родительским, а второй — подчиненным. Попробуйте сдвинуть любой из них (выделите для этого **Body**) и убедитесь в том, что они неразрывно связаны между собой. Однако попытка переместить третью кость, созданную из **Root**, приведет к тому, что она оторвется и отправится в "свободное плавание", как отдельный объект. Таким образом, ***Bone**, созданная из **Root** другой кости, является отдельным элементом, но принадлежит к одной и той же арматуре.* Это следует запомнить.

Для редактирования скелета имеется стандартный набор функций, что и для элементов **Mesh**: **Extrude**, **Duplicate**, **Subdivide**, **Delete**, **Merge**. Как и обычно, они доступны в виде горячих клавиш, на панели **Tool Shelf** или в меню **Armature**. Принцип использования их практически ничем не отличается от ранее рассмотренных объектов **Mesh**. Поэтому заострять внимание на них не будем. Вы всегда сможете

найти описание функций в соответствующем разделе книги. Конечно же, есть и специфический набор, характерный только для **Armature**.

Удалите из сцены все кости, кроме первой. Разверните **Bone** по оси X на 180 градусов, т. е. так, чтобы сфера **Root** оказалась вверху объекта. Выделите **Tip** и создайте новую кость с помощью функции **Extrude**. Еще раз выделите **Tip** первой кости и добавьте второй элемент.

На рис. 5.36, а новые кости располагаются свободно, имеют разный размер и местоположение. Вроде бы ничего необычного, но если учесть, что строение любых живых организмов симметричное, то легко представить объем работы, необходимый для создания одинаковых частей скелета.

Разработчики программы предлагают легкий способ создания симметричной арматуры, но для этого нужно учитывать важный момент — давать правильные имена звеньям. Это выполняется на специальной панели **Bone** окна **Properties** (рис. 5.37). В верхней части панели имеется окошко, где можно изменить текущее имя элемента.

По умолчанию Blender создает производные имена от слова **Bone** (Bone001, Bone002 и т. д.). Дайте следующие названия имеющимся элементам:

- ◆ первая, главная кость — **Root**;
- ◆ кость справа — **Shoulder_R**;
- ◆ кость слева — **Shoulder_L**.

С английского слово "Root" переводится, как "Корень", а "Shoulder" — "Плечо". Дело вовсе не в конкретном слове. Вы вправе называть кости как заблагорассудится, но если хотите использовать симметричное редактирование, то прибавляйте к имени через подчеркивание ключевые буквы:

- ◆ "R" или "Right" для правой стороны;
- ◆ "L" или "Left" для левой стороны.

В этом случае программа сможет определить, что данные звенья являются симметричными по отношению друг к другу.

Для включения собственно режима симметричного редактирования имеется специальная опция **X-Axis Mirror** на панели **Tool Shelf** (рис. 5.38).

Установите галочку в данной опции и попробуйте переместить **Tip** любого из элементов с именем **Shoulder**. Программа мгновенно выровняет параллельную **Bone** симметрично выделенной (рис. 5.36, б). Теперь любые операции с такими костями (добавление, манипулирование, удаление) будут дублироваться противоположной стороной. Интересно, что программа при создании нового элемента уже будет прибавлять указанный суффикс к его имени.

Устали держать руку на мышке? Поднимите и взмахните ею в воздухе. Как видите, движение начинается от плеча, которое поступательно передается всей руке. В мире Blender такое поведение костей называется *прямой кинематикой* (Forward Kinematics, FK). По умолчанию арматура работает именно так. А теперь

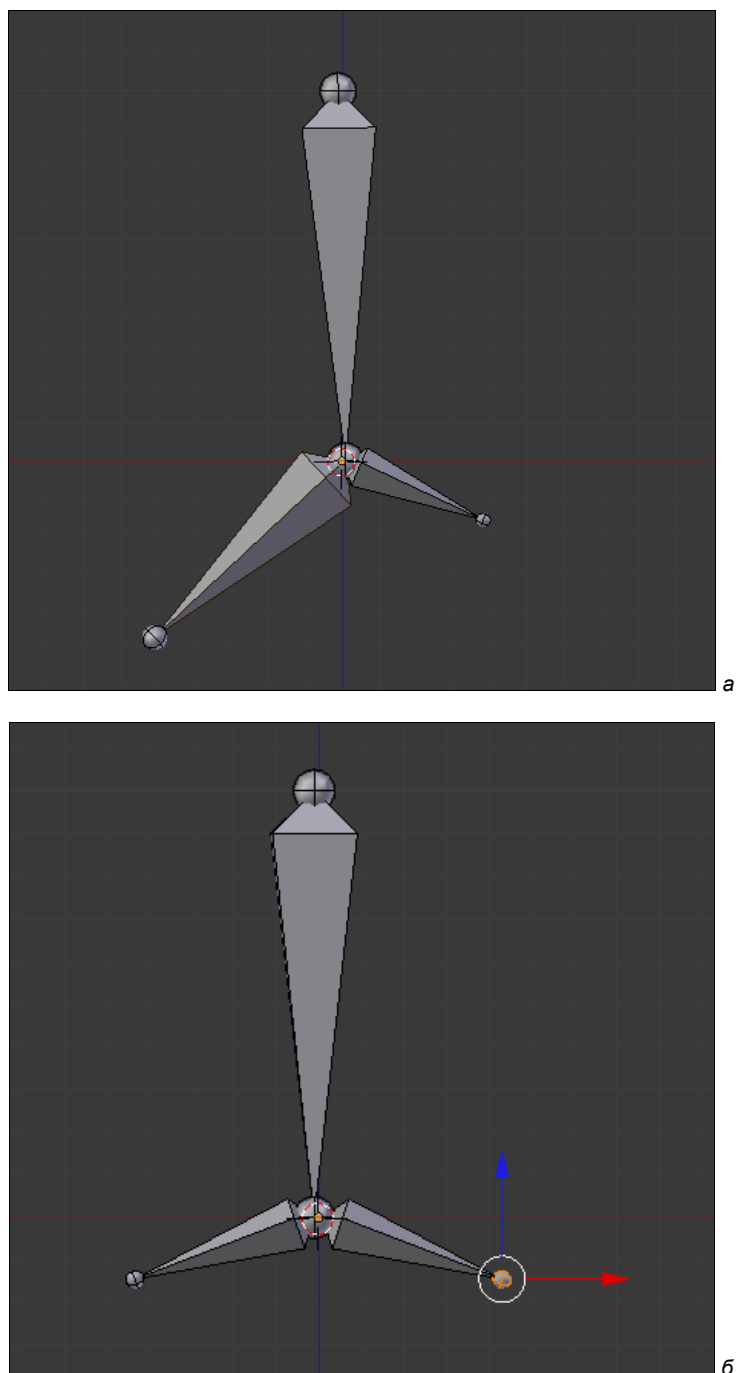


Рис. 5.36. Кости созданы в "свободном" режиме (а),
результат включения функции X-Axis Mirror (б)

представьте, что вы с кем-то здороваетесь за руку. В этом случае, движение начинается с кисти и передается по цепочке назад. Это уже *инверсная кинематика* (Inverse Kinematics, IK).

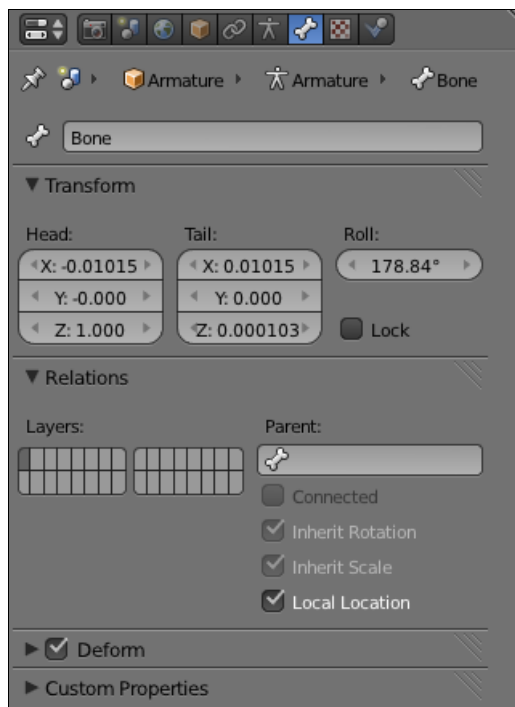


Рис. 5.37. Настройки отдельной кости

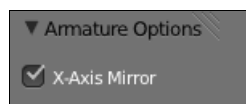


Рис. 5.38. Опция X-Axis Mirror для арматуры

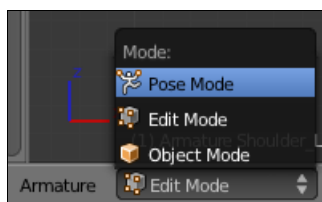


Рис. 5.39. Режим Pose Mode меню Mode

Для настройки кинематики и создания анимации используется специальный режим **Pose Mode** (Режим позы) (рис. 5.39).

Переключитесь в этот режим и передвиньте любую кость, а теперь нажмите клавишу <Tab> для возврата в **Edit Mode**. Вы увидите, что расположение костей разное, в зависимости от выбранного режима. *Манипуляции в Pose Mode не влияют на основное строение арматуры*, в то же время изменение костей в режиме **Edit Mode** отразится для **Pose Mode**.

Для понимания работы FK и IK выполним настройку руки, от кисти до плеча. Добавьте еще три кости к любой из **Shoulder** и назовите их **Forearm** (предплечье), **Arm** (рука), **Hand** (кисть) соответственно.

Переключитесь в режим **Pose Mode** и попробуйте сдвинуть кость **Forearm**. Вслед за этой **Bone** будет двигаться остальная часть "руки" (рис. 5.40).

СОВЕТ

По умолчанию в сцене Blender присутствует вспомогательная решетка, которая помогает подгонять узлы объекта при моделировании. Если она вам мешает, то ее можно временно выключить с помощью опции **Grid Floor** на закладке **Display** панели свойств (<N>).

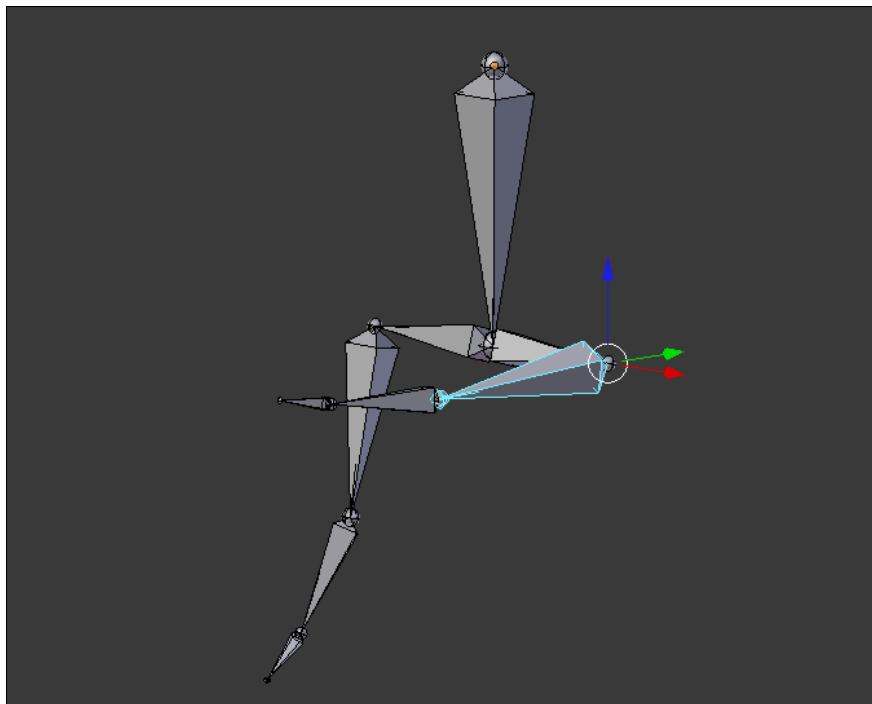


Рис. 5.40. Здесь движение руки от плеча (FK)

А теперь подготовим "руку" к крепкому, дружескому пожатию. Создание IK-связи можно выполнить разными способами, но наиболее простой — это воспользоваться специальным **Constraint**. Выделите кисть (**Hand**) в режиме **Pose Mode** и откройте панель **Constraint** в окне **Properties** (рис. 5.41).

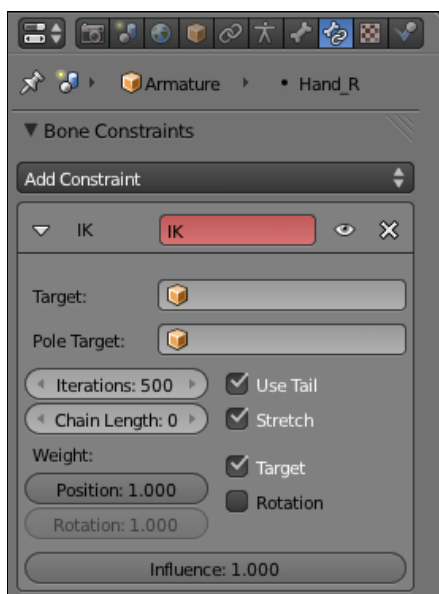


Рис. 5.41. Настройки IK

Нужный ограничитель называется **IK** (Inverse Kinematics). После его добавления к **Bone** вы увидите пунктирную линию, протянувшуюся от кисти до **Root**. Так Blender отмечает элементы, которые будут участвовать в цепочке IK. Если теперь попробовать передвинуть кисть, то весь скелет придет в движение. Нам этого не нужно. Правильным решением этой проблемы является ограничение цепочки IK костью плеча (**Shoulder**).

Это можно сделать в настройках ограничителя **IK** (см. рис. 5.41). Здесь всего лишь нужно указать в параметре **Chain Length** количество костей, участвующих в цепочке IK. По умолчанию там находится значение 0. В этом случае, ограничитель использует максимально возможное количество звеньев. Введите в поле цифру 4 и пунктир перебросится к корню **Shoulder** (рис. 5.42).

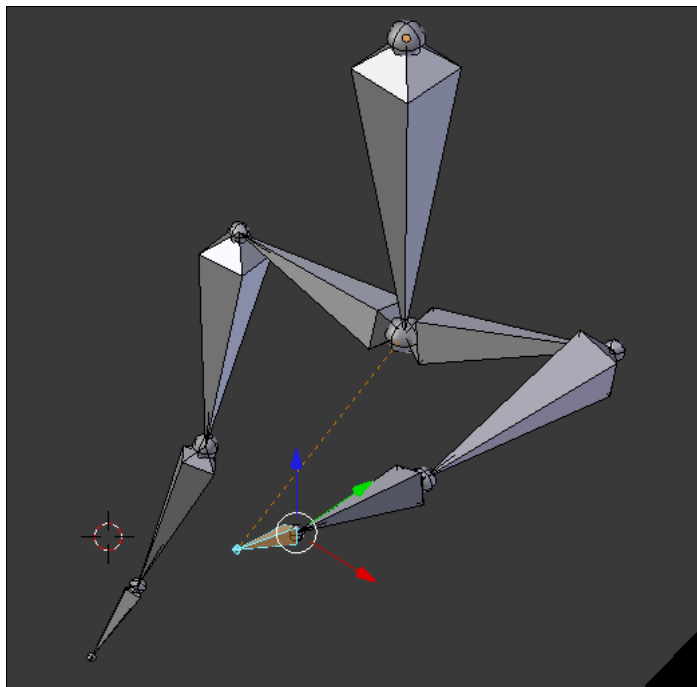


Рис. 5.42. Здесь движение руки от кисти к плечу

Теперь возможно движение костей руки, как от плеча, так и от кисти. При этом остальные кости скелета в процессе не участвуют. Запомните, в отличие от **Edit Mode** с включенной опцией **Axis Mirror**, настройка костей в **Pose Mode** осуществляется индивидуально. Не рассчитывайте, что проделанные манипуляции с IK отразятся на другой части скелета.

Внимательный читатель, испытавший на практике этот урок, заметит неправильное поведение костей руки. Так, плечо (**Shoulder**) слишком выдвигается при движении кисти вперед или назад, то же самое происходит при поднятии или опускании конечности. Попробуйте сами выполнить похожие движения рукой, чтобы понять, о чем идет речь. У реального человека эта часть тела относительно неподвижна.

Blender позволяет настроить индивидуально каждую кость, участвующую в цепочке ИК. Эти параметры находятся в группе **Inverse Kinematics** в окне **Properties** (рис. 5.43).

Установка ограничений (не путать с **Constraint**) возможна по всем осям элемента. Тут все просто, выбираете нужную ось, включаете опцию **Limit** и в соответствующих полях устанавливаете минимальный и максимальный угол отклонения.

Выделите плечо (**Shoulder**). Движение плеча может осуществляться только по двум координатам: X и Z. Соответственно, необходимо отключить перемещение его по Y. Сделать это можно, включив кнопку с замком с соответствующей буквой (см. рис. 5.43).

Установите галочки в полях **Limit** (Лимит) для осей X и Z. Программа активирует опции настройки значений.

Обратите внимание, что в окне **3D View** появились окружности красного и синего цвета вокруг выделенной кости (рис. 5.44). Так Blender отображает возможный угол движения элемента. По умолчанию минимальные и максимальные значения для обеих осей установлены в 180 градусов — это явно излишне. Пусть плечо будет двигаться лишь в пределах 20 градусов в каждом направлении. Поменяйте значение лимита для всех полей. Не забудьте сохранять знаки плюса или минуса!

Вот теперь движение конечности выглядит гораздо естественнее.

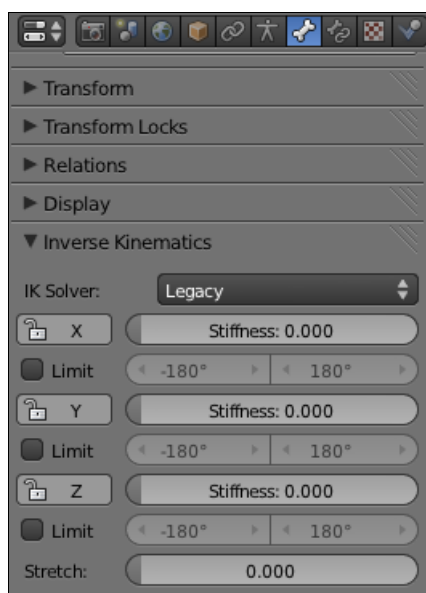


Рис. 5.43. Настройки ИК для выделенной кости

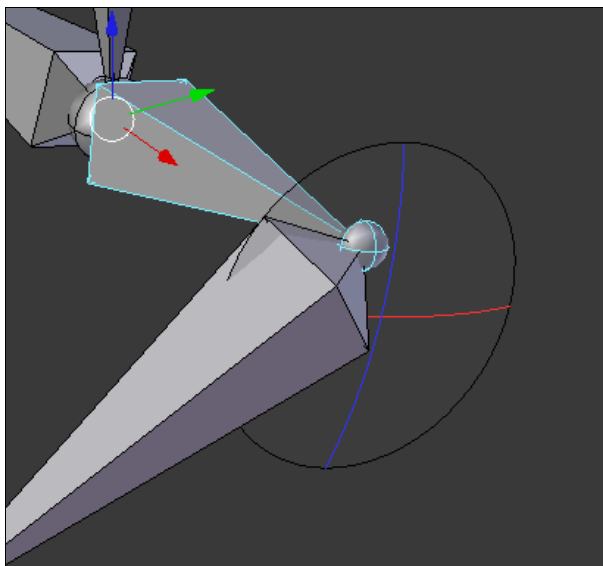


Рис. 5.44. Визуальное отображение лимита движения

5.8. Наращиваем "мясо"

Создание и настройка скелета — это всего лишь полдела. Теперь нужно правильно присоединить его к модели, для которой он создавался.

Blender предлагает несколько вариантов создания *привязки* (Skinning). Удобнее всего использовать автоматический, когда программа на основе сопоставления элементов скелета и модели создает области воздействия. Правда, все равно некоторые места придется дорабатывать вручную.

С помощью *разд.* 5.7 вы уже сделали арматуру, где имеется настроенная инверсная кинематика одной из руки. Если нет, то вы можете воспользоваться готовым файлом `Scenes\glava5\glava5_simple_armature.blend` из архива примеров к этой книге (см. приложение 2).

Однако сначала нужно создать модель для арматуры. Ограничимся созданием человеческого торса с двумя конечностями. Не будем гоняться за реалистичностью, важно научиться работать со **Skinning**.

Основой модели послужит обычный примитив **Cube**. Добавьте куб в любом удобном месте сцены и переключитесь в ортогональную проекцию **Front View**.

Для быстроты воспользуемся принципом симметричного моделирования. Нажмите `<Tab>` для редактирования куба и, используя функцию **Subdivide** (`<W>`), дважды разбейте структуру объекта. Переключитесь в режим **Wireframe** (`<Z>`) и удалите все вершины с левой стороны до центра. Добавьте к примитиву модификатор **Mirror** (рис. 5.45).

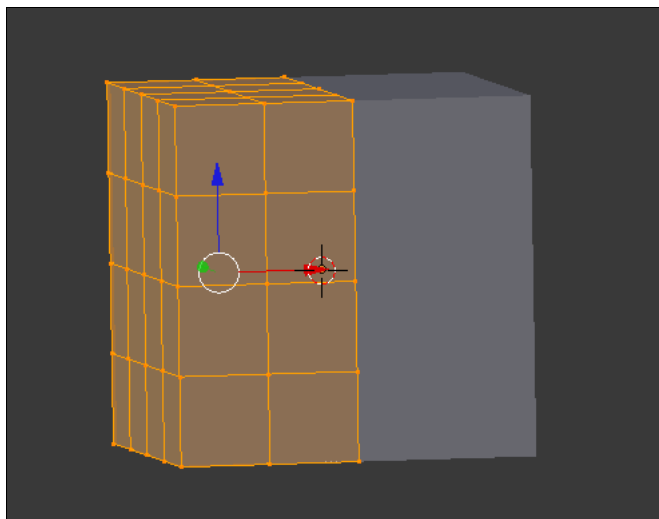


Рис. 5.45. Заготовка для торса

Сожмите немного куб по оси **Y** с помощью **Scale** (`<S>`). Это можно сделать и в режиме редактирования. Теперь ему нужно придать более обтекаемую форму. Выделите вершины, как показано на рис. 5.46. Используя инструменты масштабирования и перемещения, добейтесь округлой формы.

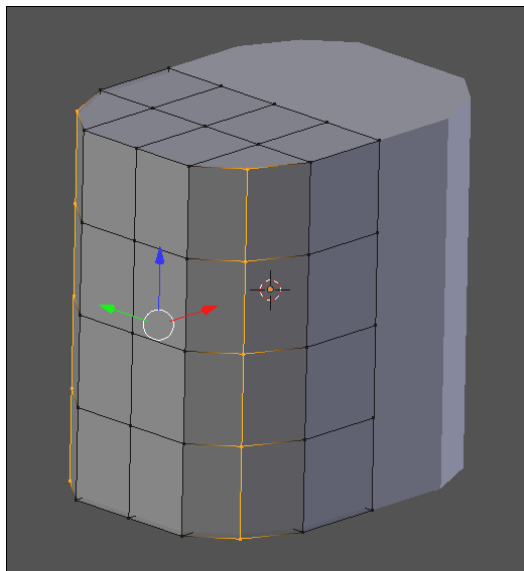


Рис. 5.46. Переместите эти вершины для сглаживания краев



Рис. 5.47. Сужение торса к основанию

Теперь выделите центральные вершины между указанными на рис. 5.46 и немного сдвиньте по оси X.

Получившийся цилиндр мало похож на человеческий торс. Включите общее сглаживание для объекта (это можно сделать в **Tool Shelf** путем нажатия кнопки **Smooth**, но только в режиме **Object Mode**) и добавьте еще один модификатор **Subdivision Surface** (Разбиение поверхности). Этот инструмент работает наподобие **Multires** и создает сглаженные поверхности путем увеличения промежуточных элементов. Установите в настройках модификатора цифру 2 в опциях **View** и **Render**.

Полученный объект больше смахивает на кусок мыла, поэтому немного доработаем его. Выделите все нижние вершины и сместите вниз по оси Z для придания объекту вытянутой формы. Используя манипулятор **Grab**, сузьте низ модели (рис. 5.47).

Теперь осталось дело за малым — добавить руки. Основой конечности послужат два полигона в верхней части торса. Выполните следующие шаги:

1. Нажмите клавишу <E> и немного вытяните новые грани. Прodelайте эту операцию трижды. Таким образом вы создадите место для изгиба плеча. Впоследствии это пригодится.
2. Выполните **Extrude** и переместите полигоны на расстояние, достаточное для создания предплечья. Уменьшите размер граней с помощью масштабирования.
3. Создайте еще три ступени граней для локтевого сгиба.
4. Нажмите <E> и выдавите следующую часть конечности. Уменьшите масштаб.
5. Прodelайте эти же операции для создания кисти.

Таким образом, у вас должна получиться грубая заготовка руки, как на рис. 5.48.

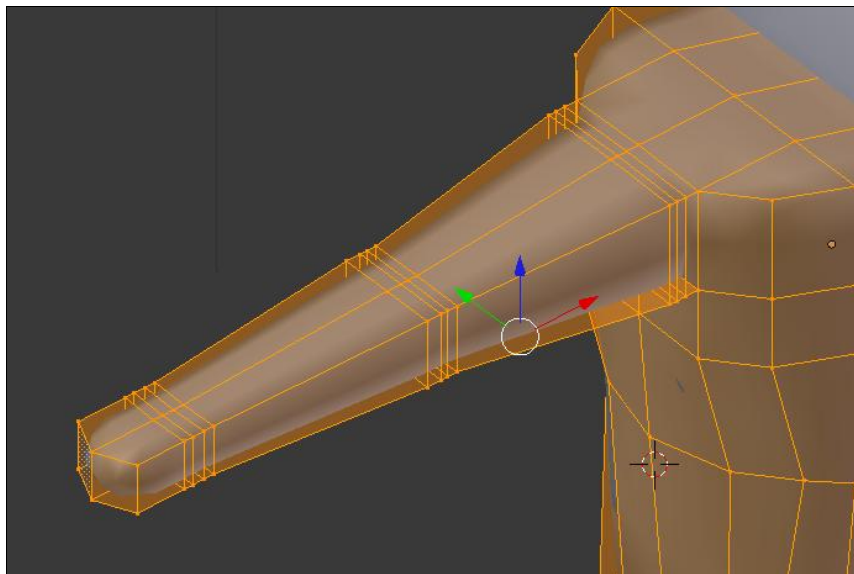


Рис. 5.48. Макет руки

По желанию вы можете довести модель торса до приемлемого вида. Используйте перемещение вершин, инструмент пропорционального редактирования и даже скульптурные кисти. Помните, если имеющейся структуры недостаточно, то вы всегда можете добавить новые ребра с помощью инструмента **Loop Cut and Slide** (Создать петлю и переместить) (<Ctrl>+<R>).

А вот теперь настало время совместить арматуру с моделью. В режиме **Object Mode** переместите скелет в центр модели. Скорректируйте масштаб, если это нужно.

Основная задача состоит в том, чтобы каждая кость и каждый сустав находился в положенном месте. Так кость **Shoulder** должна заканчиваться на суставе плеча, **Forearm** — в месте локтя, а **Hand** — покрывать кисть. Подгонка костей осуществляется в режиме редактирования арматуры.

Наверняка возникнет сложность изменения костей из-за того, что они перекрываются структурой модели. Конечно, можно переключиться в вид **Wireframe**, но есть способ поизящнее.

Blender имеет глобальные настройки **Armature**, которые позволяют управлять внешним видом скелета и не только. Все эти опции доступны в окне **Properties** группы **Display** (рис. 5.49).

Кнопки **Octahedral** (Восьмигранник), **Stick** (Палка), **B-Bone** (В-кость), **Envelope** (Оболочка), **Wire** (Проволока) отвечают за внешний вид костей. Изменение стандартного **Octahedral** на что-то другое может быть полезно при специфичных задачах, например **Stick** удобно использовать для арматуры пальцев.

Помимо кнопок в группе имеются переключатели:

- ◆ **Names** (Имена) — при включении этой опции Blender будет отображать имя каждой кости в сцене;

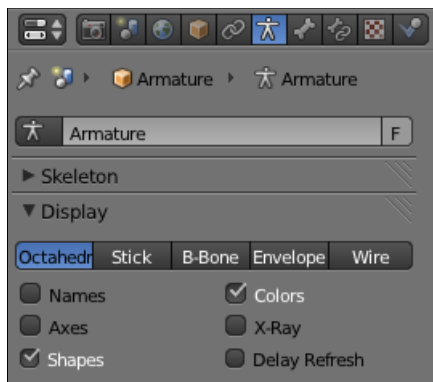


Рис. 5.49. Настройки арматуры

- ◆ **Axes** (Координаты) — показ локальных координат костей. Работает для выделенных элементов и только в режимах **Edit Mode** и **Pose Mode**;
- ◆ **Shapes** (Формы) — для удобства можно поменять внешний вид кости на любой иной объект. Опция **Shapes** позволяет это сделать;
- ◆ **Colors** (Цвета) — цветное отображение групп костей;
- ◆ **X-Ray** (Рентген) — установка данной опции заставит арматуру просвечивать сквозь модель;
- ◆ **Delay Refresh** (Задержка обновления) — опция используется только в **Pose Mode**. Отключает деформацию.

Включите **X-Ray** и установите кости, как на рис. 5.50.



Рис. 5.50. Совмещение арматуры с моделью

ЭТО ВАЖНО!

Перед выполнением подгонки арматуры к модели желательно выполнить сброс параметров трансформации костей, особенно ротации. Это можно сделать в режиме **Pose Mode**, выбрав соответствующий тип в меню **Pose | Clear Transform**. Во избежание проблем с анимацией выделите кости и нажмите <Alt>+<R>.

Для привязки арматуры к модели переведите оба объекта в режим **Object Mode**. Сначала выделите модель, затем, удерживая нажатой клавишу <Shift>, отметьте скелет. Нажмите клавиши <Ctrl>+<P> для вызова контекстного меню **Set Parent** и выберите в нем пункт **With Automatic Weights** (Автоматический вес).

Опция заставит Blender создать группы вершин для модели и привязать их к соответствующим костям.

Группы вершин — это запоминание части элементов объекта для последующего использования сторонними функциями, например модификаторами. Увидеть созданные группы можно на панели **Object Data** (рис. 5.51).

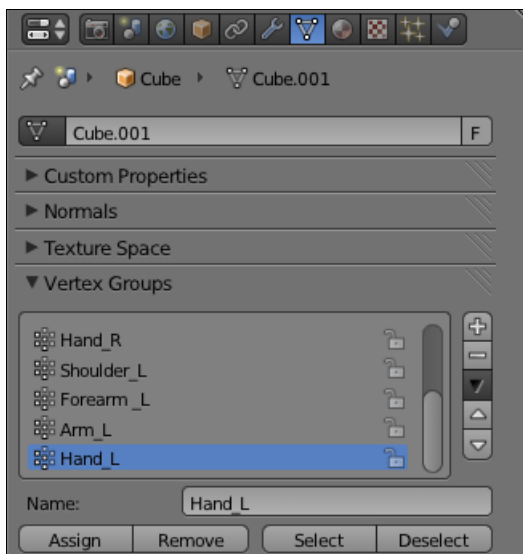


Рис. 5.51. Группы вершин

В списке **Vertex Groups** находятся группы, привязанные к определенным костям и имеющие те же названия. Работа с ними не отличается от использования мультиматериалов, которые рассматривались в *главе 4*. После выделения нужной группы вы можете просмотреть элементы (**Select**), добавить новые вершины (**Assign**) или удалить их (**Remove**).

Вернемся к нашей модели. Попробуйте в режиме **Pose Mode** подвигать костями арматуры. Вы увидите, что части модели будут послушно следовать за ними (рис. 5.52).

Хотя Blender старается в автоматическом режиме правильно создать и привязать группы вершин к костям, но не всегда это гладко получается. Бывает так, что при движении какой-либо кости перемещается ненужная часть модели или, как в дан-

ном случае, происходит наложение поверхности **Mesh**-объекта (присмотритесь внимательнее к области живота на рис. 5.52).

Вот эти искажения необходимо исправлять вручную. Есть два пути: отредактировать область вершин конкретной кости в специальном режиме или создать новую группу с привязкой к арматуре.

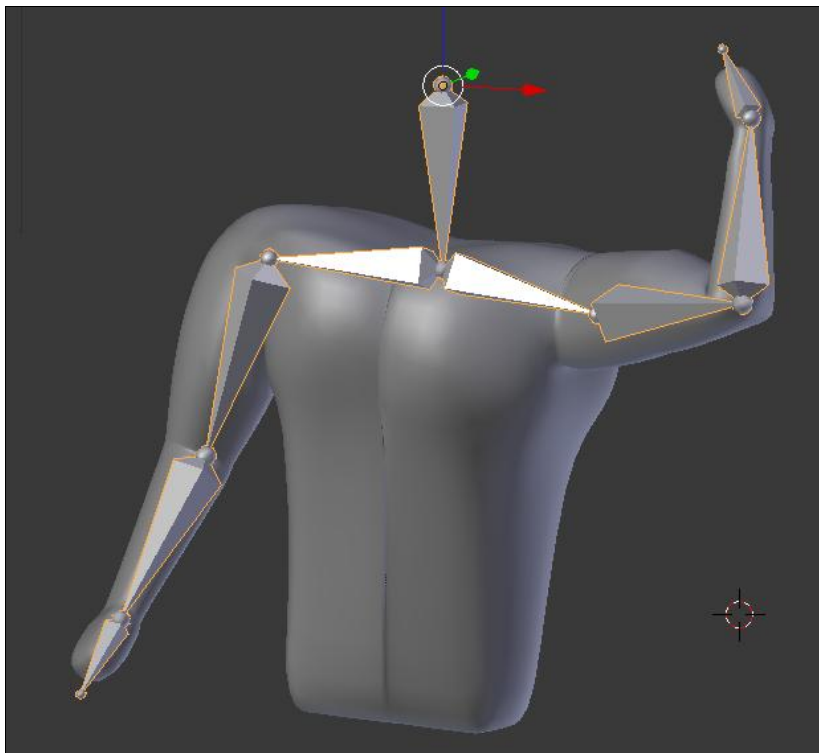


Рис. 5.52. Арматура привязана к модели

В терминологии Blender имеется понятие "*вес вершин*", которое используется в разных областях программы. В применении к **Skinning** под этим понимается степень подвижности вершин по отношению к каждой кости. Blender предлагает специальный режим **Weight Paint** (Окраска вершин) окна **3D View**, который позволяет визуальнo настраивать вес вершин. Выбрать его можно в меню **Mode**, предварительно выделив редактируемый **Mesh**-объект (рис. 5.53).

Если выделить любую кость арматуры, привязанной к модели, то Blender отобразит в цветовой гамме степень влияния **Bone** на близлежащие вершины. Чем темнее цвет, тем менее подвижна вершина. Таким образом, прощелкав по всем костям, можно определить, как программа будет деформировать модель при анимации. Однако главное назначение этого режима — редактирование веса вершин. По сути, инструментарий здесь ничем не отличается от ранее рассмотренного **Vertex Paint** (Окраска вершин). Вы можете выбрать тип кисти, силу воздействия (**Strenght**), радиус курсора (**Radius**).

Сначала нужно выделить кость, которая неправильно деформирует модель, и настроить кисть. На панели **Tool Shelf** есть опция **Weight** (Вес). Чем больше значение этого параметра, тем ярче цвет окрашиваемой вершины, и соответственно, она становится более подвижной по отношению к выделенной кости.

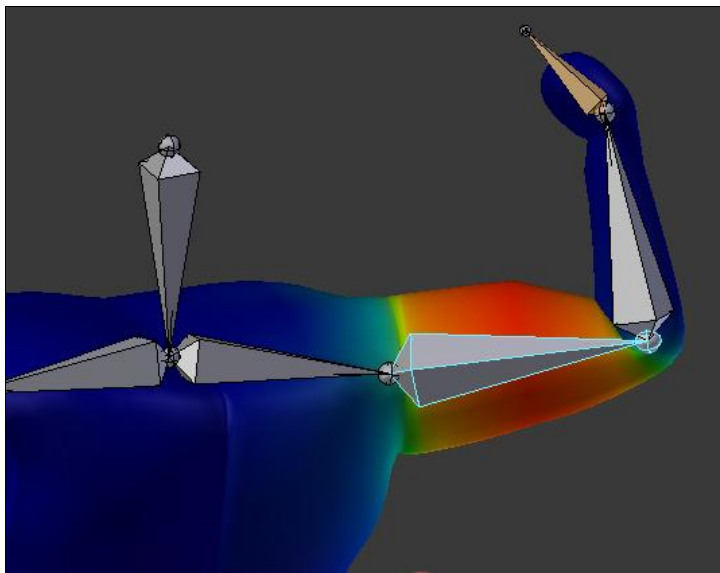


Рис. 5.53. Модель в режиме **Weight Paint**

После настройки кисти можно смело окрашивать вершины. Для удобства будет нелишним развернуть кость так, чтобы были видны нарушения. При этом Blender в реальном времени отобразит изменение деформации области вершин по мере их настройки.

Посмотрите на рис. 5.52. Разрыв в области живота вызван тем, что скелет модели сделан не полностью и программа не смогла правильно настроить вес вершин. Отредактировать нужные области для имеющихся костей, как уже оказалось, не представляет особого труда. Но можно добавить новые **Bone** к арматуре и привязать к ним группы вершин. Основная работа выполняется на панели **Object Data** в группе **Vertex Groups**. Как работать с группами, мы уже рассматривали.

После создания новой группы ее нужно привязать к кости. Это можно сделать с помощью меню **Parent**, но в этот раз нужно будет выбрать пункт **Bone**. После привязки группы к кости необходимо настроить вес вершин в режиме **Weight Paint**.

5.9. Для чего нужны "ограничители"

Вы уже знаете, что **Constraint** — это дополнительные функции, которые позволяют контролировать анимацию объекта. Некоторые из них уже рассматривались, применительно к скелетной анимации, но использование **Constraint** не ограничивается только арматурой.

Меню **Constraint** в окне **Properties** содержит большое количество пунктов (рис. 5.54). Все они разбиты на четыре группы.

- ◆ **Motion Tracking** (Отслеживание движения). Начиная с версии 2.61, в Blender появилась технология **Motion Tracking**. Это дает возможность программе отследить перемещение объекта на видео с помощью специальных меток.
- ◆ **Transform** (Трансформация). Набор ограничителей, обеспечивающих контроль над свойствами объектов: позиции, масштаб, ротация.
- ◆ **Tracking** (Слежение). Здесь предлагается набор функций, способных настроить поведение одного объекта в зависимости от другого. К примеру, с помощью ограничителя **Track To** можно с легкостью заставить камеру всегда следить за перемещением объекта.
- ◆ **Relationship** (Связь). Ограничители данной группы позволяют манипулировать связями между объектами, как, например, функция **Follow Path**.

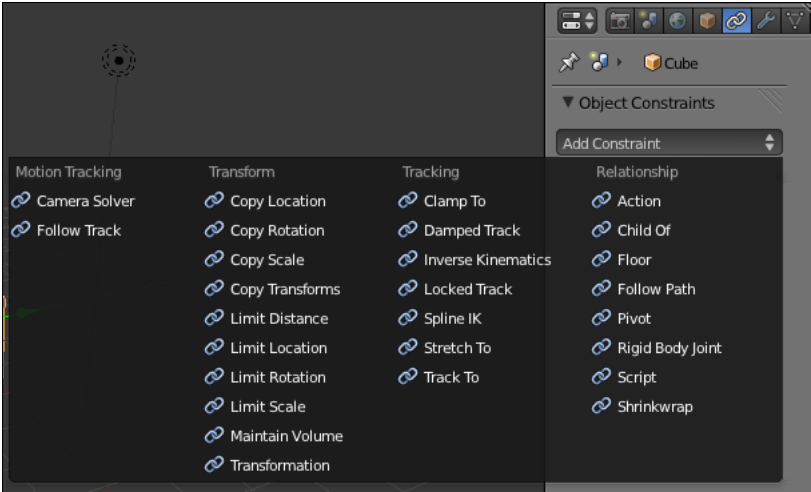


Рис. 5.54. Меню **Constraint**

Работа с **Constraint** напоминает чем-то использование модификаторов. Здесь также имеется возможность добавления нескольких ограничителей к одному объекту, но в отличие от модификатора, **Constraint** не может быть применен (рис. 5.55).

Стандартный заголовок **Constraint** позволяет: переименовать его, отключить работу (кнопка в виде глаза), переместить по стеку и удалить. Особо важная опция —

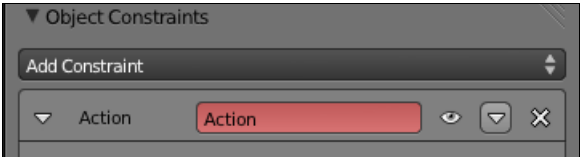


Рис. 5.55. Заголовок **Constraint**

это перемещение по стеку. Результат работы нескольких ограничителей может зависеть от порядка их расположения.

Наиболее популярная группа по использованию — это **Transform**. Начнем рассмотрение ограничителей с нее.

В эту группу входит десять **Constraint**.

- ♦ **Copy (Location, Rotation, Scale, Transforms)** (Копировать (Расположение, Вращение, Масштаб, Трансформация)). Задача этих ограничителей состоит в копировании выбранных параметров у объекта, указанного в поле **Target**, и присвоении их к текущему объекту (рис. 5.56). В ограничителях этого типа присутствуют опции выбора координатных осей и системы (локальная или глобальная).

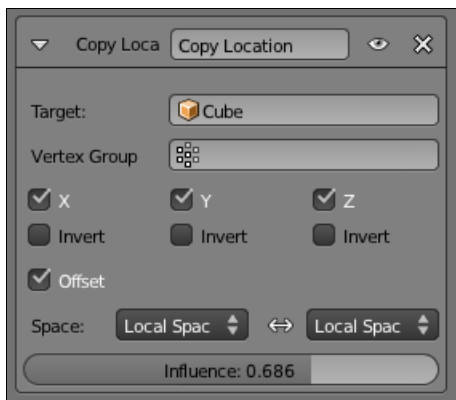


Рис. 5.56. Ограничитель Copy Location

На практике такие ограничители могут использоваться для настройки вспомогательных элементов при анимации скелета. Так, на рис. 5.57 показана арматура кисти, где для движения указательного пальца используется дополнительная кость, не привязанная к модели. Собственно, **Constraint** применен к фаланге пальца. В данном примере используется ограничитель **Copy Rotation**.

- ♦ **Limit (Distance, Location, Rotation, Scale)** (Ограничить (Расстояние, Расположение, Вращение, Масштаб)). Как вы уже догадались, назначение этих **Constraint** в ограничении движения или деформации по определенным осям. Возьмем, к примеру, анимацию пальца на руке. В реальности у живого человека пальцы могут раздвигаться или гнуться на определенный угол. По умолчанию кости кисти на рис. 5.57 способны вращаться так, как им заблагорассудится. Для установки лимита на вращение можно использовать ограничитель **Limit Rotation**. В настройках **Constraint** достаточно выбрать нужную ось вращения, установить минимальное и максимальное значения параметров движения (рис. 5.58).

Особняком в группе **Limit** находится ограничитель **Limit Distance**. Его используют для установки минимального расстояния, которое должно выдерживаться между двумя объектами. При движении главного объекта подчиненный всегда будет следовать за ним на установленном расстоянии. Примером может служить движение локомотива с вагонами.

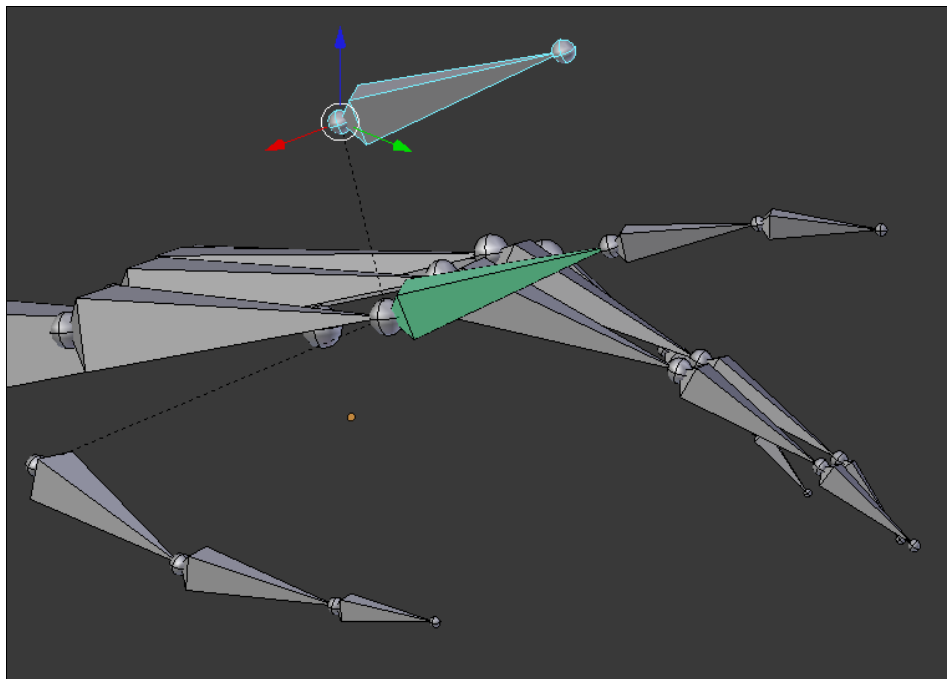


Рис. 5.57. Дополнительная кость обеспечивает удобное управление арматурой

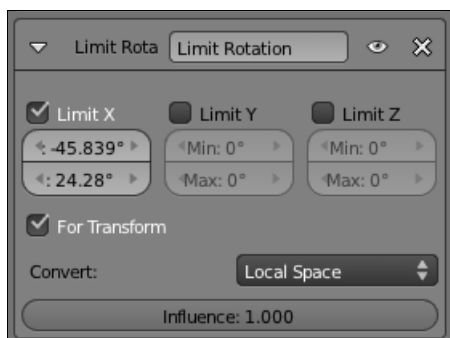


Рис. 5.58. Ограничитель
Limit Rotation

СОВЕТ

Многие **Constraint** имеют в своих настройках параметр **Influence**, который отвечает за точность выполнения поставленного условия. При уменьшении этого значения у контролируемого объекта появляется большая свобода в действии.

- ♦ **Maintain Volume** (Обслуживание объема). Очень простой ограничитель, позволяющий контролировать сжатие объекта. Не стоит путать этот эффект с изменением масштаба! Настройки тут минимальные: выбор оси и установка качества деформации (рис. 5.59).

Группа **Tracking** содержит семь **Constraint**.

- ♦ **Clamp To** (Фиксировать). По своей функциональности этот ограничитель похож на рассматриваемый ранее **Follow Path**. Однако для работы с последним нужно

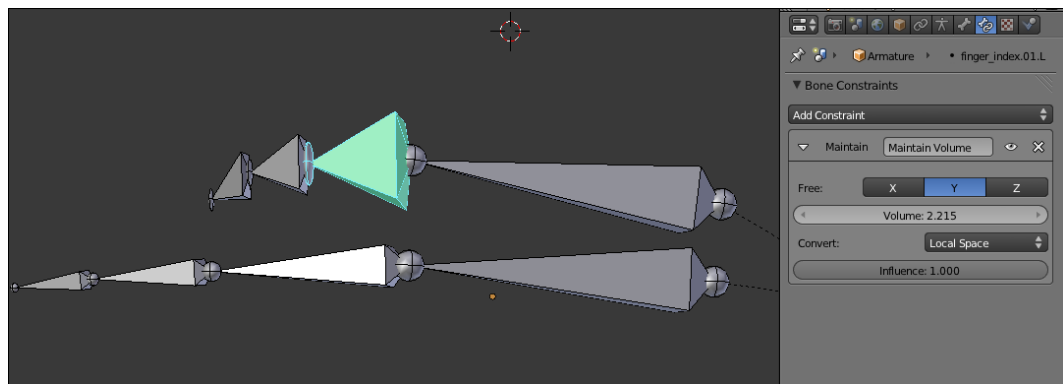


Рис. 5.59. **Maintain Volume**. Сверху результат работы ограничителя, снизу обычное масштабирование

создавать временные ключи (см. разд. 5.4). Использование ограничителя **Clamp To** намертво привязывает объект к траектории, что позволяет анимировать его с помощью простых анимационных ключей позиции. Рассмотрим несложный пример.

Создайте в сцене два объекта — **Cube** и **Curve Path**. Измените форму кривой, как вам заблагорассудится. Добавьте к кубу ограничитель **Clamp To** (рис. 5.60).

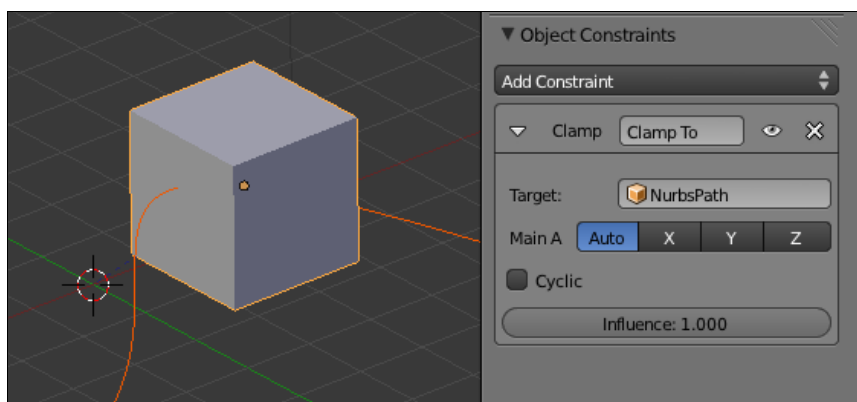


Рис. 5.60. Ограничитель **Clamp To**

Использовать этот **Constraint** просто. В поле **Target** необходимо установить кривую **Path**, выбрать ось в группе **Axis** и, при необходимости, зациклить анимацию в опции **Cyclic**.

Выполните выше описанные действия и попробуйте просто передвинуть куб. Вы увидите, что вне зависимости от выбранной оси движения объект будет перемещаться только по установленной кривой. Для создания анимации нужно установить обычные ключевые кадры **Location**.

Попробуйте сделать простейшую анимацию. Сделайте нулевой кадр ключевым, нажмите клавишу <I> и выберите пункт **Location** в появившемся меню. Устано-

вите другой активный кадр, передвиньте объект и добавьте новый ключ **Location**. Сочетание клавиш <Alt>+<A> заставит куб двигаться по кривой.

Как видите, использовать **Clamp To** гораздо удобнее, нежели **Follow Path**. Но у этого ограничителя имеется один существенный недостаток. Здесь нет возможности установки ориентации объекта по движению.

- ◆ **Damped Track** (Мягкое слежение). Используйте этот ограничитель, если хотите, чтобы ось объекта всегда "смотрела" в сторону другого объекта. В настройках имеется выбор **Target** и установки оси.
- ◆ **Inverse Kinematics** (Инверсная кинематика). Инверсная кинематика рассматривалась в разд. 5.7.
- ◆ **Locked Track** (Блокировка слежения). Смысл этого ограничителя заключается в "замораживании" одной из оси объекта. Проще всего представить работу **Locked Track** на примере обычного компаса. Вы можете переворачивать компас, как заблагорассудится, но стрелка всегда будет указывать на север, при этом вращаясь на оси шпиля.
- ◆ **Spline IK** (Кривая ИК). Ограничитель, который поможет выровнять кости арматуры при помощи кривой. Это может пригодиться, к примеру, для создания модели змеи (рис. 5.61).

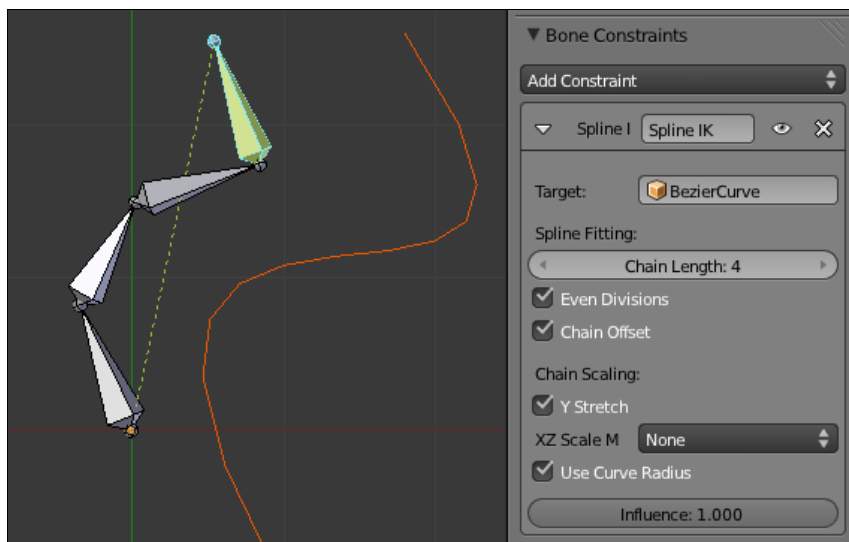


Рис. 5.61. Ограничитель **Spline IK**

- ◆ **Stretch To** (Эластичность). Объект, после выбора этого ограничителя, всегда "смотрит" на **Target** и изменяет свой масштаб по одной из оси при движении последнего.
- ◆ **Track To** (Слежение). Наиболее популярный **Constraint**, который чаще всего используется для привязки одного объекта к другому. К примеру, с помощью него можно заставить камеру следить за движущимся объектом.

СОВЕТ

Некоторые **Constraint** настолько популярны, что могут устанавливаться горячими клавишами. К таким относятся: **Damped Track**, **Track To** и **Lock Track**. Выделите первым объект, к которому должен быть присоединен ограничитель, затем с помощью клавиши <Shift> добавьте объект **Target** и нажмите <Ctrl>+<T>. Появится меню с указанными **Constraint**. Нужно заметить, что в данном случае Blender выполнит автоматическую настройку ограничителей, которая вполне оказывается работоспособной.

Группа **Relationship** (Связь) содержит еще 8 типов ограничителей.

- ◆ **Action** (Действие). Ограничитель, позволяющий использовать анимацию объекта **Target** для собственной анимации. Звучит достаточно запутанно, поэтому рассмотрим его использование на простом примере.

Добавьте в сцену примитивы **Cube** и **Sphere**. Пусть куб у нас будет иметь простейшую анимацию движения из точки А в точку Б за 25 кадров. Создайте эту анимацию с помощью ключей **Location**. По умолчанию программа добавит действие с названием **CubeAction** (вы его можете увидеть, например, в окне **DopeSheet**).

Теперь к сфере присоедините ограничитель **Action** (рис. 5.62).

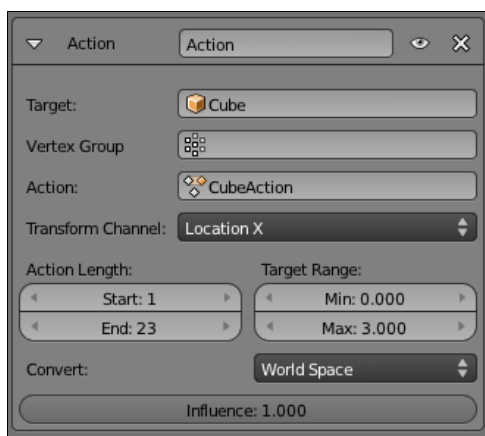


Рис. 5.62. Настройки Action

В опции **Target** выберите из списка примитив **Cube**, а в **Action** — **CubeAction**. Параметр **Transform Channel** отвечает за действие, которое будет производить объект. Выберите из списка опцию **Location X** (движение по координате X). Группа **Action Length** позволяет выбрать начальный (**Start**) и конечный (**End**) кадры анимации. Так как движение куба осуществляется с 1 по 25 кадр, то установите в соответствующие поля эти значения. **Action Length** отвечает за время анимации главного объекта.

Группа **Target Range** имеет поля для установки минимального и максимального значений смещения главного объекта. Установите их в 0.

Теперь при проигрывании анимации будет видно, что сфера совершает прыжок из точки А в точку Б, пока движется куб. Попробуйте установить в поле **Max**

значение 3 и еще раз включите анимацию. На этот раз сфера совершит плавное перемещение вместе с кубом.

- ◆ **Child Of** (Дочерний элемент). Установка родительской связи между объектами наподобие известной функции **Parent**. В отличие от последней имеет возможность более широкой настройки связи объектов.
- ◆ **Floor** (Пол). Этот **Constraint** может использоваться для создания непроходимых препятствий, таких как стены или пол. Смысл его заключается в том, что по достижении объекта **Target** ограничитель запрещает перемещение главного объекта в выбранном направлении.

Создайте в проекте **Plane** и расположите над ним **UV Sphere**. Присоедините к сфере ограничитель **Floor**. В качестве параметра **Target** выберите примитив **Plane**. Так как движение сферы осуществляется по координате Z (сверху вниз), то включите кнопку Z в настройках **Floor** (рис. 5.63).

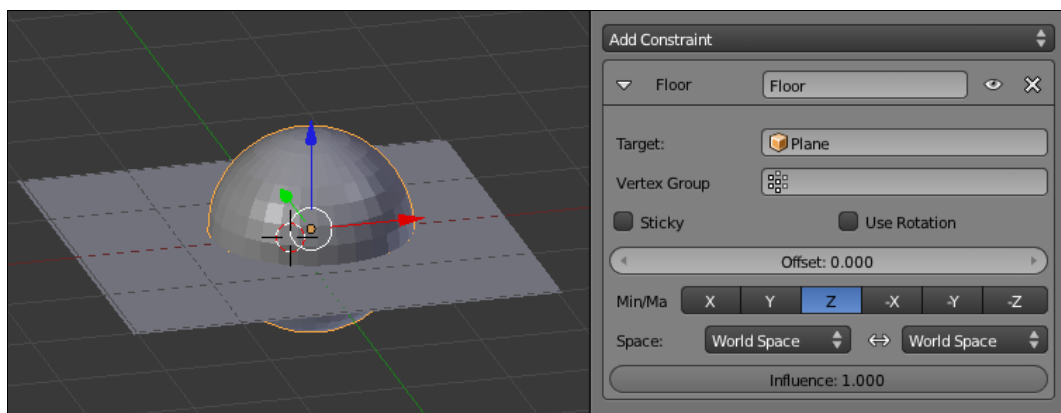


Рис. 5.63. Настройки **Floor**

Попробуйте переместить сферу вниз, и вы увидите, что она "упирается" в плоскость (своим центром). Вы можете передвигать ее по координатам X или Y, или даже по Z, но не ниже **Plane**.

В параметрах **Floor** имеется опция **Sticky**. Включите ее, если хотите, чтобы сфера при соприкосновении "прилипла" к **Target**. В этом случае остается лишь возможность движения объекта по оси Z вверх.

- ◆ **Follow Path** (Следовать маршрутом). Движение по траектории рассматривалось в разд. 5.4.
- ◆ **Pivot** (Центр). По умолчанию вращение объекта осуществляется вокруг его центра. Этот ограничитель позволяет использовать в качестве центра вращения объект, указанный в поле **Target**.
- ◆ **Rigid Body Joint** (Соединитель Rigid Body). Специфичный ограничитель, используемый для настройки физики объекта.
- ◆ **Script** (Скрипт). Позволяет выбрать скрипт для выполнения. По умолчанию версия Blender не имеет сторонних скриптов **Constraint**.

- ♦ **Shrinkwrap** (Упаковка). С помощью этого ограничителя можно получить эффект, когда один объект при соприкосновении с другим как бы обтекает его со стороны.

5.10. Работа с *Action Editor*

Вы уже знаете, что анимация арматуры создается в режиме **Pose Mode**. Принцип работы все тот же — использование ключевых кадров. Откройте проект Blender, где вы создали скелет человеческого торса, или загрузите файл `Scenes/glava5/glava5_skin_simple.blend` из архива примеров к этой книге (см. приложение 2).

Выделите арматуру и включите режим **Pose Mode**. Для настройки анимации оптимально подходит раскладка окон **Animation**. Выберите ее из главного меню программы.

В отличие от ранее рассмотренных способов, анимация арматуры выглядит несколько сложнее из-за обилия элементов скелета. Поэтому, в данном случае, удобнее использовать режим автоматической записи (см. рис. 5.2). Активируйте красную кнопку записи в окне **Timeline**.

Теперь попробуем создать первое движение скелета. Пусть это будет поднятие вверх руки, которая имеет настроенный ИК-привод. Текущим фреймом установите кадр 0. Выберите самую нижнюю **Bone**, нажмите клавишу <I> для вызова меню **Insert Keyframe** (рис. 5.64).

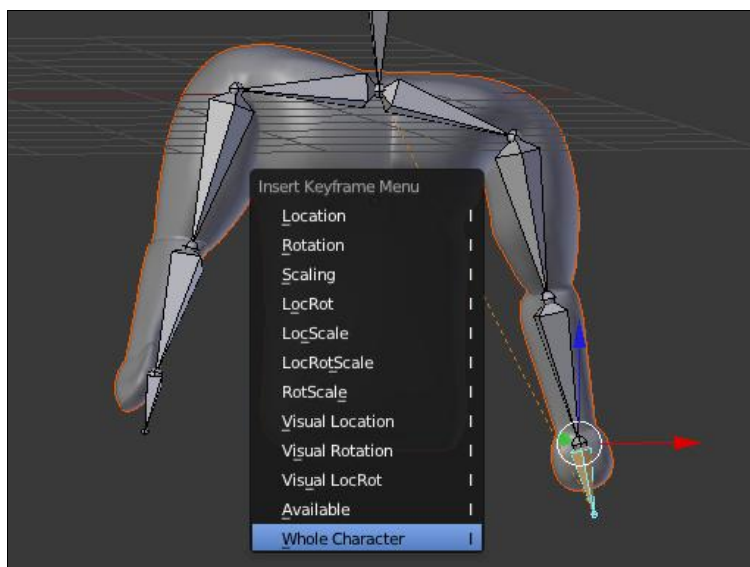


Рис. 5.64. Начальная позиция скелета

Для работы с арматурой данное меню имеет пункт **Whole Character** (Весь персонаж). При выборе этой опции Blender создаст ключевые кадры всех костей выделенного скелета. Выберите этот пункт.

Если вы посмотрите на окна **Dope Sheet** и **Curve Editor**, то увидите список всех **Bone**, участвующих в анимации.

Теперь передвиньте анимационный курсор на несколько кадров дальше. Поднимите руку. Так как кнопка записи анимации ранее была включена, то программа создаст ключи для нужных костей (5.65).

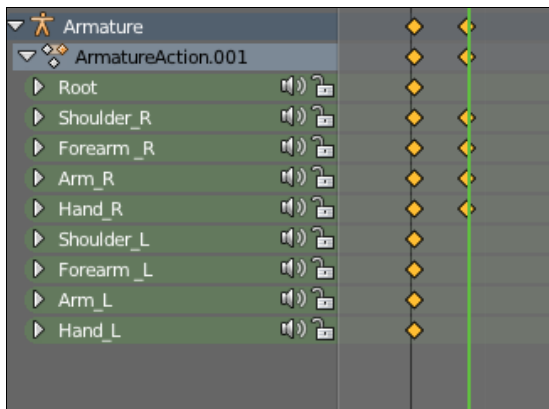


Рис. 5.65. Окно **Dope Sheet** для арматуры

Таким образом, вы можете создавать любую мыслимую анимацию своей модели. Работа с **Dope Sheet** и **Curve Editor** уже рассматривалась.

Теперь представьте ситуацию, когда необходимо выполнение разных действий персонажа. Допустим, модель должна уметь ходить, бегать, да хоть плясать вприпрыжку! Все это сделать можно, но несложно догадаться, какой объем анимации придется выполнить. Ведь даже для одного действия, скажем бега, будут задействованы почти все кости скелета. Обилием ключевых точек в анимационных редакторах вы будете обеспечены. И если вдруг понадобится скорректировать время выполнения нужной анимации, то немудрено будет запутаться.

"Действие" (**Action**) — это слово уже не раз использовалось в книге, но в данный момент здесь понимается специальная возможность, предоставляемая программой. *Действием* называется законченная анимация, объединяющая в себе множество движений элементов объекта. Таким образом, бег, ходьба, пляска — это **Action**.

Конечно, вы вправе не использовать действия в своей анимации, а просто ограничиться ключами, но посмотрим на преимущества этой техники.

- ◆ **Action** не зависят друг от друга. А это значит, что вы можете редактировать каждое действие, без боязни нарушить другие.
- ◆ **Action** могут многократно использоваться в общей анимации объекта со сдвигом во времени или даже скорости исполнения.
- ◆ **Action** могут экспортироваться в другие форматы (вместе с моделью) для использования в сторонних программах.

Даже если вы не планируете использовать действия в своей сцене, при работе с анимацией Blender автоматически их создает. Но они являются одиночными для

каждого объекта. Посмотрите на рис. 5.65, где показывается окно **Dope Sheet**. В левой части редактора список костей объединен в папку с названием **ArmatureAction**. Это и есть автоматически созданное действие для арматуры.

Забегая вперед скажу, что в Blender имеется специальный редактор для работы с **Action** — **NLA Editor**. С его помощью и контролируется глобальная анимация объекта и сцены. Но сначала нужно научиться создавать свои собственные действия.

Для этой цели редактор **Dope Sheet** имеет специальный режим **Action Editor** (Редактор действий), который доступен в меню **Mode** на заголовке окна (рис. 5.66).

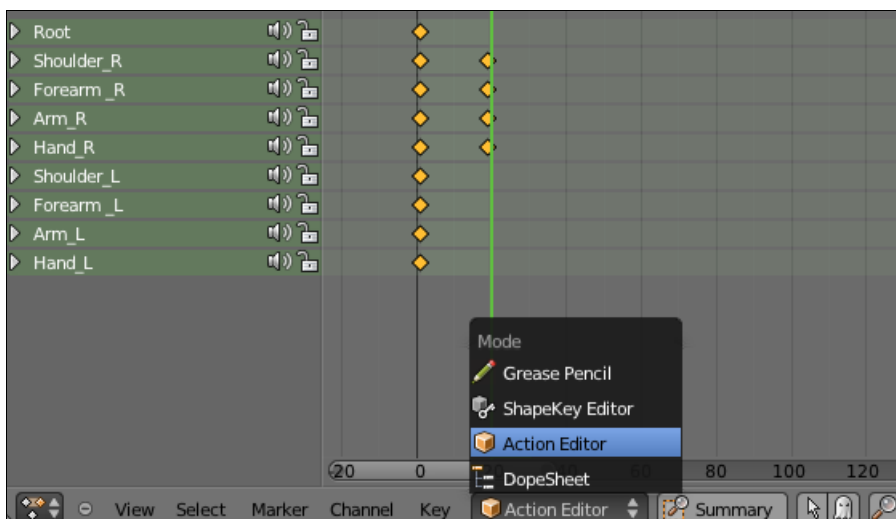


Рис. 5.66. Редактор **Dope Sheet** в режиме **Action**

Как видите, внешне **Action Editor** особо не отличается от стандартного **Dope Sheet**. По сути дела, это всего лишь смена режима работы окна **Dope Sheet**, и основные возможности редактирования анимации здесь не претерпели никаких изменений. Но несколько изменился заголовок окна, где появилось поле управления действиями (рис. 5.67).

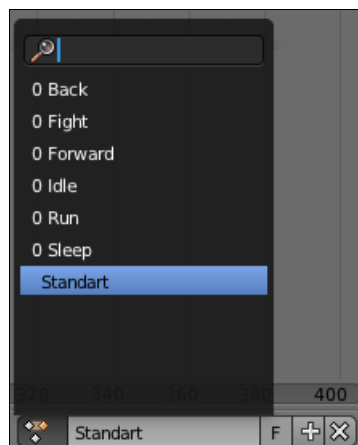


Рис. 5.67. Опции управления действиями

Это стандартный интерфейс создания новых блоков данных Blender. Такой, например, используется для мультиматериалов или текстур. Вы можете: переименовать текущее действие, щелкнув по его названию; выбрать иное из списка; добавить или удалить с помощью кнопок "+" и "×".

Так как вы сделали ранее анимацию движения руки, то действие уже существует и называется **ArmatureAction**. Теперь щелкните по кнопке со знаком плюс для добавления нового **Action**. Внешне при этом в окне **Action Editor** ничего не изменится. Дело в том, что Blender при создании действия копирует данные ключей из активной зоны. Выделите все ключи клавишей <A> и нажмите <X> для удаления. Теперь вы можете создавать новые движения персонажа.

5.11. NLA Editor — заключительный аккорд

Non-Linear Animation Editor (НеЛинейный Анимационный Редактор) — это высокоуровневый редактор, который позволяет компоновать все анимации сцены удобным и понятным для пользователя способом.

С помощью **NLA Editor** появляется возможность компоновки анимации в любой последовательности и с различными эффектами. Вся анимация в окне редактора выглядит в виде полосок, с которыми и происходит работа. Вы можете выстроить их одну за другой, изменить время запуска, увеличить или уменьшить скорость и многое другое. Если вы работали раньше с какой-нибудь программой видеомонтажа, то подобная концепция будет вам знакомой.

Рассмотрим работу с этим редактором с помощью простого примера. В нашей сцене будут участвовать два объекта: **Cube** и **UV Sphere**. Куб движется по горизонтали, сталкивается со сферой, которая отлетает в сторону.

Создайте новую сцену Blender и добавьте сферу. Переключитесь в режим просмотра **Front**. Разместите два объекта на некотором удалении друг от друга.

Сначала нужно сделать анимацию. Включите раскладку окон **Animation** и активируйте в окне **Timeline** режим автоматической записи (рис. 5.68).

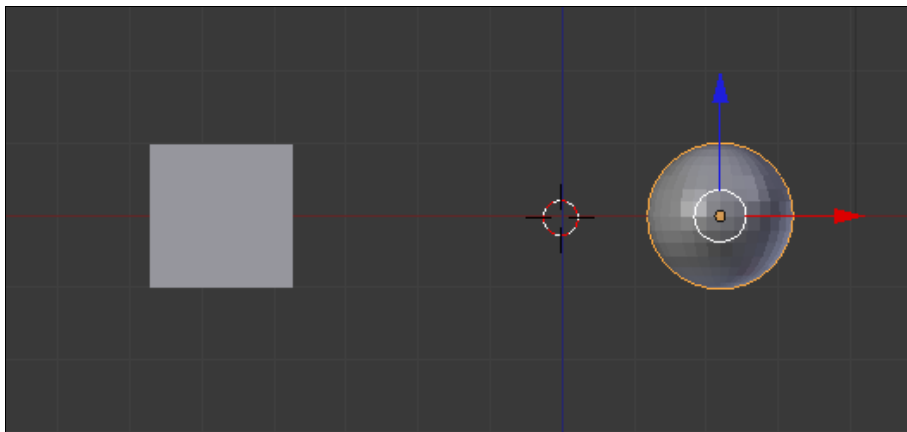


Рис. 5.68. Подготовка к анимации

Выделите куб, нажмите клавишу <I> для вызова меню **Insert Keyframe**. В этом меню выберите пункт **Location**. Установите анимационный курсор в окне **Timeline** на кадр 20. Теперь передвиньте куб вправо близко к сфере.

Займемся анимацией сферы. Установите активным первый кадр. Выберите сферу и создайте ключ **Location**. Затем в кадре 20 передвиньте сферу вправо на некоторое расстояние.

Нажмите клавиши <Alt>+<A> для проигрывания анимации. Вы увидите, что оба объекта одновременно уедут в правую сторону. Это, конечно, не тот результат, который нужно достичь, но все можно исправить в **NLA Editor**.

Откройте в любом удобном месте окно **NLA Editor**, например на месте **Dope Sheet**. Окно редактора разбито на две области: в левой отображаются объекты, а в правой — собственно ключи (рис. 5.69).

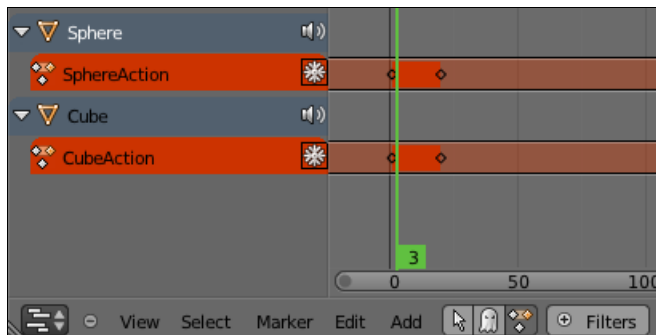


Рис. 5.69. Окно **NLA Editor**

В левой части редактора имеется список объектов сцены, у которых есть анимация. Сейчас там находятся **Cube** и **Sphere**. Для каждого примитива **NLA Editor** показывает действие, которое было последним в окне **Action Editor**. По умолчанию Blender создал действия: **SphereAction** и **CubeAction**.

Выглядит просто, но если вы попытаетесь передвинуть любую точку в правой области окна, то потерпите поражение. Дело в том, что **NLA Editor** в данный момент показывает лишь наличие действий для объектов, но работать он умеет только с собственными данными. Поэтому названия анимаций выделены красным цветом, как неактивные.

Присмотритесь к **Action** в окне редактора (см. рис. 5.69). Справа у каждого названия имеется небольшая круглая кнопка. Если ее нажать, то редактор станет работать с анимацией, как с данными NLA. Щелкните по этим кнопкам у обоих объектов (рис. 5.70).

Давайте рассмотрим, что же получилось. Во-первых, каждый объект получил дополнительные полосы с названиями **NlaTrack**. Во-вторых, в главной области окна появились полосы желтого цвета с названиями действий.

В терминологии Blender полосы желтого цвета называются **Strip** (стрип, полоска). Проще говоря, каждая из них является тем или иным действием объекта. Теперь вы можете настроить анимацию так, как необходимо.

Щелкните правой кнопкой мыши по **Strip** с названием **SphereAction**. Вы увидите, что полоска с именем **CubeAction** станет серого цвета. Таким образом, Blender помечает выделенный **Strip** желтым фоном.

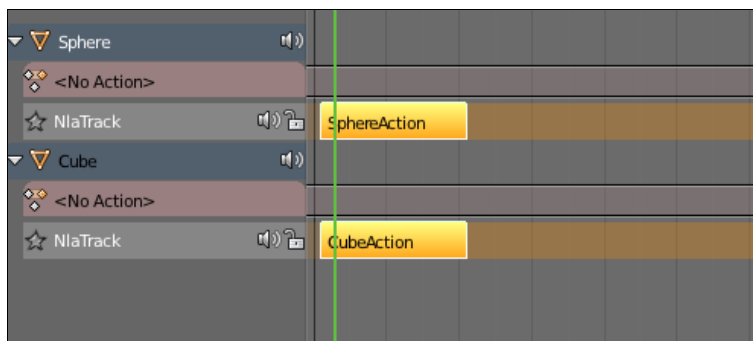


Рис. 5.70. NLA Editor в режиме редактирования

Нажмите правой кнопкой мыши по **SphereAction** и, не отпуская ее, сдвиньте в сторону. **Strip** изменит название на **TempMeta** и станет фиолетового цвета. Но самое главное, что теперь движением мыши вы можете перетаскивать полосу в любое место. Переместите ее в конец полосы **CubeAction** и нажмите левую кнопку мыши для фиксации результата (рис. 5.71). Вот теперь анимация столкновения куба со сферой будет работать корректно.

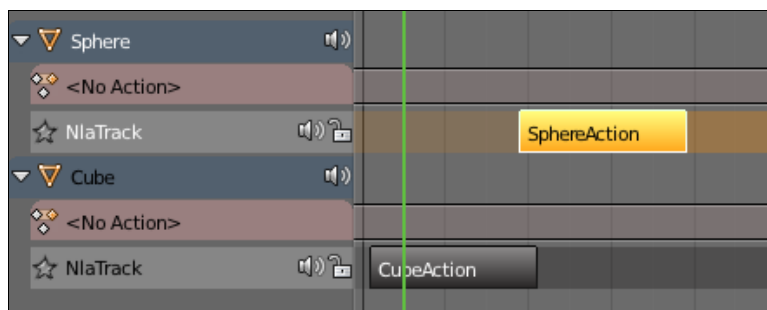


Рис. 5.71. Правильно настроенная анимация

Полосы с названиями **NlaTrack** являются слоями. В каждом таком слое могут находиться несколько **Strip** с различными **Action**.

Передвиньте анимационный курсор в окне редактора так, чтобы он не находился ни на одном из **Strip**. Выделите **Strip** с именем **SphereAction** и нажмите клавиши <Shift>+<A>. Появится меню, где программа выдаст список имеющихся действий в сцене. Выберите пункт **CubeAction** (рис. 5.72).



Рис. 5.72. Слой с несколькими анимациями

Как видите, к анимации **SphereAction** присоединилась **CubeAction**. Попробуйте проиграть то, что получилось. Результат вас наверняка обескуражит.

Несмотря на то, что вторая анимация называется **CubeAction**, она расположена в канале примитива **Sphere**. Логично предположить, что произойдет повтор движения куба. В действительности, вторично будет двигаться сфера, причем при этом дублировать анимацию куба. Такое возможно из-за того, что **NLA Editor** считает **Action** независимыми от их объектов. Этот фокус позволяет использовать чужую анимацию на разных объектах (правда, не всегда). Попробуйте расставить **Strip**, как на рис. 5.73, и вы получите своеобразные догонялки объектов друг за другом.

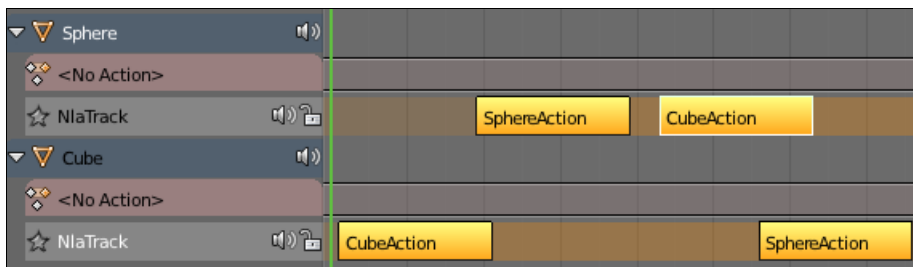


Рис. 5.73. Это расположение **Strip** поменяет местами объекты в анимации

По умолчанию **NLA Editor** создает по одному слою для каждого объекта и помещает в него **Strip** с последним активным действием. Вы можете управлять количеством таких слоев.

Для этого нужно сначала выделить имя последнего слоя левой кнопкой мыши. После этого буквы станут белого цвета. Для создания нового слоя воспользуйтесь меню **Add | Add Track** в заголовке окна **NLA Editor**. Чтобы удалить слой пустой или со всеми **Strip**, переведите курсор мыши в левое поле, выделите название нужного канала и нажмите клавишу **<X>**.

Это базовые возможности использования **NLA Editor**, но есть еще у него немало дополнительных функций.

Допустим, вам захотелось один из **Strip** проиграть в обратном направлении и с удвоенной скоростью.

Выделите любой **Strip** и нажмите клавишу **<N>**. В правой части окна редактора появится панель **Properties** для данного слоя и объекта на нем (рис. 5.74).

Здесь вы можете переименовать сам слой или отдельно выбранный **Strip**. Изменить привязанное действие к **Strip** (опция **Action** закладки **Action Clip**). Вообще отключить **Strip** от участия в анимации (опция **Muted**) и многое другое.

В нашем случае понадобится задействовать всего две функции:

- ◆ **Reversed** (закладка **Active Strip**) — при включении данной опции анимация выделенного **Strip** будет проигрываться реверсивно;
- ◆ **Scale** (закладка **Action Clip**) — масштабирование **Strip** приводит к тому, что изменяется время проигрывания анимации.

Рис. 5.74. Панель свойств окна **NLA Editor**

Попробуйте в поле **Scale** установить значение 0.5 и включить опцию **Reversed**. Вы увидите, что данный **Strip** будет проигрываться с конца с удвоенной скоростью.

5.12. Практика. Жарим яичницу

Помните процесс жарки яичницы? Берется яйцо, разбивается, и содержимое выливается на сковородку. Нечто подобное будем делать в этом уроке.

Рассмотрим этапы решения этой задачи:

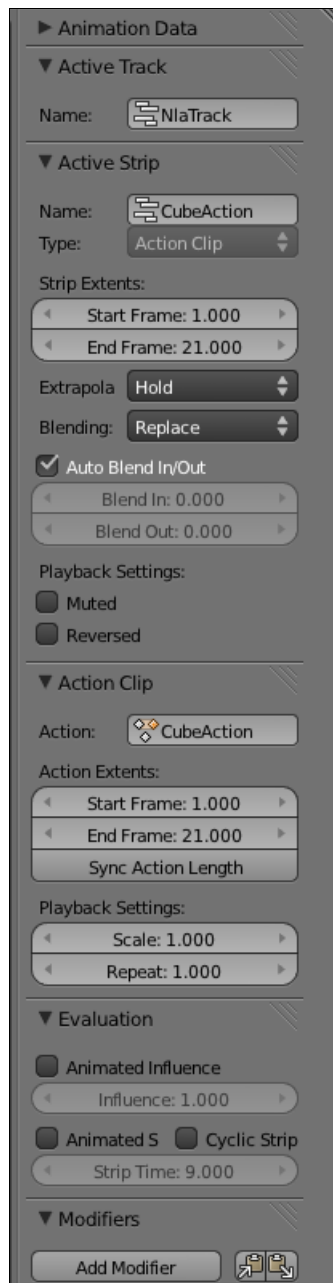
1. Создание моделей: скорлупы, содержимого яйца, сковородки.
2. Анимация раскрытия скорлупы.
3. Анимация вытекания жидкости яйца.
4. Анимация растекания яйца по сковородке.

Если с первыми двумя пунктами вопросов возникнуть не должно, то последние вызывают сомнение. По сути, работу с жидкостями нужно выполнять с помощью физики Blender, но т. к. эта тема пока не раскрывалась, то придется обойтись теми инструментами, которые уже известны. В действительности, использовать физику для такого простого действия не нужно. Дело в том, что расчет физики жидкости задача весьма трудоемкая и потребует немалого процессорного времени.

Вам придется вспомнить принципы моделирования, работу с модификаторами и, конечно же, использование анимационных редакторов. Так как урок посвящен работе с анимацией, то настройка материалов будет самой примитивной.

Начнем с создания модели яйца. Проще всего ее сделать из примитива **UV Sphere**. Откройте новый проект, удалите из сцены куб и создайте сферу. Выделите верхнюю вершину сферы и переключитесь в режим просмотра **Front View**.

Для придания остроконечной формы нужно вытянуть выделенную вершину по координате Z вверх. Вот только делать это нужно с включенным режимом пропор-



ционального редактирования. Нажмите клавишу <O>, затем <G> и <Z>. Переместите вершину вверх, одновременно регулируя колесиком мыши зону охвата (рис. 5.75).

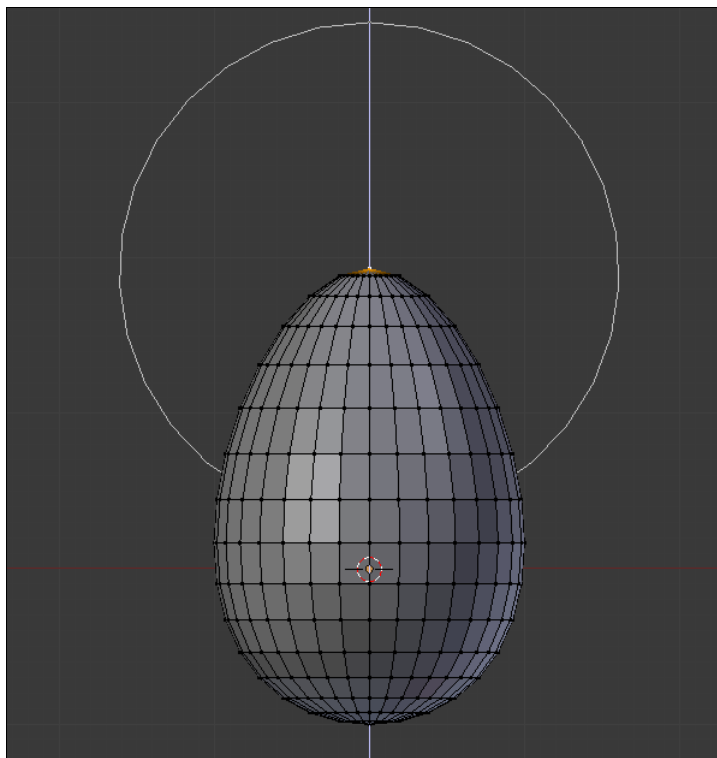


Рис. 5.75. Создание внешней оболочки яйца

Включите функцию **Smooth** в режиме **Object Mode**, и внешняя форма яйца готова. Нажмите клавишу <H>, чтобы временно скрыть модель в сцене.

Теперь займемся созданием сковородки. В качестве основы для нее послужит примитив **Circle** из группы **Mesh**. Переключитесь в режим просмотра **Top View** (<Num Pad 7>) и добавьте объект.

На панели **Tool Shelf** найдите группу **Add Circle**, которая содержит настройки нового примитива. Включите в них опцию **Fill** (Заливка).

Для создания сковородки выполните следующие шаги:

1. Переключитесь в режим **Front View** (<NumPad 1>) и войдите в режим редактирования (<Tab>).
2. Нажмите кнопку **Extrude** на панели **Tool Shelf** (или просто <E>) и вытяните немного вверх выделение. Не забудьте отключить перед этой операцией режим пропорционального редактирования. Теперь с помощью масштабирования (<S>) немного расширьте верхнюю часть объекта. Таким образом, получилась объемная модель (рис. 5.76).

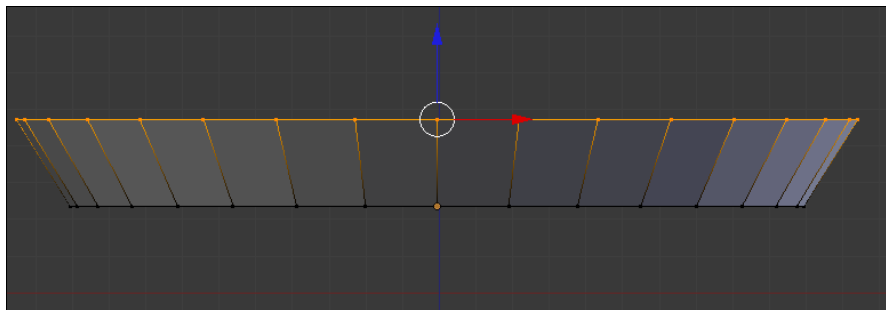


Рис. 5.76. Создание формы

3. Не снимая выделение, переключитесь в **Top View**. Нажмите клавишу <E> для выполнения операции выдавливания и сразу же зафиксируйте результат щелчком левой кнопки мыши. Немного сожмите выделение с помощью инструмента **Scale** (<S>). Этим вы сделаете заготовку для дна сковородки (рис. 5.77).

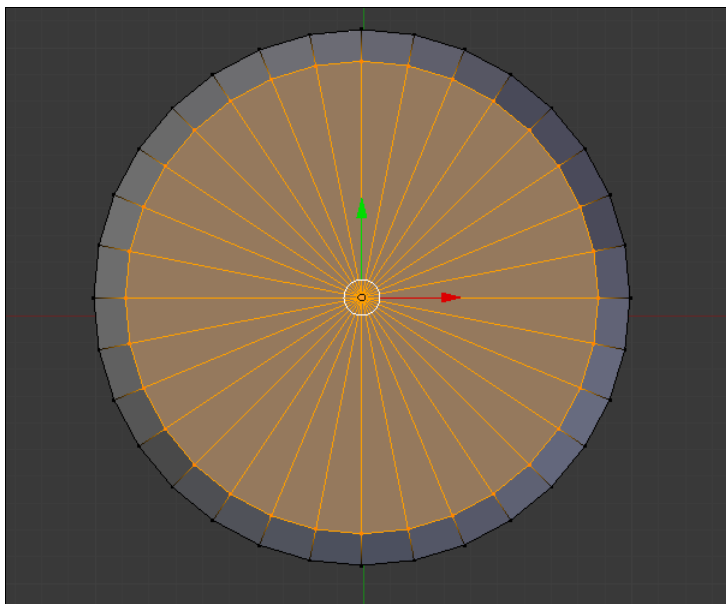


Рис. 5.77. Заготовка для создания углубления

4. Разверните сцену для удобного просмотра. Опустите выделение до дна сковородки. Возможно, придется немного подогнать масштаб опускаемой части.
5. Выйдите из режима редактирования и добавьте к объекту модификатор **Subdivision Surface**. Установите значение 2 для опции **View** (рис. 5.78).

Займемся созданием содержимого яйца. Роль его будет играть примитив **UV Sphere**, но сначала нажмите клавишу <H>, чтобы временно скрыть сковородку.

Добавьте сферу в сцену из меню **Add | Mesh | UV Sphere**. Сразу же включите для объекта сглаживание кнопкой **Smooth** на панели **Tool Shelf**. В принципе редакти-

ровать объект нет надобности, но немного настроим материал для большей правдоподобности.

Перейдите в настройки материала в окне **Properties** и нажмите кнопку **New**. Этот материал будет играть роль белка, поэтому он должен быть белого цвета. Установите цвет **Diffuse** белым. Переименуйте материал как **White**.

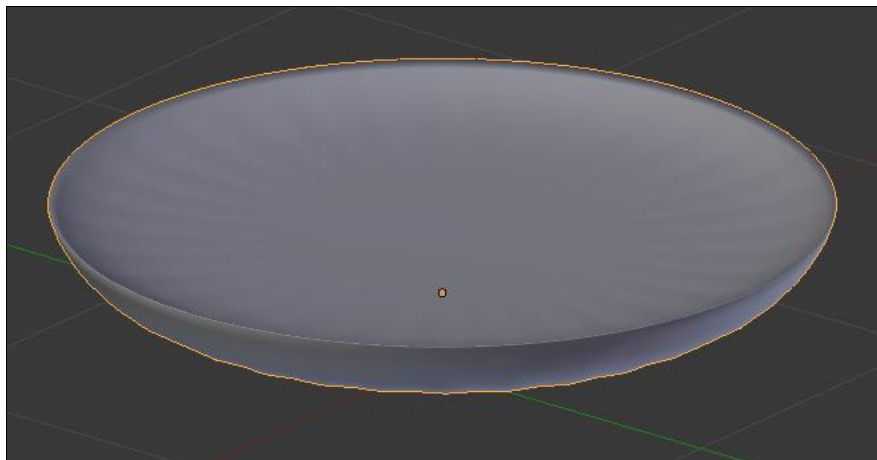


Рис. 5.78. Готовая модель сковородки

Добавьте новый слот в списке материалов, создайте материал, измените его цвет на желтый и назовите его **Yellow** (рис. 5.79).

Перейдите в режим редактирования и выделите верхнюю часть яйца, как на рис. 5.80.

СОВЕТ

Используйте инструмент **Circle** (<C>) для кругового выделения в просмотре **Top View**.

На панели **Material** выберите из списка материал с именем **Yellow** и нажмите кнопку **Assign**. Выделенная область сразу же окрасится в желтый цвет, все остальное останется белым. Если что-то не получается, еще раз прочтите *разд. 4.6*.

На этом создание моделей завершено. Нажмите клавиши <Alt>+<H> для возвращения в сцену скрытых объектов.

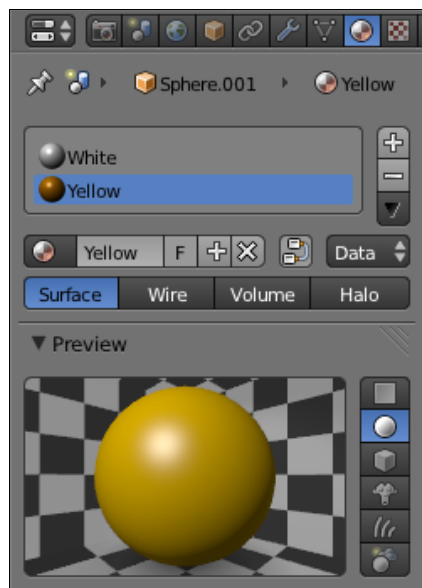


Рис. 5.79. Мультиматериалы для содержимого яйца

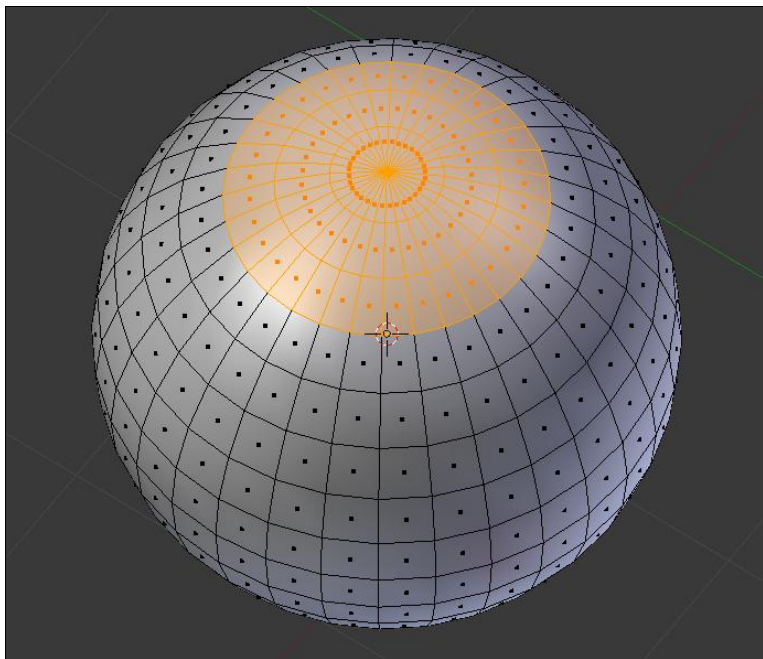


Рис. 5.80. Выделение желтка

Выделите скорлупу и уменьшите ее масштаб в соответствии с размером сковородки. Переместите ее немного выше. Нажмите клавишу <N> для вызова свойств объекта. Установите значение -90 в параметр **Rotation Y** для разворота яйца по горизонтали (рис. 5.81).

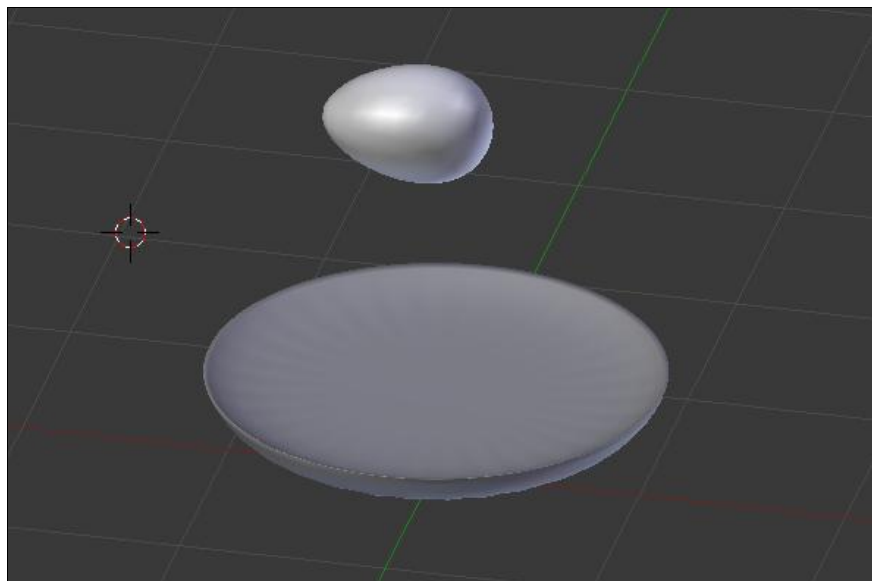


Рис. 5.81. Подготовка сцены

Нужно разделить скорлупу на две части так, чтобы края у них были рваные. Нечто подобное уже делалось во второй главе с ножкой гриба.

Выделите скорлупу, нажмите клавишу <NumPad .>, затем <NumPad 1> — активный объект займет всю рабочую область окна в режиме просмотра **Front View**. Выделите в центре объекта круговую линию разлома. Это можно сделать в режиме работы с ребрами с удерживаемой нажатой клавишей <Alt> (рис. 5.82).

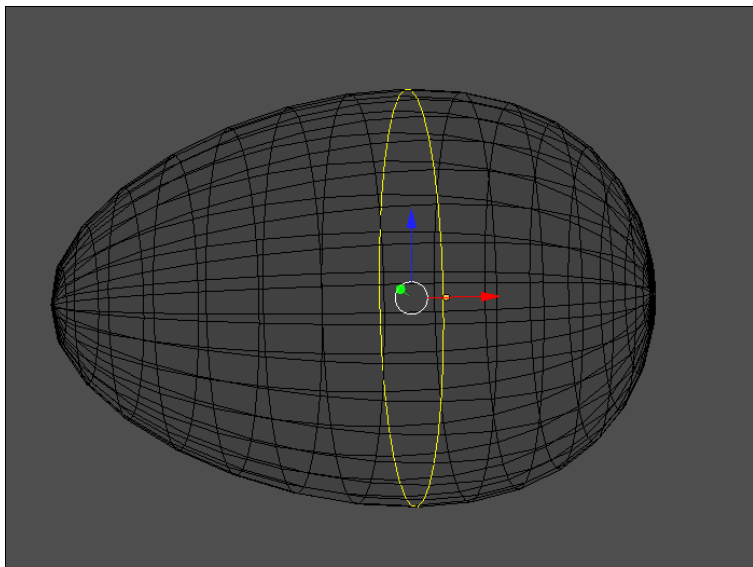


Рис. 5.82. Линия разлома

Для удобства работы нужно скрыть все остальные элементы объекта. Инвертируйте выделение с помощью функции **Select | Inverse**. Нажмите клавишу <H>, чтобы скрыть выделение.

Выделите одну из вершин оставшегося в сцене кольца. Для создания изломанной линии необходимо включить пропорциональное редактирование в режиме **Random**. Это можно сделать в меню **Proportional Editing** в заголовке окна **3D View** (рис. 5.83).

Нажмите клавишу <G>, затем <X> и передвиньте влево мышь. С помощью колесика мыши отрегулируйте область воздействия так, чтобы функция **Random** охватила все кольцо (рис. 5.84).

Восстановите скрытые элементы с помощью <Alt>+<H> и выделите полигоны одной половины скорлупы так, чтобы захватить линию разлома. Теперь можно воспользоваться функцией разделения объекта **Mesh | Verticles | Separate**. Вот так и получилось аккуратно разломанное яйцо (рис. 5.85).

Пришло время заняться анимацией — начнем со скорлупы. Выделите одну половинку и попробуйте развернуть ее (<R>). Получится нечто несуразное, как будто одна половина яйца вдавливается в другую (рис. 5.86).

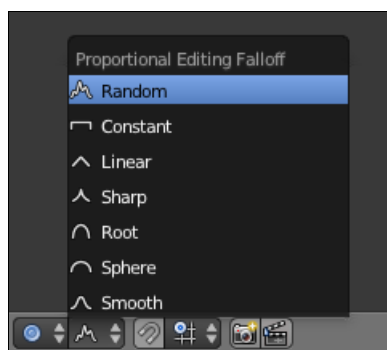


Рис. 5.83. Включение Random

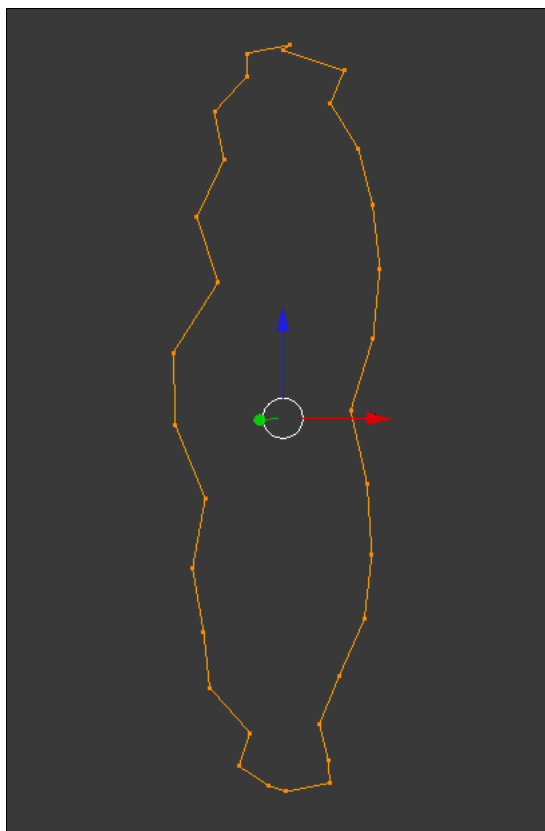


Рис. 5.84. Изломанная окружность

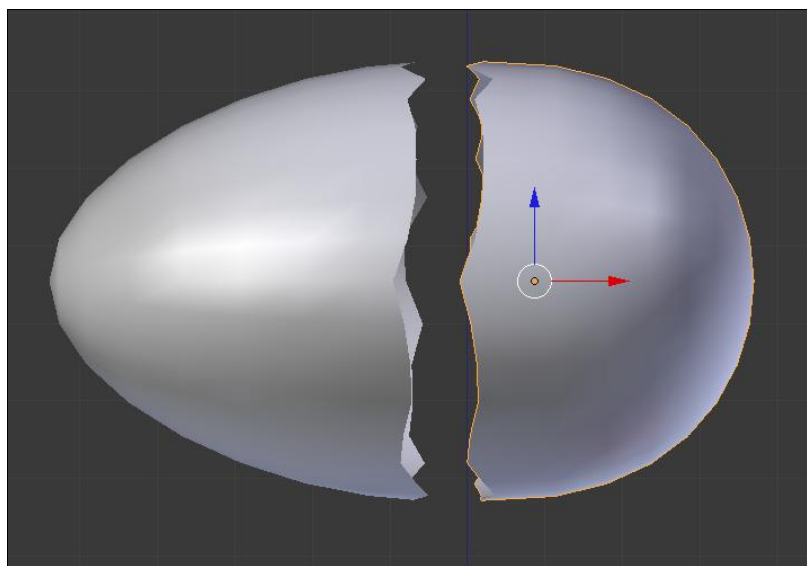


Рис. 5.85. Две половинки скорлупы

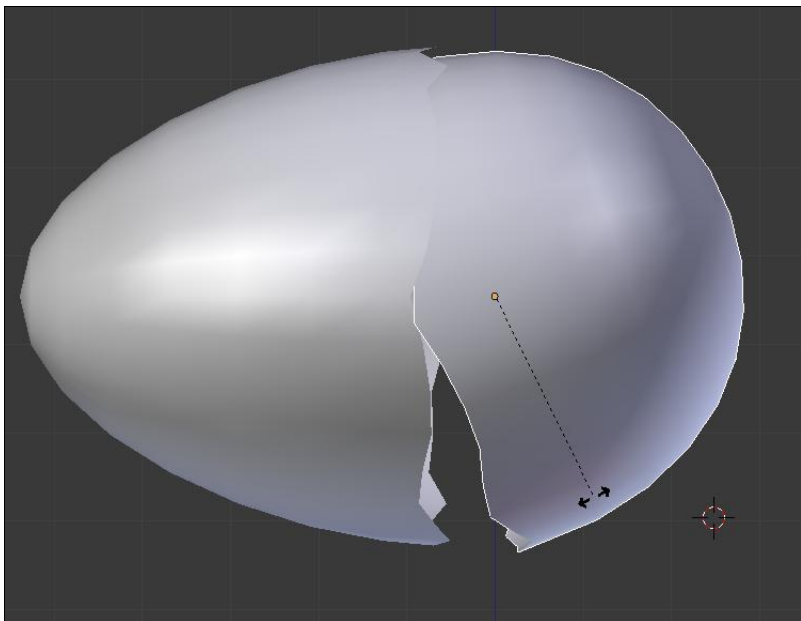


Рис. 5.86. С центром по умолчанию анимация будет неправильной

Проблема в том, что точка вращения объекта располагается в его центре. Нам же нужно сместить центр к одной из верхних вершин объекта. Это можно сделать с помощью установки **3D Cursor** в нужном месте и функции перерасчета **Origin To 3D Cursor**.

Вручную установка **3D Cursor** на месте одной из вершин — дело трудоемкое и неблагодарное. Поэтому воспользуемся функцией **Snap**.

В режиме редактирования (**Front View**) выделите одну из верхних вершин на линии разлома. Выполните функцию **Mesh | Snap | Cursor to Selected**. **3D Cursor** переместится в нужное место. Выйдите из режима редактирования. Вызовите функцию **Mesh | Transform | Origin to 3D Cursor**. Пропеделайте то же самое со второй половинкой скорлупы (рис. 5.87).

Чтобы в дальнейшем не запутаться в объектах, необходимо их переименовать в нечто более понятное. Проще всего это сделать в окне **Outliner**. Принцип следующий, выделяете примитив, находите его в **Outliner**, щелкаете левой кнопкой мыши по названию, одновременно удерживая <Ctrl>, и изменяете имя.

Используйте следующие имена для объектов:

- ◆ левая половинка скорлупы — **egg_left**;
- ◆ правая половинка скорлупы — **egg_right**;
- ◆ сковородка — **Target**;
- ◆ внутренность яйца — **egg_in**.

Для анимации раскрытия скорлупы воспользуемся обычными ключами. Выберите первый фрейм активным кадром.

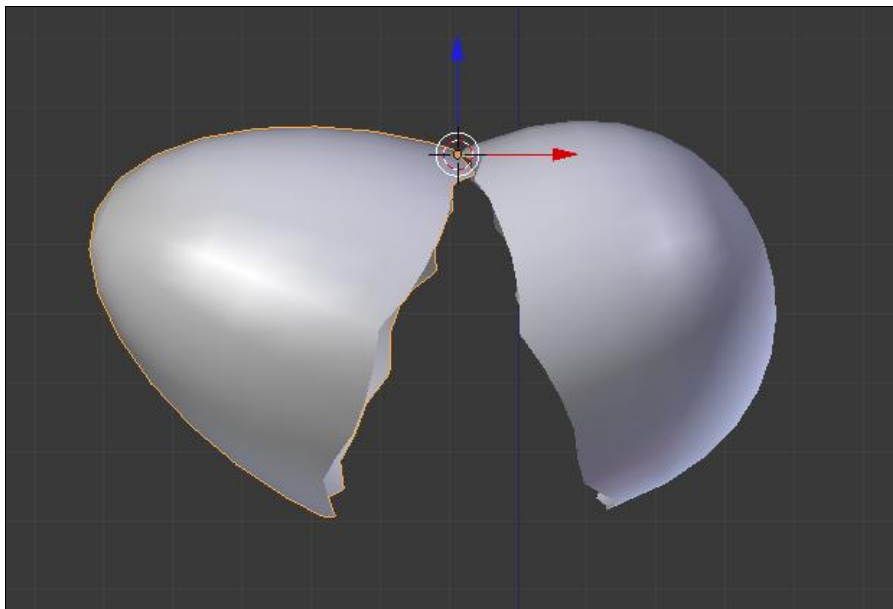


Рис. 5.87. Правильно настроенные центры объектов для анимации

Выделите объект **egg_left**, нажмите клавишу <I> и установите первый ключ **Rotation**. Следующий ключ будет находиться в 25 кадре. Нажмите <R> для разворота объекта и установите еще один ключ **Rotation**. Пропеделайте то же самое с объектом **egg_right**. В итоге у вас должна получиться анимация длительностью 25 кадров, где происходит раскрытие яйца. Нажмите <Alt>+<A> для проверки проигрывания анимации (см. рис. 5.87).

Теперь самое интересное. Нужно заставить вытекать жидкость из яйца на сковородку. Сначала поместите объект **egg_in** так, чтобы он полностью скрывался в закрытой скорлупе. Придайте ему овальную форму. Используйте для этого инструменты масштабирования и перемещения.

Для искажения формы содержимого яйца воспользуемся модификатором **Lattice**. Однако сначала добавьте в сцену сам объект **Lattice**. Зайдите в настройки **Lattice** и установите следующее разрешение по осям:

- ◆ **U** = 3
- ◆ **V** = 3
- ◆ **W** = 4

Большее количество ребер позволит более точно настроить анимацию.

Как вы уже знаете, с помощью изменения формы **Lattice** можно управлять формой примитива. Добавьте к объекту **egg_in** модификатор **Lattice**. В его настройках в поле **Object** выберите ранее добавленный объект.

Теперь, что вам нужно сделать. Отмасштабируйте объект **Lattice** так, чтобы он начинался от скорлупы, а заканчивался у поверхности сковороды. В режиме редактирования измените форму объекта так, чтобы он выглядел своеобразным туннелем

с расширением вначале и в конце. Для точной подстройки деформации вы можете передвигать **egg_in** по оси Z и корректировать его форму с помощью **Lattice** (рис. 5.88).

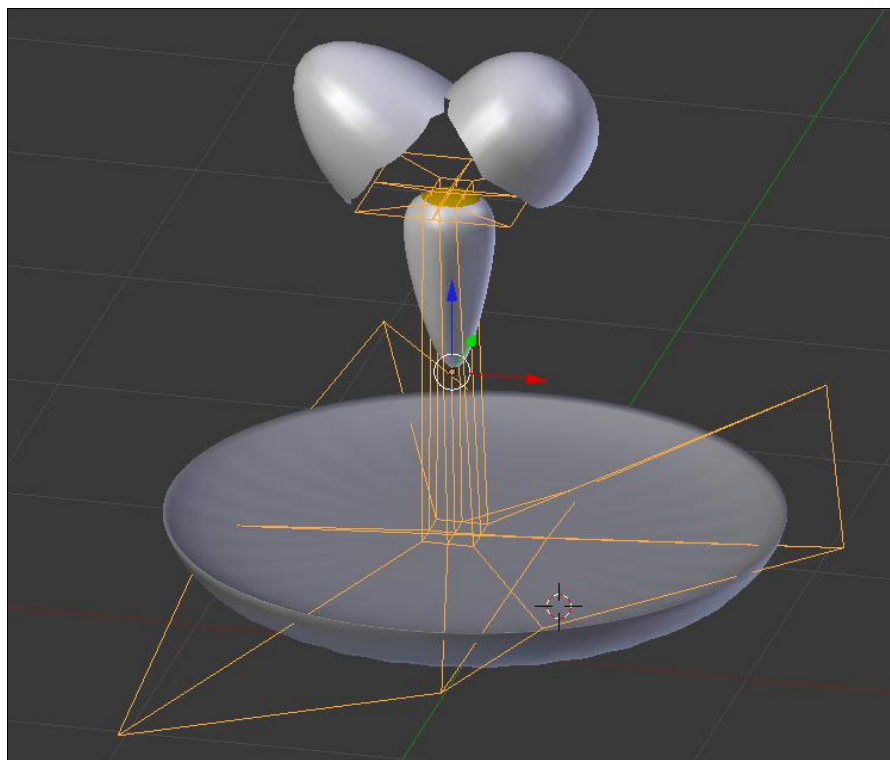


Рис. 5.88. Примерная форма **Lattice**

Пора расставлять ключи для **egg_in**. Переместите анимационный курсор в окне **Timeline** на позицию 1. Установите яйцо в верхней точке. Нажмите клавишу <I> для вызова анимационного меню, а в нем выберите пункт **Location**. Установите активным кадром 25, переместите объект на низшую позицию и добавьте второй ключ **Location**. Попробуйте проиграть анимацию для проверки (рис. 5.89).

Возможно, анимация в сцене будет выглядеть несинхронизированной или слишком медленной. Это можно исправить в **NLA Editor**.

Перейдите в раскладку окон **Animation**. Откройте окно **NLA Editor** вместо любого другого. Щелкните мышью по круглой кнопке рядом с каждым названием анимации для включения режима **NLA**. Вот теперь вы можете смело корректировать анимацию на свой вкус (рис. 5.90).

Конечно, с помощью **Lattice** удалось добиться весьма грубой анимации движения жидкости. Так как содержимое яйца достаточно вязкое по консистенции, то в этом случае использование **Lattice** оправданно, а вот смоделировать воду так уже не получится. Для этого придется использовать физику Blender, но это уже тема другой главы.

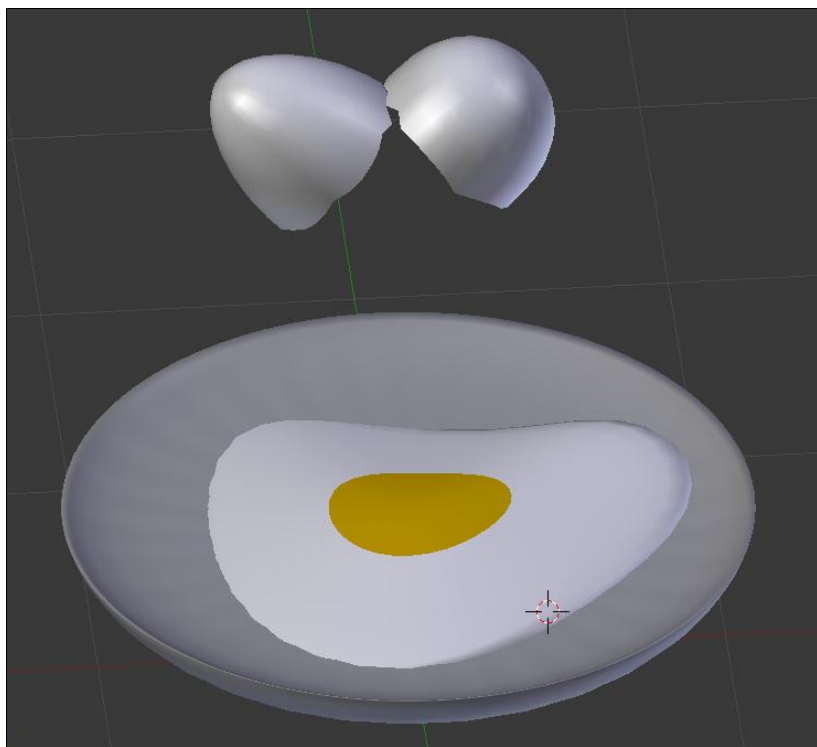
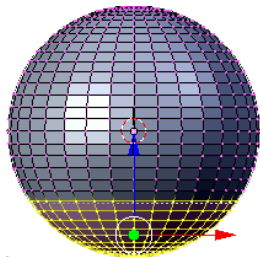


Рис. 5.89. Вид сцены в кадре 25



Рис. 5.90. Настройка NLA Editor

ГЛАВА 6



Физика

Все что нас окружает, подчиняется различным физическим законам: родник, бьющий из земли; огонь, пылающий в печи; пылинки, танцующие в воздухе. И если вы хотите создать действительно реальный мир, то без использования физики не обойтись.

Конечно, многое можно сделать в Blender без участия физических законов, заменяя их подсобными средствами. Так, например, практический урок предыдущей главы продемонстрировал использование модификатора **Lattice** для имитации движения жидкости. Но куда реалистичнее этот же пример выглядел бы с использованием физики Blender.

Но за все нужно расплачиваться. Используя физику, вы получите качественное поведение анимации в ущерб производительности. Время обработки даже простой сцены может быть очень длительным, и если есть возможность подмены реальной физики другими средствами, то, вероятно, лучше этим воспользоваться.

Есть и еще один минус использования физики. Как правило, настройки этих функций очень сложны и обширны. Малейшая неточность приведет к неправильному результату. Поэтому прочтите эту главу внимательно, повторяя все примеры на практике.

6.1. Физический мир Blender

Любая физическая функция в программе подчиняется гравитации, которая описывается в законе всемирного тяготения Ньютона. Не будем вдаваться в дебри фундаментальной физики. Важно знать, что ускорение притяжения предметов к Земле в реальном мире равно 9.8 м/с^2 . Это же значение установлено в Blender по умолчанию.

В качестве оси притяжения используется Z , т. е. объект с включенной физикой будет падать вниз по оси. Однако многие функции Blender имеют свое значение гравитации, которое является относительным по отношению к глобальному. В свою очередь глобальный параметр можно изменить на панели **Scene** окна **Properties**

(рис. 6.1). Изменение этого параметра будет касаться всех локальных **Gravity** в сцене, поэтому корректируйте его с осторожностью.

Физический мир Blender обширен. Вы можете с его помощью имитировать движение жидкостей, горение огня, дым, взрывы и многое другое. Все эффекты разбиты по тематическим группам в окне **Properties**. Основные настройки сконцентрированы на панели **Physics** (Физика) (рис. 6.2).



Рис. 6.1. Глобальная переменная **Gravity**

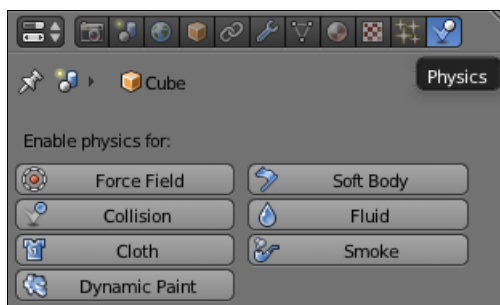


Рис. 6.2. Панель **Physics** окна **Properties**

- ◆ **Force Field** (Силовые поля). Здесь сконцентрированы функции, оказывающие динамичное воздействие на остальные физические объекты. К примеру, с их помощью можно имитировать дуновение ветра.
- ◆ **Collision** (Столкновение). Настройки поведения объектов при взаимодействии друг с другом. Например, каким образом будет вести себя ткань, положенная на обычный **Mesh**-объект.
- ◆ **Soft Body** (Мягкие тела). Этот инструмент можно использовать для создания тканей, волос, пружин. Он прекрасно взаимодействует с полями и столкновениями, поэтому оптимально подходит к имитации раскачивания чего-либо, например флага.
- ◆ **Cloth** (Ткань). Специальные настройки, оптимальные для создания одежды. Имеются заготовки различных тканей. Конечно, для этого можно использовать **Soft Body**, но **Cloth** предпочтительней.
- ◆ **Fluid** (Жидкость). Здесь вы найдете обширные настройки для имитации жидкостей. С помощью данного инструмента можно с легкостью создать: струйку воды, льющейся из крана; водяные брызги; тающий лед и многое другое.
- ◆ **Smoke** (Дым). Инструмент для имитации дыма, например от костра.
- ◆ **Dynamic Paint** (Динамичная краска). Уникальный инструмент, позволяющий создавать динамичные детали, такие как след краски от мазка кистью по стене, следы от шин в пыли, отпечатки сапог в снегу, рябь на воде.

Есть и еще одна панель физики в окне **Properties**, которая носит название **Particles** (Частицы) (рис. 6.3).

С них мы и начнем постигать столь сложный и прекрасный мир физики Blender.

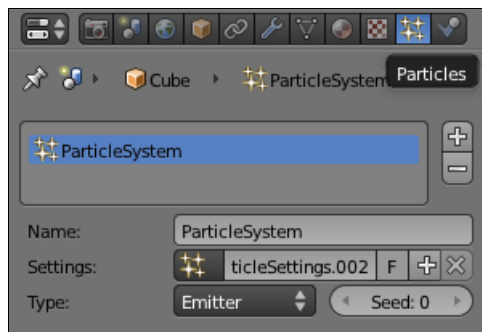


Рис. 6.3. Настройки **Particles** в окне **Properties**

6.2. Создание и настройка частиц

Частицы (**Particles**) представляют собой множество мелких объектов, подчиняющихся физическим законам. Они прекрасно могут взаимодействовать с другими системами физики Blender. С помощью частиц можно создавать как динамические объекты, так и статичные. Например: фейерверк, огонь, дым, фонтан, волосы, шерсть, мех и т. д.

Для того чтобы частицы могли функционировать, им нужен родительский объект, который называют эмиттером (**Emitter**). В качестве эмиттера могут служить примитивы **Mesh**.

Создать частицы очень легко. Нужно выделить **Mesh** в сцене, перейти на панель **Particles** окна **Properties** (см. рис. 6.3) и нажать кнопку с плюсом. Это действие создаст новую систему частиц. К одному и тому же объекту может быть присоединено сразу несколько систем частиц с разными настройками. Например, для создания огня понадобятся частицы, имитирующие пламя, и частицы, создающие дым.

Попробуйте добавить частицы к кубу, который содержится в сцене Blender. Уже по умолчанию они настроены на непрерывную генерацию и падение вниз под силой тяжести (рис. 6.4).

Частицы могут быть статичными и динамичными. Переключение режимов генерации осуществляется в меню **Type** (рис. 6.5).

Статичные (пункт меню **Hair**) оптимально подходят для создания волос, меха, травы. Слово "статичные" не значит, что они не могут двигаться. Например, волосы имеют одинаковую длину, но способны шевелиться под воздействием ветра.

Динамичные (пункт меню **Emitter**), наоборот, уже изначально участвуют в анимации. С их помощью можно создавать объекты, меняющие свою форму со временем: дождь, снег, огонь, дым, взрывы. Этот тип является активным по умолчанию.

В зависимости от выбранного типа изменяются все последующие настройки **Particle**. Сначала рассмотрим работу с динамичными частицами.

Самый главный параметр, точнее группа опций, — это **Emission** (Излучение). Именно здесь выполняется начальная настройка генератора частиц (рис. 6.6).

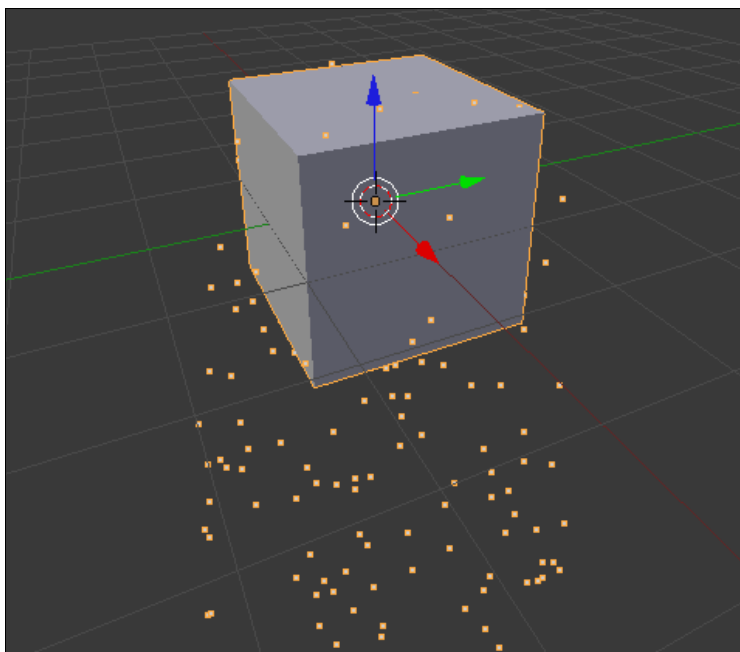


Рис. 6.4. Генерация частиц по умолчанию

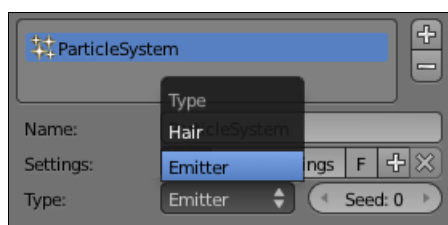
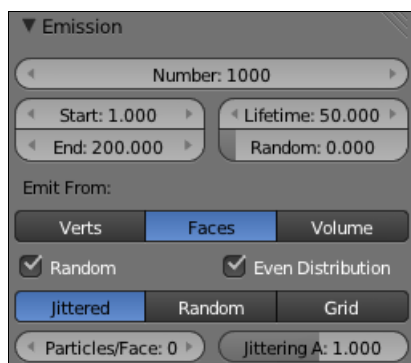


Рис. 6.5. Глобальные настройки системы частиц

Рис. 6.6. Закладка **Emission** — настройка генератора

Система динамических частиц может функционировать только определенное время, которое устанавливается в параметрах **Start** (Начало) и **End** (Конец). По умолчанию там установлены значения анимации сцены в целом.

Переведите анимационный курсор на нулевой кадр и нажмите <Alt>+<A>. Вы увидите, что частицы начинают генерироваться постепенно. В некоторых случаях это приемлемо, например сцена включения душа. Если вам понадобится изначально полный поток частиц, то сдвиньте значение параметра **Start** на несколько кадров назад. Так для сцены по умолчанию вы можете указать "-50".

Если вы отъедете от куба несколько дальше, то увидите, что движение частиц обрывается в некоторой точке и дальше они просто исчезают. Возникает ощущение,

что поток резанули острым ножом. Здесь действует еще одно очень важное свойство частиц — это время жизни (**Lifetime**). На рис. 6.6 поле **Lifetime** содержит значение 50. Это значит, что каждая испускаемая частица живет ровно 50 кадров. А так как частицы проходят за отведенное время жизни одно и то же расстояние, то и получается эффект резкого обрыва.

Но есть еще один параметр регулирования времени жизни — это **Random** (Случайный). По умолчанию он выключен и содержит 0. Измените его в большую сторону, чтобы варьировать случайным образом время жизни каждой частицы. Попробуйте установить в поле **Random** значение 1 и проиграть анимацию. Вот теперь исчезание частиц будет крайне неравномерным.

ЭТО ВАЖНО!

Настраивая частицы, приходится часто проверять их в действии путем запуска анимации. Желательно установить первым активным кадром в анимации значение поля **Start**, хотя бы временно. Дело в том, что если проигрывание анимации начинается с того же кадра, что и запуск частиц, то Blender автоматически сбрасывает по новой генератор частиц. В противном случае вы, возможно, не увидите результата от своих манипуляций с настройками частиц.

Поле **Number** (Число) указывает, сколько частиц будет создано в соответствии с параметрами **Start** и **End**. Будьте осторожны с ним, излишне большое значение приведет к медленной работе системы.

На закладке **Emission** присутствует еще одна группа **Emit From** (Излучать из), которая ответственна за выбор источников генерации. Дело в том, что частицы могут испускаться разными частями объекта. По умолчанию эмиттерами являются плоскости примитива, но вы можете использовать в качестве генераторов вершины объекта. За это отвечают кнопки **Verts** (Вершины) и **Faces** (Грани) (рис. 6.7).

В зависимости от выбранного типа источника становятся доступными дополнительные настройки. Так, для вершин имеется всего один параметр **Random**. С активной опцией частицы генерируются, произвольно выбирая вершины объекта в качестве эмиттеров. Если отключить параметр **Random**, то генерация частиц будет происходить поочередно из каждой вершины.

Совсем другой объем настроек появится, если выбрать тип источника **Faces** (см. рис. 6.6). Рассмотрим эти опции.

- ◆ **Random** (Случайный). Аналогично такой же опции, как и в случае с **Verts**.
- ◆ **Even Distribution** (Распределение). Включите эту опцию, если хотите, чтобы интенсивность испускания частиц зависела от размера конкретной плоскости.
- ◆ Кнопки **Jittered**, **Random**, **Grid** позволяют дополнительно настроить эмиттеры объекта.
 - **Jittered** (Дрожание). Включено по умолчанию. Указывает, каким образом каждая плоскость будет испускать частицы. Имеется два параметра:
 - **Particles/Face** — отвечает за количество источников на одной плоскости, установка значения в ноль включает автоматическое определение;
 - **Jittering Amount** (Значение дрожания) — качество смещения.

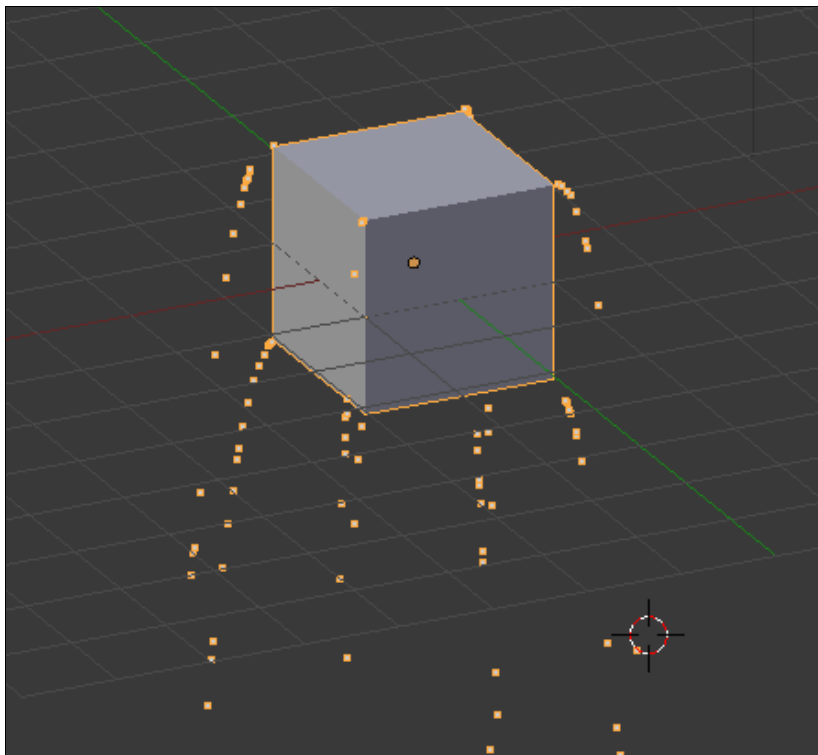


Рис. 6.7. Здесь вершины являются эмиттерами

- **Random** (Случайный). Источники определяются случайным способом. Дополнительные настроек нет.
- **Grid** (Сетка). Полигоны рассматриваются в виде сетки. Это позволяет получать интересные эффекты генерации. По умолчанию сетка прямоугольная, но с помощью опции **Hexagonal** ее можно сделать шестиугольной.

Закладка **Cache** (Кеш) предназначена для создания и использования заранее рассчитанных частиц. По умолчанию частицы генерируются в реальном времени. **Cache** позволяет сохранять результаты генерации в памяти компьютера или на жестком диске — это значительно экономит время при просчете всей сцены.

Если до этого мы рассматривали, как создавать частицы, то пришло время разобраться, как ими управлять и что они должны делать. По умолчанию частицы просто осыпаются по оси Z. Разумеется, Blender имеет средства для управления движением частиц, и они очень обширные.

Закладка **Velocity** (Скорость) содержит параметры, отвечающие за направление движения частиц (рис. 6.8).

Группа **Emitter Geometry** (Геометрия излучателя) позволяет задать начальную скорость генерации. Здесь всего два ключевых параметра: **Normal** (Нормаль) и **Tangent** (Тангенс). В первом случае частицы будут двигаться вдоль нормалей граней. Увеличение значения в **Normal** приведет к более высокой скорости (рис. 6.9).

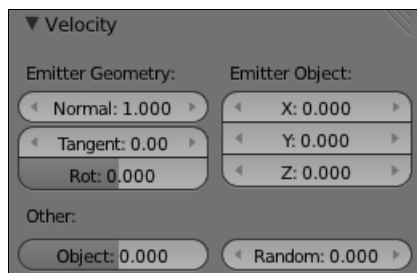
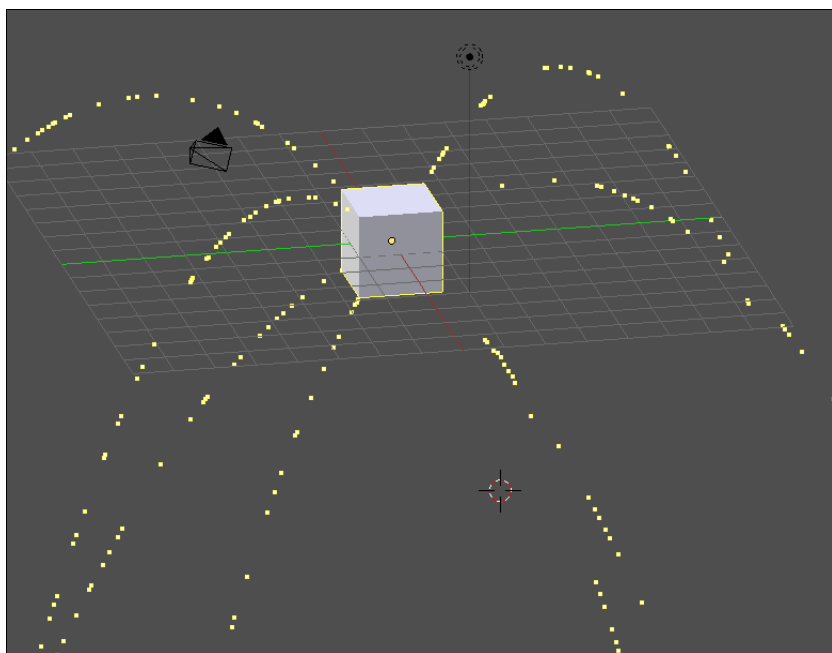


Рис. 6.8. Настройки движения

Рис. 6.9. Используйте **Normal** для получения эффекта фонтана

Во втором случае испускание частиц будет производиться под углом, причем наклон можно регулировать опцией **Rot**.

Группа **Emitter Object** (Излучатель) позволяет выбрать направление движения всех частиц, вне зависимости от расположения источников. Для этого имеются стандартные поля **XYZ** (рис. 6.10). Используйте вспомогательное поле **Random**, чтобы внести разнообразие в движение частиц.

Закладка **Rotation** (Вращение) содержит параметры для управления ротацией частиц (рис. 6.11).

Группа **Rotation Axis** (Оси вращения) позволяет выбрать из меню ось вращения каждой частицы относительно координаты **X**. Опция **Phase** (Фаза) отвечает за начальный угол поворота. Вы можете использовать **Random** для добавления случайности выборы.

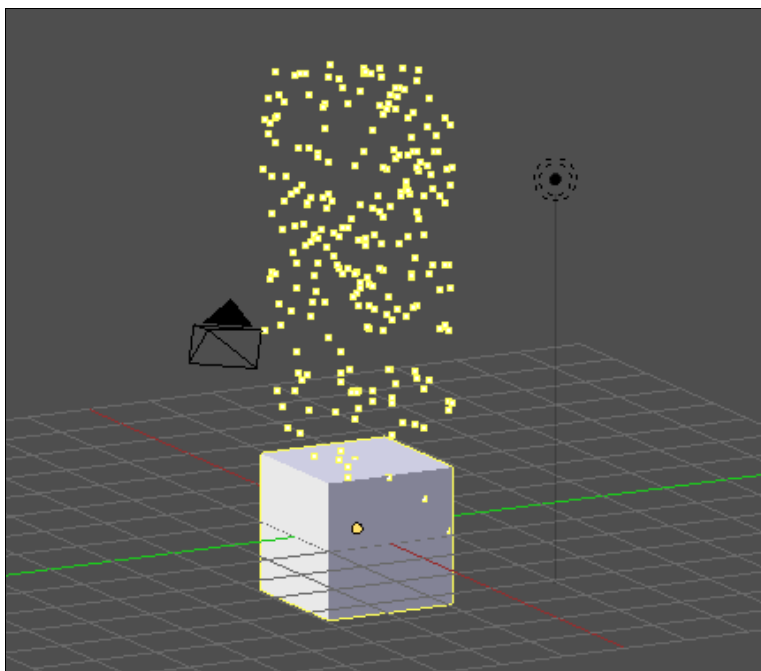


Рис. 6.10. Здесь частицы двигаются вдоль положительной оси Z

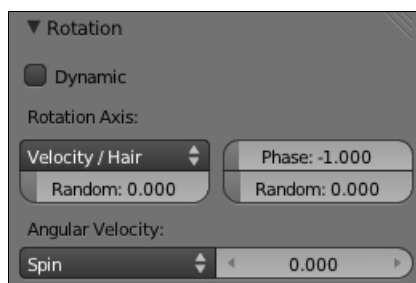


Рис. 6.11. Настройки вращения

Группа **Angular Velocity** (Угловая скорость) содержит настройки скорости вращения частиц. В меню три пункта:

- ◆ **None** (Ни одного) — вращения не будет;
- ◆ **Spin** (Вращение) — скорость вращения устанавливается рядом находящимся полем;
- ◆ **Random** (Случайный) — генерируется случайный вектор.

Очень интересна опция **Dynamic** (Динамика). По умолчанию она отключена, но если ее активировать, то частицы будут взаимодействовать между собой. Она не влияет на начальное положение элементов, а вот угловая скорость и вектор будут зависеть от столкновения частиц.

ЭТО ВАЖНО!

Не путайте эти две группы! **Rotation Axes** отвечает за то, под каким углом частицы будут генерироваться, а **Angular Velocity** устанавливает дальнейшее вращение частиц.

Следующая закладка **Physics** позволяет выбрать алгоритм, который будет управлять поведением частиц (рис. 6.12).

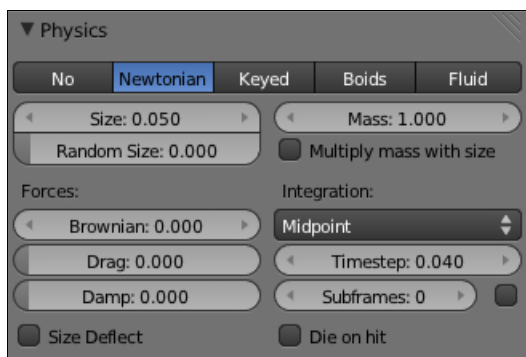


Рис. 6.12. Физика частиц

Очень сложная и насыщенная группа опций частиц. Рассматривать все их просто нецелесообразно. В дальнейшем некоторые из них будут рассмотрены на практических примерах. А пока коснемся главных особенностей.

Группа кнопок позволяет выбрать главный алгоритм физики частиц:

- ◆ **None** (Ни одного) — физика движения полностью отсутствует и происходит только генерация частиц;
- ◆ **Newtonian** (Ньютонов) — кнопка активна по умолчанию. Движение частиц осуществляется в соответствии с физическими законами;
- ◆ **Keyed** (Ключевая) — текущая система частиц будет взаимодействовать с другими системами, что позволяет создавать цепочки действий. Это подойдет, к примеру, для анимации движения волокон в ткацком станке;
- ◆ **Boids** (Имитация) — позволяет создавать своего рода искусственный интеллект, который характерен для группового поведения. Например, с его помощью можно анимировать стаю птиц, рыбный косяк или рой пчел;
- ◆ **Fluid** (Жидкость) — имеет настройки для генерации движения частиц по законам жидкостей.

Вне зависимости от выбранного алгоритма закладка **Physics** имеет общие настройки:

- ◆ **Size** (Размер) — размер частиц;
- ◆ **Random Size** (Случайный размер) — вариация размера относительно указанного в поле **Size**;
- ◆ **Mass** (Масса) — вес частиц. Этот параметр имеют все алгоритмы, кроме **None**;

- ♦ **Multiply mass** (Множественная масса) — изменение веса частицы происходит в соответствии с ее размерами. При этом берется за эталон значение опции **Mass**.

Система частиц имеет большое количество нерассмотренных настроек. С некоторыми из них вы познакомитесь в остальных разделах книги, а что-то придется искать в официальной документации Blender, но главная часть генерации и поведения освещена. Тем не менее хотелось бы вкратце остановиться еще на двух часто используемых группах опций — это **Display** (Экран) и **Render** (Рендер, Обработка) (рис. 6.13).

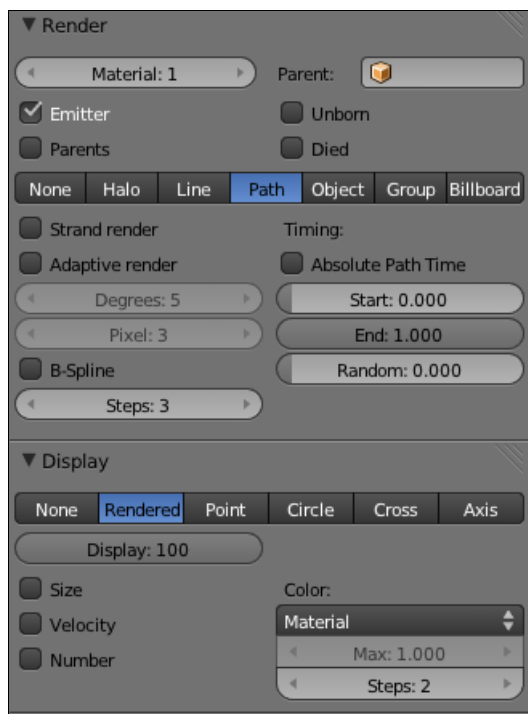


Рис. 6.13. Настройки визуализации частиц

Закладка **Display** (Экран) позволяет настроить внешний вид частиц в окне **3D View**. Это не будет влиять на результат рендера, а служит лишь вспомогательным средством контроля. Так группа кнопок **None** (Ни одного), **Rendered** (Обработанный), **Point** (Точка), **Circle** (Окружность), **Cross** (Крест), **Axis** (Оси) позволяет выбрать внешний вид частиц от нулевого (их вообще не видно в окне) до результативного (кнопка **Rendered** покажет частицы так, как они будут выглядеть после рендера). Кроме того, можно использовать опции **Size** (Размер), **Velocity** (Скорость), **Number** (Число), которые, соответственно, выведут дополнительную информацию к каждому элементу системы о ее размере, направлении и номеру (рис. 6.14).

Закладка **Render** (Обработка) отвечает за внешний вид частиц при окончательной визуализации сцены. По умолчанию объект-эмиттер будет присутствовать в рендере картинки, но можно отключить его с помощью опции **Emitter**.

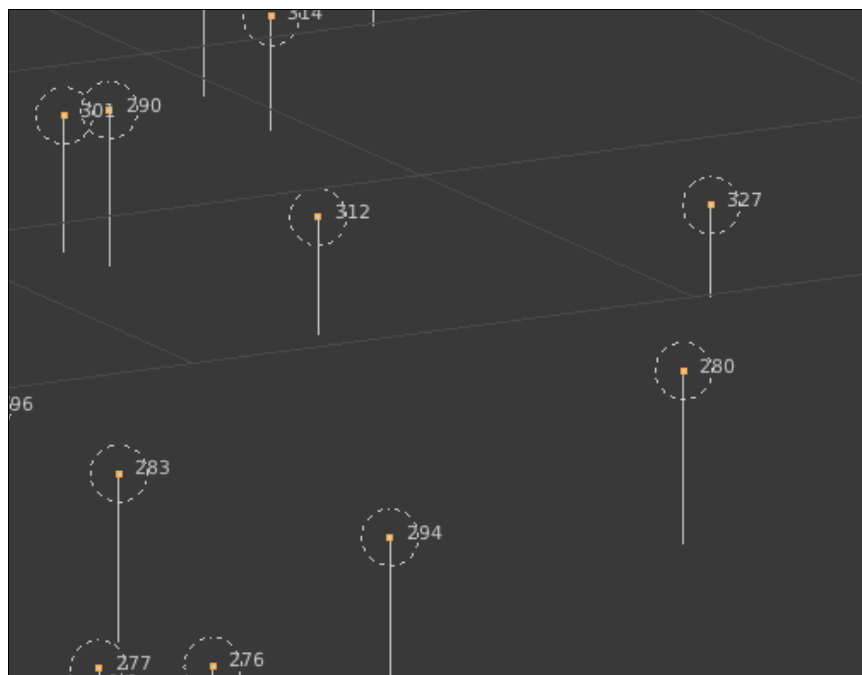


Рис. 6.14. Результат включения **Size**, **Velocity**, **Number**

А вот с помощью имеющейся группы кнопок в данной закладке можно выбрать, что будет присвоено каждой частице при обработке. По умолчанию рендер частиц будет осуществляться в виде **Halo**, но можно установить другой объект. Рассмотрим, как это можно сделать.

Добавьте в сцену обычную сферу. Отключите опцию **Emitter**, чтобы вывод куба не мешал просмотру. Нажмите кнопку **Object** в настройках **Render** (рис. 6.15).

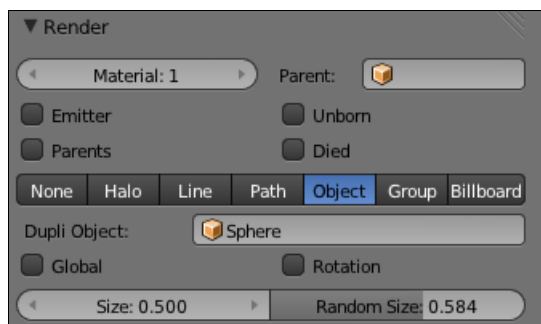


Рис. 6.15. Настройки **Object**

Появившееся поле **Dupli Object** позволяет выбрать из списка любой объект в сцене. Выберите сферу. Установите параметр **Size** (Размер) равным 0.5, а **Random Size** (Случайный размер) равным 0.584. Нажмите клавишу <F12> для рендера картинки (рис. 6.16).

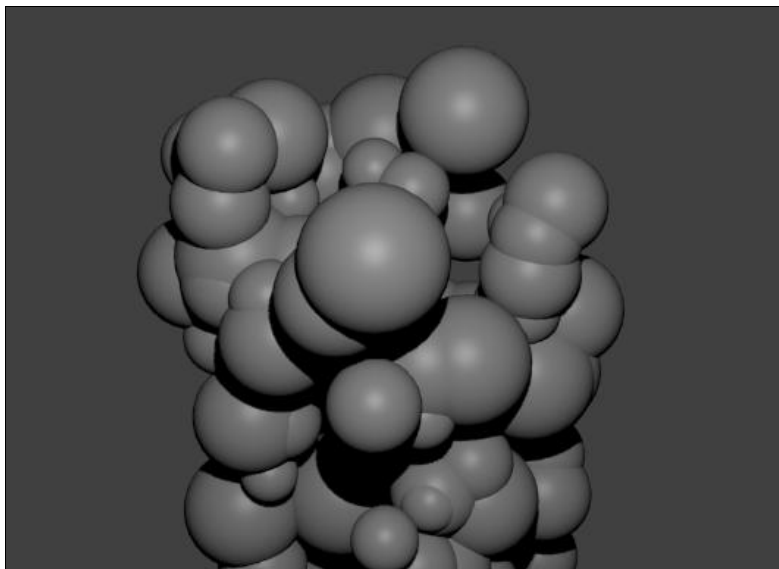


Рис. 6.16. Прimitives сферы используется как эталон для частиц

6.3. Моделирование волос и меха

Системы частиц оптимально подходят для создания большого количества мелких деталей, например шерсти. Мало того, Blender предлагает специальный режим генерации частиц **Hair** (Волосы), который имеет соответствующие настройки.

Рассмотрим работу с **Hair** на простом примере. Добавьте сферу в сцену. Она будет служить эмиттером для частиц. Создайте для нее систему частиц, нажав кнопку "+" на панели **Particles**.

Выберите в меню **Type** пункт **Hair** (см. рис. 6.5). Внешний вид и поведение частиц тут же изменится. Первое, что бросается в глаза, — это неработающая анимация. Все правильно, ведь у данного типа частиц нет полей **Start**, **End** или **Lifetime** (рис. 6.17).

Зато появился новый параметр **Hair Length** (Длина волос). Собственно говоря, он и отвечает за длину частиц. Только имеющееся значение по умолчанию явно слишком большое, измените его на 0.2 (рис. 6.18).

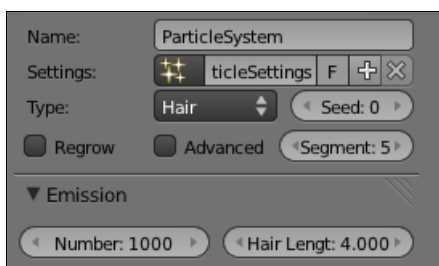


Рис. 6.17. Настройки генерации частиц в режиме **Hair**

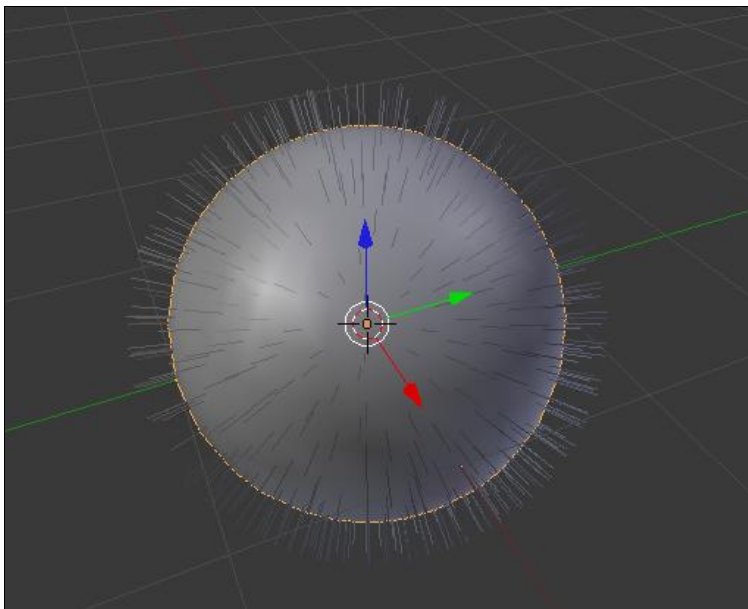


Рис. 6.18. Частицы с уменьшенной длиной

Если вы попытаете обработать сцену так, как она есть, то получите нечто похожее на свернувшегося малость облысевшего ежа. Конечно, можно увеличить количество частиц в поле **Number**, но есть вариант поизящней — это использование вторичных частиц (**Children**).

Рассмотрим особенности использования **Children**-частиц.

- ♦ Вторичные частицы наследуют все признаки, материалы своего родителя.
- ♦ Для визуализации **Children** не требуется перерасчета физики, т. к. они основаны на родительских частицах.
- ♦ В дополнение к основным элементам вторичные могут прибавить свои настройки, что позволяет получать интересные эффекты.

В настройках **Particles** имеется закладка, которая так и называется **Children** (рис. 6.19). По умолчанию эта функция отключена (активна кнопка **None**).

Нажмите кнопку **Simple**, чтобы создать вторичные частицы (рис. 6.20).

По результату отчетливо видно, что частиц получилось слишком много, и они буквально наслаиваются друг на друга. Не беда, исправить это легко.

В настройках **Children** имеется два параметра: **Display** (Экран) и **Render** (Обработка). Оба отвечают за количество вторичных частиц относительно родительских. Только первый служит для визуализации частиц в окне **3D View**, а второй — рендера. Установите в обоих полях значение 20.

Если вы еще раз обработаете сцену, то увидите, что хотя количество элементов было уменьшено, результат далек от идеала и выглядит слишком грубо. Проблема заключается в настройках рендера программы.



Рис. 6.19. Настройки вторичных частиц

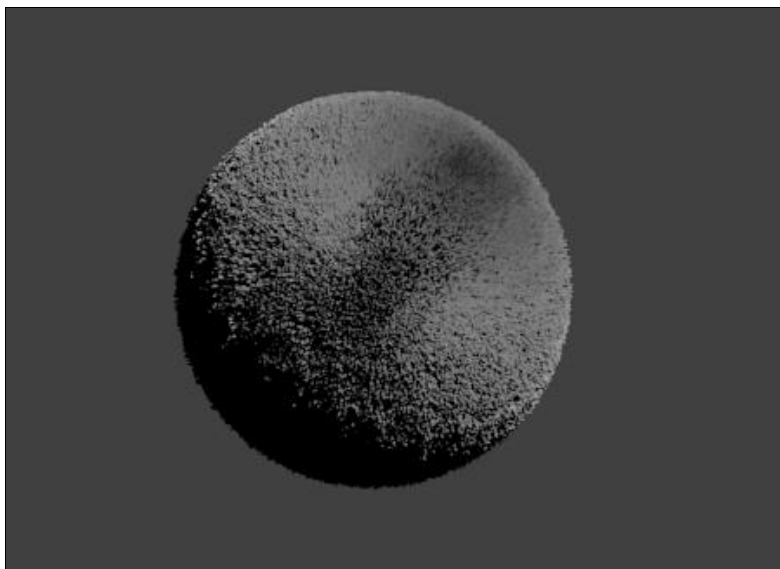


Рис. 6.20. Результат рендера с использованием Children

По умолчанию Blender обрабатывает материал объекта (а значит, и частиц) так, как он есть. Но разработчики предусмотрели специальные настройки материала **Strand**, которые предназначены как раз для визуализации меха или волос.

Откройте закладку **Render** панели **Particles** и включите опцию **Strand render** (рис. 6.21).

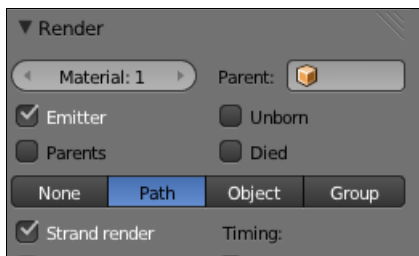


Рис. 6.21. Закладка **Render** для режима **Hair**

Это заставит программу обрабатывать частицы в соответствии с правилами **Strand** (рис. 6.22).

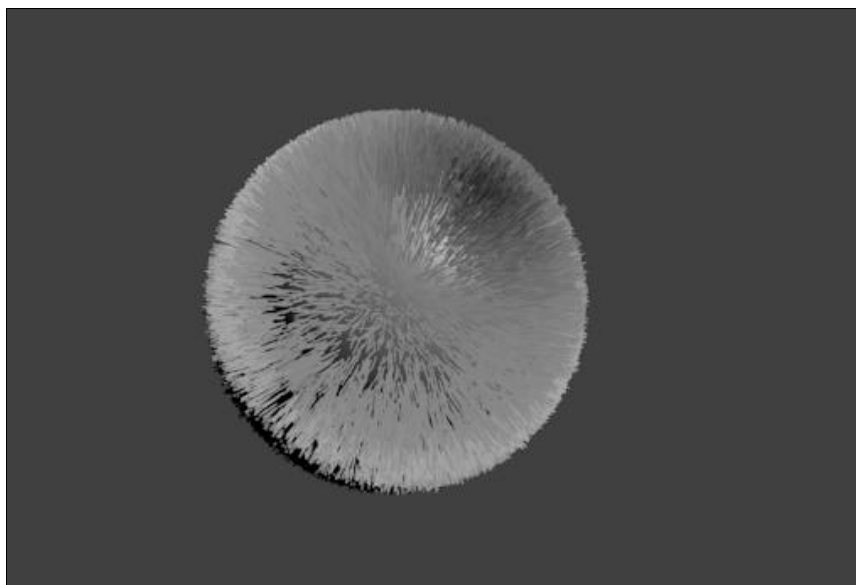


Рис. 6.22. Результат обработки сферы с параметром **Strand**

Наверное, вы уже догадались, как с помощью частиц генерировать волосы. Действительно, что может быть проще, увеличили длину элементов, добавили смещение в параметрах **Velocity**, **Rotation**, **Physics**, и все готово. Но не тут-то было, хотя бы потому, что этих групп в настройках **Hair** не имеется.

Взамен этого разработчики программы предлагают волшебную возможность творить прически непосредственно в окне **3D View**. Опять-таки рассмотрим создание волос с помощью несложного примера.

Опыты будем проводить на примитиве **Monkey**, который расположен в группе **Mesh** меню **Add**. Создайте новый проект, удалите куб и добавьте обезьянку.

Для удобства работы требуется объект развернуть лицевой частью к камере. Нажмите клавишу <N> и введите значение 90 для параметра **Rotation X**. Добавьте для красоты сглаживание (**Smooth**) и модификатор **Subdivision Surface** (рис. 6.23).

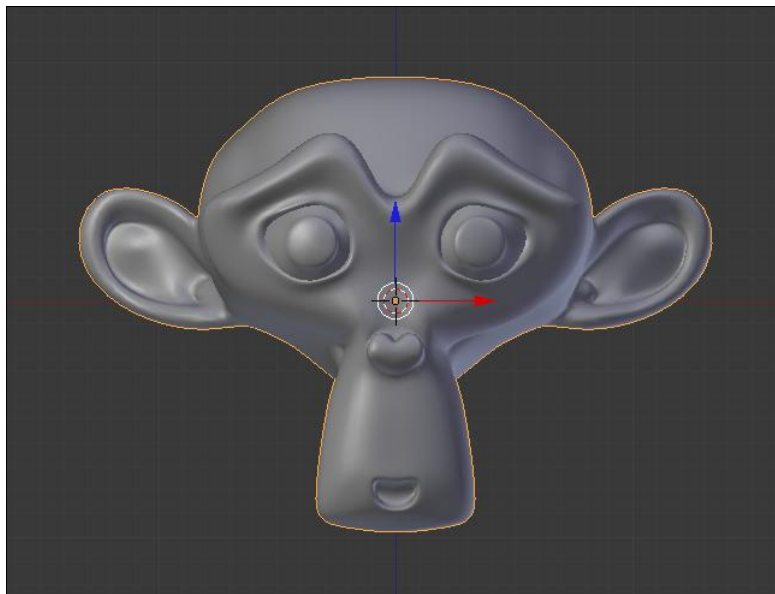


Рис. 6.23. Подопытный образец

Если сейчас прибавить к объекту систему **Particle**, то частицы покроют всю модель, что совсем не нужно. Необходимо отвести часть элементов на голове **Monkey** для создания группы. Выделите их, как на рис. 6.24.

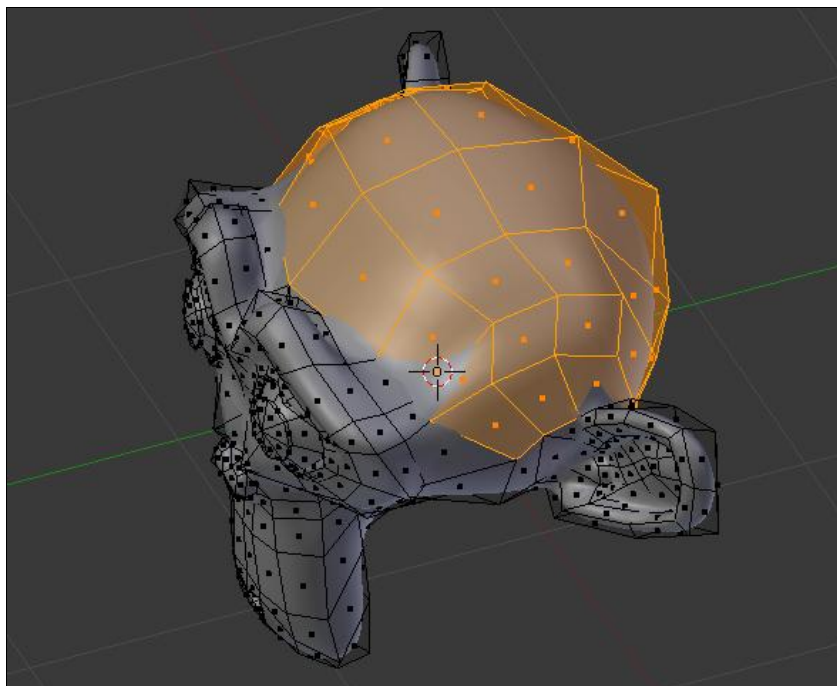


Рис. 6.24. Выделение полигонов для группы

Необходимо создать группу вершин под выделение. Это можно сделать на панели **Object Data** окна **Properties**. Откройте закладку **Vertex Groups**, нажмите кнопку с изображением плюса, а затем **Assign** (рис. 6.25).



Рис. 6.25. Создание группы вершин

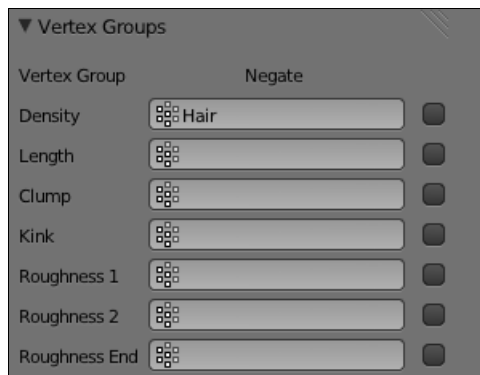


Рис. 6.26. Использование групп вершин для частиц

Вот теперь можно добавить к объекту систему частиц. Сделайте это и включите режим **Hair** с помощью меню **Type**.

По умолчанию частицы будут генерироваться всеми гранями объекта. Чтобы задействовать группу вершин, откройте закладку **Vertex Groups** в настройках системы частиц и в поле **Density** выберите группу вершин (рис. 6.26).

На рис. 6.26 видно, что частицы могут задействовать сразу несколько групп вершин. Это сделано для привязки к имеющимся настройкам вторичных элементов **Children**. Если вы не хотите, чтобы какая-нибудь группа участвовала в процессе генерации, то установите галочку в поле напротив имени группы.

Откройте закладку **Emission** и установите длину частиц в поле **Hair Length**, равную 0.5.

Теперь нужно активировать генерацию **Children**. Включите в данной группе кнопку **Simple**, а в параметрах **Display** и **Render** введите значение 10. В настройках **Render** активируйте опцию **Strand Render** (рис. 6.27).

А вот теперь обещанное волшебство. Откройте меню **Mode** окна **3D View**, и вы увидите новый пункт **Particle Mode** (Режим частиц) (примитив с частицами должен быть выделен). Этот режим позволяет в реальном времени с помощью специальных кистей управлять укладкой частиц на объекте.

Обратите внимание на панель **Tool Shelf**, которая содержит все необходимые инструменты (рис. 6.28).

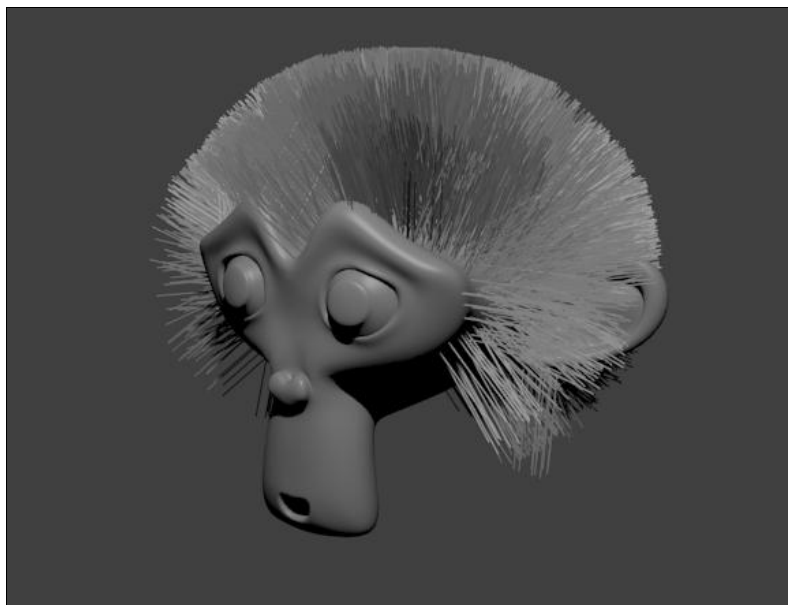


Рис. 6.27. Monkey явно нуждается в стрижке

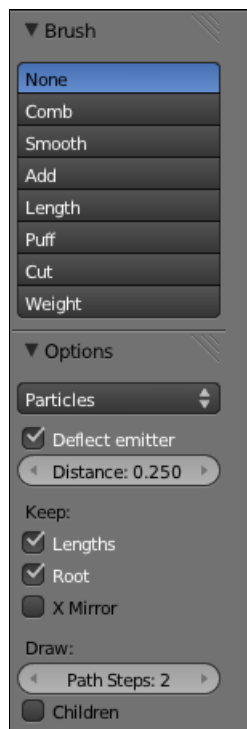


Рис. 6.28. Панель Tool Shelf в режиме Particle Mode

Принцип работы простой. Выбираете нужную кисть на закладке **Brush** и редактируете частицы. Всего имеется 7 вариантов:

- ◆ **Comb** (Расческа) — эффект равноценен действию реальной расчески. Частицы будут укладываться в сторону движения этой кисти;
- ◆ **Smooth** (Сглаживание) — эта кисть позволяет выровнять изгибы волос;
- ◆ **Add** (Добавить) — добавление новых частиц;
- ◆ **Length** (Длина) — управление длиной волос;
- ◆ **Puff** (Стяжка) — используйте ее, если хотите, чтобы волосы располагались в соответствии с нормалью поверхности;
- ◆ **Cut** (Вырезать) — удаление частиц;
- ◆ **Weight** (Вес) — используется при анимации мягких тел. Вес 1 — волосы не движутся, вес 0 — полностью подвержены анимации.

Набор небольшой, но вполне достаточный.

Вы, наверное, уже заметили, что частицы в этом режиме выглядят как-то не так. Дело в том, что по умолчанию выключен вывод вторичных элементов **Children**. При необходимости можете включить опцию **Children** на панели **Tool Shelf** (см. рис. 6.28).

Попробуйте сделать свою уникальную прическу. Выберите кисть **Comb** и причешите обезьянку. Проще всего это сделать в режиме просмотра **Top View** (<NumPad 7>). Вы можете изменить размер кисти в параметре **Radius** и силу нажатия в поле **Strength**.

Проведите кистью по волосам **Monkey** над любым ухом. Вы увидите, что кисть воздействует только на верхний слой частиц. Если это вызывает неудобство, то отключите кнопку **Limit Selection**, расположенную в заголовке окна **3D View** (рис. 6.29).

В заключение этого раздела рассмотрим способ подключения материала к системе частиц. По умолчанию частицы используют материал объекта-эмиттера. Это не всегда бывает удобно.

Создайте материал для **Monkey**. Он будет являться основным для объекта. Теперь в окне мультиматериалов добавьте слот и создайте еще один материал. Измените его цвет на черный.

Откройте настройки частиц и найдите поле **Material** в группе **Render**. По умолчанию там находится единица. Поменяйте ее на двойку и обработайте результат (рис. 6.30).



Рис. 6.29.
Кнопка
Limit Selection



Рис. 6.30. Прическа для **Monkey**

6.4. Работа с *Soft Body*

Физика *мягких тел*, как еще называют **Soft Body**, находит широкое применение при создании анимации в Blender. Это краеугольный камень многих инструментов программы. С помощью **Soft Body** вы сможете создать качественную анимацию дви-

жения тканей, раскачивание подвесного моста, колебание веревки и многое другое. Практически все, что окружает нас в реальном мире, может быть имитировано с помощью **Soft Body**. Но хватит разговоров, пора познакомиться с этим зверем поближе.

Как и положено, настройки **Soft Body** обширные, загадочные и непостижимые, но только на первый взгляд. Попробуем физику в действии.

Создайте новый проект, удалите куб и добавьте примитив **Plane**. Сделайте его прямоугольным. Для физики очень важно качество структуры модели. Войдите в режим редактирования и, используя инструмент **Subdivide**, разбейте **Plane** несколько раз.

Откройте панель **Physics** в окне **Properties** и нажмите кнопку **Soft Body** (см. рис. 6.2). Ничего не меняйте в настройках. Попробуйте проиграть анимацию в окне **3D View**. Вы увидите, что плоскость начала плавно раскачиваться по вертикали.

Это, конечно, не смахивает на демонстрацию внушительных возможностей **Soft Body**. Давайте немного улучшим результат.

Представьте, что произойдет, если углы этой плоскости прибить виртуальными гвоздями. Правильно, получится гамак, батут и все в этом роде.

Роль гвоздей будут играть угловые вершины, объединенные в одну группу. Выделите их в режиме редактирования и откройте закладку **Object Data**. Найдите опции **Vertex Groups** (Группы вершин), нажмите кнопку с плюсом, а затем кнопку **Assign**. Вернитесь в настройки **Soft Body** и найдите закладку **Soft Body Goal** (Целевой Soft Body) (рис. 6.31).



Рис. 6.31. Настройки **Soft Body Goal**

Слово "goal" переводится как "цель". Эти опции можно рассматривать, как своего рода ограничители, привязанные к определенной группе. Выберите свою группу в поле **Vertex Group**. Нажмите <Alt>+<A> для проигрывания анимации (рис. 6.32).

Вы увидите, что плоскость совершает все те же колебательные вертикальные движения, но углы уже неподвижны. Это выглядит более похожим на гамак, но движение его слишком заторможенное.

Soft Body имеет глобальные параметры, которые определяют поведение объекта. Они располагаются в закладке с одноименным именем (рис. 6.33).

Рассмотрим их:

- ◆ **Friction** (Трение) — настройка вязкости вершин. Чем больше значение этого параметра, тем более вязкой становится среда;

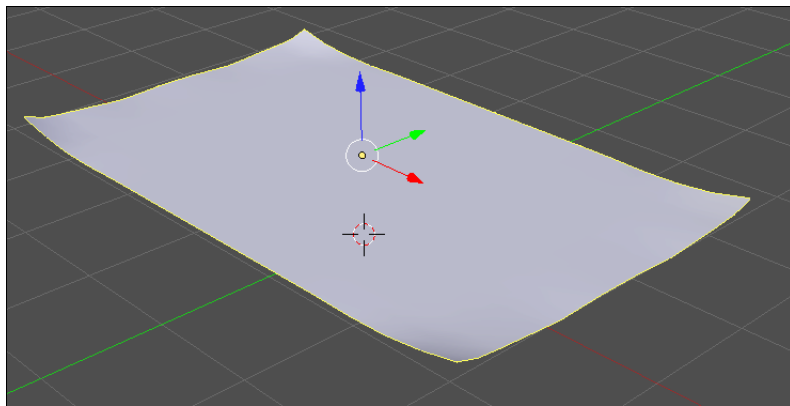


Рис. 6.32. Плоскость, припиленная по углам, стала больше похожа на гамак

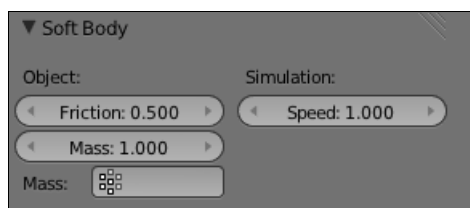


Рис. 6.33. Главные настройки **Soft Body**

- ◆ **Mass** (Масса) — чем больше значение, тем более инертным становится объект;
- ◆ **Speed** (Скорость) — скорость выполнения симуляции.

В этой группе имеется еще один параметр **Mass**, который позволяет выбрать из меню группу вершин объекта. Если вы это сейчас сделаете, то не получите никакого видимого эффекта. Дело в том, что данная опция позволяет в имитации использовать вес вершины группы, но так как они по умолчанию все одинаковые, то не будет и эффекта. По желанию, вы можете изменить вес вершин с помощью режима **Weight Paint** окна **3D View**.

Давайте опробуем действие этих настроек на практике. Введите значение 10 в поле **Mass** и запустите анимацию (рис. 6.34).

Вы увидите, что **Plane** выполняет мощные колебания и практически без затухания. А теперь попробуйте изменить параметр **Friction**. Установите в нем значение 15 и еще раз проиграйте анимацию. В этот раз плоскость значительно быстрее успокоится.

ЭТО ВАЖНО!

Как и система частиц, **Soft Body** требует для корректного отображения измененных настроек запуск с самого первого кадра анимации.

Усложним задачу. Добавим в сцену сферу и заставим прыгать ее на батуте. Причем сфера должна вести себя, как обычный упругий мяч, т. е. деформироваться от удара о поверхность.

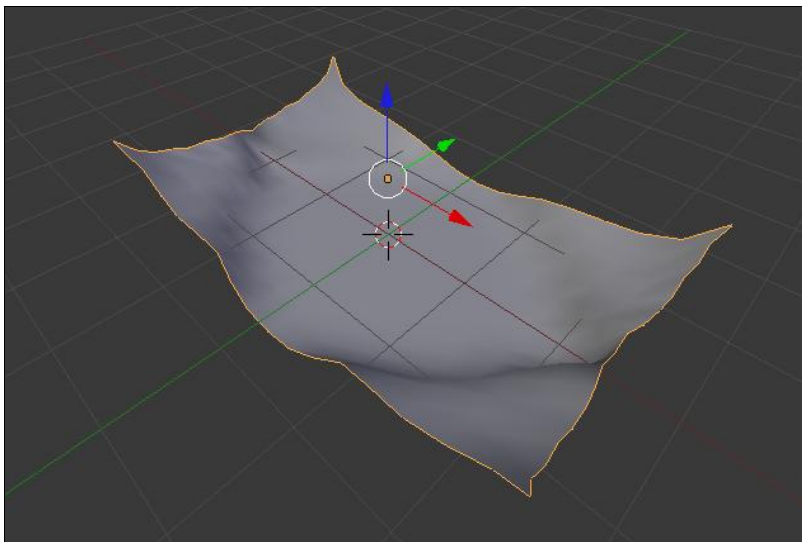


Рис. 6.34. С увеличенной массой плоскость совершает более сильные движения

Добавьте в сцену примитив **UV Sphere** и расположите ее над плоскостью. Включите **Soft Body**. Наш мячик будет иметь другие параметры трения и массы, в отличие от батута. Введите следующие значения:

◆ **Friction** = 0.1

◆ **Mass** = 0.5

Так как сфера должна совершать свободное падение на плоскость, то необходимо отключить группу опций **Soft Body Goal**, которые выполняют привязку объекта в пространстве, и включить группу **Soft Body Edges** (Ребра Soft Body) (рис. 6.35).

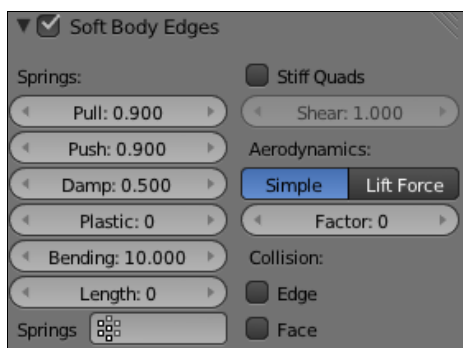


Рис. 6.35. Настройки **Soft Body Edges**

Эти опции позволяют управлять внутренней деформацией объекта при столкновении с другими объектами:

◆ **Pull** (Напряжение) — настройка растяжения ребер. Чем меньше значение, тем более эластичный материал;

- ◆ **Push** (Упругость) — сопротивление объекта при попытке его сжатия. Минимальное значение характерно для тканей;
- ◆ **Damp** (Дампинг) — сила трения для ребер. Чем меньше значение, тем больше трение;
- ◆ **Plastic** (Пластик) — используйте этот параметр, если хотите, чтобы сохранялась деформация объекта после столкновения;
- ◆ **Bending** (Изгиб) — нечто наподобие **Damp**, но создает связи между вершинами по диагонали;
- ◆ **Length** (Длина) — что будет происходить с объектом. Если установить слишком маленькую цифру, то произойдет разрыв ребер. Значение в 0 отключает данную функцию, значение в 100 не оказывает влияния.

Установите следующие значения для данных опций:

- | | |
|---------------------|-----------------------|
| ◆ Pull = 0.9 | ◆ Plastic = 0 |
| ◆ Push = 0.9 | ◆ Bending = 10 |
| ◆ Damp = 0.5 | ◆ Length = 0 |

Если вы попробуете проиграть анимацию, то увидите, что сфера пролетает сквозь **Plane**. Все правильно, ведь плоскость не имеет возможности просчитывать столкновения. Для этого нужно включить кнопку **Collision** вверху панели **Physics** (см. рис. 6.2). Вот теперь физика будет работать должным образом (рис. 6.36).

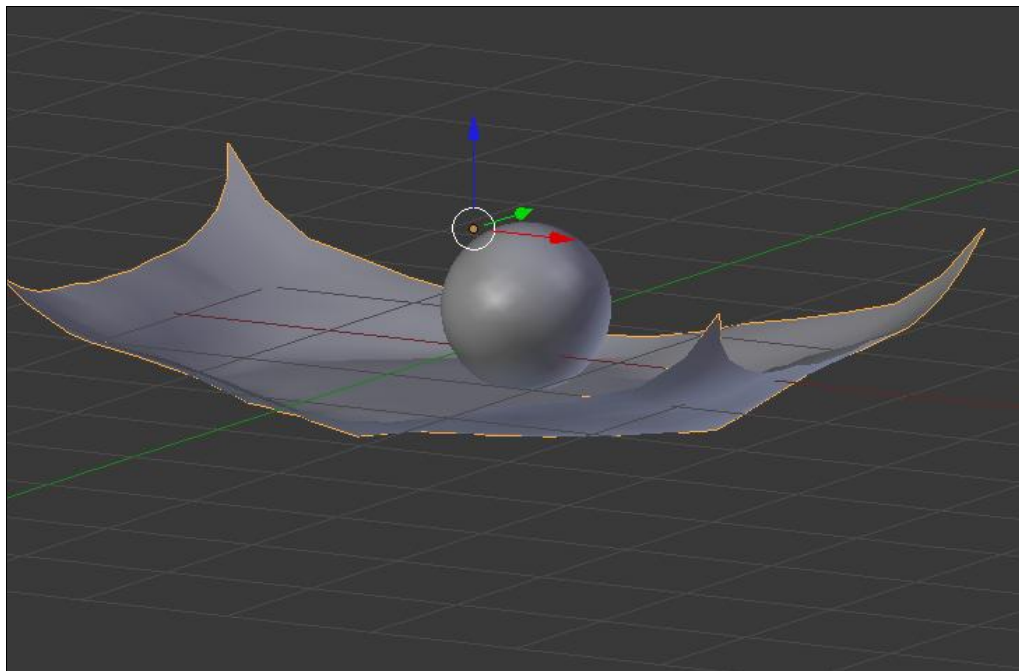


Рис. 6.36. Кадр из анимации **Soft Body**

СОВЕТ

Вероятно, вы столкнетесь с тем, что анимация физики выполняется слишком медленно. Можно воспользоваться функцией **Cache** для создания предварительного просчета. Откройте закладку **Soft Body Cache** (Кеширование Soft Body) у любого объекта, участвующего в анимации. Найдите в этой группе кнопку **Bake All Dynamics** (Выпечка всей динамики) и нажмите ее. Запаситесь терпением. После некоторого времени кеш будет создан. Теперь вы можете просматривать анимацию физики с приемлемой скоростью. Для сброса кеша рядом имеется кнопка **Free All Bakes** (Сбросить всю выпечку). Это может понадобиться, если вы решите редактировать параметры физики объектов.

6.5. Создание ткани

Для создания имитации движения ткани можно использовать **Soft Body**, но лучше воспользоваться специальным подразделом **Cloth** (Ткань), который располагается на той же панели **Physics** (см. рис. 6.2).

Откройте новый проект, добавьте в сцену примитив **Plane** и расположите его над кубом. Плоскость будет играть роль ткани, которая при падении должна будет повиснуть на кубе. Используя масштабирование, увеличьте размер **Plane**. Разбейте структуру куба несколько раз при помощи функции **Subdivide**.

Выделите плоскость и включите для нее подраздел физики **Cloth** (рис. 6.37).

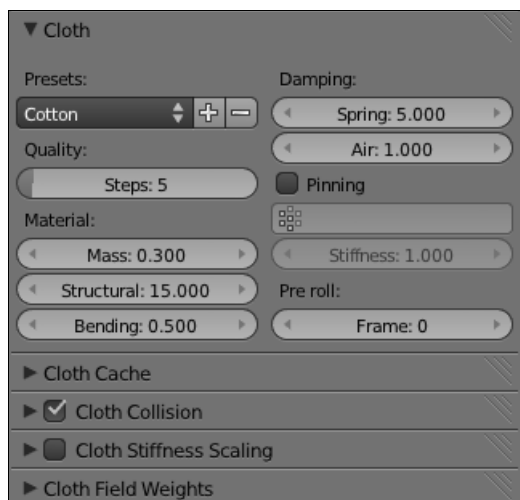


Рис. 6.37. Настройки Cloth

Главное достоинство раздела **Cloth** в том, что он предназначен исключительно только для имитации физики тканей, поэтому настроек здесь мало.

По умолчанию **Cloth** предлагает заготовки параметров нескольких типов тканей. Если ни один из них вам не подходит, то можно выполнить самостоятельную настройку и сохранить ее, как отдельный блок данных.

Заготовки располагаются в меню **Presets** (Заготовки):

- ◆ **Cotton** (Хлопок);
- ◆ **Denim** (Джинсы);
- ◆ **Leather** (Кожа);
- ◆ **Rubber** (Каучук);
- ◆ **Silk** (Шелк).

Выберите любой из пресетов для плоскости. Хотя настройка ткани завершена, не стоит нажимать кнопку **Play** в окне **Timeline** — ткань пролетит сквозь куб. Все правильно, к кубу нужно добавить физику столкновений. Нажмите кнопку **Collision** в настройках физики примитива **Cube** (рис. 6.38).

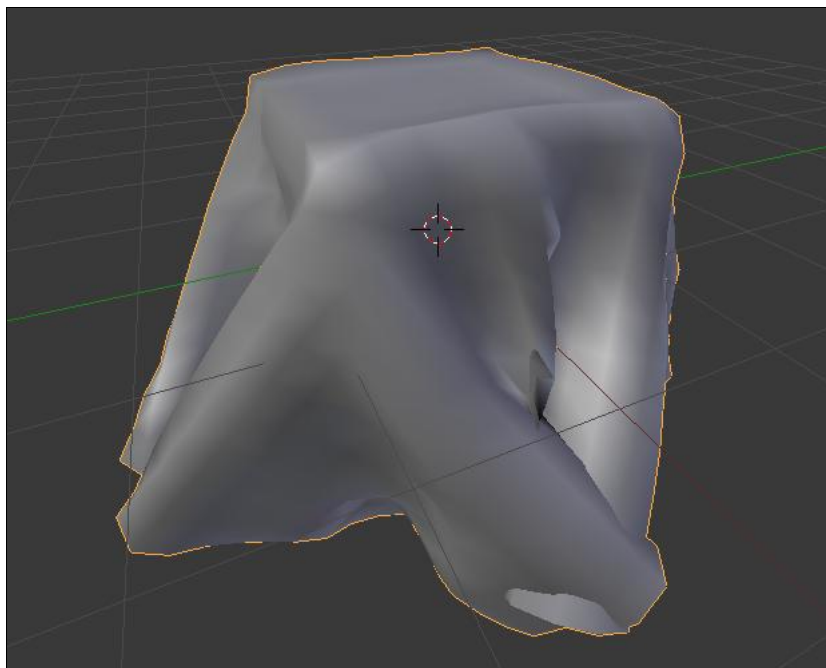


Рис. 6.38. Ткань при падении облегает куб

Хотя имеющихся заготовок вполне достаточно, но, возможно, понадобится создать что-то уникальное. Поэтому рассмотрим настройки ткани.

- ◆ **Quality** (Качество). Установка качества обработки. Чем выше, тем лучше, но и медленнее.
- ◆ Группа **Damping** (Затухание):
 - **Spring** (Пружина) — скорость деформации ткани;
 - **Air** (Воздух) — увеличение параметра замедляет падение ткани.
- ◆ Группа **Material** (Материал):
 - **Mass** (Масса) — масса ткани;
 - **Structural** (Структура) — определяет жесткость ткани;
 - **Bending** (Изгиб) — большее значение приведет к более сильным изгибам.

- ♦ **Pinning** (Крепление). Необходимо выбрать группу вершин, которая не будет участвовать в имитации. Нечто подобное имеется у **Soft Body** под названием **Goal**.

6.6. Силовые поля

Возможности Blender в создании анимации почти безграничны и все же бывают ситуации, когда этого явно недостаточно. Возьмем, к примеру, такое природное явление, как метель. Возникает закономерный вопрос, а как это сделать? Конечно, можно создать частицы, настроить соответствующим образом динамику и получить красивый, падающий снег, но не метель с ее порывами, резкими переменами направления ветра. В этом случае удобнее будет воспользоваться силовыми полями.

Force Fields (Силовые поля) — это еще одна часть физики Blender, предлагающая специальные алгоритмы для внесения динамики в сцену. Разумеется, это применимо только к объектам, использующим физику **Soft Body**, **Particles**, **Cloth**.

Силовые поля доступны на закладке **Physics** (см. рис. 6.2). Как и большинство физических систем, они имеют глобальные настройки, характерные для всех типов полей (рис. 6.39).

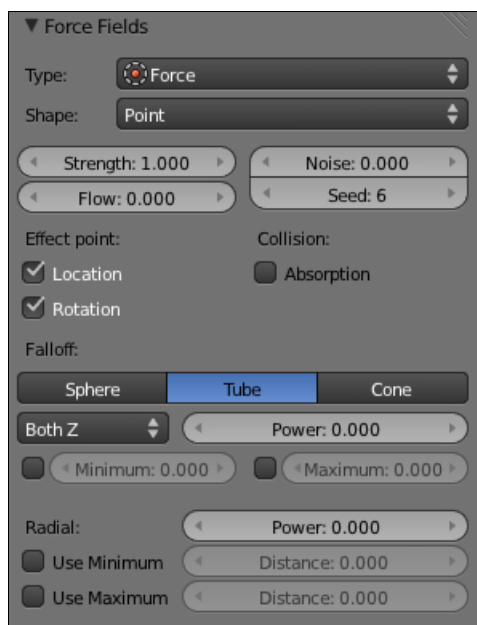


Рис. 6.39. Настройки силовых полей

В настройках **Force Fields** имеется меню **Type**, которое содержит богатый набор алгоритмов полей. Рассмотрим их действие применительно к частицам:

- ♦ **Force** (Сила) — простейшее поле, не имеющее дополнительных настроек. Эффект сдувания частиц от центра **Force** к его границам;

- ◆ **Wind** (Ветер) — наиболее часто употребляемое поле, имитирующее ветер. В зависимости от ориентации объекта-носителя поля, меняется направление движения потока;
- ◆ **Vortex** (Вихрь) — эффект закручивания частиц вокруг центра поля по спирали;
- ◆ **Magnetic** (Магнетизм) — используйте это поле для получения эффекта магнетизма. Зависит от скорости частиц;
- ◆ **Harmonic** (Гармония) — визуально действие поля выглядит как притягивание частиц к своему центру с последующим выбросом;
- ◆ **Charge** (Заряд) — привлечение или отражение частиц, если установлена область влияния **Charge**;
- ◆ **Lennard-Jones** — данное поле старается удерживать частицы на одинаковом расстоянии друг от друга, при условии, что между ними изначально небольшая дистанция;
- ◆ **Texture** (Текстура) — сложный алгоритм, который позволяет создавать свое поле с уникальными параметрами. При этом информация об искажениях хранится в RGB текстуры;
- ◆ **Curve Guide** (Направляющая кривая) — еще одно поле, позволяющее произвольно настраивать его влияние. В данном случае это осуществляется с помощью кривой. Причем носителем этого поля должна быть кривая. Это позволяет создавать необычные эффекты;
- ◆ **Boid** (Имитация) — поле, обратное действию **Force**;
- ◆ **Turbulence** (Турбулентность) — действие поля основано на шумовом алгоритме **Noise** в 3D-пространстве;
- ◆ **Drag** (Перемещение) — замедляет скорость частиц.

Как видите, набор типов полей достаточно обширный.

По умолчанию действие полей осуществляется по всем направлениям, но можно ограничить его только по координате **Z**. Это выполняется с помощью меню **Shape**. В нем содержится два пункта: **Point** (по всем направлениям) и **Plane** (ось **Z**).

Кроме этих двух меню в настройках **Force Fields** присутствуют глобальные опции:

- ◆ **Strength** (Сила) — сила воздействия;
- ◆ **Flow** (Поток) — скорость потока;
- ◆ **Noise** (Шум) — фактор случайности, добавляемый в действие поля;
- ◆ **Seed** (Стартовое значение) — настройка для **Noise**, позволяющая перенастроить генератор шума;
- ◆ **Effect point** (Точка воздействия) — здесь можно выбрать, на что будет влиять данное поле: **Location** (Позиция) и **Rotation** (Ротация).

Для использования полей в сцене необходимы объекты-носители. В качестве них чаще всего используют **Empty**. Рассмотрим работу с полями на небольшом примере.

Создайте новую сцену, добавьте в нее примитивы **Plane** и **Empty**. Снабдите плоскость системой частиц. Ничего не меняйте в ее настройках.

Активируйте для **Empty** кнопку **Force Field** и выберите в меню **Type** пункт **Wind**.

По умолчанию частицы сыплются вниз по координате Z. Однако под воздействием ветра их будет сносить в сторону. Установите силу **Wind**, равную 4 (параметр **Strenght**). Теперь при изменении ориентации **Empty** вы можете управлять направлением ветра (рис. 6.40).

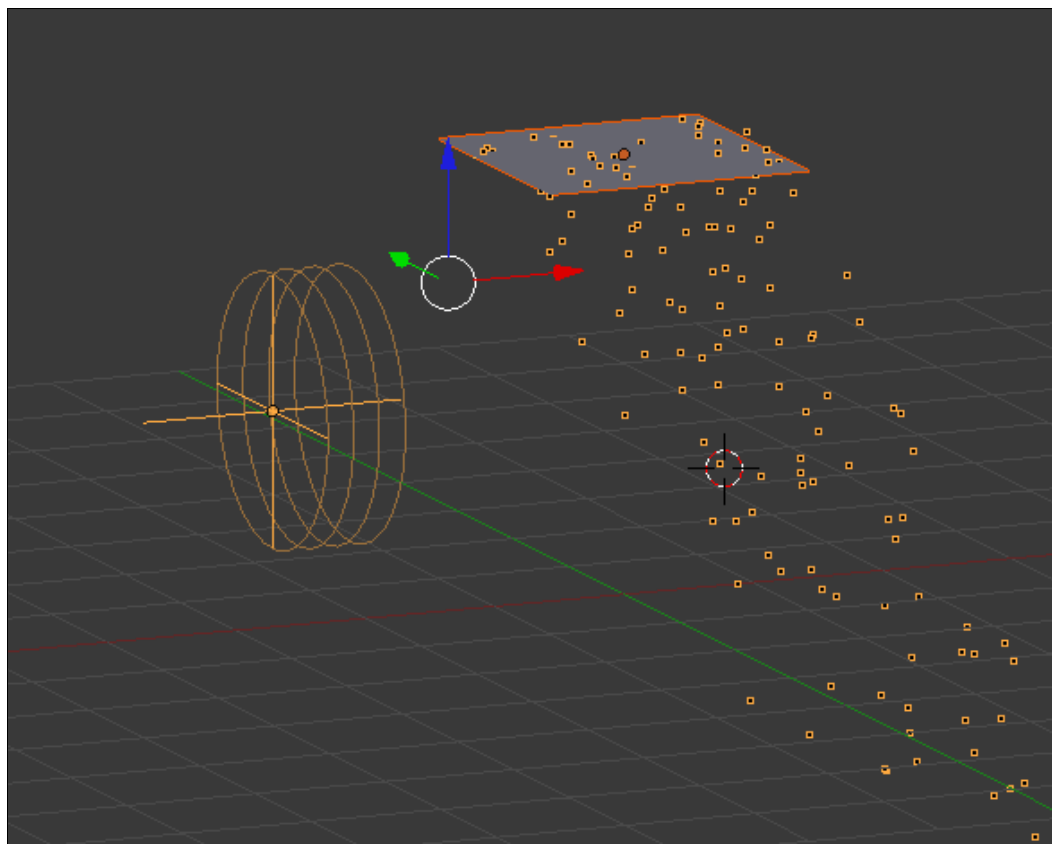


Рис. 6.40. Частицы сносит ветром

По умолчанию поля воздействуют на все физические объекты, имеющиеся в сцене. При необходимости, можно установить зону влияния поля. Эта функция доступна в группе **Falloff** (см. рис. 6.39).

Программа предлагает три формы зон: **Sphere** (Сфера), **Tube** (Труба), **Cone** (Конус). После выбора любой из этих кнопок необходимо будет установить минимальное и максимальное значения границ области. При включении этих опций в окне **3D View** будут визуально отображаться границы влияния, в соответствии с выбранной формой области (рис. 6.41).

Всего имеется 4 пункта:

- ◆ **Water** (Вода);
- ◆ **Oil** (Масло);
- ◆ **Honey** (Мед);
- ◆ **Manual** (Ручная настройка).

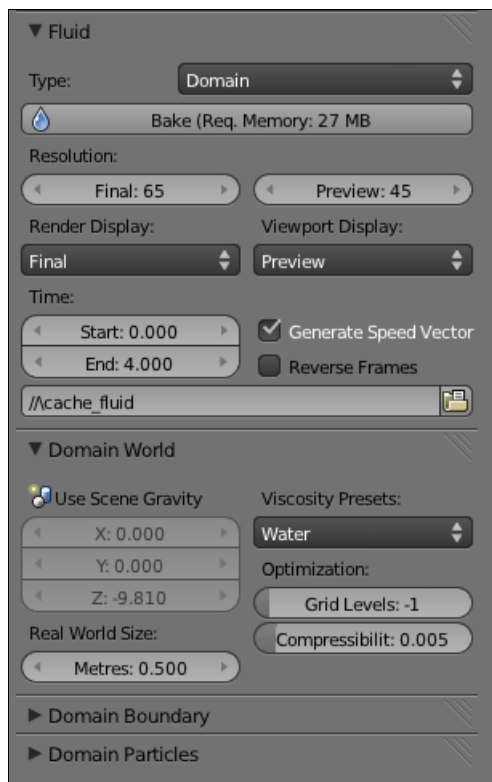


Рис. 6.42. Настройки домена

Так как разная жидкость имеет свои параметры вязкости, среди имеющихся трех вполне можно подобрать себе подходящий тип. Пусть наша жидкость будет обычной водой (**Water**).

Следующий участник в имитации — это источник или **Fluid**. В качестве него должен выступать уже другой примитив. Добавьте в сцену сферу, уменьшите ее размер и разместите в центр куба (рис. 6.43).

Включите для сферы физику жидкостей (см. рис. 6.2), а в меню **Type** выберите пункт **Fluid**.

В принципе все готово, но не спешите нажимать <Alt>+<A> для проигрывания анимации — ее просто нет. Сначала нужно сделать предварительный просчет, и это является обязательным. Прочие разделы физики в Blender предлагают функцию кеширования только как дополнительную, но не **Fluid**.

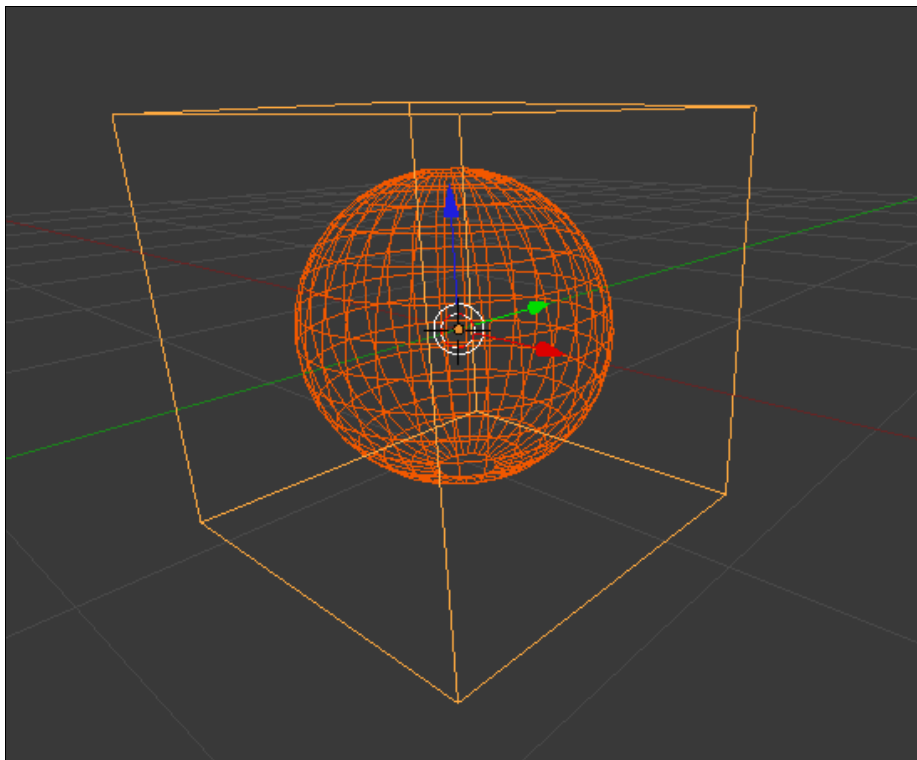


Рис. 6.43. Сцена с объектами **Domain** и **Fluid**

Выделите домен (примитив **Cube**) и откройте закладку **Fluid** в настройках физики (см. рис. 6.42). По умолчанию тут установлены вполне корректные параметры просчета, но можно отрегулировать их на свой вкус.

Перечислим настройки домена.

- ◆ Кнопка **Bake** (Выпечка) предназначена для запуска просчета.
- ◆ Группа **Resolution** (Разрешение) устанавливает разрешение в процентном соотношении для окна **3D View** (поле **Preview**) и окончательного результата (**Final**). Учтите, что изменение параметра **Final** может привести к необоснованно большому времени вычисления без улучшения видимого качества. Изменяйте его только в случае необходимости.
- ◆ Меню **Render Display** (Экран рендера) позволяет выбрать, в какой форме будет представлен окончательный результат: в максимальном качестве (**Final**), в форме предварительного просмотра (**Preview**) или в виде решетки геометрии (**Geometry**).
- ◆ Меню **Viewport Display** (Экран просмотра) имеет те же параметры, что и рассмотренные ранее, но только применительно к окну **3D View**.
- ◆ В группе параметров **Time** (Время) вы можете выставить начальный (**Start**) и конечный (**End**) кадры имитации.

По умолчанию программа сохраняет файлы кеша в папке с проектом, но вы можете изменить путь, щелкнув кнопку с характерным рисунком папки в нижней части настроек домена.

Теперь нажмите кнопку **Bake** и идите спокойно пить чашечку кофе — расчет даже столь простой имитации достаточно длительный. Заметьте, что во время просчета Blender позволяет работать с другими объектами в сцене. А следить за ходом выполнения вы можете в статусной строке главного заголовка программы.

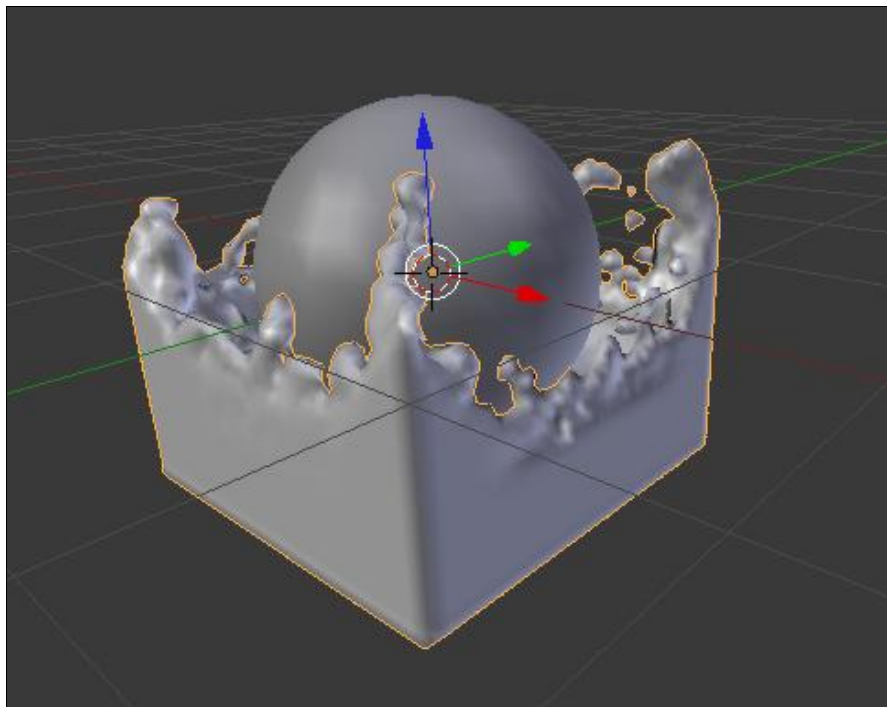


Рис. 6.44. Результат просчета системы **Fluid**

Если вы не желаете ждать окончания просчета физики, то нажмите клавишу <Esc>. Это приведет лишь к остановке кеширования, но не удаления уже обработанных кадров. Поэтому вы сможете оценить качество имитации по нескольким имеющимся кадрам.

На рис. 6.44 отчетливо видно, что жидкость располагается только в пределах границ домена-куба. Что же делать с этим дальше? Можно создать модель емкости и подогнать форму домена под нее (потребуется перерасчет), а можно добавить емкость непосредственно внутрь **Domain**.

СОВЕТ

Объект **Fluid** служит лишь для обозначения формы, объема и начального местоположения жидкости в имитации. После обработки он в сцене является явно лишним. Не удаляйте его. Либо скройте с помощью клавиши <N>, либо спрячьте в свободный слой.

Создадим несложную модель квадратной емкости. Добавьте в сцену еще один куб. Нажмите клавишу <S>, затем <Z> для уменьшения масштаба по оси Z. Сделайте куб прямоугольником.

Войдите в режим редактирования (<Tab>) и трижды разбейте структуру инструментом **Subdivide**.

Перейдите в просмотр **Top View** (<NumPad 7>) и выделите полигоны, как на рис. 6.45.

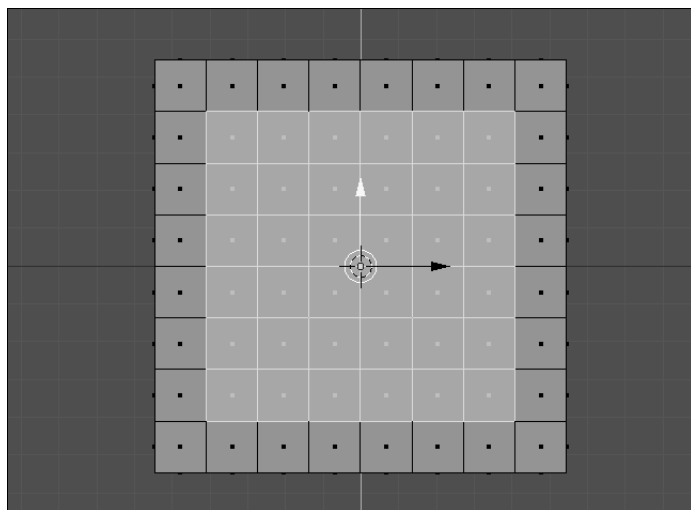


Рис. 6.45. Выделение для создания емкости

Разверните сцену и, используя манипулятор **Grab**, переместите выделенные элементы вглубь объекта по оси Z. Для улучшения качества модели включите сглаживание (**Smooth**) и добавьте модификатор **Subdivision Surface**. Поместите полученную ванночку в домен (рис. 6.46).

Не спешите выполнять перерасчет имитации — ничего не получится. Хотя емкость находится в домене, но Blender не знает, как данный объект должен взаимодействовать с жидкостью. Для этого к ванночке необходимо добавить свой экземпляр физики **Fluid**.

Откройте панель **Physics**, включите кнопку **Fluid**, а в меню **Type** выберите пункт **Obstacle** (Препятствие). Теперь выделите объект домена и смело жмите кнопку **Bake** (рис. 6.47).

СОВЕТ

Иногда бывает нужно удалить результат кеширования физики **Fluid**. К сожалению, такой функции в Blender почему-то не предусмотрено. Простейшим выходом является физическое удаление файлов кеша из папки с проектом.

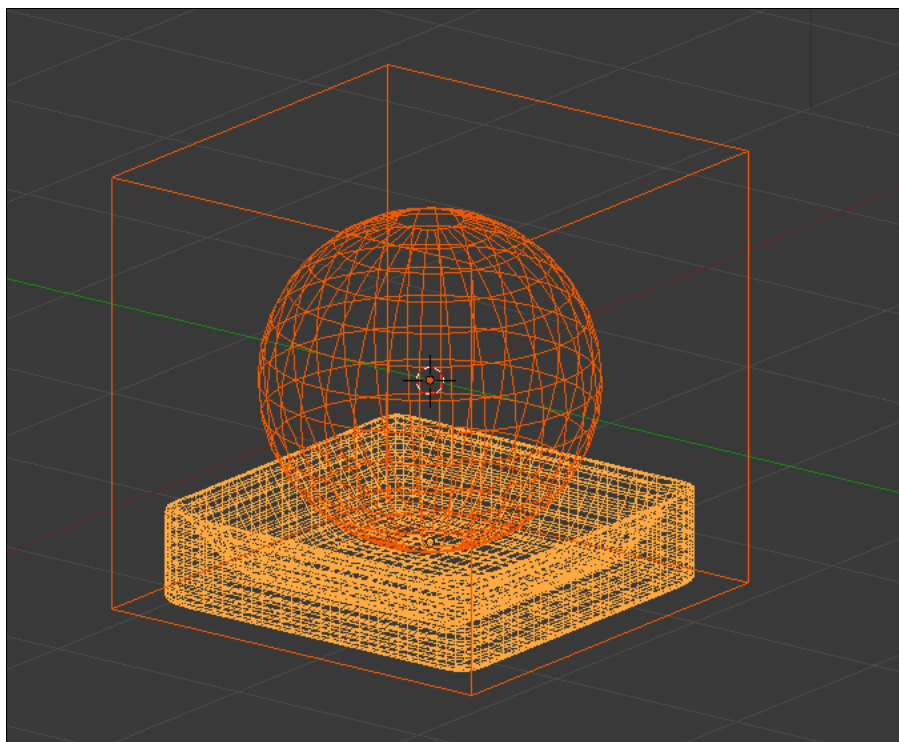


Рис. 6.46. Домен, ванночка и источник

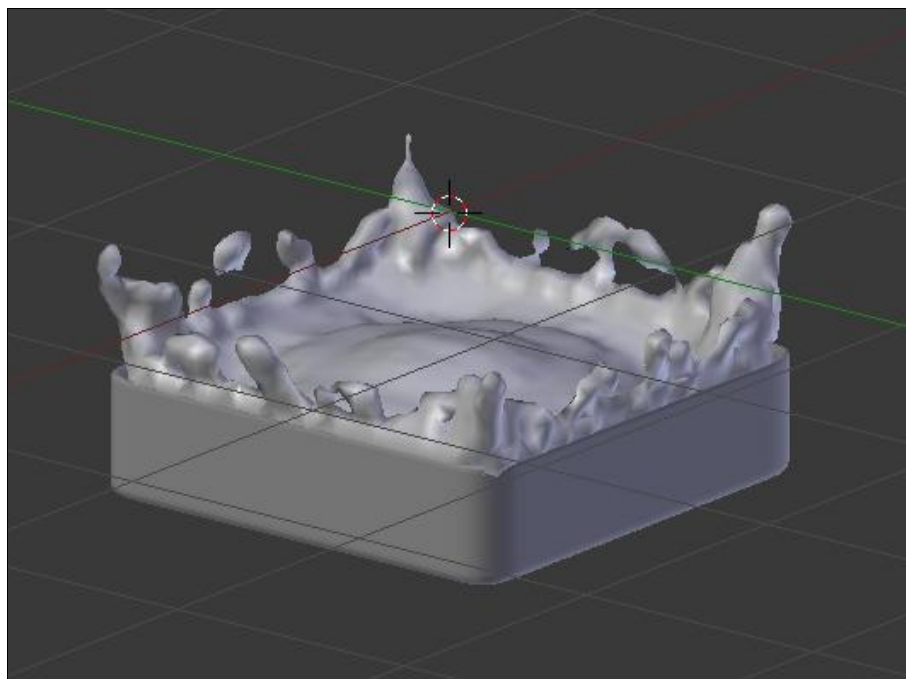


Рис. 6.47. Вода скапливается в емкости

6.8. Как сделать дым

Единственный способ создания дыма — это использовать системы частиц. Как работать с ними, рассказывалось в *разд.* 6.2. Но есть и еще один вариант, который также базируется на частицах, зато предлагает более удобное управление ими.

На панели **Physics** имеется кнопка **Smoke** (Дым) (см. рис. 6.2). Выделите куб в новой сцене и включите **Smoke** (рис. 6.48).

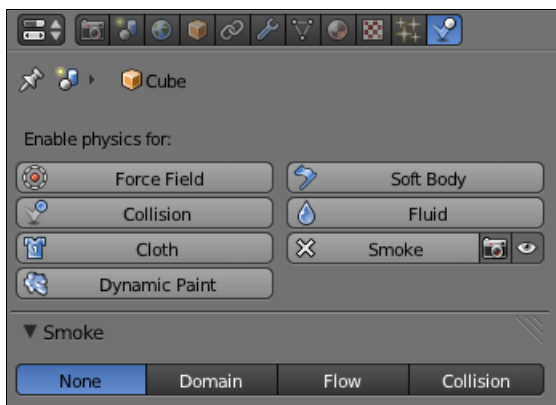


Рис. 6.48. Глобальные кнопки настройки **Smoke**

В настройках **Smoke** вы увидите несколько кнопок с подозрительно знакомыми названиями. Действительно, раздел генерации дыма использует ту же систему доменов, что и **Fluid**.

Пусть текущий куб в сцене будет являться доменом, нажмите соответствующую кнопку в настройках **Smoke**.

Теперь добавьте еще один куб, уменьшите его и поместите внутрь домена. Второй примитив будет играть роль генератора. Выделите его, включите физику **Smoke** и выберите кнопку **Flow**.

Программа автоматически добавит к этому примитиву систему частиц с определенными настройками. Вы можете проверить это на закладке **Particles** окна **Properties**.

В отличие от физики **Fluid** здесь не требуется обязательного предварительного просчета имитации, поэтому просто нажмите клавиши <Alt>+<A> (рис. 6.49).

Вы увидите, что вокруг куба-источника начинают скапливаться частицы дыма, которые совершенно не желают устремляться вверх или куда-либо еще.

Конечно, можно отрегулировать соответствующие настройки в системе частиц, прикрепленной к объекту-источнику, но тогда нет смысла вообще возиться с доменом. В действительности, дело обстоит гораздо проще. Вы можете управлять направлением и силой скорости потока частиц путем изменения формы домена. Попробуйте вытянуть куб **Domain** по оси Z и снова нажать <Alt>+<A>. Вот теперь результат больше похож на дым (рис. 6.50).

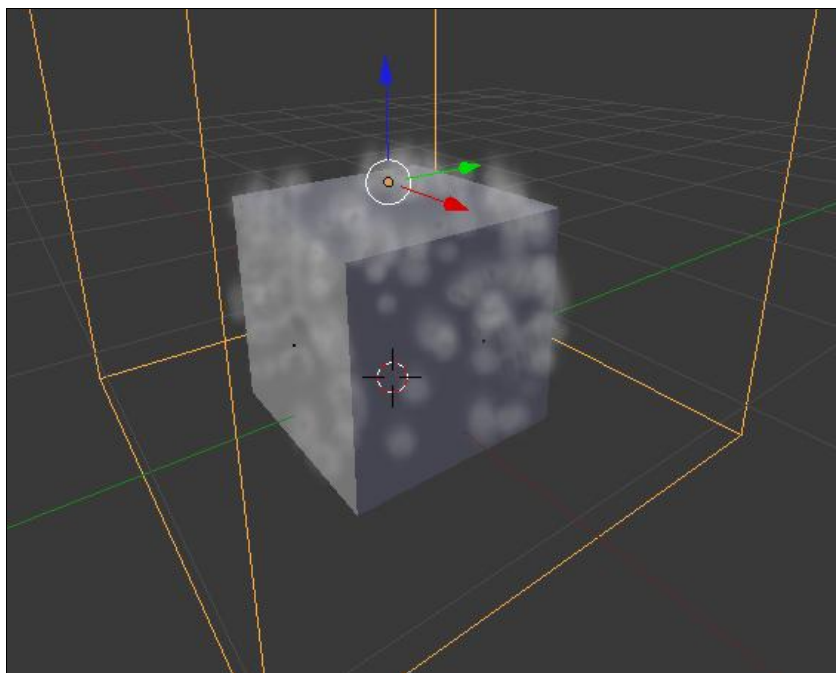


Рис. 6.49. Пробная генерации дыма

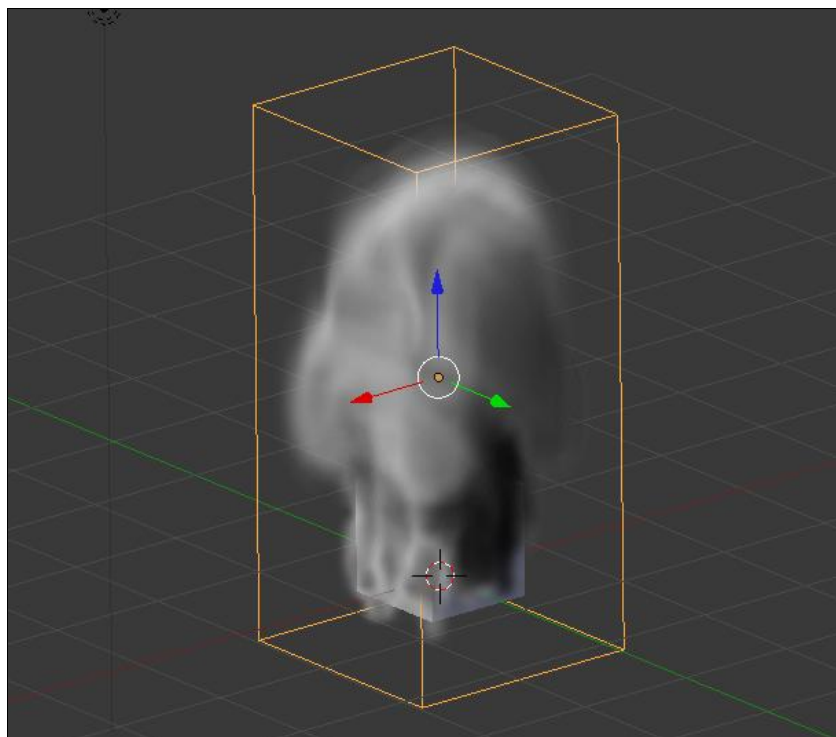


Рис. 6.50. Частицы устремляются вверх

Нажмите клавишу <F12> для рендера сцены, и вы увидите на картинке гордо торчащий куб безо всякого намека на дым. Дело в том, что хотя **Smoke** создан, настроен и корректно отображается в окне программы, система рендера не в курсе, как ей обрабатывать эти частицы. Объекту домена необходимо прикрепить материал со специальной текстурой.

Выделите домен, откройте вкладку **Material**, и если материала нет, то создайте его. Установите тип материала как **Volume**, а в поле **Density** (Плотность) введите 0 (рис. 6.51).

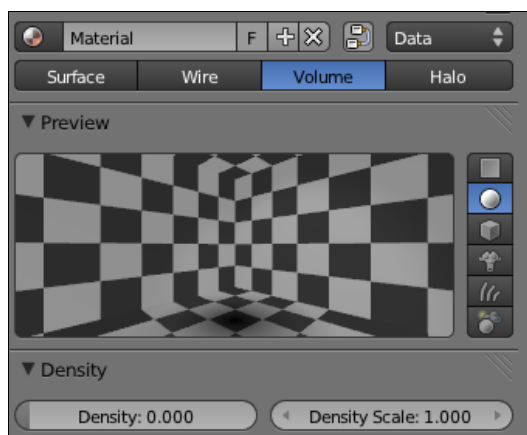


Рис. 6.51. Настройка материала для дыма

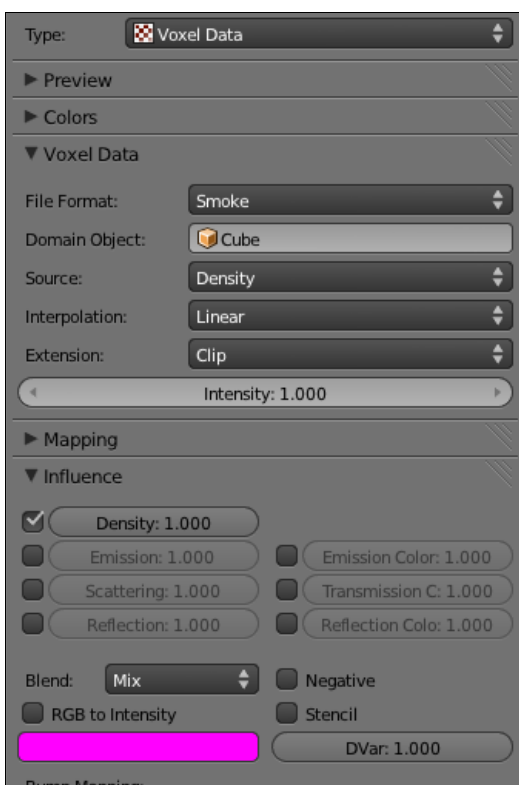


Рис. 6.52. Настройка текстуры

Перейдите на панель **Texture** и выберите из меню **Type** пункт **Voxel Data** (Данные объема). Выполните следующие шаги по настройке текстуры (рис. 6.52):

1. Укажите в поле **Domain Object** текущий куб.
2. На закладке **Influence** включите опцию **Density** и отключите **Emission Color**.

Теперь нужно отключить вывод куба-источника при рендере результата. Это можно сделать в настройках частиц. Найдите вкладку **Render** и снимите галочку у опции **Emitter**.

Возможно, результат обработки картинки вас не удовлетворит из-за излишне светлого цвета дыма. Попробуйте изменить параметр **Density Scale** у материала домена (рис. 6.53).



Рис. 6.53. Рендер дыма с параметром **Density Scale** = 5

6.9. Практика. Создание торнадо

Когда заходит разговор о создании сцены с воздействием стихии природы, все ожидают очень сложной и долгой работы. С Blender это не так. В этом уроке мы рассмотрим, как можно смоделировать такое природное явление, как торнадо (рис. 6.54).

Торнадо можно рассматривать с обывательской точки зрения, как вихревой воронкообразный поток, вращающийся с большой скоростью вокруг эпицентра. Торнадо бывают разными, но мы попытаемся смоделировать такой, как на рис. 6.54.

Несмотря на свой грозный вид, смерч создать очень просто. В качестве его основы будет служить система **Particle**, в которой частицы будут двигаться вверх по координате **Z** и одновременно закручиваться вокруг этой оси.

Чтобы не создавать систему частиц с нуля, воспользуемся уже готовой. Откройте новый проект и выделите куб.

В меню **Object** окна **3D View** имеется замечательный пункт **Quick Effects**, который содержит простые заготовки некоторых видов эффектов:

- ◆ **Quick Fur** (Мех);
- ◆ **Quick Explode** (Взрыв);
- ◆ **Quick Smoke** (Дым);
- ◆ **Quick Fluid** (Жидкость).



Рис. 6.54. Торнадо. Фото с сайта theepochtimes.com

При выборе любого из пунктов программа добавит соответствующие функции физики выделенному объекту. Сейчас нас интересует опция **Quick Smoke**. Выберите ее.

Вы увидите на экране стандартную систему **Domain** и **Flow**, которая рассматривалась в предыдущем разделе книги. Эта заготовка оптимально подходит для задуманной задачи, но кое-что придется настроить.

Во-первых, растяните домен так, чтобы он занял всю вспомогательную решетку окна **3D View**. Лучше это делать индивидуально по осям X и Y . Также немного уменьшите куб-источник (рис. 6.55).

Заготовка **Quick Smoke** уже имеет настроенный материал, который присоединен к домену. Оставим его без изменений.

Если вы сейчас проиграете анимацию, то увидите, что дым спокойно поднимается вверх по оси Z , и это абсолютно не подходит. Для создания вращения воспользуемся силовым полем **Vortex**.

Добавьте в сцену объект **Empty** из меню **Add**. Расположите его так, чтобы он находился внизу домена под центром куба-источника. Откройте вкладку **Physics** окна **Properties** и нажмите кнопку **Force Field**. В меню **Type** выберите пункт **Vortex**.

По умолчанию **Vortex** создает слишком слабое вращение. Пусть сила воздействия (**Strenght**) будет равна 10, а **Flow** = 5. На этом настройка поля завершена.

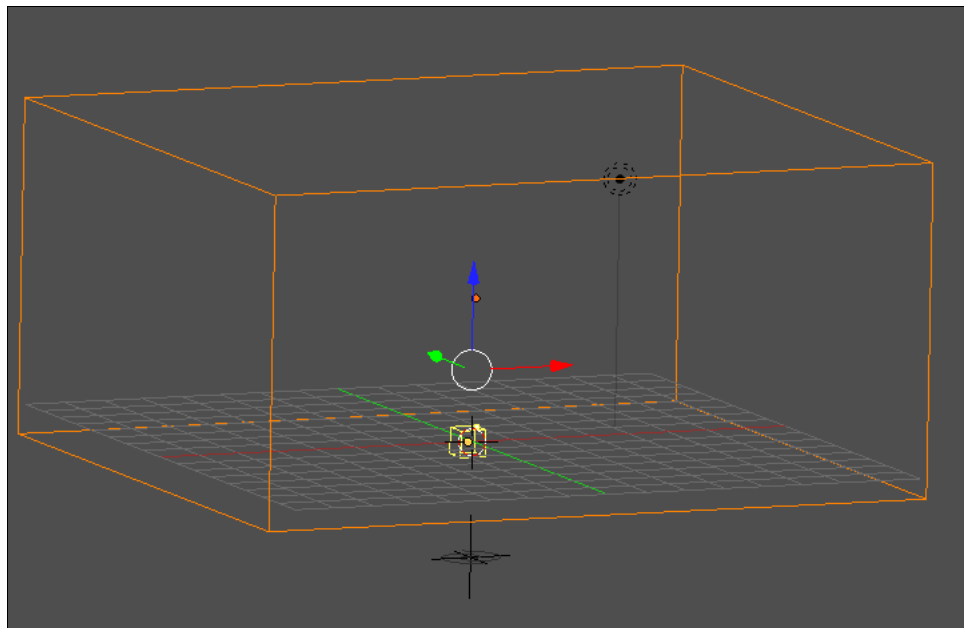


Рис. 6.55. Расположение объектов в сцене

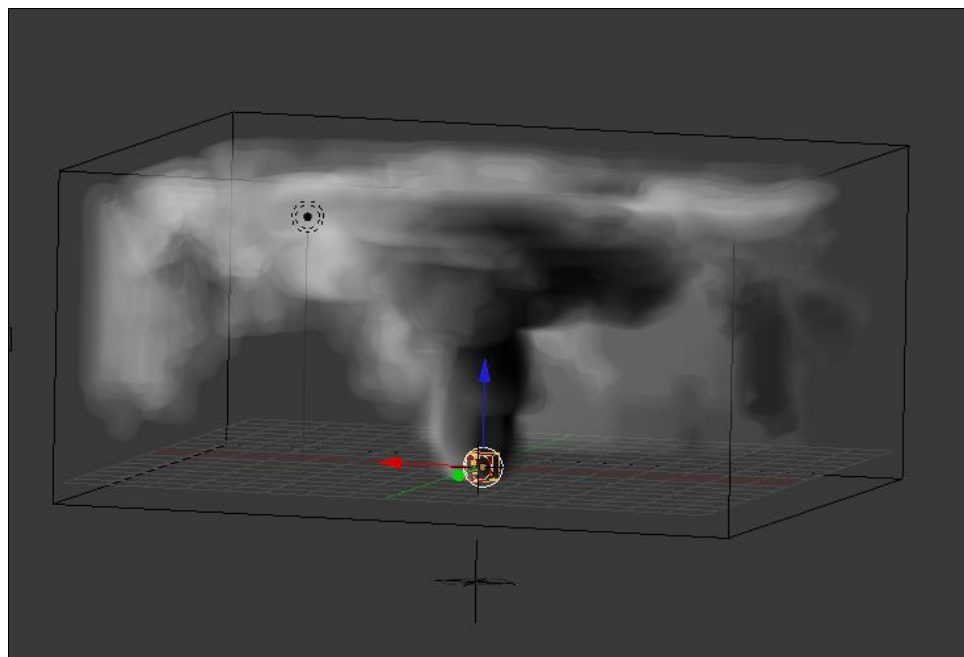


Рис. 6.56. Искусственный смерч

По умолчанию частицы **Smoke** имеют слишком продолжительное время жизни. Выделите объект домена и в настройках физики найдите опцию **Dissolve** (Затухание). Включите ее. Под ней имеется поле **Time**. Измените его значение на 25. Таким образом, каждая частица будут жить не более 25 кадров.

Вот и все, нажмите клавиши **<Alt>+<A>** и наслаждайтесь неистовым смерчем (рис. 6.56).

6.10. Практика. Следы на воде

Все, наверное, хоть раз в жизни кидали камни в воду и с интересом наблюдали за появляющимися кругами волн. Рассматриваемая далее техника позволяет с легкостью создавать подобные следы от падения или движения объекта в воде.

Несмотря на свою кажущуюся сложность, эта задача решается очень просто, а поможет в этом уникальный инструмент **Dynamic Paint** (Динамичное рисование).

Создайте в новой сцене два объекта: **Plane** и **UV Sphere**. Плоскость будет играть роль поверхности воды, а сфера, соответственно, возмутителя спокойствия. Увеличьте масштаб плоскости по своему усмотрению.

Инструмент **Dynamic Paint** (Динамичное рисование) не зря получил такое свое название. Принцип использования этого инструмента очень напоминает работу художника, когда тот мазками кисти по холсту создает свое произведение. Нечто подобное можно наблюдать и в Blender.

В динамичной сцене рисования обязательно должны участвовать два объекта. Первый играет роль холста и к нему применяются соответствующие настройки **Dynamic Paint**, а другой — роль кисти.

Выделите **Plane**, откройте панель **Physics** и включите кнопку **Dynamic Paint** (рис. 6.57).

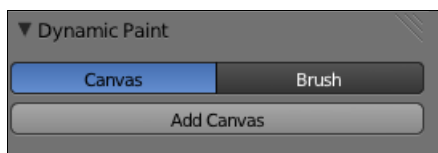


Рис. 6.57. Опции выбора глобальных настроек **Dynamic Paint**

Так как примитив **Plane** является холстом, то нажмите кнопку **Canvas**, а затем **Add Canvas**, чтобы открыть настройки (рис. 6.58).

Несмотря на обширный ряд вкладок и опций, настроить **Canvas** достаточно просто. Самое главное здесь определиться, какой след будет оставлять кисть на холсте, ведь **Dynamic Paint** способен на гораздо большее, нежели создавать круги на воде.

Это можно сделать в меню **Surface Type** (Тип поверхности) закладки **Dynamic Paint Advanced**:

- ♦ **Paint** (Рисование) — кисть будет оставлять на поверхности холста цвет или текстуру в виде следа от движения;

- ◆ **Displace** (Перемещение) — происходит деформация поверхности холста при столкновении с ней кисти;
- ◆ **Waves** (Волны) — создание эффекта волн, характерных для соприкосновения предмета с водой;
- ◆ **Weight** (Вес) — изменение веса вершины в месте столкновения объектов.

В нашем случае выберите пункт **Waves**.

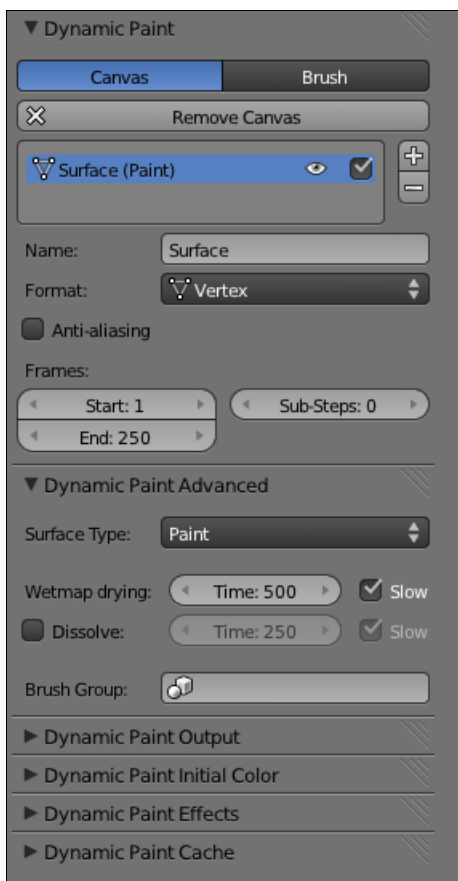


Рис. 6.58. Настройки **Canvas**

Теперь займемся настройкой кисти. Выделите сферу, включите для нее кнопку **Dynamic Paint**. Активируйте кнопку **Brush**, а затем нажмите **Add Brush**. Вот и вся настройка кисти.

Осталось только изменить структуру примитива холста. Дело в том, что качество работы инструмента **Dynamic Paint** очень сильно зависит от качества **Mesh**. Чем больше элементов в решетке, тем лучше результат. Поэтому выполните разбиение структуры **Plane** с помощью инструмента **Subdivide** несколько раз.

Проверим **Dynamic Paint** в действии. Для этого не нужно выполнять предварительный просчет физики, хотя такая возможность у **Canvas** имеется. Достаточно включить анимацию сцены ($\langle \text{Alt} \rangle + \langle \text{A} \rangle$) и провести сферой по поверхности **Plane** (рис. 6.59).

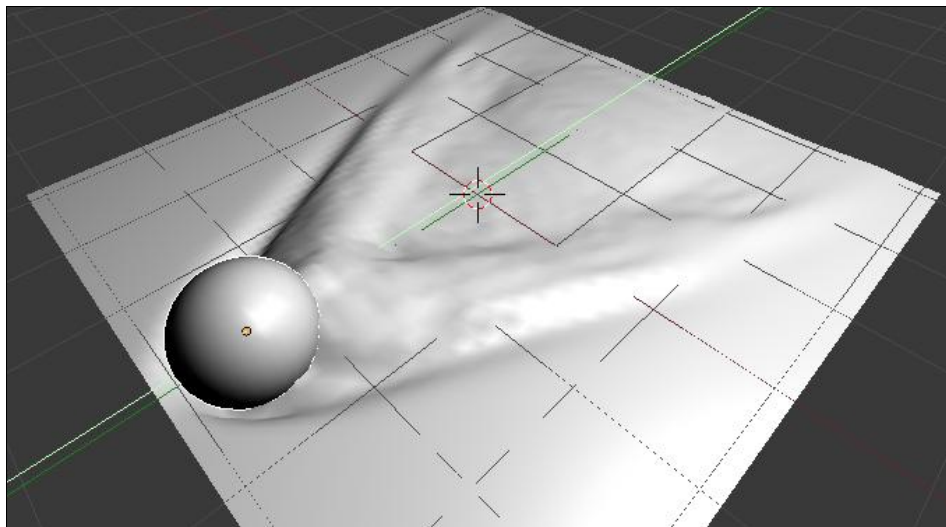
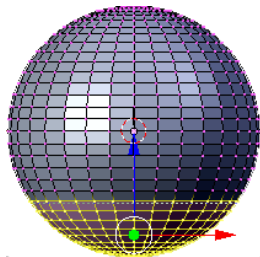


Рис. 6.59. Следы на воде очень реалистичные

ГЛАВА 7



Свет, камеры и окружение

Готовые модели, настроенная анимация и физика — это не завершение работы над сценой. Большое значение имеет правильное расположение источников света, их тип и цвет, а также сопутствующее окружение. Даже цвет фона может влиять на общее впечатление от сцены.

Blender предлагает многочисленные возможности для окончательной доработки проекта: глобальный свет, туман, градиентный фон, атмосферные эффекты и многое другое.

7.1. Источники света

В новой сцене Blender по умолчанию уже имеется один источник света и это крайне необходимо, ведь иначе вы бы ничего не увидели при обработке результата (рис. 7.1).

Источник света или лампа (**Lamp**) — это стандартный объект Blender, который подчиняется основным правилам манипуляции. Вы его можете перемещать, вращать, масштабировать, а также использовать в анимации.

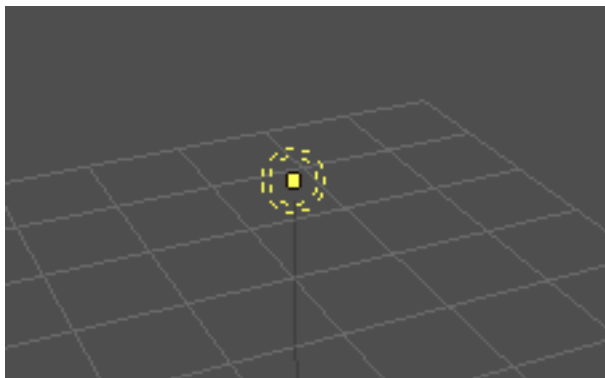


Рис. 7.1. Источник света по умолчанию

Конечно, не все манипуляторы применимы для лампы, по крайней мере к той, что находится в сцене по умолчанию. Так, например, вы не сможете ее развернуть. Это связано с тем, что она является точечным источником света, который равномерно освещает объекты по всем направлениям. Но вот с другими типами такое действие возможно.

Всего Blender предлагает пять типов ламп, которые доступны для создания в меню **Add | Lamp**. Интересно, что в любой момент можно изменить тип уже присутствующей в сцене лампы в ее настройках.

Типы ламп:

- ◆ **Point** (Точечный) — точечный источник, равномерно освещающий объекты вокруг себя. Используется по умолчанию;
- ◆ **Sun** (Солнце) — это специальный тип лампы, оптимально подходящий для имитации освещения поверхности сцены так, как происходит в реальном мире. Имеет уникальные настройки, позволяющие учитывать атмосферные искажения;
- ◆ **Spot** (Спот) — прожекторный тип, который характеризуется наличием круговой или квадратной области освещенности. Имеет возможность ротации для изменения направления излучения. Обычно используется в сцене для освещения конкретного объекта;
- ◆ **Hemi** (Направленный) — эта лампа предлагает широкое, направленное освещение;
- ◆ **Area** (Область) — позволяет выделять в сцене прямоугольные области для освещения. Например, с ее помощью можно сделать свет в окне.

Нужно знать, что вне зависимости от типа лампы и ее действия, визуально она не будет присутствовать в картинке рендера (есть одно исключение). Так, если вы захотите сделать обычную лампочку в сцене, то понадобится модель с соответствующими настройками материала. Лампы — это источники света, но не реально видимые объекты!

Настройка ламп осуществляется на специальной закладке **Lamp** окна **Properties** (рис. 7.2).

Конечно, большинство опций здесь являются уникальными для каждого типа источника, но есть и общие параметры. Во-первых, панель **Lamp** имеет область предпросмотра. Это, скорее всего, для примерного ознакомления с воздействием лампы. Во-вторых, здесь можно изменить тип при помощи соответствующих кнопок.

Любая лампа имеет параметр **Energy** (Энергия) и поле для установки цвета излучения. С помощью **Energy** вы можете контролировать силу источника.

По умолчанию лампы имеют белый цвет излучения. Но ничто не мешает вам заменить его на другой из стандартной палитры, которая появляется после щелчка мышью по образцу. Цвет лампы является сильным средством для подчеркивания атмосферы сцены!

Кроме энергии и цвета имеются и другие общие параметры:

- ◆ **Negative** (Негатив) — инвертирование освещенности участков на поверхности моделей;

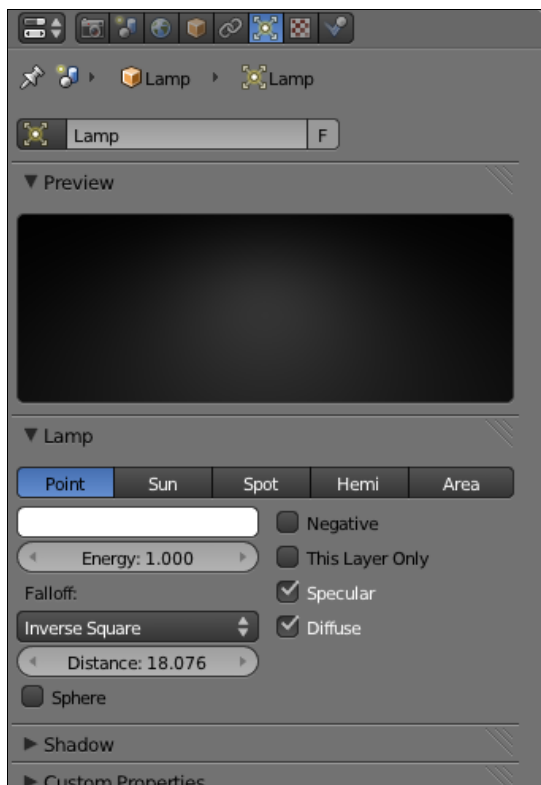


Рис. 7.2. Свойства лампы

- ◆ **This Layer Only** (Только этот слой) — включите эту опцию, если хотите, чтобы лампа освещала объекты только своего слоя;
- ◆ **Specular** (Зеркальный) — влияние на бликовые шейдеры;
- ◆ **Diffuse** (Диффузный) — влияние на основной цвет модели.

Наиболее популярной лампой после **Point** является **Spot**. Часто для выгодного выделения объекта в сцене применяют направленный на него источник света. Как раз для этих целей предназначен **Spot**.

Действие этого типа лампы сравнимо с прожектором, когда в сцене освещаются только те объекты, что попадают в конус света (рис. 7.3).

Очень удобным является визуальное отображение в окне программы границ и направления источника **Spot**. Это позволяет с легкостью позиционировать лампу так, как нужно.

Spot в силу своей особенности подчиняется изменению масштаба и ротации. Причем первое сделано не просто для галочки. Обратите внимание на окружность тыловой части конуса. Так Blender отмечает границы света (рис. 7.4).

Изменяя масштаб объекта **Spot**, можно корректировать диаметр окружности, но для этой цели все же лучше воспользоваться соответствующими настройками на панели **Lamp** (рис. 7.5).

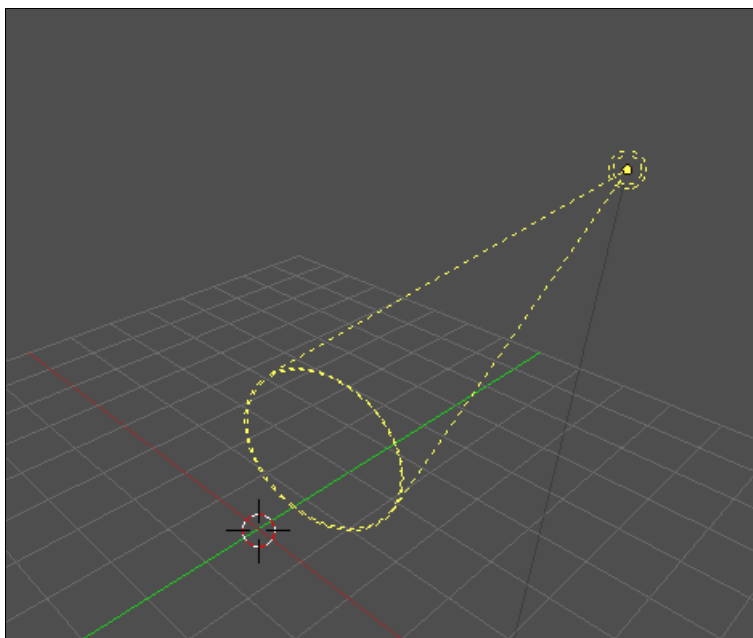


Рис. 7.3. Объект Spot

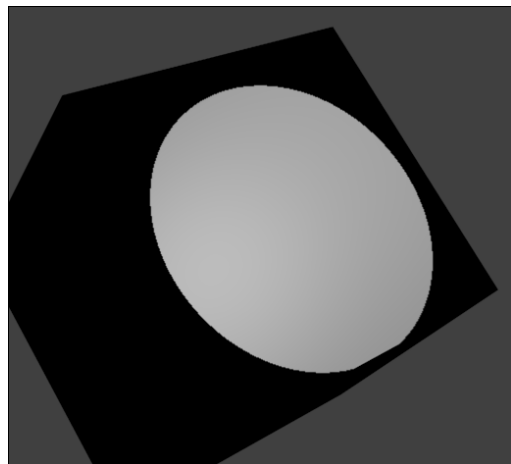


Рис. 7.4. Участок освещенности от лампы Spot

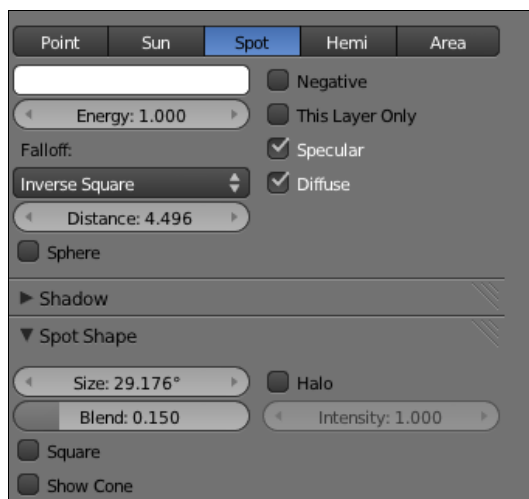


Рис. 7.5. Настройки Spot

Закладка **Spot Shape** (Форма спота) содержит опции, характерные только для этого типа лампы. Параметр **Size** (Размер) как раз и предназначен для изменения диаметра окружности конуса.

По умолчанию лампа генерирует свет с резким переходом в темноту за пределами конуса. В некоторых случаях это нежелательно. Размыть границы освещенности поможет опция **Blend**. Чем выше ее значение, тем более мягкий получается переход (рис. 7.6).

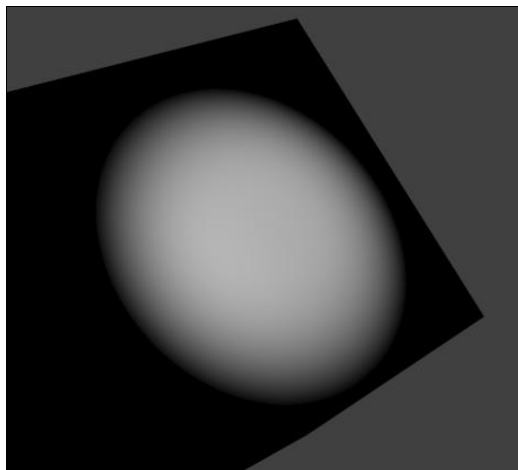


Рис. 7.6. Участок освещенности с увеличенным значением **Blend**

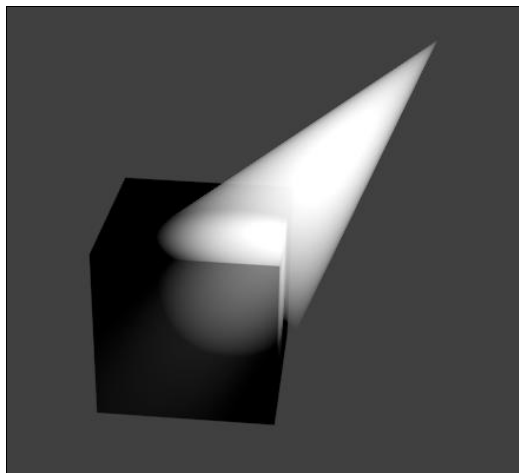


Рис. 7.7. Эффект **Halo** для источника

В настройках **Spot Shape** имеется опция **Halo** (Свечение), которая по умолчанию отключена. Это не что иное, как придание видимости конусу света в результатах рендера. Похожий эффект наблюдается у реального прожектора. После включения **Halo** становится доступно поле **Intensity** (Интенсивность), с помощью которого можно управлять силой эффекта свечения (рис. 7.7).

Еще одним важным параметром этой лампы (и некоторых других) является понятие "дистанция". **Spot** и **Point** — это лампы, мощность свечения которых начинает ослабевать после определенной точки. Расстояние от источника до этой точки называется **Distance** (Дистанция). Эту опцию вы сможете найти в группе **Falloff** (см. рис. 7.2).

СОВЕТ

По умолчанию **Spot** генерирует конус света в виде окружности. Включите опцию **Square** (Квадрат) в группе **Spot Shape**, чтобы получить квадратную область.

Вы уже знаете, что белый цвет излучения лампы можно заменить на любой другой. Но Blender предлагает возможность использования текстуры. Это можно сделать на стандартной панели **Textures**. Например, если использовать процедурную текстуру с анимацией, то получится своеобразная игра света, характерная для дискотек (рис. 7.8).

В реальном мире предметы всегда отбрасывают тени под воздействием любого источника света, но в Blender вы можете управлять тенью по своему усмотрению.

Лампы, кроме **Hemi**, имеют в своих настройках закладку **Shadow** (Тень). Опции в них практически одинаковые для всех типов **Lamp** (рис. 7.9).

Вы можете отключить тень, генерируемую текущим источником света с помощью кнопки **No Shadow** (Без тени) и, соответственно, включить кнопкой **Ray Shadow** (Отбрасывать тень).

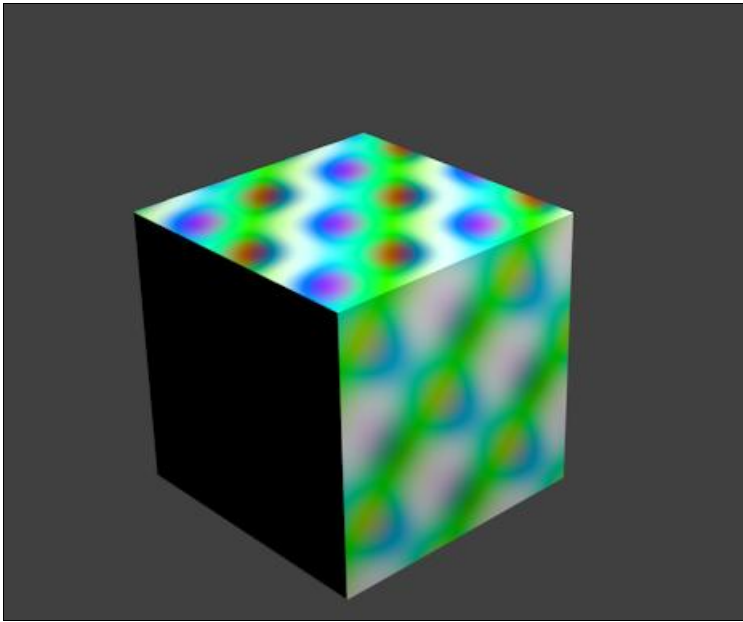


Рис. 7.8. Здесь лампа проецирует на объект текстуру

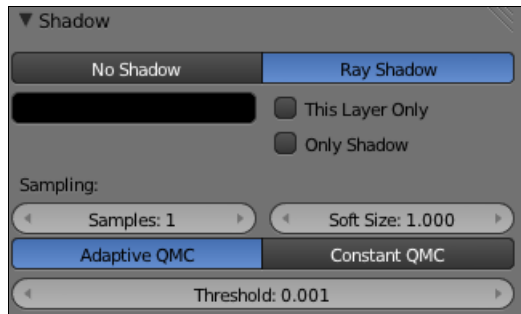


Рис. 7.9. Настройки Shadow

Группа **Sampling** (Выборка) содержит настройки, которые отвечают за качество теней. Так, чем выше значение поля **Samples**, тем выше качество теней.

По умолчанию тень имеет достаточно резкие границы перехода к свету. Вы можете воспользоваться опцией **Soft Size** (Размер смягчения) для смягчения переходных границ.

7.2. Солнце и атмосфера

В палитре источников света Blender имеется лампа, которая позволяет создавать сцены с реалистичным дневным светом. Речь идет о **Sun** (Солнце). Это действительно уникальная лампа, хотя бы потому, что ее можно видеть визуально после рендера картинки.

Особенностью этого источника является независимость силы излучения от его местоположения в сцене. Роль играет только угол наклона. Лампа дает очень насыщенное излучение, которое не изменяется в зависимости от расстояния (рис. 7.10).

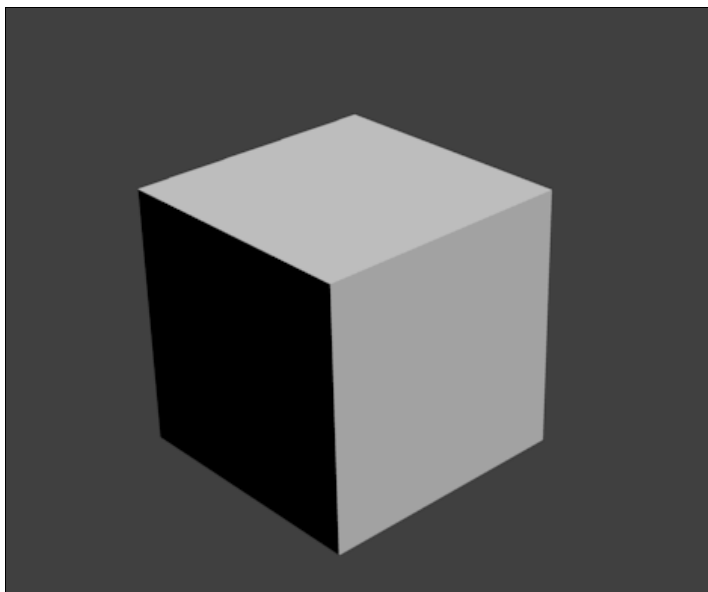


Рис. 7.10. Куб, освещенный источником **Sun**

Лампа **Sun** используется для освещения открытых сцен, например горного ландшафта. Причем ее настройки позволяют генерировать атмосферу с горизонтом.

Обычно для создания неба используют **Skybox** (бесшовные текстуры, натянутые на внутреннюю поверхность куба), но с учетом возможностей **Sun** в этом больше нет необходимости.

По умолчанию генерация атмосферы и неба отключены. Вы их можете найти на закладке **Sky & Atmosphere** (Небо и атмосфера) (рис. 7.11).

Blender уже предлагает готовые заготовки нескольких типов атмосфер, которые можно найти в меню **Sky Presets**, но давайте разберемся с параметрами для ручной настройки.

Включение **Sky** создаст небо с видимым солнцем, которые будут смешаны с основным цветом фона (по умолчанию он серый). Перечислим опции этой группы.

- ◆ **Turbidity** (Мутность). Управление чистотой неба. Чем выше параметр, тем более туманно небо, которое приобретает красный оттенок, а вокруг солнца создается характерный ореол.
- ◆ **Blending** (Смешивание). Это меню содержит стандартный список функций для смешивания эффекта **Sky** с фоном сцены. Степень смешивания устанавливается с помощью параметра **Factor**.

- ♦ Группа **Horizon** (Горизонт) содержит два параметра: **Brightness** (Яркость) и **Spread** (Распространение). Управляя этими двумя опциями, можно добиться разного вида горизонта.
- ♦ Группа **Sun**, ответственна за вид солнца. Так с помощью параметра **Size** можно установить размер солнца, а параметром **Brightness** задать его яркость. Еще одно поле в этой группе **Back Light** отвечает за степень дополнительного свечения вокруг солнца.

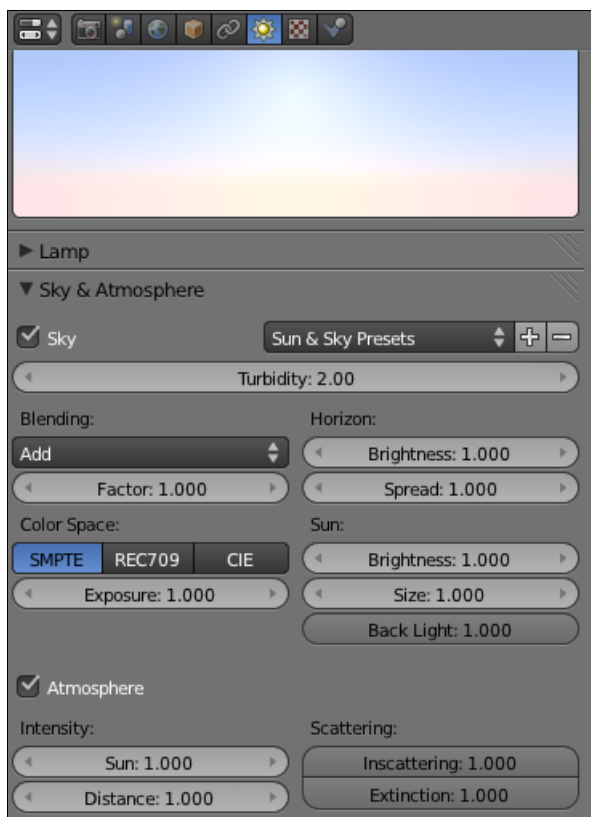


Рис. 7.11. Дополнительные опции Sun

Закладка опций **Atmosphere** позволяет использовать эффект рассеивания лучей солнца, которые проходят через атмосферу. Например, объекты на горизонте окажутся в дымке.

Здесь группа параметров **Intensity** поможет настроить интенсивность солнечного света. Так, при увеличении значения поля **Sun**, далекие объекты будут выглядеть более синими. А группа параметров **Scattering** отвечает за рассеивание солнечных лучей.

Набор этих в целом несложных параметров поможет создать прекрасное небо с реальным солнцем (рис. 7.12).

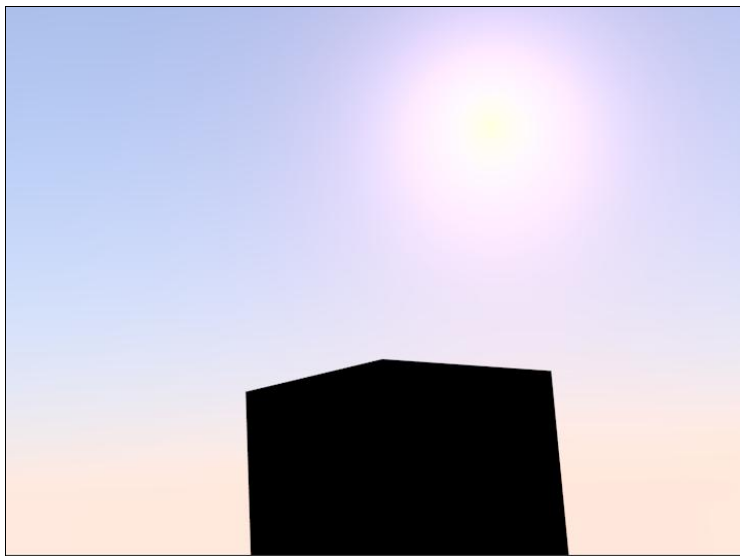


Рис. 7.12. Пример рендера неба

7.3. Работа с камерой

Камера — это окно в мир Blender. Именно с ее ракурса осуществляется конечная обработка сцены (рендер). После открытия программы в новой сцене всегда присутствует одна камера (рис. 7.13).

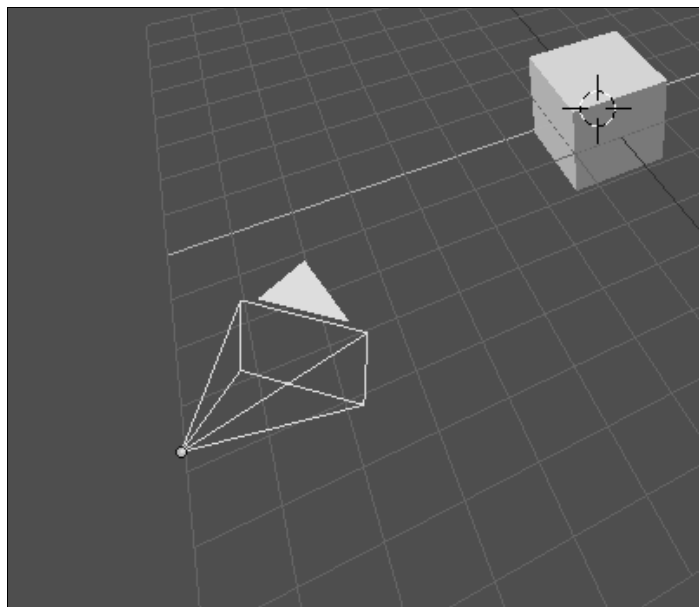


Рис. 7.13. Камера в сцене

Камера — это типичный объект сцены, которым можно управлять стандартными манипуляторами. Она может участвовать в анимации, использовать модификаторы и **Constraint**. Так, очень часто для автоматического слежения камеры за объектом применяется ограничитель **Track To**.

Если нажать клавишу <NumPad 0>, то Blender переключит просмотр активного окна **3D View** в режим камеры (рис. 7.14).

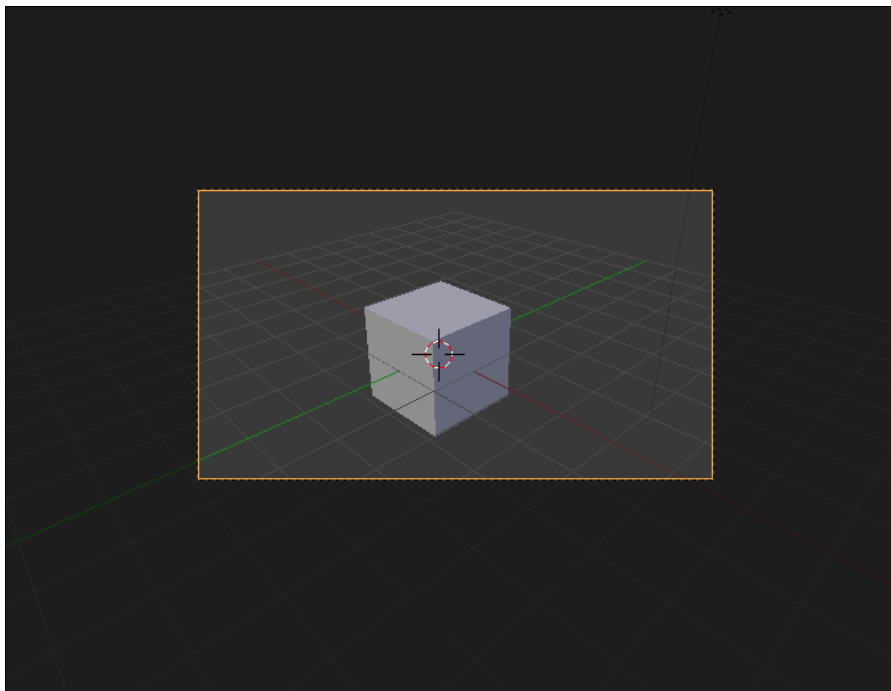


Рис. 7.14. Типичный вид просмотра **Camera View**

То, что находится в рамке просмотра, попадет в кадр обработки, вся остальная часть сцены затемнена. Степень затемнения неактивной зоны можно регулировать с помощью поля **Passepartout** в группе **Display**.

Обратите внимание на то, что активная рамка камеры по умолчанию является широкоформатной (в пропорции 16:9). Ее вид в первую очередь зависит от установленных параметров рендера (рассматриваются в следующей главе), но немаловажное значение имеют собственные настройки. Они располагаются в окне **Properties** (рис. 7.15).

СОВЕТ

Помимо наличия активной рамки при включении просмотра **Camera View**, программа может выводить дополнительную рамку **Title Safe** (Безопасная зона титров). Она очерчивает зону, которая является безопасной для вывода ключевой информации, например текста. Это сделано для того, чтобы такая информация обязательно была видна на любых типах телевизоров, которые могут произвольно обрезать картинку. Рамка **Title Safe** доступна после включения аналогичной опции в группе **Display** панели **Camera**.

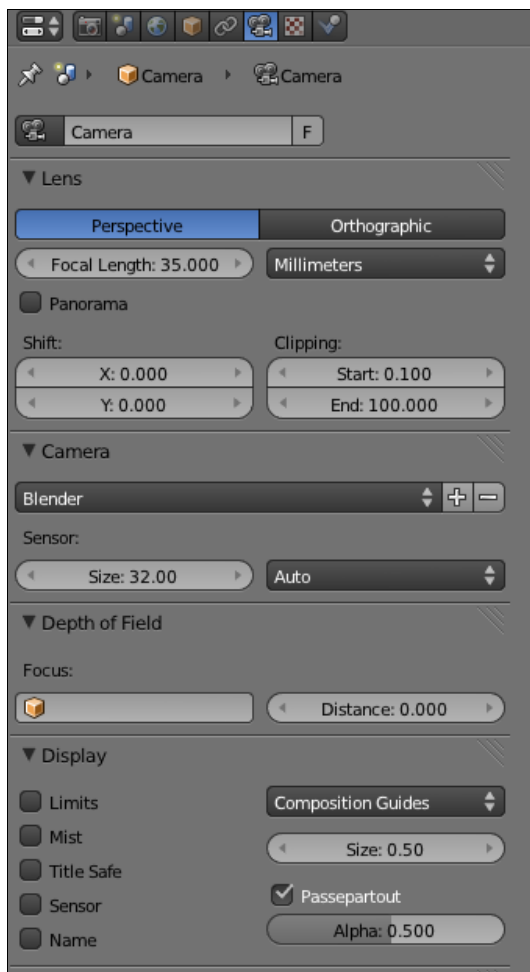


Рис. 7.15. Настройки камеры

Стандартная настройка камеры позволяет ей "видеть" объект на очень большом расстоянии. Вы можете ограничить видимость сцены с помощью группы опций **Clipping** (Обрезка). Здесь имеются параметры **Start** (Начало) и **End** (Конец).

Для удобства регулирования расстояния просмотра объект **Camera** может визуально отобразить в окне **3D View** соответствующую прямую. Для этого включите опцию **Limits** (Лимиты) на закладке **Display** и перейдите в любой другой вид просмотра (рис. 7.16).

После установки зоны просмотра нужно определиться с линзой для камеры. Обычно используется стандартная 35-миллиметровая линза. Она же и установлена по умолчанию в параметре **Focal Length** (Фокусное расстояние). Изменяя его значение, можно управлять видимым охватом сцены.

Как и в реальной камере, вы можете управлять ее фокусом, т. е. способностью максимально четко просматривать нужный объект. Все остальное обычно при этом расплывается.

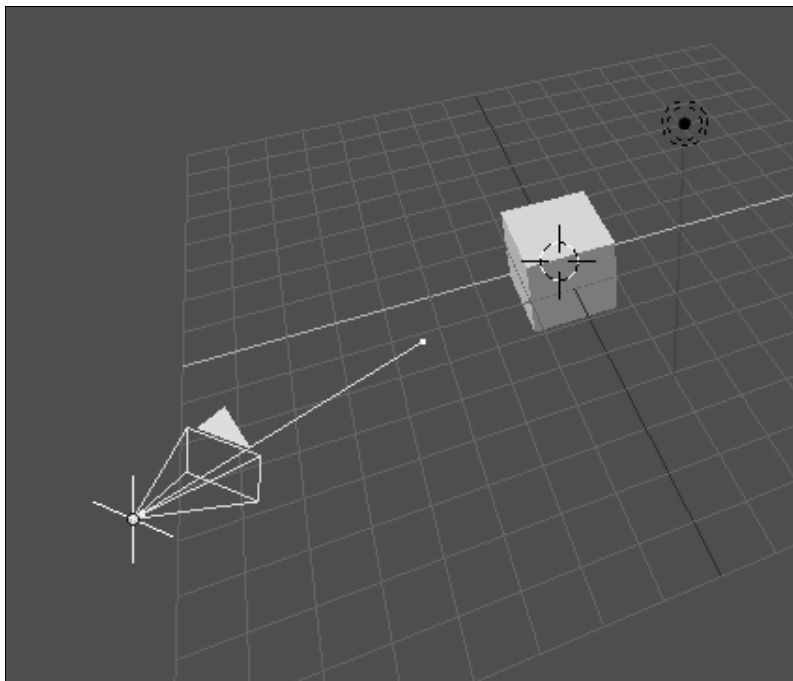


Рис. 7.16. Здесь куб не попадает в поле зрения камеры

Параметры фокуса находятся в группе опций **Depth of Field** (Глубина резкости). Вы можете установить постоянное значение для расстояния фокуса с помощью **Distance** (Дистанция), но куда удобнее функция автоматического фокуса для определенного объекта. Для этого служит поле **Focus** (Фокус), в котором нужно выбрать объект из списка.

СОВЕТ

В настройках камеры имеется группа **Camera**, которая позволяет определить зону чувствительности. Обычно здесь не требуется выполнять какой-либо корректировки. Но для максимального эффекта можно попытаться выбрать тип реальной камеры из соответствующего списка. Он насчитывает пару десятков камер известных брендов.

В проекте могут быть одновременно сразу несколько камер. Добавить их можно, как обычно, из меню **Add**. Но только одна должна быть активной и с ее позиции будет проводиться рендер сцены. Это можно сделать, если выделить нужную камеру и нажать клавиши <Ctrl>+<NumPad 0>.

7.4. Окружение: звезды, туман, глобальный свет

Когда в проекте расставлены объекты, настроена анимация и источники света, хочется быстрее обработать сцену и насладиться полученной картинкой. Но спешить не стоит! Blender имеет в своей копилке еще несколько интересных эффектов,

которые сделают выразительной любую сцену. Здесь мы рассмотрим последний этап работы над проектом — настройку параметров окружения.

По умолчанию фон сцены имеет светло-серый цвет. Его можно с легкостью поменять на иной другой цвет и даже добавить градиент. В некоторых случаях без корректировки фона просто не обойтись. Например, если результат обработки сцены в дальнейшем будет использован в какой-нибудь программе видеомонтажа. Современные редакторы видео позволяют совмещать кадры нескольких файлов с использованием различных эффектов. Наиболее популярным из них является **Green Screen** или производные от него. Смысл последнего заключается в том, что программа убирает из кадра все пиксели указанного цвета (обычно используется зеленый или синий). Так, обработав сцену с зеленым фоном, вы получите видео, которое с легкостью может быть использовано для совмещения с другим изображением. Настройки окружения размещаются на панели **World** (Окружение) окна **Properties** (рис. 7.17).

Здесь имеется привычное окно **Preview**, с помощью которого можно контролировать настройки окружения.

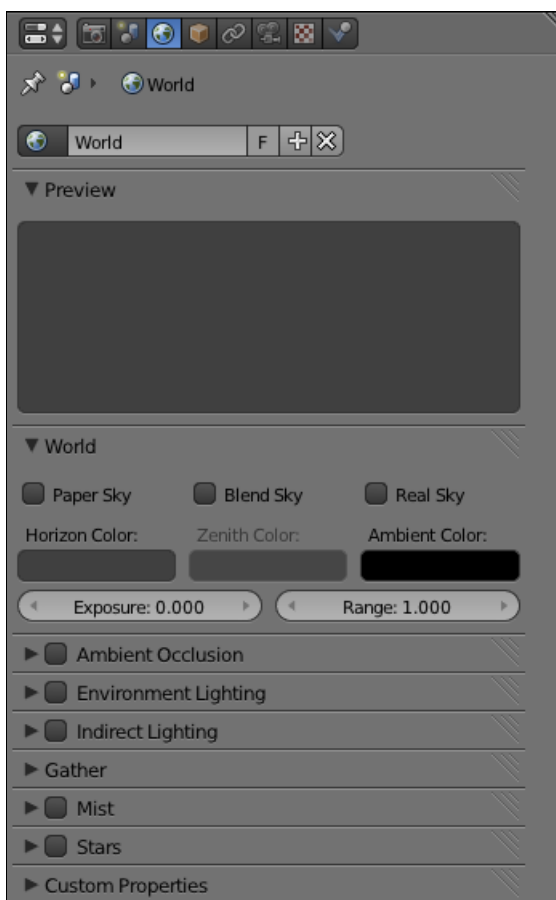


Рис. 7.17. Общий вид панели **World**

Фон сцены может быть изменен щелчком мыши по образцу с названием **Horizon Color** (Цвет горизонта). При этом он будет сплошным. Но есть еще возможность добавления градиента, который будет учитывать поворот и местоположение камеры. Так можно создать простое небо с горизонтом.

Опции управления фоном:

- ♦ **Blend Sky** (Смешивание фона) — режим смешивания, где участвуют цвета **Horizon Color** (Цвет горизонта) и **Zenith Color** (Цвет зенита). Это простой градиент, который создает переход по вертикали между двумя цветами;
- ♦ **Paper Sky** (Обертка фона) — его действие равноценно первому пункту, но при этом учитывается поворот камеры;
- ♦ **Real Sky** (Реальный фон) — двойной градиентный переход, где центр экрана занимает цвет **Horizon Color**. В этом случае учитываются расположение и ротация камеры.

Эти опции можно комбинировать между собой для получения нужного эффекта (рис. 7.18).

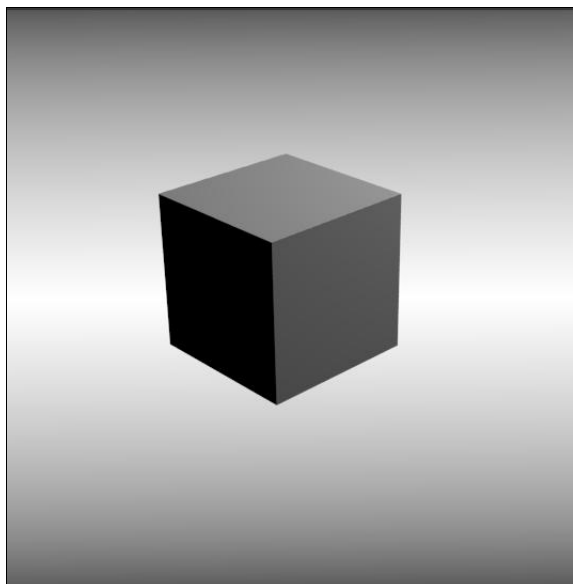


Рис. 7.18. Пример градиента фона

Обычно в сцене размещают несколько источников света для выгодной демонстрации модели, т. к. одной лампы явно для этого недостаточно. Blender предлагает несколько алгоритмов освещения, способных придать любой сцене яркость и насыщенность (рис. 7.19).

Наиболее популярный из них — это **Ambient Occlusion** (Объемный свет). Его можно рассматривать как дополнительное освещение сцены. **Ambient Occlusion** может прекрасно работать без источников света, но наличие последних необходимо для создания теней. Этот алгоритм позволяет получить качественную картинку,

правда за счет возросшей нагрузки на систему. Особо это касается прозрачных материалов с использованием **Ray Tracing**.

Настройки **Ambient Occlusion** расположены в одноименной группе. Достаточно установить галочку у названия группы и выбрать степень смешивания в параметре **Factor** (рис. 7.20).

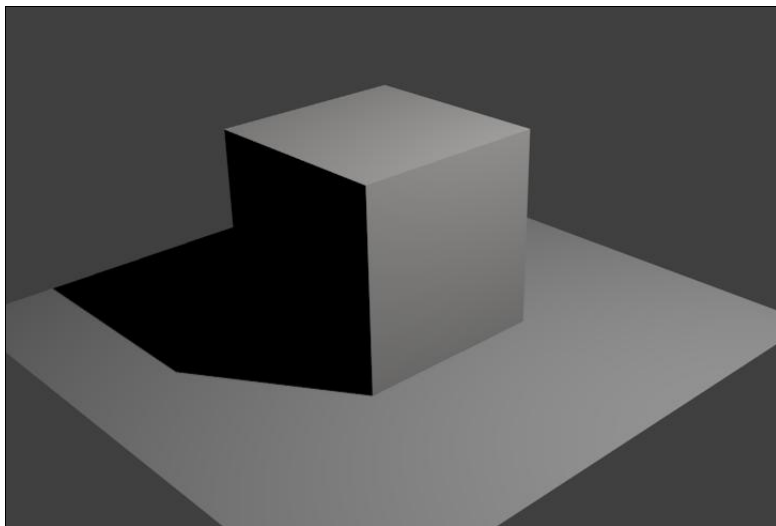


Рис. 7.19. Рендер сцены с одним источником света

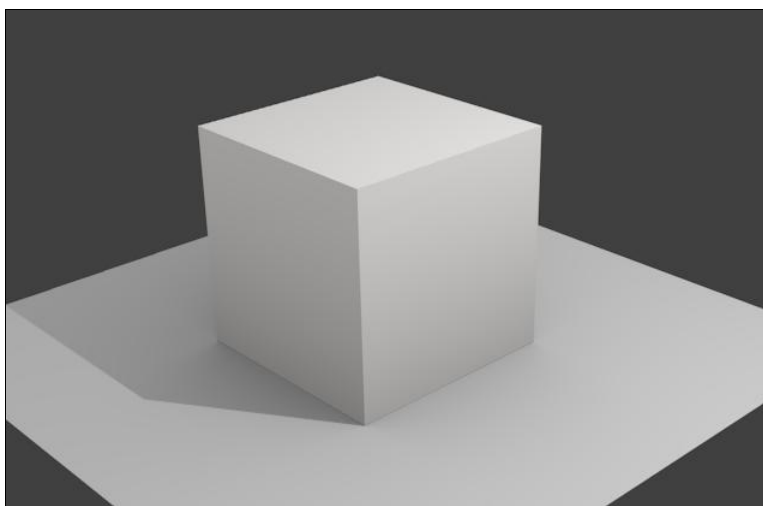


Рис. 7.20. Рендер сцены с **Ambient Occlusion** (**Factor** = 0.5)

Если **Ambient Occlusion** создает глобальное освещение сцены, то действие еще одного алгоритма **Indirect Lighting** (Ненаправленное освещение) позволяет создавать удивительные эффекты светящихся объектов. Подобно первому он также пре-

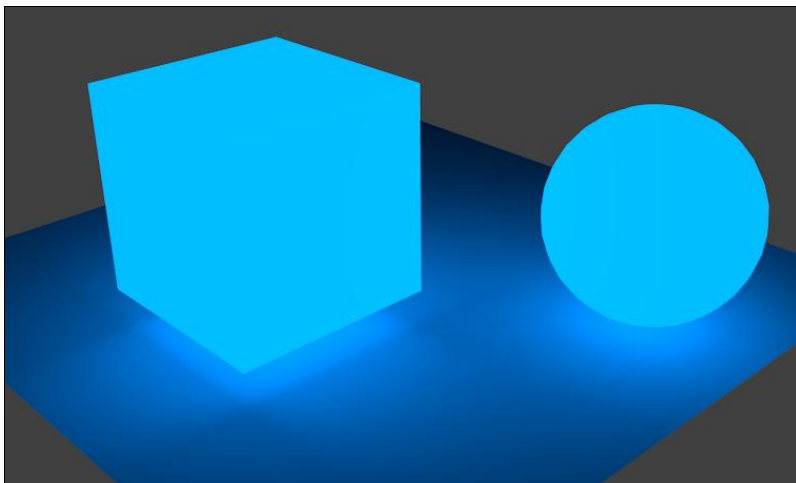


Рис. 7.21. Результат включения **Indirect Lighting**

красно обходится без основных источников света, но и тени не генерирует. Смысл **Indirect Lighting** заключается в отражении от объекта световых лучей (рис. 7.21).

В настройках материалов есть опция **Emit** (Излучение) в группе **Shading**, которая отвечает за свечение объекта. Но только в совокупности с **Indirect Lighting** можно получить эффект светового отблеска на близлежащих объектах.

Для использования этой функции обязательно включите параметр **Emit** в настройках материала. **Indirect Lighting** работает в связке с **Gather** (Сборка), где нужно выбрать метод **Approximate** (Приблизительный) (рис. 7.22).

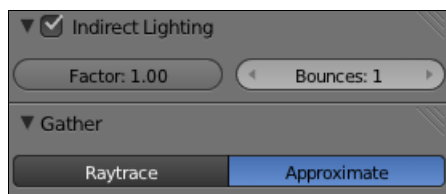


Рис. 7.22. Настройки **Indirect Lighting**

В настройках **Indirect Lighting** имеется всего два параметра:

- ◆ **Factor** (Коэффициент) — чем выше значение, тем выразительней эффект;
- ◆ **Bounces** (Возвраты) — увеличение этого параметра создает более мягкие переходы отражений.

Blender умеет создавать туман, который охватывает всю сцену (рис. 7.23). При чем имеется возможность регулирования его плотности, высоты и расстояния (рис. 7.24).

Параметры **Mist** (Туман):

- ◆ **Minimum** (Минимум) — минимальная интенсивность тумана;
- ◆ **Start** (Начало) — расстояние от камеры, откуда начинается туман;

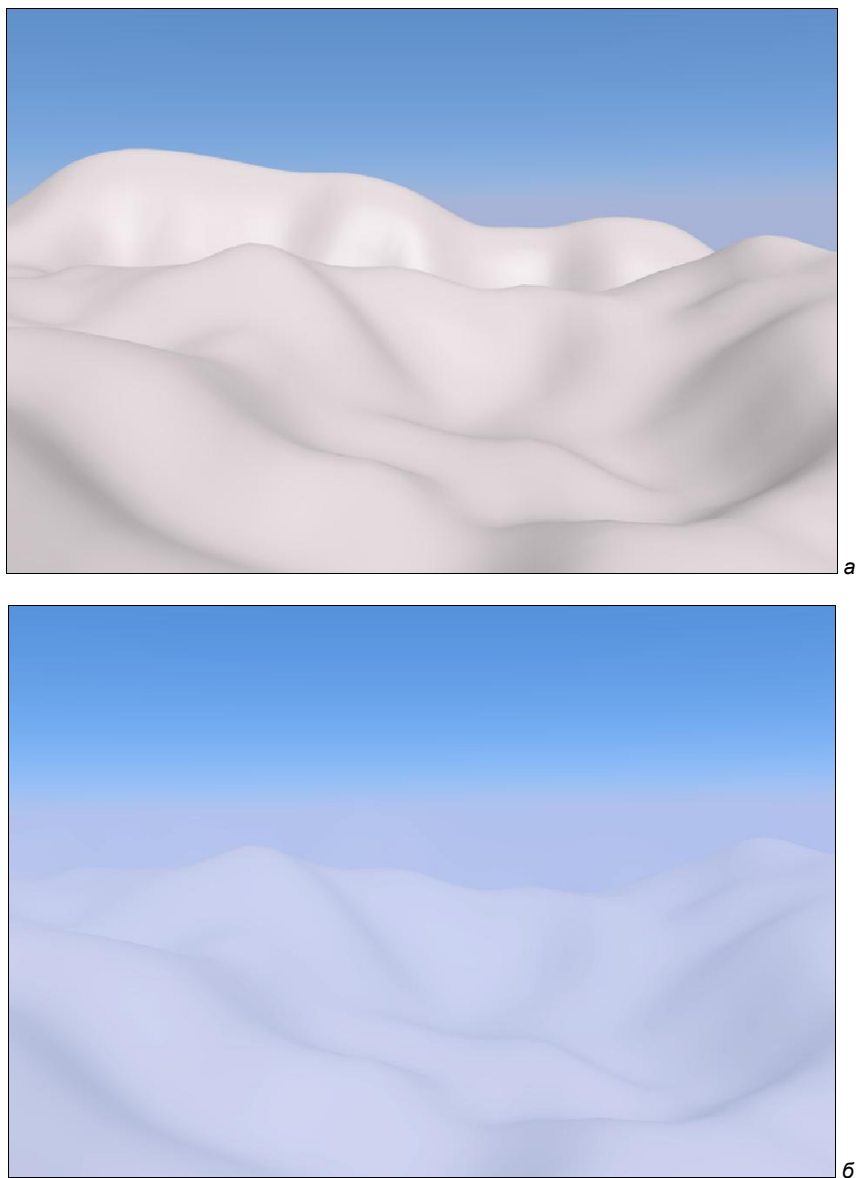


Рис. 7.23. Сцена без тумана (а) и с туманом (б)

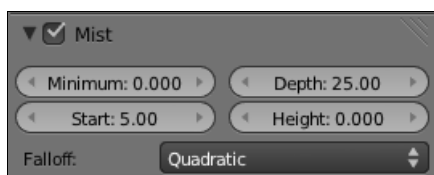


Рис. 7.24. Настройка тумана

- ◆ **Depth** (Глубина) — глубина тумана. Отсчет начинается от точки **Start**. Чем дальше от нее, тем хуже видимость;
- ◆ **Height** (Высота) — высота тумана. По умолчанию его плотность является неизменной. С помощью этого параметра можно установить наивысшую точку, где туман будет полностью отсутствовать. Это позволяет создавать более реалистичные сцены.

Можно вручную опытным путем добиться необходимой консистенции тумана, изменяя параметры **Mist** и проверяя результат рендера. А можно воспользоваться способностью камеры отображать в окне **3D View** отрезок видимости. Для этого нужно открыть параметры камеры, найти группу **Display** и включить пункт **Mist**.

Еще одним выразительным средством в Blender является возможность генерации звездного неба. Эти настройки располагаются на закладке **Stars** (рис. 7.25).

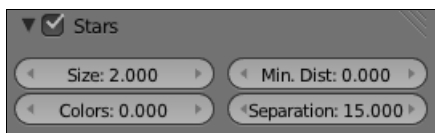


Рис. 7.25. Настройка звезд

Опции **Stars**:

- ◆ **Size** (Размер звезд);
- ◆ **Colors** (Цветность) — при минимальном значении цвет у звезд один — белый. Чем выше этот параметр, тем насыщенней и разнообразней становится цвет точек;



Рис. 7.26. Пример звездного неба

- ♦ **Min. Dist.** (Минимальная дистанция) — расстояние от камеры до звезд. Следите за тем, чтобы этот параметр был больше, нежели удаленность самого дальнего объекта в сцене от камеры. В противном случае вы рискуете получить звезду на самом объекте;
- ♦ **Separation** (Разделение) — расстояние между точками. Этим параметром можно регулировать плотность звездного неба.

Звезды — это не что иное, как объекты, подобные **Halo**. Blender выполняет смешивание их с цветом фона или атмосферными эффектами **Sun**. С помощью **Stars** и градиентного фона можно с легкостью добиться прекрасного звездного неба. Еще одной особенностью **Stars** является то, что они не привязаны к камере и при вращении последней остаются на месте (рис. 7.26).

7.5. Практика. Закат солнца

В этом уроке рассмотрим, как можно сделать солнечный закат с использованием расширенных настроек источника света **Sun**. Создайте новый проект и удалите куб.

Чтобы картинка была более интересной, добавим в сцену простой ландшафт на основе примитива **Plane**. Создайте плоскость и откройте панель свойств в окне **3D View** (<N>). Зрелищность заката сильно зависит от расположения всех объектов участников сцены. Поэтому большая часть манипуляций будет совершаться с помощью панели свойств.

Создать форму ландшафта можно разными способами, но наиболее простой — это воспользоваться процедурной текстурой по технологии **Displacement Map** (см. разд. 4.10).

Сначала нужно увеличить размер плоскости. Введите значение 8.300 в параметры **Scale** для всех осей объекта (рис. 7.27).

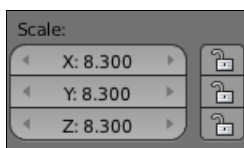


Рис. 7.27. Масштабирование **Plane** на панели свойств

После этой корректировки плоскость должна занять всю вспомогательную решетку в окне **3D View**. Займемся созданием ландшафта.

Откройте панель **Textures** в окне **Properties**. Создайте новую текстуру, нажав кнопку **New**.

В качестве текстуры будем использовать функцию **Cloud**, она должна быть по умолчанию в меню **Type**. В настройках текстуры найдите закладку **Clouds** и установите в поле **Size** значение 1.50. Это увеличит узор рисунка, т. к. имеющийся по умолчанию слишком мелкий.

Откройте панель модификаторов в окне **Properties** и добавьте к плоскости **Displace**. В его настройках, в свойстве **Texture** выберите созданную текстуру. При этом визуально в окне **3D View** ничего не изменится. Для того чтобы произошло смещение вершин, необходимо увеличить структуру решетки **Plane** с помощью инструмента **Subdivide**. Выполните пятикратную разбивку **Mesh**-объекта. Можете и больше, ведь чем качественней решетка, тем лучше будет выглядеть ландшафт. Но не переусердствуйте!

Включите сглаживание **Smooth** для плоскости в режиме **Object Mode**. Ландшафт готов (рис. 7.28).

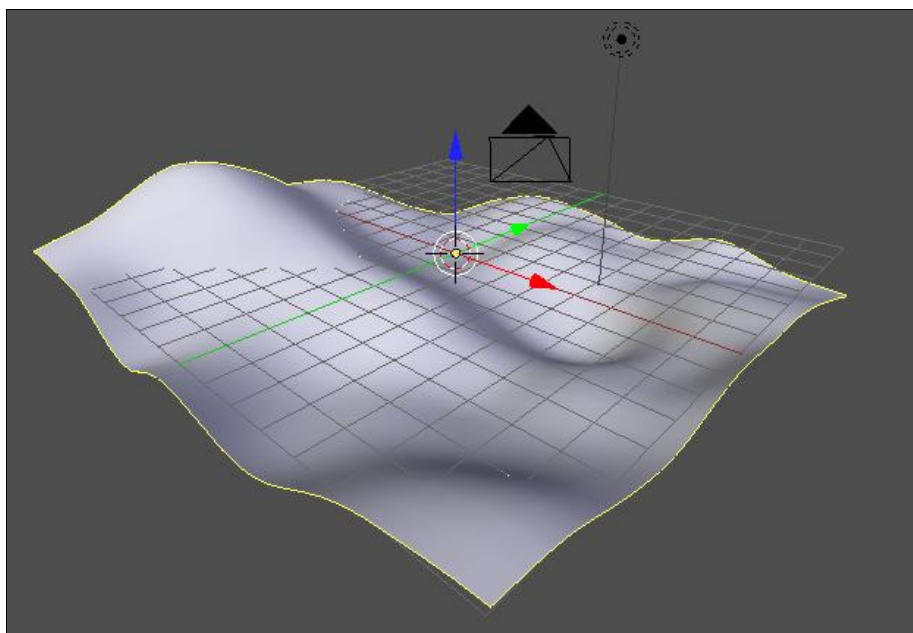


Рис. 7.28. Готовый ландшафт

Теперь поместим камеру так, чтобы она смотрела вдоль ландшафта в сторону горизонта. Выделите в сцене объект **Camera** и внесите следующие изменения на панели свойств:

- | | |
|---------------------------|--------------------------|
| ◆ Location X = 0 | ◆ Rotation X = 90 |
| ◆ Location Y = -12 | ◆ Rotation Y = 0 |
| ◆ Location Z = 2 | ◆ Rotation Z = 0 |

Если все сделано правильно, то вид из камеры (<NumPad 0>) выдаст картинку, как на рис. 7.29.

Конечная картинка неба зависит от ротации источника света. В случае с **Sun** местоположение объекта не принципиально. Выделите источник света, откройте его настройки (закладка **Light** в окне **Properties**) и поменяйте его тип на **Sun**, нажав соответствующую кнопку. Включите группы **Sky** и **Atmosphere**.

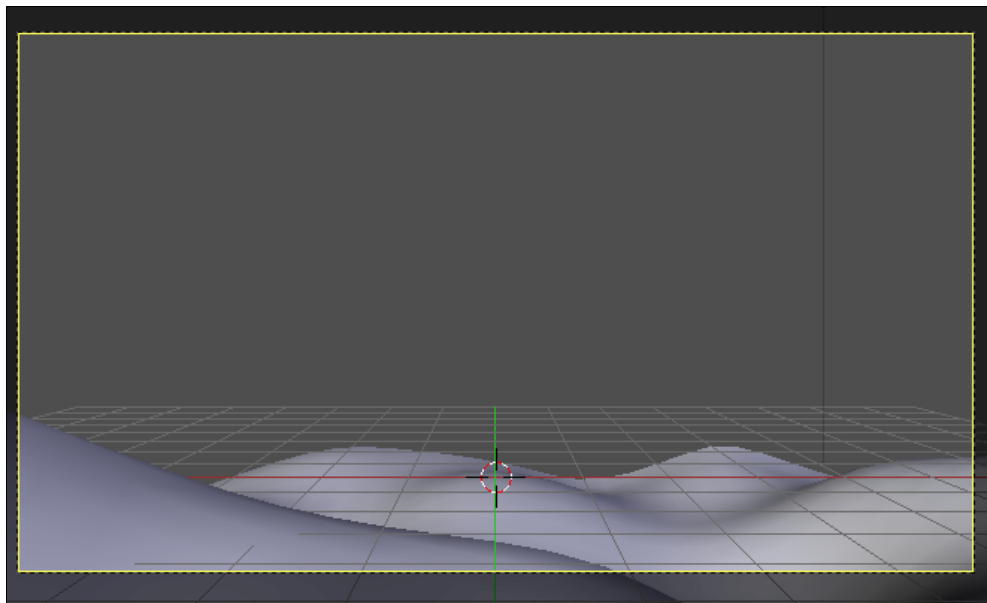


Рис. 7.29. Ландшафт с позиции камеры

Теперь развернем источник света так, чтобы лучи проецировались вдоль ландшафта под углом соответствующему закату. Внесите следующие коррективы на панели свойств объекта:

- ◆ **Rotation X** = 54
- ◆ **Rotation Y** = -83
- ◆ **Rotation Z** = -161

Попробуйте обработать сцену с помощью клавиши <F12> (рис. 7.30).



Рис. 7.30. Солнце размещено верно, но это не закат

На закат то, что вы получите, похоже очень мало. Солнце не видно, небо какое-то желтое. Здесь явно требуется корректировка параметров **Sun**.

Не будем трогать горизонт, займемся видом солнца. На закладке **Sky** имеется группа **Sun**, которая отвечает за внешний вид светила (см. рис. 7.11). Увеличим его размер и яркость:

◆ **Brightness** = 3

◆ **Size** = 10

Если вы обработаете сцену, то результат будет лучше, но все же не то. Слишком яркое небо, слишком насыщенное солнце.

Обратимся к опциям **Color Space**, которые позволяют контролировать цветовое пространство света. Нажмите кнопку **CIE** и установите в поле **Exposure** значение 0.2. Это значительно смягчит свечение солнца и насыщенность фона.

Последним штрихом в создании заката будет изменение фона сцены. Вы ведь помните, что **Sky** смешивается с этим цветом?

Откройте параметры **World** в окне **Properties**. По умолчанию цвет фона является серым. Щелкните по образцу **Horison Color** и замените цвет на черный. Вот теперь у вас получился настоящий закат (рис. 7.31).

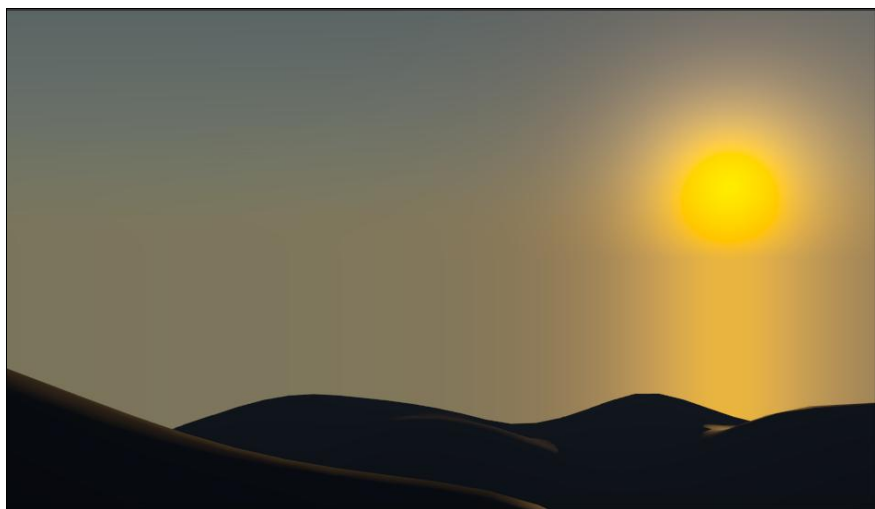
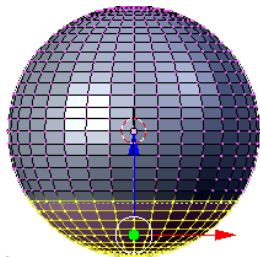


Рис. 7.31. Закат солнца

ГЛАВА 8



Система рендеринга Blender

Рендер — вот оно сладкое слово для каждого моделлера. Это момент истины, заключительный этап работы над проектом. И хотя создание сцены завершено, от правильной настройки системы обработки зависит очень многое. Малейшая неточность и результат будет испорчен. А ведь визуализация даже одного кадра может длиться часами.

В этой главе вы узнаете, что такое рендер, как его настраивать, как сбалансировать качество и быстроту для достижения наилучшего результата.

8.1. Основы обработки

На протяжении всей книги вы часто встречали слово "рендер" и даже выполняли быстрый прогон сцены с помощью клавиши <F12>.

Рендер (Render) — это система визуализации проекта и сохранение результата в виде графических или видеофайлов. В то же время рендером называется собственно результат обработки.

В мире существует много систем визуализации 3D и некоторые из них поддерживаются программой. Поэтому различают встроенные рендеры и сторонние. В этой версии Blender уже имеется три встроенных обработчика, которые можно найти и выбрать в меню **Engine** (Движок), расположенном в заголовке главного окна программы (рис. 8.1).

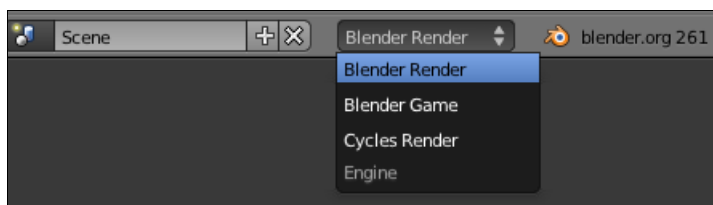


Рис. 8.1. Встроенные рендеры Blender

Рендеры Blender:

- ◆ **Blender Render** — это самый первый визуализатор программы, он используется по умолчанию;
- ◆ **Blender Game** — движок для визуализации сцены в режиме реального времени. Blender — это не только программа для создания 3D, но и мощный игровой конструктор. Выбирайте этот пункт, если вы создаете игру или приложение реального времени;
- ◆ **Cycles Render** — новый рендер, появившийся в версии программы 2.61. Разработчики пророчат ему великое будущее и в дальнейшем предполагают использовать его по умолчанию. Движок сырой, но имеет очень хорошие возможности.

Помимо встроенных обработчиков вы можете использовать некоторые другие. Наиболее популярными являются: **YafRay**, **LuxRender**, **Pov-Ray**, **Rendermfarm**. Эти движки подключаются к программе с помощью встроенного механизма плагинов.

Каждый рендер хорош по-своему, у каждого есть свои сильные и слабые стороны. Так, к примеру, новейший **Cycles** имеет возможность ускорения просчета за счет ресурсов графического GPU (видеоплаты). **YafRay** советуют использовать для сцен, содержащих материалы с преломлением и отражением. Но нужно знать, что в большинстве случаев сцену и материалы объектов придется подгонять под требования конкретного рендера. Поэтому в данной главе будет рассмотрен лишь стандартный **Blender Render**. Его возможностей вполне достаточно для визуализации сложнейших сцен.

Обработчик Blender способен сохранять результат в различных форматах графических и видеофайлов. Так, при нажатии клавиши <F12> выполняется обработка одного текущего кадра, который можно сохранить на диск. Это делается либо с помощью меню **Image | Save As Image** в окне результата обработки, либо просто нажав клавишу <F3>. По умолчанию программа предложит сохранить картинку в формате PNG. А вот изменить его можно в настройках рендера (рис. 8.2).

В верхней части панели **Render** присутствуют две большие кнопки, которые позволяют запустить процесс обработки — это **Image** (Картинка) и **Animation** (Анимация). Уже по названию понятно, что за что отвечает. Процесс обработки Blender демонстрирует в окне **Image Editor**, которое автоматически запускается при старте. Но вы можете выбрать иной тип вывода в меню **Display**.

Однако перед обработкой сначала нужно правильно определиться с выбором выходного формата и качеством картинки. Вообще, подобные вещи лучше делать на самом первом этапе создания сцены. Так как даже банальное изменение разрешения картинки может привести к необходимости корректировки объектов в сцене и анимации.

Настройки изображения выполняются на закладке **Dimensions** (Размеры) (рис. 8.3).

При запуске программа предлагает обрабатывать картинку в разрешении Full HD, т. е. 1920 на 1080 пикселей. Это, конечно, великолепно, но в большинстве случаев излишне. Вы можете вручную установить разрешение в полях **Resolution**. Только учтите, что необходимо правильно установить соотношение сторон (**Aspect Ratio**)

и частоту кадров (**Frame Rate**). Ведь существуют определенные стандарты телевизионного вещания (на них ориентируются форматы видео). Так, у нас в стране используется PAL, который имеет разрешение 720×576 пикселей с частотой 25 кадров в секунду. В странах Европы популярен NTSC с совсем другими пропорциями и частотой, а ведь есть еще всевозможные варианты HD. Если вы не знаете конкретных данных необходимого стандарта, то лучше воспользоваться заготовками Blender в меню **Dimensions**. Здесь есть практически все популярные телевизионные стандарты разных стран.

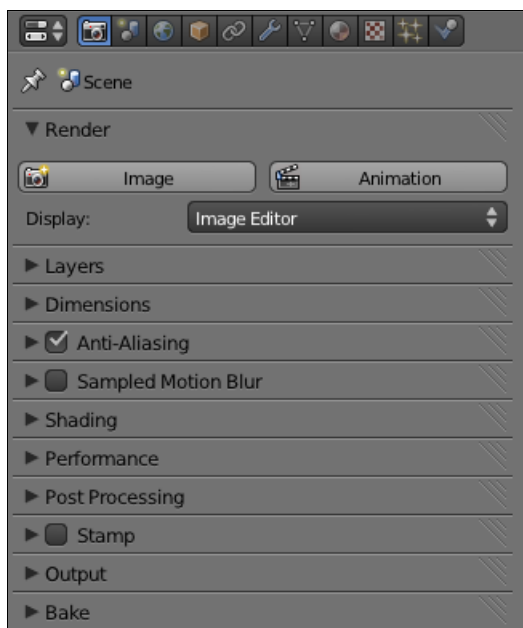


Рис. 8.2. Настройка рендера

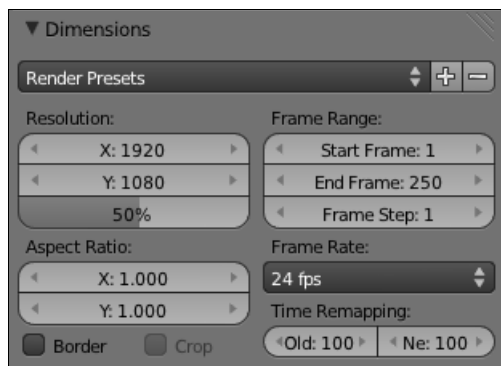


Рис. 8.3. Настройки разрешения

После того как вы определились с разрешением картинки, нужно выбрать, в каком формате будет сохранен результат. Это делается на закладке **Output** (Выход) (рис. 8.4).

По умолчанию Blender сохраняет результат в папке Tmp. В зависимости от операционной системы, она может находиться в разных местах. Вы можете указать другой путь, щелкнув по кнопке с характерным рисунком папки.

Главный выбор осуществляется в меню **File Format** (Формат файла) (рис. 8.5).

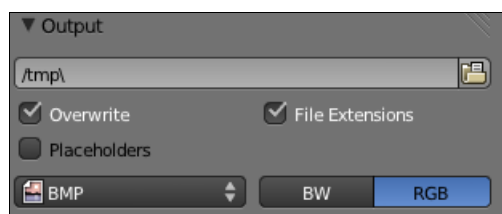


Рис. 8.4. Настройка вывода

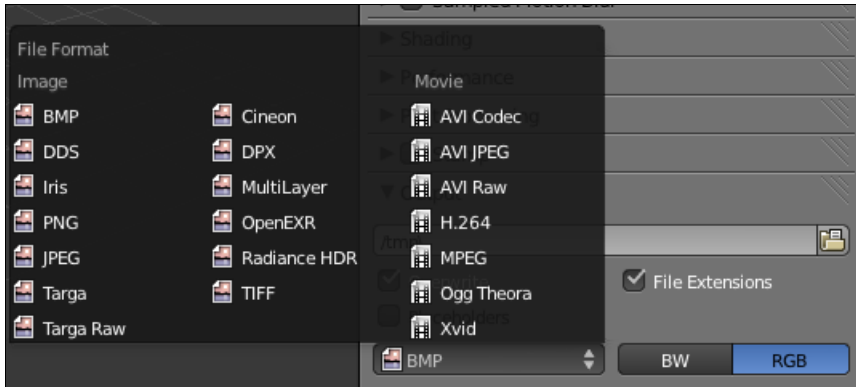


Рис. 8.5. Форматы файлов, поддерживаемых программой

Форматы разбиты на две группы: **Image** (Картинки) и **Movie** (Фильм). При выборе нужного пункта программа предложит дополнительные настройки, характерные для него. Если с графическими файлами все просто, то видео рассмотрим подробнее.

Список форматов видео достаточно обширен: AVI Codec, AVI JPEG, AVI Raw, H.264, MPEG, Ogg, Xvid. Многие, наверное, вам знакомы. Но что же выбрать из этого обилия?

Это действительно очень сложный этап, ведь в большинстве случаев для обработки Blender будет использовать сторонние кодеки. Конечно, многое зависит от установленных параметров на закладке **Dimensions**. Некоторые кодеки способны выдавать приемлемое качество только в определенном разрешении.

Первые три варианта AVI являются ничем иным, как собранными в общий контейнер графическими файлами соответствующего типа. Так, AVI Codec создает видеофайл без какого-либо сжатия, один к одному, с максимальным качеством и большого объема.

Все остальные типы — это кодеки, осуществляющие сжатие видео по определенным алгоритмам. Если выбрать любой из них, то откроются дополнительные настройки с большим количеством опций (рис. 8.6).

Нужно быть специалистом, чтобы разобраться в этих параметрах или идти опытным путем. Правда, разработчики озаботились созданием заготовок для кодеков, которые содержатся в меню закладки **Encoding**.

По личному опыту замечу, что лучше всего обрабатывать видео вообще без сжатия с использованием AVI Codec. А готовый вариант уже можно "пережарить" с помощью любой сторонней программы. Так получается гораздо быстрее.

Но есть вариант лучше. Blender, как и любая программа, может дать сбой в процессе обработки. Будет очень обидно, если несколько часов рендера пойдут насмарку. Если в качестве выходного формата выбрать графический файл (например PNG), а запустить обработку анимации, то Blender выполнит рендер каждого кадра в отдельный графический файл. Причем все они будут именованы порядковыми цифрами. В этом случае не страшен сбой программы, ведь после перезапуска можно

продолжить обработку с сорвавшегося места. В дальнейшем это множество файлов можно поместить в виде последовательности в одну из монтажных программ. Например, популярный видеоредактор Adobe Premier может импортировать порядковые графические файлы, как единый поток.

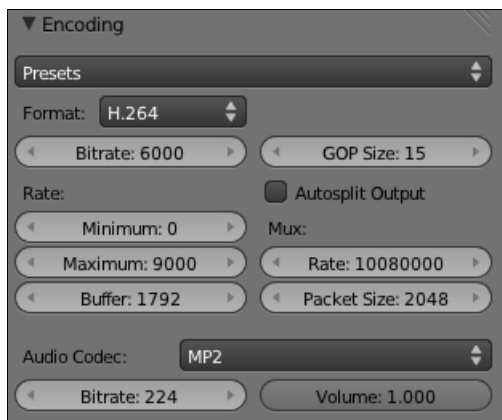


Рис. 8.6. Настройки кодеков

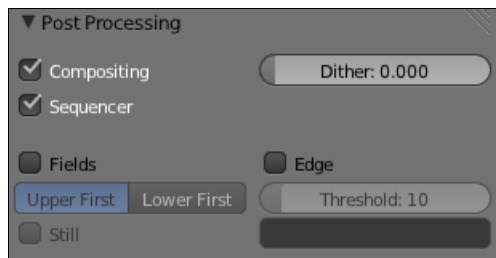


Рис. 8.7. Настройка Fields

Вроде бы с форматами определились и можно приступать к обработке, но не тут-то было. Есть еще такое страшное понятие, как поля (**Fields**).

Для того чтобы человеческий глаз перестал фиксировать смену картинок и ощущал их как единое целое, они должны чередоваться со скоростью не менее 25 кадров в секунду. Но при резких движениях предметов в кадре этого явно недостаточно. Поэтому придумали разбивать каждый кадр на два полукадра или поля. Они получили названия: верхнее поле (**Upper Field**) и нижнее поле (**Lower Field**). Во время проигрывания поля опять смыкаются в определенном порядке. Этот фокус позволяет показывать удвоенное количество кадров, т. е. для PAL физические 25 превращаются в 50. Все бы ничего, но существующие форматы видео используют разные поля, и если вы ошиблись в выборе правильного первого поля, то полученное видео может неприятно дергаться. Опять-таки необходимо определиться, в каком конечном формате должен получиться ролик и затем выставлять нужный порядок полей. Настройка полей осуществляется на закладке **Post Processing** (Постобработка) в группе **Fields**. Установите галочку в названии группы и нажмите кнопку с нужным полем (рис. 8.7).

По умолчанию Blender обрабатывает анимацию в соответствии с параметрами **Start** и **End** окна **Timeline**. Но можно выбрать иные кадры в настройках рендера. На закладке **Dimensions** имеются поля **Start Frame** и **End Frame**, которые как раз и указывают программе, что нужно обработать (см. рис. 8.3). В этой же группе есть еще один важный параметр — **Frame Step** (Шаг кадра), отвечающий за смещение. Так если в нем установлена единица, то программа выполнит обработку каждого кадра без пропуска.

Вот теперь рендер полностью настроен, и можно смело жать кнопку **Animation!**

8.2. Дополнительные возможности

В первом разделе главы были рассмотрены настройки, являющиеся обязательными для визуализации сцены. Но на этом возможности рендера не заканчиваются. Имеется еще целый ряд параметров, которые могут улучшить или изменить результат обработки.

Прodelайте небольшой опыт. Поднимите руку и быстро опустите ее. Вы заметите, что чем выше скорость, тем более смазанным выглядит движение руки. Это относится к любым быстро двигающимся предметам.

Если в сцене имеется модель, которая быстро перемещается из точки в точку, то при обработке вы увидите неприятное подергивание видео. Глаза вполне могут проследить все этапы движения объекта. Да, можно использовать поля, чтобы увеличить объем картинок, проходящих за секунду времени (с полями Blender обрабатывает в два раза больше кадров). Но и этого бывает недостаточно.

Blender поддерживает технологию **Motion Blur** (Смазывание движения). Технически это выглядит, как создание дополнительных шагов движения объектов в одном и том же кадре. Использование **Motion Blur** очень сильно отражается на времени обработки, но это единственный способ отобразить движение в кадре без рывков.

Настройки **Motion Blur** находятся на одноименной закладке панели **Render** (рис. 8.8).



Рис. 8.8. Настройки **Motion Blur**

В поле **Motion Samples** нужно указать количество дубликатов в кадре, а параметр **Shutter** отвечает за расстояние между ними. К сожалению, дать рекомендации по настройкам **Motion Blur** невозможно, т. к. все зависит от текущего проекта. Неплохое качество могут дать значения **Samples** = 6 и **Shutter** = 1. Но это опять-таки очень приблизительно.

Motion Blur относится к так называемым постэффектам. Это означает, что он выполняется после обработки самой сцены.

Рендер программы имеет еще несколько полезных возможностей. Например, вы можете придать мультяшный, нарисованный вид своим героям в сцене. Всего несколько настроенных параметров и программа обработает объекты с дополнительной обводкой ребер (рис. 8.9).

Эти настройки выполняются на закладке **Post Processing** (Постобработка). Включите опцию **Edge** (Ребро), установите требуемую толщину обводки в **Threshold** и выберите нужный цвет (рис. 8.10). Если установить для шейдеров материалов алгоритм **Toon**, то результат станет еще лучше.

В параметрах рендера есть группа опций, объединенная закладкой **Anti-Aliasing** (Сглаживание). Это не что иное, как дополнительная обработка для удаления характерных ступенек на изображении (рис. 8.11).

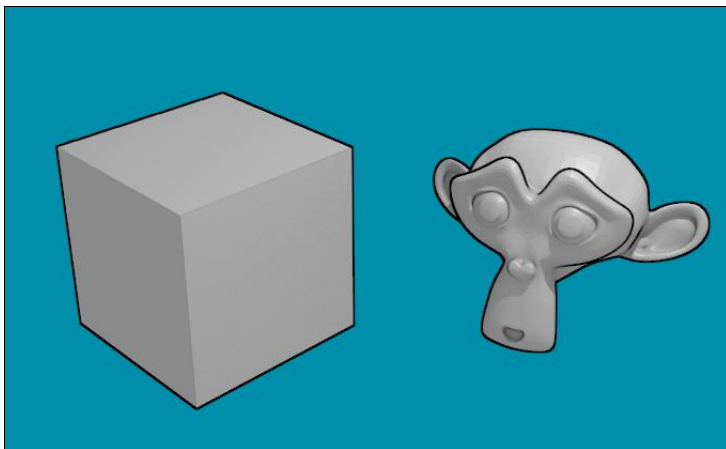


Рис. 8.9. Пример мультяшной обработки

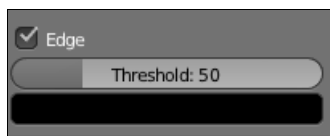


Рис. 8.10. Настройки Edge

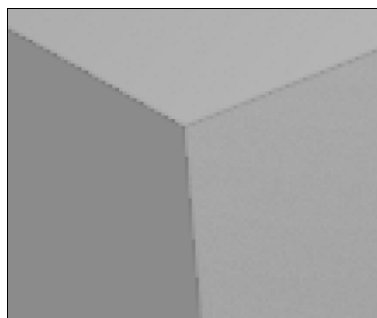


Рис. 8.11. На ребрах куба видны ступеньки

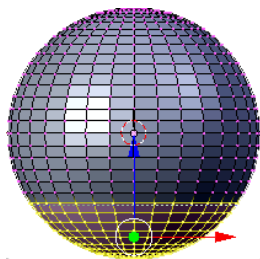
По умолчанию функция **Anti-Aliasing** включена, но вы можете подкорректировать ее настройки для получения оптимального результата. Группа кнопок с числами 5, 8, 11, 16 отвечает за качество шагов обработки. Чем больше значение, тем лучше качество. Хотя, даже стандартно выбранная пятерка дает хороший результат. Если вы хотите обрабатывать сцену в большом разрешении, то стоит выбрать более высокое значение **Anti-Aliasing** (рис. 8.12).



Рис. 8.12. Настройки Anti-Aliasing

Помимо визуализации сцены, рендер программы способен выпекать различные текстуры: **Normal Map**, **Displacement**, **Shadow** и др. Работа с некоторыми из них описана в *главе 5*. Настройки выпечки находятся на закладке **Bake**.

ГЛАВА 9



Что еще умеет Blender

В течение восьми глав вы прошли все этапы работы со сценой, начиная от создания моделей и заканчивая настройкой рендера. Казалось, что еще можно рассказать об этой программе. Но Blender многогранен и, чтобы описать все его нераскрытые возможности, понадобится еще не одна сотня страниц.

Эта глава посвящена нескольким уникальным инструментам, которые могут облегчить труд моделлера. Использовать в своей работе их вовсе необязательно, но знать, что они существуют — полезно.

9.1. Изучаем *Node Editor*

Node Editor (Редактор узлов) — чрезвычайно мощное средство для создания композиции проекта. С его помощью можно настраивать как поведение и свойства объектов, так и рендер сцены целиком. Вы можете редактировать материалы, текстуры, управлять слоями, постэффектами и многое другое. Все настройки в этом режиме отображаются в виде иерархии графических блоков, которые соединяются между собой проводниками и которые можно с легкостью редактировать (рис. 9.1).

Как и обычно, окно **Node Editor** можно открыть в любой области Blender, выбрав соответствующий пункт из меню **Editor Type** (см. рис. 1.3). Но этот редактор настолько популярен среди пользователей Blender, что разработчики программы подготовили специальную раскладку окон, которая называется **Compositing** (Композиция) и находится в меню **Screen Layout** (см. рис. 1.5). В макет входят пять окон: **Node Editor**, **UV/Image Editor**, **Properties**, **3D View**, **Timeline**.

В этом редакторе любые действия базируются на узлах, или нодах. Это самостоятельные единицы, которые содержат настройки какой-либо функции в Blender. Нода может быть создана из специального меню, перенесена в любое место окна редактора, свернута или удалена из проекта. Управлять нодой очень просто. Для перемещения ее в окне нужно поместить курсор мыши в область ноды, нажать левую кнопку и, удерживая ее, передвинуть на новое место. При необходимости узел можно свернуть, если щелкнуть по стрелке в заголовке около названия (рис. 9.2).

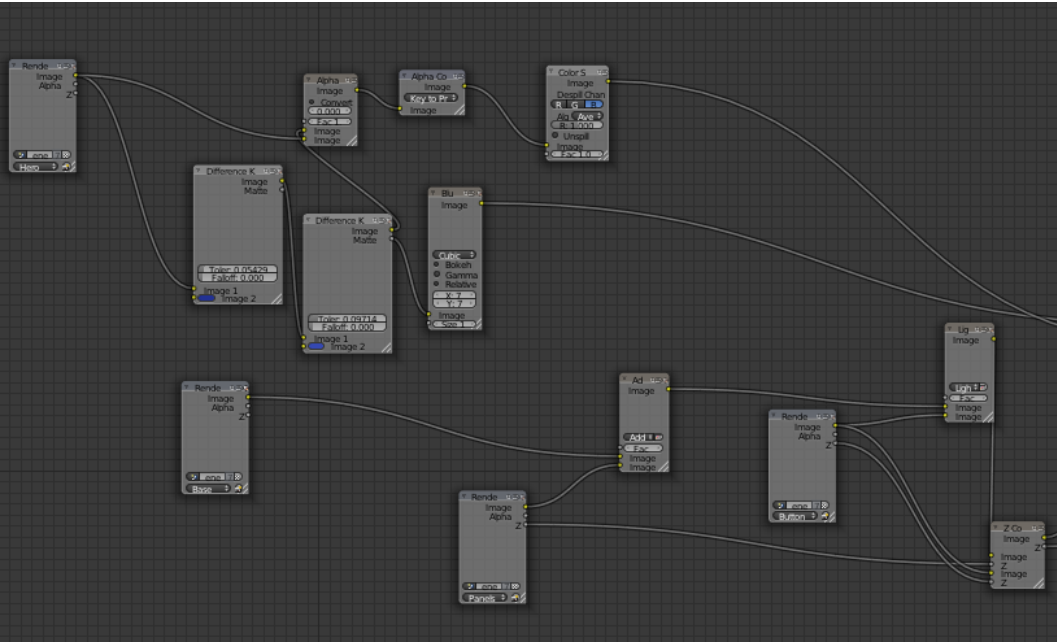


Рис. 9.1. Типичный вид окна Node Editor

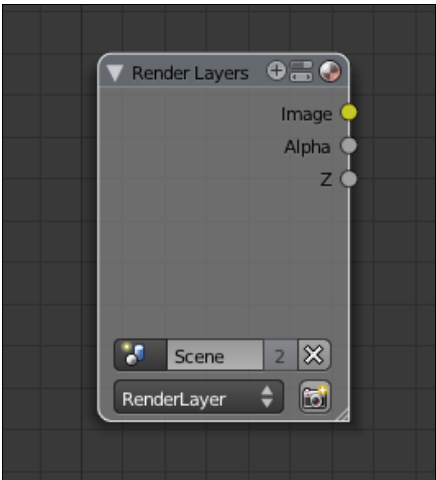


Рис. 9.2. Нода редактора



Рис. 9.3. Опция активации Node Editor

Все настройки, выполненные в этом редакторе, являются уникальными и никак не пересекаются с настройками, сделанными обычным путем. Так если к объекту уже присвоен материал в окне **Properties**, то в **Node Editor** вам придется сделать это еще раз. Поэтому в заголовке **Node Editor** имеется специальная опция **Use Nodes**, включение которой заставляет Blender выполнять любые действия в соответствии с настройками узлов (рис. 9.3).

Ноды могут использоваться для создания и контроля материалов, текстур и, собственно, композиции. Поэтому в заголовке окна **Node Editor** имеется группа кнопок для выбора нужного режима (рис. 9.4).

Первая кнопка (они рассматриваются слева направо) включает режим материалов (**Shader Nodes**). Здесь можно создать и настроить материал. Вторая кнопка (**Texture Nodes**) дает полный контроль за текстурами сцены, причем не только объектов, но и окружения или кистей. Третья кнопка ответственна за композицию (**Compositing Nodes**). Эти возможности позволяют создавать конвейер рендера, настраивать постэффекты, управлять слоями.

Сила **Node Editor** в том, что узлы могут быть связаны между собой в единый конвейер, который на выходе выдаст нужный результат. Рассмотрим простой пример создания материала.

Откройте макет окон **Compositing**. Переключите **Node Editor** в режим работы с материалами (см. рис. 9.4). Установите галочку в опции **Use Nodes**.

После этих действий в окне редактора появится минимальная заготовка, необходимая для функционирования материала (рис. 9.5).



Рис. 9.4. Выбор режима работы редактора

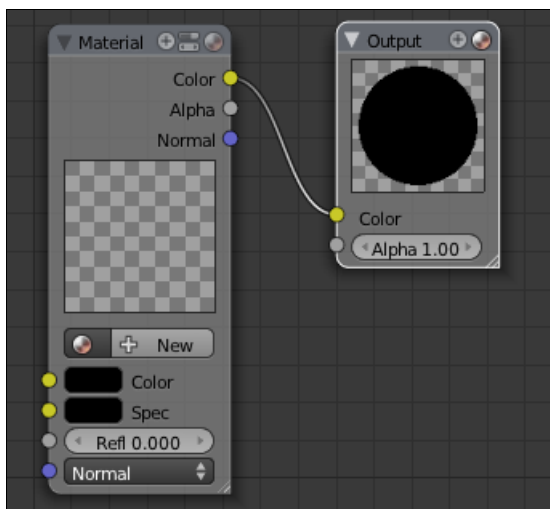


Рис. 9.5. Режим материалов

Узел **Material** предназначен для создания, настройки или выбора материала. Обратите внимание, что на одноименной панели окна **Properties** появилась предупреждающая надпись "No material node selected" (не выделен материал ноды).

Выделите куб в сцене. Вы помните, что при создании новой сцены куб по умолчанию уже имеет материал, но данная нода никак себя не проявила после выделения. Все правильно, с точки зрения **Node Editor** данный объект не имеет материала.

Настройки большинства узлов соответствуют таким же на панелях окна **Properties**. Так, область ноды **Material** содержит минимальные параметры управления материалами: кнопка создания, основные цвета **Diffuse** и **Specular**.

Нажмите кнопку **New** в узле **Material**. Вот теперь к объекту присоединен новый материал. Заметьте, что одноименная панель в окне **Properties** содержит полные настройки материала ноды, и вы можете их редактировать на свое усмотрение.

Кроме **Material** в окне **Node Editor** присутствует еще одна нода — **Output**. Этот узел показывает конечное состояние материала.

Ноды должны быть соединены между собой проводниками, как, например, на рис. 9.5. Так результат работы ноды **Material** передается **Output**. Если узел не включен в конвейер, то он не участвует в действии.

Для связи между собой ноды имеют по бокам панелей точки — каналы. Причем с правой стороны располагаются выходы, а с левой — входы. Наведите курсор на точку с названием **Color** у ноды **Output** и нажмите левую кнопку мыши. Кривая проводника тут же пропадет, а нода **Output** потемнеет. В этом случае было выполнено разъединение конвейера (рис. 9.6).

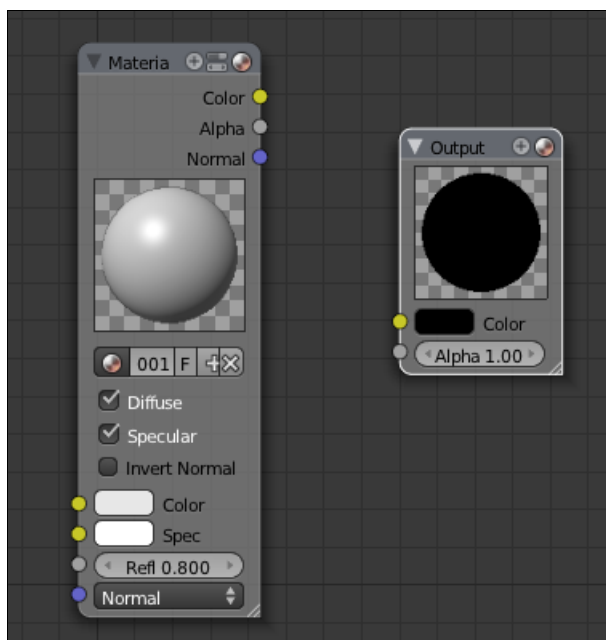


Рис. 9.6. Конвейер разъединен

Наведите курсор на точку канала **Color** узла **Material**. Нажмите левую кнопку мыши и подведите появившуюся кривую к входу **Color** ноды **Output**. Связь восстановлена.

Возможно, эта демонстрация вас не впечатлила, но не спешите скидывать **Node Editor** со счетов! Рассмотрим пример добавления еще одной ноды, которая обесцветит материал.

Щелкните левой кнопкой мыши по образцу цвета **Color** ноды **Material**. Выберите из палитры любой цвет, отличный от белого. Теперь разъедините оба узла.

Меню **Add**, расположенное в заголовке редактора, содержит большое количество нод, которые могут быть созданы в текущем режиме **Node Editor**. Выберите из него пункт **Convertor | RGB to BW**. Поместите новую ноду между имеющимися. Соедините выход **Color** узла **Material** с входом **RGB to BW**, а его выход с каналом **Color** ноды **Output** (рис. 9.7). Нажмите клавишу <F12> для проверки результата.

Если нода вам больше не нужна, то вы можете удалить ее. Выделите щелчком левой кнопки мыши и нажмите клавишу <X>.

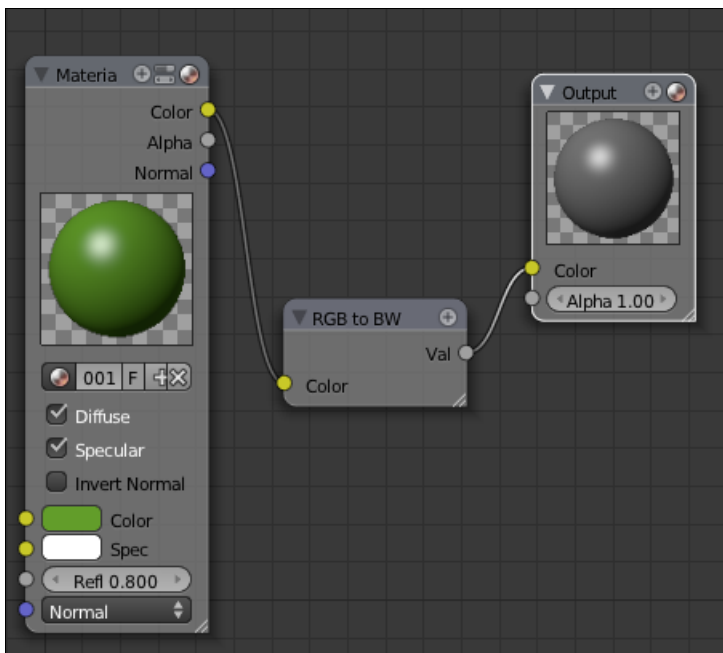


Рис. 9.7. Конвейер для обесцвечивания

9.2. Встроенный редактор видео

В главе 5 рассказывалось об **NLA Editor** — системе для глобального управления анимацией сцены. Казалось бы, зачем Blender нужен еще один высокоуровневый редактор?

Иногда трехмерная сцена создается с учетом того, что она будет использована в качестве эффекта для основного видео, снятого на камеру. В этом случае сцену обрабатывают с фоном, который можно будет в дальнейшем легко убрать средствами сторонней системы монтажа. А теперь представьте, как было бы удобно, если бы в самом Blender можно было выполнять компоузинг (*от англ. compositing*) трехмерной сцены и обычных видеофайлов. Такая система в программе есть и называется она **Video Sequence Editor** (Редактор видеопоследовательностей).

Профессиональные программы видеомонтажа — это целый комплекс функций для работы с видео. Встроенный редактор Blender, конечно, проще, но с его помощью

также можно с легкостью компоновать видео, использовать переходы или эффекты наложения.

Video Sequence Editor (VSE) имеет три режима отображения (рис. 9.8):

- ◆ **Sequence** (Секвенсор) — все окно редактора занимают проектные дорожки;
- ◆ **Preview** (Предпросмотр) — вывод видео для контроля секвенсора;
- ◆ **Sequence/Preview** (Секвенсор/Предпросмотр) — в этом случае окно редактора разбито на две области, где одна секвенсор, а другая превью.



Рис. 9.8. Выбор режима отображения VSE

Вы можете вывести в разных областях программы несколько окон **VSE** и включить соответствующие режимы, но можно просто выбрать заготовку **Video Editing** (Редактирование видео) из меню **Screen Layout** с удобным расположением окон.

Система монтажа видео в Blender является многодорожечной. Каждый загруженный файл представляет собой полосу (**Strip**), которая и подвергается редактированию (рис. 9.9).

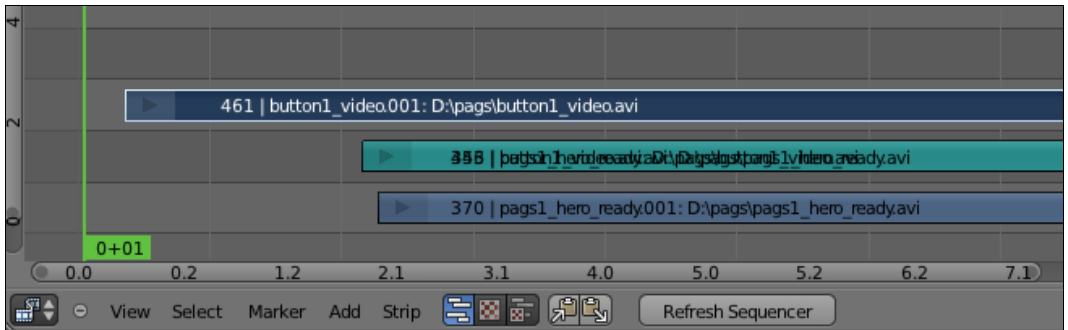


Рис. 9.9. Так выглядит секвенсор

VSE может работать с разными форматами файлов: видео, звуковые, графические. Для добавления файла в проект служит меню **Add** в заголовке окна редактора:

- ◆ **Effect Strip** (Эффект);
- ◆ **Sound** (Звук);
- ◆ **Image** (Картинка);
- ◆ **Movie** (Видео);
- ◆ **Scene** (Сцена).

После выбора нужного типа и загрузки файла в окне **VSE** появится **Strip**. Если нажать кнопку **Play** в окне **Timeline**, то редактор начнет воспроизведение видео.

СОВЕТ

Если вы обработали анимацию сцены как последовательность множества графических файлов, то можете загрузить эти файлы в виде одной **Strip** в окно **VSE**. Выберите **Image** в меню **Add**, выделите все файлы рендера и загрузите в проект.

Strip можно перемещать как в пределах дорожки, так и с одной на другую. Просто щелкните правой кнопкой мыши на нужной полоске и, удерживая ее, переместите в другое место. Для фиксации результата служит левая кнопка мыши. Разумеется, в окне **VSE** поддерживается стандартная клавиша <G>, которую вы можете использовать для перемещения **Strip**.

Видеоредактор Blender позволяет создавать переходы между **Strip**. Например, используя стандартный переход **Cross**, картинка одного видеофайла будет плавно заменяться картинкой другого. Кроме переходов можно использовать видеоэффекты, с помощью которых можно настроить прозрачность видео (**Alpha Over**), корректировать его цвет (**Color**), управлять скоростью потока (**Speed Control**) и многое другое. Чтобы добавить эффект, нужно выделить один файл или два с помощью клавиши <Shift> (для переходов) и выбрать соответствующий пункт в меню **Add**.

СОВЕТ

Каждая **Strip** имеет свои уникальные настройки, которые содержатся на дополнительной панели, вызываемой клавишей <N> в окне **VSE**. Здесь можно изменить дорожку (**Channel**), настроить прозрачность (**Opacity**), выбрать наложение (**Blend**) и многое другое.

И наконец, вы можете загрузить текущую сцену в качестве **Strip**. Для этого нужно выбрать пункт **Scene** меню **Add**. Откроется дополнительное меню, содержащее список имеющихся сцен в проекте. После выбора нужной, сцена появится в виде **Strip** в окне секвенсора. Теперь вы можете управлять ею, как обычной **Strip**, т. е. перемещать, накладывать эффекты и переходы. Только учтите, что обработка конечного результата станет дольше, т. к. будет выполняться просчет 3D.

По умолчанию Blender уже настроен на обработку последовательностей в **VSE**. Если это не происходит, то включите опцию **Sequencer** на закладке **Post Processing** панели **Render** (рис. 9.10).

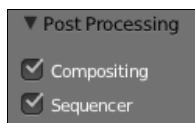


Рис. 9.10. Управление выводом **VSE**

9.3. Восковой карандаш

В наборе функций Blender нашлось место необычному инструменту — **Grease Pencil** (Восковой карандаш). С его помощью можно рисовать произвольные линии в окне **3D View**. Нечто подобное предлагают двухмерные редакторы (Gimp, Photoshop). Казалось бы, зачем трехмерной программе такое изобразительное средство?

Управление карандашом располагается на двух панелях **3D View**. Опции переключения режимов работы находятся в **Tool Shelf** (рис. 9.11, а), а конкретные настройки — на панели **Properties** (рис. 9.11, б).

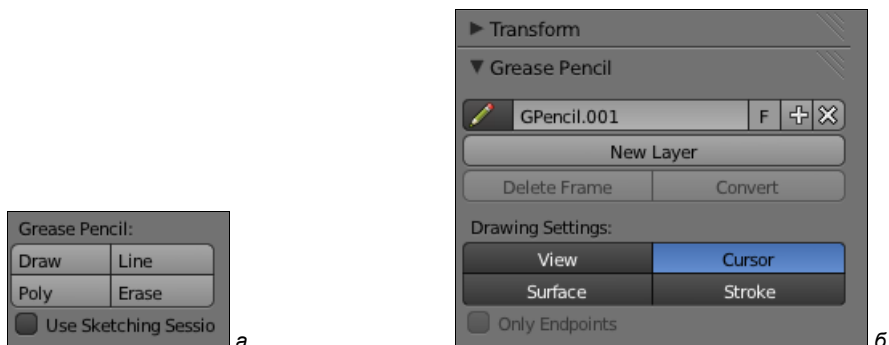


Рис. 9.11. Режимы работы **Grease Pencil** (а) и настройки карандаша (б)

Основные режимы **Grease Pencil**, которые находятся в **Tool Shelf**:

- ◆ **Draw** (Рисование) — произвольное рисование, линия появляется в местах движения курсора;
- ◆ **Line** (Линия) — рисование одиночных прямых линий;
- ◆ **Poly** (Множество) — рисование прямых линий, когда каждая новая линия является продолжением предыдущей;
- ◆ **Erase** (Стирание) — удаление рисунка или его частей.

Работать с карандашом просто. Включаете нужный тип инструмента и, удерживая левую кнопку мыши, занимаетесь художеством. После отпускания кнопки программа возвращается в привычный режим работы. Вы можете включить опцию **Use Sketching Session** (Использовать режим скетча) на панели **Tool Shelf** (см. рис. 9.11, а) для закрепления режима рисования. Чтобы выйти из него, нужно нажать клавишу <Esc>.

С одной стороны, **Grease Pencil** — это инструмент для создания двухмерных рисунков, а с другой стороны, результат его работы располагается в трехмерном пространстве. Blender предлагает несколько вариантов привязки рисунка к пространству. Эти настройки находятся на закладке **Grease Pencil** панели **Properties** (рис. 9.12).

Когда вы рисуете первую свою линию с помощью карандаша, Blender создает для него отдельный слой. Конечно, вы можете управлять слоями вручную. Кнопка **New Layer** добавляет слой в виде отдельного блока настроек на закладку **Grease Pencil** (см. рис. 9.12). Щелкнув по характерному крестику около названия слоя, вы выполните удаление блока.

Использование слоев для конкретных частей рисунка имеет свои выгоды. Во-первых, вы можете махом убрать ненужную часть, просто удалив блок, во-вторых, слои — это единственный способ безопасного редактирования рисунка. Допустим,

вы нарисовали в окне несколько пересекающихся линий. Если вы захотите с помощью **Erase** (Стереть) стереть одну из них, то наверняка повредите другую. А вот расположив эти линии в разных слоях, вы этого никак не сделаете.

В настройках **Grease Pencil** имеется группа кнопок **Drawing Settings**, с помощью которых можно выбрать режим привязки рисунка к 3D-пространству.

- ◆ **View** (Просмотр). Рисунок позиционируется относительно окна **3D View**. В этом режиме нарисованные линии остаются на своих позициях при вращении сцены.
- ◆ **Surface** (Поверхность). Рисование в соответствии с углом просмотра сцены. Полученный рисунок имеет трехмерные координаты, жестко привязанные к сцене. В этом режиме удобно рисовать линии, располагающиеся относительно плоскостей трехмерных объектов.
- ◆ **Cursor** (Курсор). Создание трехмерных линий, привязанных к координатам **3D Cursor**. Такой вариант годится, если нужно точно провести линию в определенной точке пространства. Чем ближе проходит рисунок к **3D Cursor**, тем более точно он позиционируется. Можно использовать **3D Cursor**, как центр общего рисунка.
- ◆ **Stroke** (Движение). Включите этот режим, если хотите создавать замкнутые кривые.

Используя эти варианты, можно рисовать объемные фигуры в окне **3D View** (рис. 9.13).

Как рисовать с помощью **Grease Pencil**, мы рассмотрели, но в чем же практическая ценность этого инструмента?

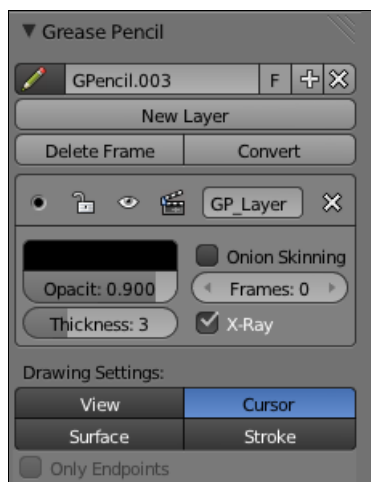


Рис. 9.12. Расширенные настройки карандаша

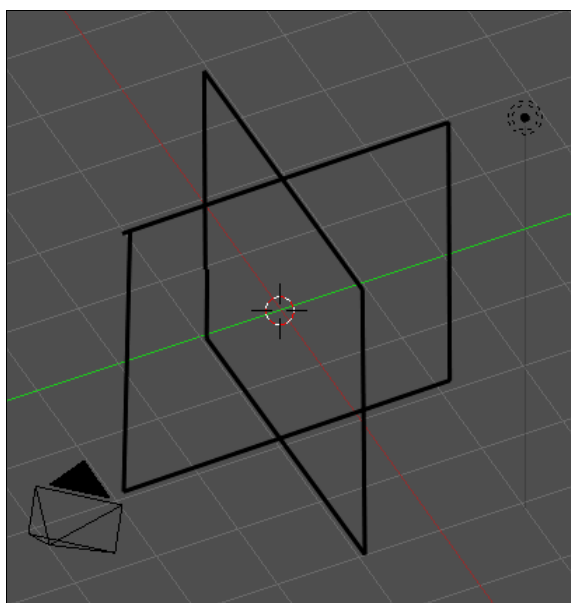


Рис. 9.13. Это трехмерный рисунок

Приведем варианты использования карандаша.

Вы можете использовать **Grease Pencil** для пояснительных записей к своим объектам, ведь при обычном рендере рисунков не будет видно.

Рисунок можно конвертировать в трехмерный сплайн. В этом случае **Grease Pencil** становится удобным способом рисования произвольных кривых. Для конвертации нужно нажать кнопку **Convert** в настройках инструмента (см. рис. 9.11, б). Программа выведет запрос, в какой формат должен быть переведен рисунок (**Bezier Curve** или **Path**).

Вы можете создать серию рисунков и выводить их в виде анимации.

Если с первыми двумя пунктами все понятно, то последний очень интересный. В детстве, наверное, каждый пробовал делать простейшую мультипликацию в виде фаз рисунка на каждой страничке блокнота. При быстром перелистывании создавался эффект движения. Нечто подобное предлагает **Grease Pencil**.

В окне настроек карандаша имеется поле **Frames** (см. рис. 9.12). Оно предназначено для создания фаз рисунка. Принцип работы заключается в следующем:

1. Создать рисунок.
2. Изменить активный кадр анимации в **Timeline**.
3. Переключить следующую фазу в поле **Frames**.

Таким образом создается целая серия рисунков, где фазы движения привязаны к кадрам анимации. Нажмите **<Alt>+<A>** и рисунок оживет.

А теперь небольшая хитрость. По умолчанию рендер программы не умеет выводить рисунки, созданные с помощью **Grease Pencil**. Конечно, их можно конвертировать в сплайны и добавить объема, но это не самый лучший вариант. В заголовке окна **3D View** имеются две кнопки для запуска OpenGL-рендера (рис. 9.14). В первом случае будет обработан активный кадр, а во втором — полностью анимация. Параметры OpenGL-рендера настраиваются в стандартных опциях панели **Render** (установите размер, путь вывода, количество кадров). Самое главное — это позволяет выводить на картинке рисунки **Grease Pencil** совместно с трехмерными объектами!

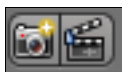


Рис. 9.14. Кнопки запуска OpenGL-рендера сцены

9.4. Скрытые возможности

Если вы считаете, что уже все узнали про Blender, то глубоко ошибаетесь. В меню выбора окон имеется одно с названием **User Preferences** (Пользовательские настройки). Конечно, это не что иное, как настройки Blender, и они хранят целые залежи полезных функций.

Окно **User Preferences** может быть вызвано также из главного меню программы **File** или путем нажатия горячих клавиш <Ctrl>+<Alt>+<U> (рис. 9.15).

Все настройки программы разбиты на группы:

- ◆ **Interface** (Интерфейс) — глобальные параметры, отвечающие за поведение интерфейса;
- ◆ **Editing** (Редактирование) — настройки, относящиеся к процессу редактирования;
- ◆ **Input** (Управление) — настройки горячих клавиш и привязка их к функциям программы;
- ◆ **Addons** (Дополнения) — список всех плагинов программы;
- ◆ **Themes** (Темы) — визуальное оформление и палитра окон;
- ◆ **File** (Файл) — настройка файловой системы, установка каталогов по умолчанию;
- ◆ **System** (Система) — тонкие настройки ядра программы.



Рис. 9.15. Окно **User Preferences**

Самое интересное из всего этого — группа **Addons** (Дополнения). Скачивая Blender с сайта разработчика, вы получаете полнофункциональную программу, способную очень на многое. Но, оказывается, еще большая часть возможностей просто отключена. Не забывайте, что Blender — это программа Open Source, которая очень динамично развивается сообществом и Blender Foundation. Те функции, что прошли официальное подтверждение, автоматически включаются в текущую версию Blender. Множество перспективных плагинов сторонних разработчиков также на-

ходятся в Blender, правда, в выключенном состоянии. Причина этого может быть разная, например, плагин дублирует какие-нибудь уже имеющиеся функции или находится в тестовом статусе, или это слишком редко используемая функция не для большого круга пользователей.

При выборе панели **Addons** появляется список возможных категорий плагинов (рис. 9.16).

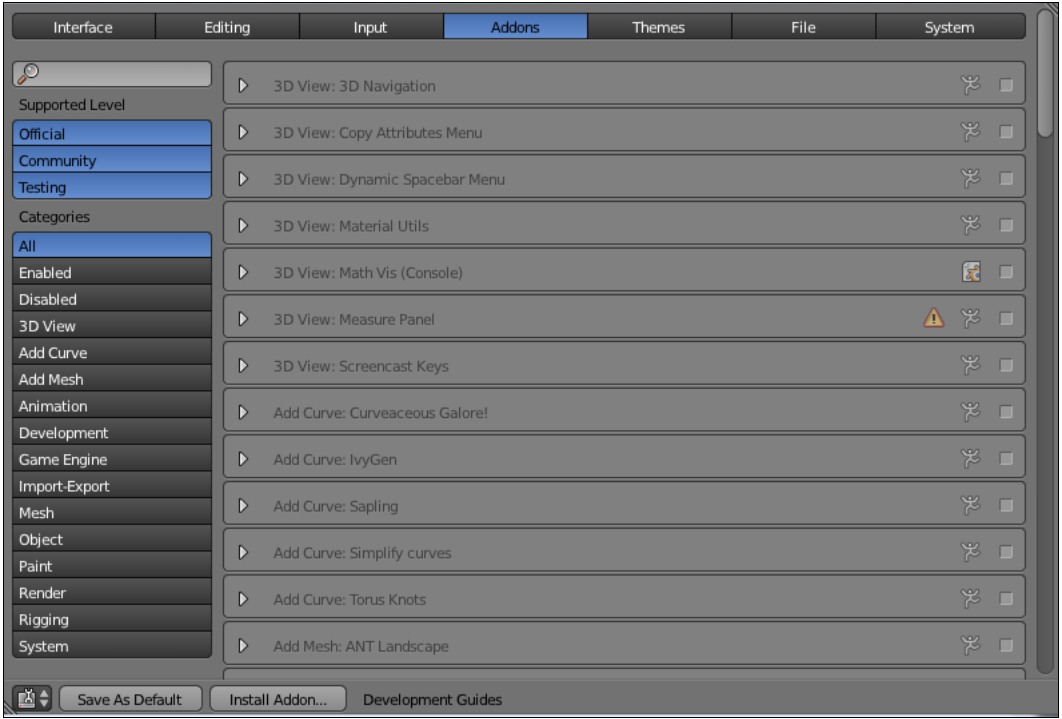


Рис. 9.16. Панель Addons

Обратите внимание на группу кнопок **Supported Level** (Поддерживаемый уровень). С помощью них вы можете сортировать плагины:

- ◆ **Official** (Официальные);
- ◆ **Community** (Сообщество);
- ◆ **Testing** (Тестируемые).

Названия неактивных плагинов в списке имеют светло-серый цвет. Вы можете открыть описание любой функции с помощью щелчка по треугольнику рядом с названием. Возможно, краткое описание будет малоинформативно, тогда просто нажмите кнопку **Link to the Wiki**, чтобы перейти на сайт, содержащий исчерпывающий обзор плагина. Чтобы включить или отключить нужную функцию, поставьте галочку напротив названия.

Рассмотрим наиболее интересные и популярные плагины. В группе **Add Mesh** имеется плагин с названием **BoltFactory**. Это не что иное, как конструктор для созда-

ния моделей болтов и гаек. Конечно, такие объекты можно сделать самому, но зачем мучаться, если есть уже готовые (рис. 9.17).

На рис. 9.17 показаны всего три объекта. В действительности, **BoltFactory** имеет широкие настройки для создания разного вида гаек, шурупов и болтов. После включения плагина в меню **Add | Mesh** появляется пункт **Bolt**. После его выбора в **Tool Shelf** открывается панель настроек (рис. 9.18).

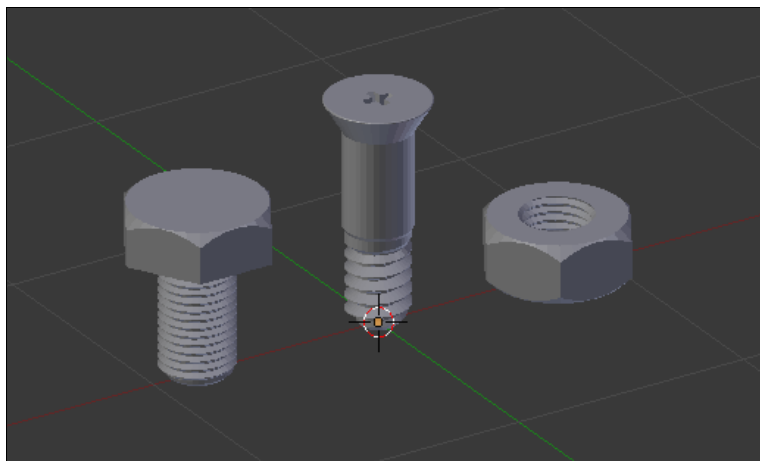


Рис. 9.17. Образцы моделей, созданных с помощью **BoltFactory**



Рис. 9.18. Настройки **BoltFactory**

Плагин **ANT Landscape** в группе **Add Mesh** — генератор ландшафтов. Основывается на шумовых функциях, применительно к плоскости. После добавления его в сцену появляется **Plane** с деформацией, а также целая группа настроек в **Tool Shelf** (рис. 9.19).

Если у вас сложности с созданием скелета для модели или просто нет времени, воспользуйтесь плагином **Rigify** из группы **Rigging**. В меню **Add | Armature** появится пункт **Human**, который позволит добавить в сцену полноценную арматуру для человеческой фигуры. Никаких настроек здесь нет, просто скелет (рис. 9.20).

Вас не устраивает встроенный рендер программы? Тогда загляните в список плагинов в группе **Render**. Там имеется набор нескольких популярных рендеров, адаптированных к Blender: **Network Renderer**, **Pov-Ray**, **Renderfarm**.

СОВЕТ

Обязательно ознакомьтесь с плагинами в **User Properties** и их функционалом. Возможно, найдете полезные инструменты и для своего проекта.

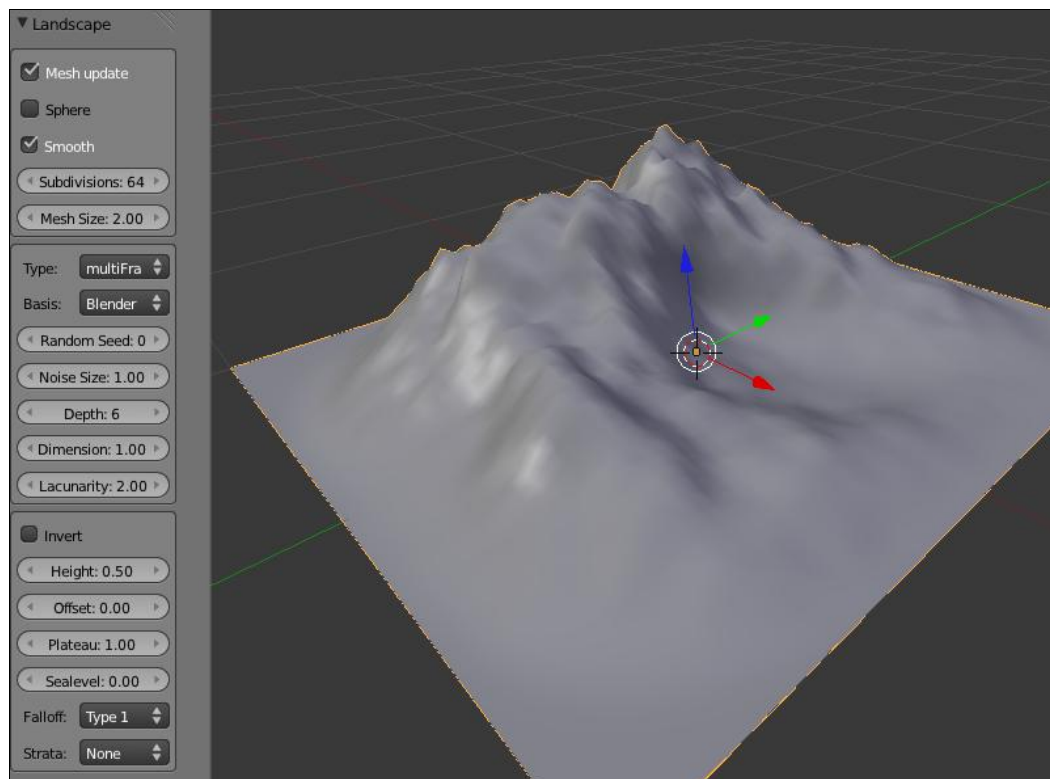


Рис. 9.19. Настройки и пример Landscape

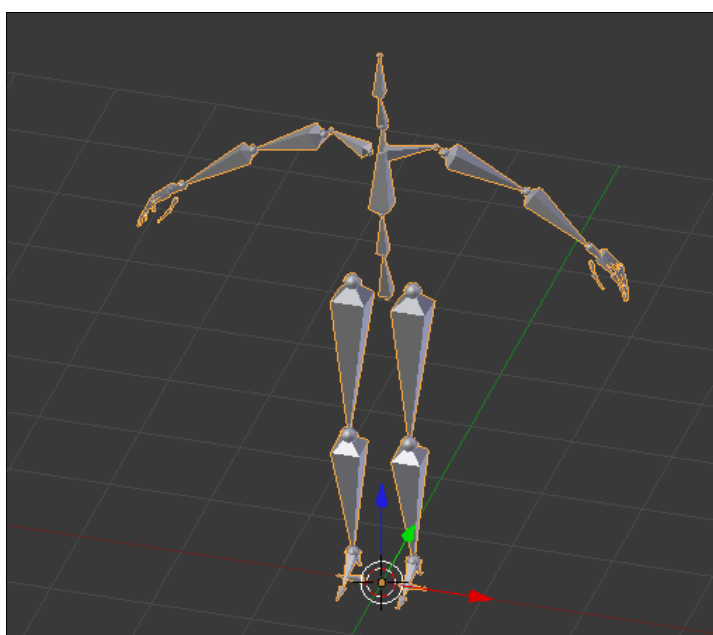


Рис. 9.20. Арматура, созданная с помощью Rigify

9.5. Практика. Приемы работы с нодами

Начинающие пользователи обычно стараются не работать с **Node Editor**, несмотря на его мощь и гибкость. Возможно, это вызвано недостатком информации по ключевым нодам в официальной документации Blender. В этом разделе приведены несколько полезных примеров практического использования нод.

Рассмотрим работу с текстурами на примере создания текстуры мрамора. Самое простое, что приходит в голову, — это воспользоваться процедурной функцией **Marble** (Мрамор). Пусть в качестве испытуемого объекта будет куб.

Создайте новую сцену, откройте заготовку окон **Compositing** и активируйте режим работы **Textures Nodes** (см. рис. 9.4). Выделите куб и поставьте галочку в опции **Use Nodes**.

Программа автоматически создаст заготовку для работы с текстурой — ноды **Checker** и **Output**. Первая является процедурной текстурой, генерирующей в шахматном порядке два указанных цвета. Она нам не нужна. Поэтому щелкните левой кнопкой мыши по узлу и нажмите клавишу <X> для удаления. Окошко **Output** служит конечной точкой конвейера.

Добавьте из меню **Add | Textures** ноду **Marble**. Соедините каналы **Color** обоих узлов (рис. 9.21).

Получившаяся текстура является черно-белой (**Output** показывает конечный вид текстуры), что и не удивительно, ведь такая она изначально. Нужно добавить к ней

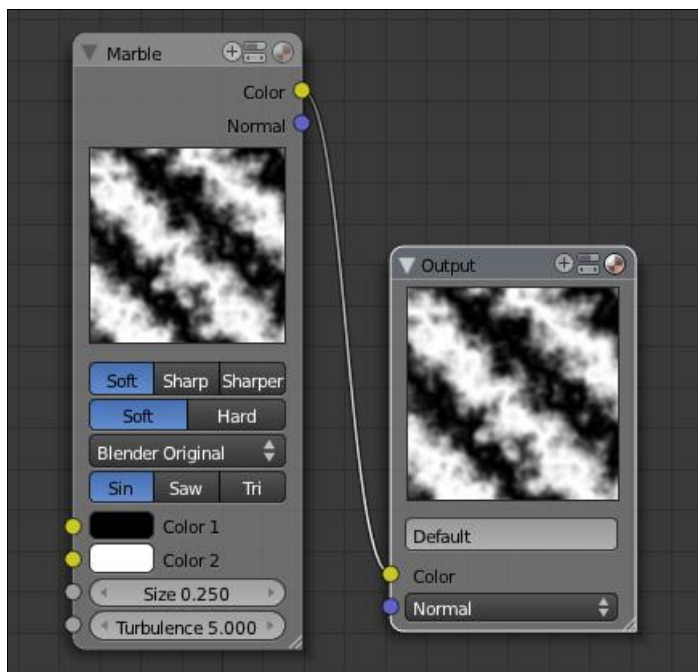


Рис. 9.21. Простой вывод текстуры

немного цвета. Для этого понадобится еще два узла из группы **Color** меню **Add: Mix** и **Compose RGBA**.

Нода **Mix** предназначена для смешивания двух источников. Она позволяет выбрать стандартный алгоритм смешивания, установить баланс с помощью опции **Factor**.

Нода **Compose RGBA** является ничем иным, как простым генератором цвета с помощью ползунков **RGBA**.

Разъедините ноды в окне редактора. Итак, узел **Mix** будет смешивать выход текстуры **Marble** с выходом **Compose RGBA**, а результат подавать на **Output**. Сделайте эти связи и настройте цвет **RGBA** (рис. 9.22):

- ◆ **R** = 0.68
- ◆ **G** = 0.40
- ◆ **B** = 0.40
- ◆ **A** = 1.00

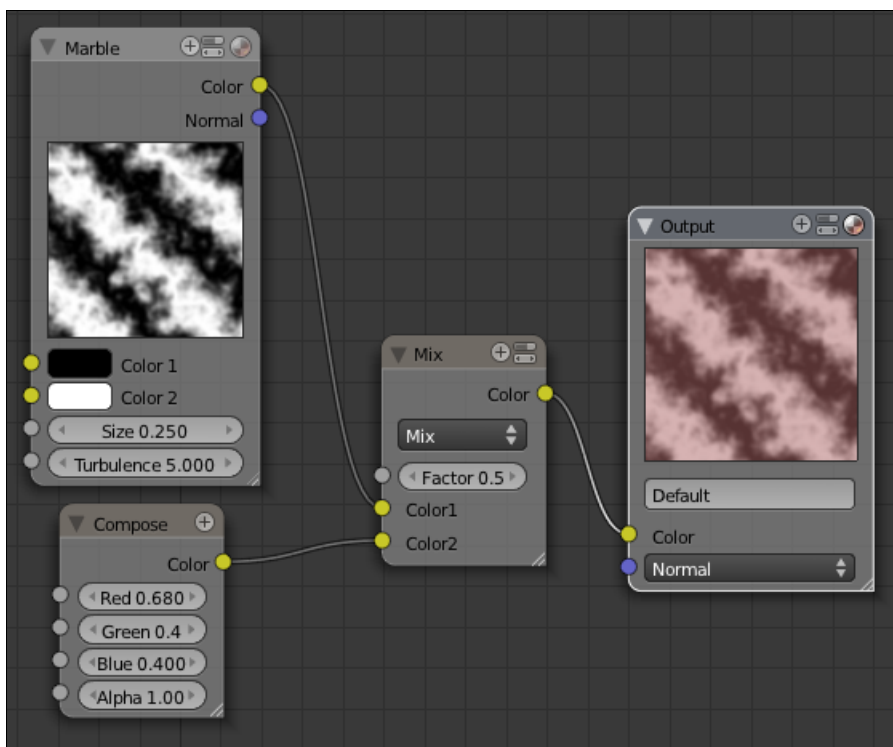


Рис. 9.22. Смешивание цвета с текстурой

Выглядит лучше, но недостаточно. Можно попробовать добавить еще одну процедурную текстуру для создания зернистости. Для этой цели лучше всего подойдет **Cloud**. Добавьте ноду **Cloud** из меню **Add | Textures** и еще один узел **Mix**.

По умолчанию **Cloud** генерирует слишком большой узор. Установите в параметрах ноды значение **Size** равным 0.05.

Задача в следующем. Нужно вывод первого узла **Mix** послать на вход второго **Mix** и к нему же прикрепить **Cloud** (рис. 9.23).

Теперь можно обработать сцену и полюбоваться на результат (рис. 9.24).

Рассмотрим еще один более сложный пример объединения изображения **Green Screen** с остальной сценой.

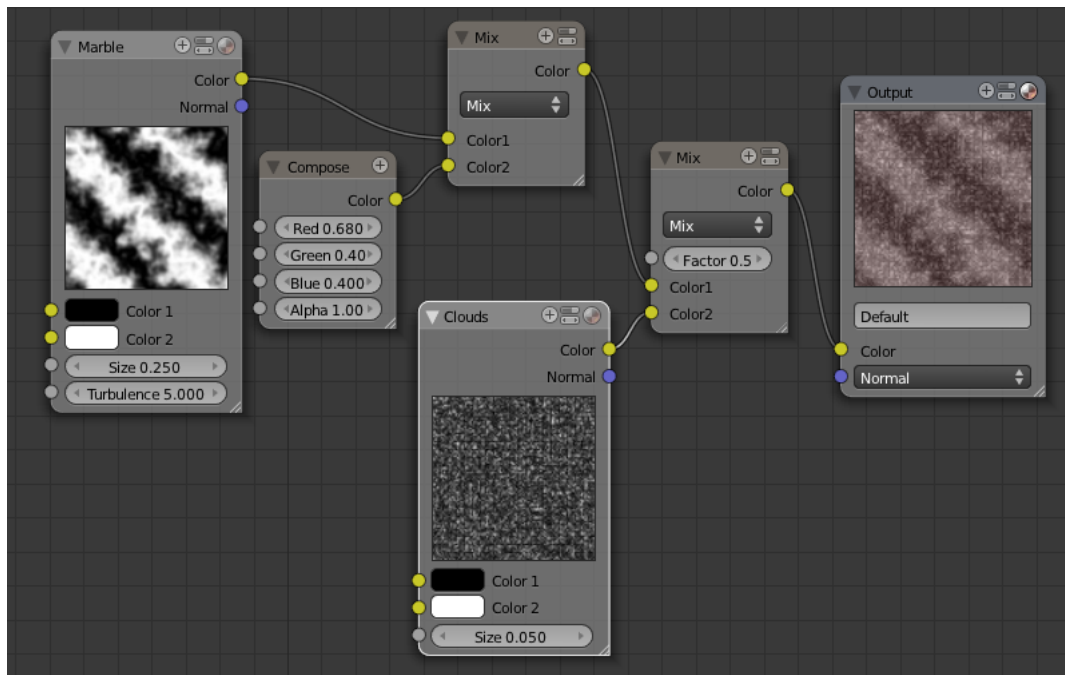


Рис. 9.23. Добавление зернистости к текстуре

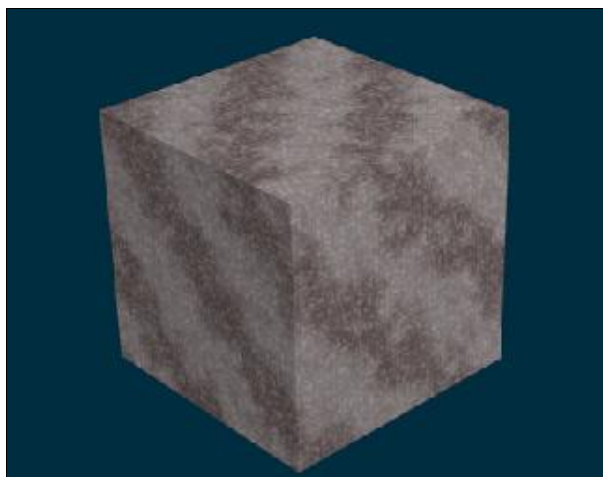


Рис. 9.24. Мраморная текстура, созданная с помощью нод

Green Screen (Зеленый экран) — это название для техники вырезания однотонного фона с целью последующей замены его другой картинкой. Так, например, делается прогноз погоды на некоторых телеканалах, когда ведущий стоит на фоне движущихся картинок. В действительности, диктор снимается на фоне задника сплошного зеленого или синего цвета. В дальнейшем этот цвет убирается и подставляется нужное изображение.

Создадим простую сцену, сквозь текстуру **Green Screen**, натянутую на плоскость, должны просвечивать другие объекты сцены.

Создайте новый проект, удалите куб из сцены. Добавьте плоскость и сферу. Расположите их так, чтобы сфера скрывалась за **Plane** с точки зрения камеры (рис. 9.25).

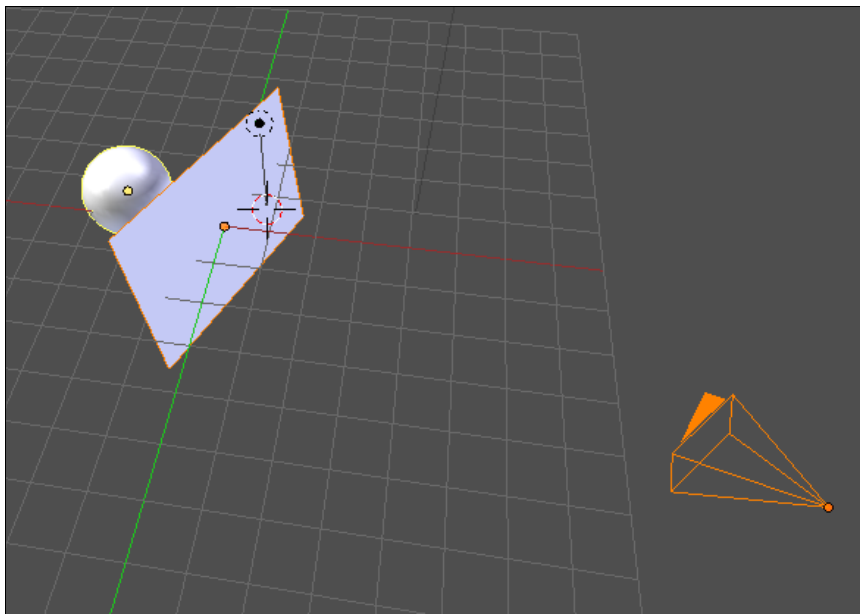


Рис. 9.25. Подготовка сцены

В качестве изображения **Green Screen** можете обработать сцену с кубом. Только установите в параметрах **World** цвет фона зеленым и сохраните картинку в файл.

Создайте материал для **Plane**, добавьте текстуру **Image** и загрузите в нее подготовленное изображение (рис. 9.26).

Задача отнюдь не простая. Нужно будет создать сразу несколько конвейеров для вырезания фона, добавления альфа-канала и микширования с остальной сценой. Удобнее всего будет разместить ключевые объекты по разным слоям. Пусть плоскость с картинкой остается в первом слое по умолчанию, а сферу переместите в рядом находящийся слой. Для отображения всех объектов в сцене активируйте оба слоя (рис. 9.27).

Откройте заготовку окон **Compositing**, включите режим **Compositing Nodes** и установите галочку в опцию **Use Nodes**.

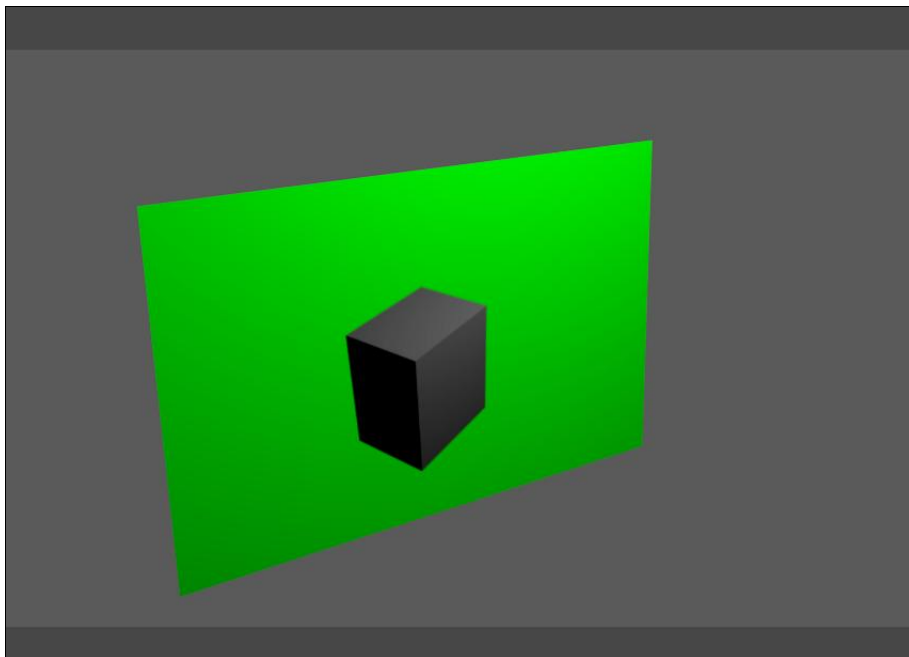


Рис. 9.26. При рендере сцены сферы не видно



Рис. 9.27. Активные слои

Программа создаст заготовки для этого режима:

- ◆ **Render Layers** (Слои обработки) — метаслой, устанавливаемый на панели **Render** окна **Properties**, который может иметь свои уникальные настройки. По умолчанию Blender всегда имеет один такой слой. Эта нода позволяет выбрать слой рендера в качестве исходящей точки конвейера;
- ◆ **Composite** (Композиция) — конечная точка выхода.

Нажав клавишу <F12>, вы можете проверить работоспособность данной заготовки. А теперь разъедините эти узлы.

Чтобы вырезать фон у картинки, понадобится специальная нода **Difference Key** (Дифференцированная прозрачность). Добавьте ее из меню **Add | Matte**. Соедините выход **Image** узла **Render Layers** с входом **Image 1** новой ноды, но при этом ничего не изменится. Нода **Difference Key** упорно не хочет показывать содержимое в своем окне. Можно соединить ее выход **Image** с одноименным входом **Composite** и, используя <F12>, постоянно обновлять результат. Конечно, это не удобно. Если включить опцию **Auto Render** в заголовке окна **Node Editor**, то программа будет автоматически выполнять рендер при любых выполненных изменениях. Вторым вариантом — это воспользоваться специальной нодой **Viewer** (меню **Add | Output**), которая демонстрирует работу конвейера без просчета в виде фона окна **Node**

Editor. Подключите ее к выходу **Image** ноды **Difference Key**, включите опцию **Backdrop** в заголовке окна и спокойно редактируйте параметры. В дальнейшем вы можете подключать **Viewer** к любому сомнительному узлу для контроля (рис. 9.28).

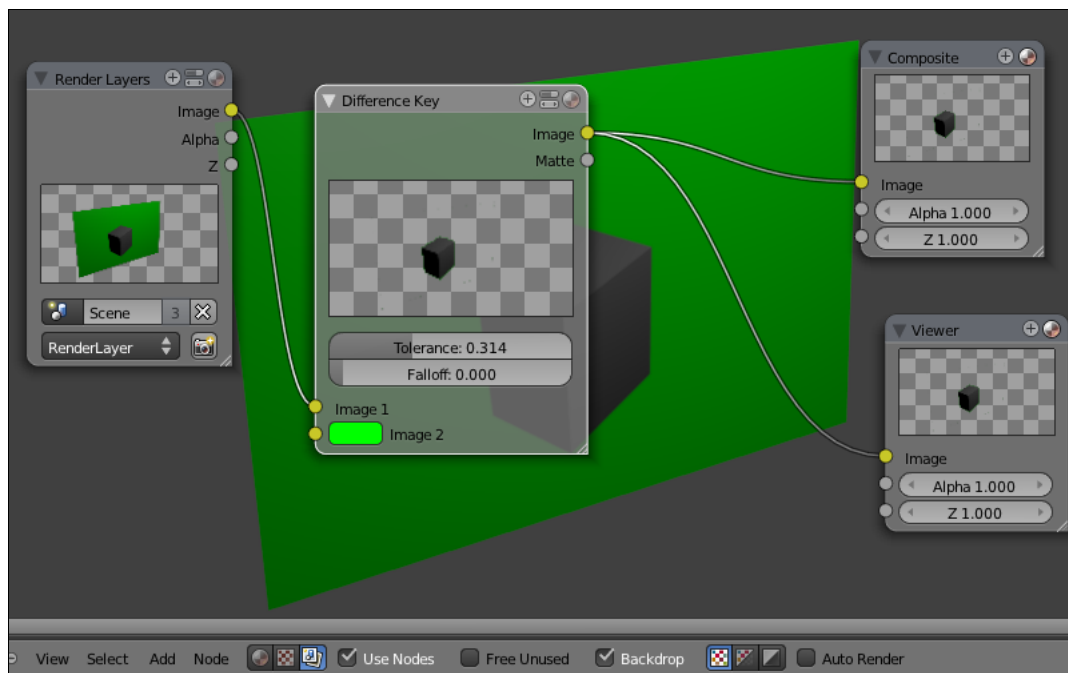


Рис. 9.28. Проверка работы конвейера

Настройка ноды **Difference Key** заключается в следующем:

1. Щелкните по образцу цвета **Image 2**. Выберите зеленый цвет.
2. С помощью изменения параметров **Tolerance** (Допуск) и **Falloff** (Затухание) добейтесь исчезновения фона в окне просмотра ноды. Вы можете подключить выход **Matte** (Маска) к **Viewer** для более точного регулирования.

Итак, на выходе узла **Difference Key** образовалось два изображения, где **Image** — оригинальное, а **Matte** его маска.

Теперь добавим ноды для создания альфа-канала: **AlphaOver** (Прозрачность) из группы **Color** и **Color Spill** (Заливка цветом) из группы **Matte**. Создайте еще узел **Mix** для окончательного смешивания результата. Выполните подключение нод, как на рис. 9.29.

Итак, что мы имеем. Изображение с помощью **Difference Key** расщепляется на основное и маску. Затем содержимое канала **Image** подается на верхний вход **Image** узла **AlphaOver**, а выход **Image** ноды **Render Layers** соединен с нижним входом **Image** узла **AlphaOver**. Получается этакое перекрестное соединение. В результате изображение с альфа-каналом поступает узлу **Color Spill**, который удаляет зеленый цвет (в его настройках нажмите кнопку **G** в группе **Despill Channel**). Готовое изображение входит во второй канал **Mix**.

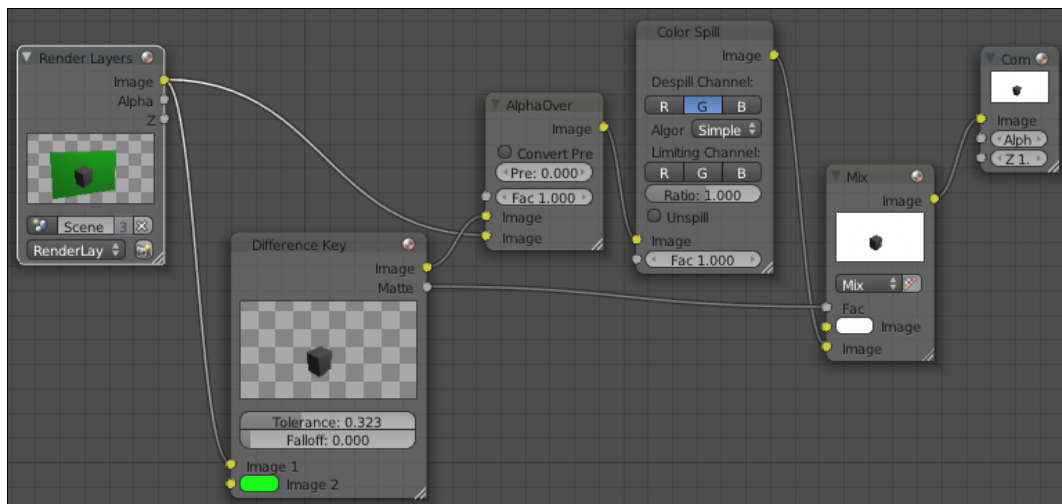


Рис. 9.29. Конвейер обработки альфа-канала

Теперь при рендере вы увидите куб на черном фоне. Это конечно совсем не то, что ожидалось получить, но все еще впереди. У нас еще осталась неподсоединенная маска. Подключите канал **Matte** к входу **Fac** (Factor, Коэффициент) узла **Mix**. После этого фон изображения станет белым.

Сейчас производится рендер обоих слоев с удалением цвета, настройки прозрачности и т. д. Нам же надо отделить от основного конвейера слой со сферой и подать его на оставшийся вход **Mix** без изменений. Нужно создать еще один слой для рендера.

Откройте панель **Render** окна **Properties**. На закладке **Layers** уже имеется слой с названием **RenderLayer**, для которого был создан конвейер. Переименуйте его как **GreenScreen** (рис. 9.30).

Так как сделанный конвейер должен обрабатывать только слой с плоскостью, то в группе **Layer** установите его активным (см. рис. 9.30). Снимите галочку с опции **ZTransp** в группе **Include**.

Нажмите кнопку с плюсом для создания еще одного слоя. Назовите его "Scene" и укажите второй слой в группе **Layers**.

Перейдите в окно **Node Editor**. Добавьте из меню **Add | Input** ноду **Render Layers**. Выберите из меню узла слой с названием **Scene**. Соедините его выход **Image** с оставшимся свободным входом **Mix**. Вот теперь все, настройка нод закончена.

Добавьте источник света во второй слой для освещения сцены и нажмите клавишу <F12> (рис. 9.31).

СОВЕТ

Если при использовании **Difference Key** не удастся полностью убрать фон, то добавьте после него еще один такой узел для более тонкой настройки.

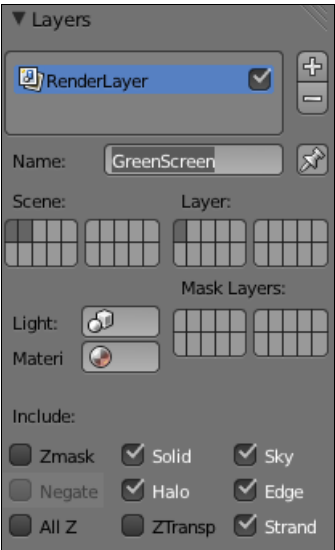


Рис. 9.30. Настройки слоев рендера

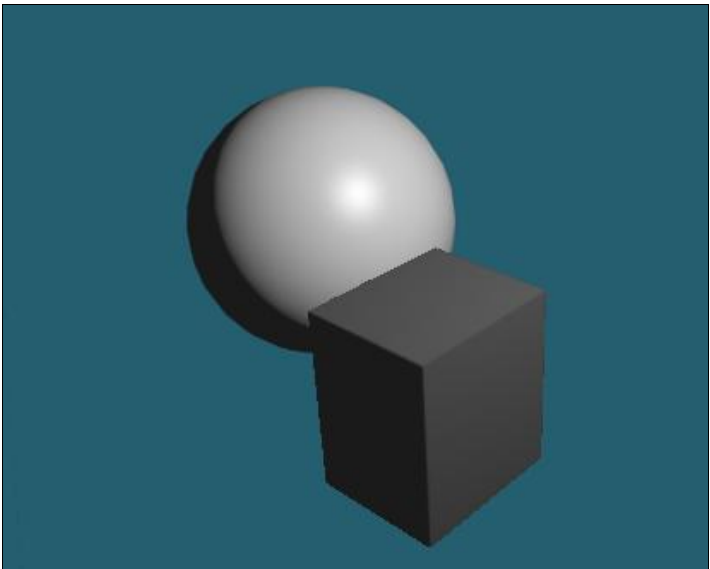


Рис. 9.31. Зеленый фон удален и сцена просвечивает сквозь текстуру

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1

Инсталляция Blender

Разработчики программы на своем официальном сайте (www.blender.org) выкладывают разные сборки Blender для популярных операционных систем: Windows, Linux, Mac Os X и FreeBSD. Причем имеются версии как 32-битные, так и 64-битные. Если вы используете 64-битную систему, то предпочтительно устанавливать именно ее, для улучшения производительности.

П1.1. Установка программы в Windows

В зависимости от выбранной ОС, поставка программы может осуществляться в виде упакованного архива или исполняемого файла. В целом установка особо не отличается на разных ОС, но есть некоторые нюансы. Рассмотрим пример инсталляции версии Blender для Windows.

После запуска скачанного файла инсталлятор потребует подтвердить соглашение о лицензии. Конечно, нажмите кнопку **I agree!**

Следующим окном будет выбор компонентов. В действительности здесь выбирать особо не из чего. Набор функций привычный и все они являются активными:

- ◆ **Add Start Menu Shortcuts** (Добавить ссылку в стартовое меню);
- ◆ **Add Desktop Shortcuts** (Добавить ссылку на рабочий стол);
- ◆ **Open .blend files with Blender** (Открывать файлы .blend с помощью Blender).

А вот к следующему окну с банальным предложением выбрать место установки, стоит отнестись более внимательно. Версии Blender выходят очень часто, бывает, что разработчики выкладывают на сайт, так называемые development build, т. е. тестовые сборки. Разве неинтересно попробовать свежее испеченные версии программы? Конечно, интересно, но стоит помнить, что они могут быть нестабильными и лучше, если рядом находится старый релиз программы. Итого вывод, устанавливайте программу в папку с уникальным именем. Например, путь по умолчанию в Windows:

C:\Program Files\Blender Foundation\Blender

Выберите, что-то вроде:

```
C:\Program Files\Blender Foundation\Blender2.62
```

Нажатие кнопки **Next** начнет установку программы в указанную папку.

П1.2. Установка программы в Linux

Linux славится большим количеством дистрибутивов с различными менеджерами окон, поэтому проще всего будет объяснять процесс установки, опираясь на командную строку.

Допустим, вы скачали файл `blender-2.62-linux-i686.tar.bz2` в папку текущего пользователя. Это двойной архив и его нужно распаковать:

◆ `$ bunzip2 blender-2.62-linux-i686.tar.bz2` — распаковка архива `bzip`;

◆ `$ tar xvf blender-2.62-linux-i686.tar` — распаковка архива `tar`.

После отработки последней команды в папке пользователя будет находиться каталог с одноименным названием. Теперь самое время решить, куда установить программу. Для этого чаще всего используется каталог `/opt`. Выполните перемещение, но уже с правами суперпользователя:

```
# mv blender-2.62-linux-i686/opt
```

Для работы программы, возможно, понадобится установить Python версии 3.2 (для указанной сборки Blender). Выполните это в пакетном менеджере вашего дистрибутива. Пример для дистрибутива, основанного на Debian:

◆ `# aptitude search python` — ищем пакет с подходящим именем;

◆ `# aptitude install python3.2` — установка.

Вот и все. Для запуска Blender используйте команду:

```
$ /opt/blender-2.62-linux-i686/blender
```

ПРИЛОЖЕНИЕ 2

Файловый архив

Электронный архив к книге выложен на FTP-сервер издательства по адресу: **ftp://85.249.45.166/9785977508230.zip**. Ссылка доступна и со страницы книги на сайте **www.bhv.ru**.

Архив содержит проекты Blender рассмотренных примеров и практических уроков. В определенных случаях вам могут понадобиться вспомогательные файлы для выполнения задания. Все они расположены в папках, соответствующих номеру главы. Например, модель меча, создание которого рассматривалось во второй главе, содержится в папке: `Scenes/glava2`.

Предметный указатель

A

Action 239, 262
Action Editor 263
Air 303
AlphaOver 372
Ambient 139
Ambient Occlusion 336
Angular Velocity 286
ANT LandScape 365
Anti-Aliasing 350
Append 41
Area 324
Armature 237
Array 81
Aspect Ratio 346
Atmosphere 329

B

Background Images 57
Bevel Object 102
Blend Sky 336
Blinn 143
Boolean 67
Bump Mapping 181

C

Cache 284
Clipping 333
Cloth 280, 302
Collision 280
Color Space 344
Color Spill 372
Composite 371
Constraint 253

Convert To 122
CookTorr 143
Current Frame 217
Curve 95

D

Damp 301
Delete 51
Depth of Filed 334
Difference Key 372
Diffuse 142
Dimensions 346
Displacement Map 192
Domain 307
Duplicate 28
Dynamic Paint 280, 319

E

Emission 283
Emit 139
Encoding 348
Extrude 49

F

Fluid 280, 307
Focal Lenght 333
Force Field 280, 304
Frame Rate 347
Fresnel 143
Friction 298

G

Grease Pencil 359
Green Screen 370

H

Hair 281, 290
Halo 147
Hemi 324
Hide 54
Horizon Color 336

I

Image Mapping 168
Image Sampling 169
Indirect Lighting 338
Insert Keyframe 218
Interpolation 221
Inverse 54
IOR 157

J

Join 48

K

Key 216
Knife 52

L

Lambert 142
Lamp 323
Lattice 70, 276
Layers 38
Lens Flares 150
Link 42
Loop Cut and Slide 50
Lower Field 349

M

Mass 299
Mesh 43
Minnaert 143
Mist 338
Motion Blur 350

N

NLA Editor 264
Normal Mapping 183
NURBS 96, 110

O

Oren-Nayar 142
Outliner 35

P

Paper Sky 336
Particle Mode 295
Particles 281, 290
Path 229
Phong 143
Pinning 304
Pose Mode 243
Post Processing 349
Properties 22, 36
Pull 300
Push 301

R

Ramp 144, 211
Ray Tracing 155
Real Sky 336
Render 345
Render Layers 371
Resolution 346
Rigify 365

S

Scene Selector 18
Screen Layout 17
Sculpt Mode 77
Seam 198
Separate 54
Shadeless 139
Shading 139
Shadow 141, 327
Shape Keys 233
Show 54
Skinning 247
Sky 329, 342
Smoke 280, 313
Snap 23, 30
Soft Body 280, 298, 300
Spin 109
Spot 324
Spring 303
Stars 340
Strand 140
Strip 265

Subdivide 53
Subdivision Surface 248
Sun 324
Surface 95
Sync Mode 219

T

Taper Object 102
Text 115
Timeline 216
Toggle Cyclic 101
Tool Shelf 22
Toon 142
Translucency 139
Transparency 155

U

Upper Field 349
UV 194
UV Mapping 195

V

Velocity 284
Vertex Groups 251, 298
Vertex Paint 201
Video Sequence Editor 358
Viewport Shading 22, 23
VSE 358

W

Wardiso 143
World 335

Z

Zenith Color 336

Б

Безье 96

В

Выдавливание 49

Г

Группы 36
◇ вершин 251

Д

Дублирование 28

И

Инверсная кинематика 243

К

Ключ 216
Контрольные точки 96
Координатная система 25
Кривые 95

М

Макет Экрана 17
Материал 135
Модификатор 60
◇ boolean 67
◇ Lattice 276
◇ mirror 60
◇ multires 73
◇ subdivision surface 248
Мультиматериалы 153

Н

Нормаль 183

О

Окна
◇ DopeSheet 235
◇ Graph Editor 220
◇ NLA Editor 265
◇ Node Editor 353

Окна (*прод.*)

◇ properties 36

◇ timeline 217

◇ User Preferences 362

П

Привязка 23

Примитивы

◇ curve 97

◇ mesh 43

◇ path 229

◇ surface 97

◇ text 115

Пропорциональное редактирование 48

Просмотр сцены

◇ вращение 23

◇ панорамирование 23

◇ масштабирование 23

Процедурные текстуры 170

Прямая кинематика 241

Р

Разбиение 53

Развертка UV 194

Разделение 54

Рамповый шейдер 144

Рендер 345

Рычаги 96

С

Связи 33

Симметричное моделирование 57

Скелетная анимация 237

Слои 38

Сплайны 95

Т

Тени 141