

Как вы знаете, это ещё не поле ввода, а ссылка — переменная, в которой можно хранить адрес какого-нибудь поля ввода. Создавать сам объект удобнее всего в обработчике события `OnCreate` формы (он будет вызван при создании формы в самом начале работы программы):

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  decEdit:=TIntEdit.Create(Self);
  decEdit.Text:='100';
  decEdit.Left:=6;
  decEdit.Top:=6;
  decEdit.Width:=115;
  decEdit.Parent:=Self
end;
```

В первой строке создаётся новый компонент, и его адрес записывается в поле `decEdit`. Слово `Self` при вызове конструктора `Create` означает «этот объект». Поскольку метод `FormCreate` — это метод формы, в данном случае этот объект — сама форма. Такой вызов конструктора говорит о том, что владельцем (англ. *owner*) нового компонента будет форма, и когда форма будет удалена из памяти, вместе с ней будет уничтожен и компонент `decEdit`.

В следующих строках устанавливаются начальные значения для свойств компонента (`Text` — содержимое, `Left` и `Top` — координаты левого верхнего угла, `Width` — ширина).

В последней строчке мы меняем свойство `Parent` (в переводе с англ. — родитель), записывая в него адрес формы (`Self`). Это очень важно, потому что «родитель» отвечает за показ всех «дочерних объектов» на экране; если этого не сделать, то поле ввода останется невидимым.

Остаётся определить для нового компонента обработчик события `OnChange`: при изменении содержимого поля ввода нужно показать соответствующее шестнадцатеричное число с помощью метки `hexLabel`. Сначала создадим сам обработчик. Вручную добавим в описание формы заголовок процедуры:

```
procedure decEditChange(Sender: TObject);
```

а в секцию **implementation** — её текст:

```
procedure TForm1.decEditChange(Sender: TObject);
begin
  hexLabel.Caption:=IntToHex(decEdit.Value,1);
end;
```

Сначала мы запрашиваем числовое (!) значение у поля ввода, используя новое свойство `Value`, а затем переводим его в шестнадцатеричную систему счисления с помощью функции `IntToHex`. Её второй параметр — количество шестнадцатеричных цифр; если этот параметр равен 1, выбирается минимально возможное количество цифр (без лидирующих нулей).

Теперь нужно как-то подключить этот обработчик к вновь созданному компоненту. Для этого нельзя использовать Инспектор объектов, потому что в режиме разработки нашего компонента на форме нет. Но существует другой способ — в программе записать в свойство `OnChange` (оно наследуется от базового класса `TEdit`) адрес процедуры обработки события. Это нужно сделать при создании формы, добавив в конец обработчика `OnCreate` строку

```
decEdit.OnChange:=@decEditChange;
```

Символ `@` перед именем процедуры обозначает её адрес.

Теперь программа готова и её можно запустить. Фактически мы вручную сделали все операции, которые обычно выполняет среда Lazarus, если компонент добавляется на форму из палитры компонентов. Установить компонент в палитру можно с помощью меню **Пакет**, но это выходит за рамки нашего курса.

Вопросы и задания

1. В каких случаях имеет смысл разрабатывать свои компоненты?
2. Подумайте, в чём достоинства и недостатки использования своих компонентов.
3. Почему программисты редко создают свои компоненты «с нуля»?
4. Объясните, как связаны классы компонентов `TIntEdit` и `TEdit` из примера в параграфе. Чем они различаются?
5. В каких секциях модуля нужно расположить описание нового класса и его реализацию (программный код методов)? Объясните, почему нежелательно располагать всё в одной секции.
6. Какие функции используются для преобразования числового значения в текстовое и обратно?
7. Какая функция применяется для перевода числа в шестнадцатеричную систему счисления?
8. Объясните, как работает свойство `Value` у компонента `TIntEdit` из примера в параграфе?
9. Почему в приведённом в параграфе примере для обработки вводимых символов мы не устанавливали свой обработчик `OnKeyPress`?
10. Что означает слово `override` при описании метода?
11. Как создать компонент во время выполнения программы?



12. Почему компоненты обычно создаются в обработчике OnCreate формы?
13. Чем различаются роли владельца компонента и его родительского объекта?
14. Почему свойства нового компонента в данном примере устанавливаются только из программы, а не в Инспекторе объектов?
15. Как установить обработчик события во время выполнения программы?
16. Почему можно использовать обработчик события OnChange, который не был объявлен в классе TIntEdit?



Подготовьте сообщение

«Создание компонентов в программе на C#»



Задачи

1. Измените рассмотренную в параграфе программу так, чтобы в самом начале метка показывала шестнадцатеричный код числа, которое записано в поле ввода.
2. Разработайте компонент, который позволяет вводить шестнадцатеричные числа.
- *3. Используя дополнительные источники, разберитесь, как установить новый компонент в палитру среды Lazarus. Переделайте свою программу так, чтобы компонент добавлялся на форму из палитры.

§ 56 Модель и представление

Одна из важнейших идей технологии быстрого проектирования программ (RAD) — повторное использование написанного ранее готового кода. Чтобы облегчить решение этой задачи, было предложено использовать ещё одну *декомпозицию*: разделить **модель**, т. е. данные и методы их обработки, и **представление** — способ взаимодействия модели с пользователем (интерфейс) (рис. 7.22).

Пусть, например, данные об изменении курса доллара хранятся в виде массива; требуется искать в массиве максимальное и минимальное значения, а также строить приближённые зависимости, позволяющие прогнозировать изменение курса в ближайшем будущем. Это описание задачи на уровне модели.

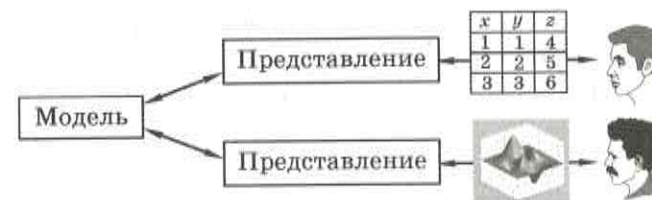


Рис. 7.22

Для пользователя эти данные могут быть представлены в различных формах: в виде таблицы, графика, диаграммы и т. п. Полученные зависимости, приближённо описывающие изменение курса, могут быть показаны в виде формулы или в виде кривой. Это уровень представления или интерфейса с пользователем.

Чем хорошо такое разделение? Его главное преимущество состоит в том, что модель не зависит от представления, поэтому одну и ту же модель можно использовать без изменений в программах, имеющих совершенно различный интерфейс.

Вычисление арифметических выражений: модель

Построим программу, которая вычисляет арифметическое выражение, записанное в символьной строке. Для простоты будем считать, что в выражении используются только:

- целые числа;
- знаки арифметических действий $+$ $-$ $*$ $/$.

Предположим, что выражение не содержит ошибок и посторонних символов.

Какова **модель** для этой задачи? По условию, данные хранятся в виде символьной строки. Обработка данных состоит в том, что нужно вычислить значение записанного в строке выражения.

Вспомните, что аналогичную задачу мы решали в § 43, где использовалась структура типа «дерево». Теперь мы применим другой способ.

Как вы знаете, при вычислении арифметического выражения последней выполняется крайняя справа операция с наименьшим приоритетом (см. § 43). Таким образом, можно сформулировать следующий алгоритм вычисления арифметического выражения, записанного в символьной строке s :

1. Найти в строке s последнюю операцию с наименьшим приоритетом (пусть номер этого символа записан в переменной k).

2. Используя дважды этот же алгоритм, вычислить выражения слева и справа от символа с номером k и записать результаты вычисления в переменные $n1$ и $n2$ (рис. 7.23).

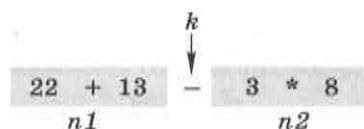


Рис. 7.23

3. Выполнить операцию, символ которой записан в $s[k]$, с переменными $n1$ и $n2$.

Обратите внимание, что в п. 2 этого алгоритма нужно решить ту же самую задачу для левой и правой частей исходного выражения. Как вы знаете, такой прием называется *рекурсией*.

Основную функцию назовём *Calc* (от англ. *calculate* — вычислить). Она принимает символьную строку и возвращает целое число — результат вычисления выражения, записанного в этой строке. Алгоритм её работы на псевдокоде:

```

k:=номер символа, соответствующего последней операции
если k=0 то
    знач:=перевести всю строку в число
иначе
    n1:=результат вычисления левой части
    n2:=результат вычисления правой части
    знач:=применить найденную операцию к n1 и n2
все

```

Для того чтобы найти последнюю выполняемую операцию, будем использовать функцию *LastOp* из § 43. Если эта функция вернула 0, то операция не найдена, т. е. вся переданная ей строка — это число (предполагается, что данные корректны).

Теперь можно написать функцию *Calc*:

```

function Calc ( s: string ) : integer;
var k, n1, n2: integer;
begin
    k:=LastOp (s);
    if k=0 then Calc:=StrToInt(s) {вся строка - число}
    else begin
        n1:=Calc(Copy(s, 1, k-1)); {левая часть}

```

```

n2:=Calc(Copy(s, k+1, Length(s)-k));
    {правая часть}
    {выполнить операцию}
case s[k] of
    '+': Calc:=n1+n2;
    '-': Calc:=n1-n2;
    '*': Calc:=n1*n2;
    '/': Calc:=n1 div n2
end
end
end;

```

Обратите внимание, что функция *Calc* — рекурсивная, она дважды вызывает сама себя.

Функции *Calc* и *LastOp* (а также функцию *Priority*, которая вызывается из *LastOp*) удобно объединить в отдельный модуль *Model* (модуль модели):

```

unit Model;
interface
function Calc(s: string): integer;
implementation
uses SysUtils;
function Priority(op: char): integer;
...
function LastOp(s: string): integer;
...
function Calc(s: string): integer;
...
end.

```

Секция *interface* этого модуля содержит только заголовок функции *Calc* — это всё, что доступно другим модулям программы. В секции *implementation* подключается модуль *SysUtils* (в котором находится функция *StrToInt*) и записаны все функции (многоточие обозначает тело функции). Таким образом, наша модель — это функции, с помощью которых вычисляется арифметическое выражение, записанное в строке.

Вычисление арифметических выражений: представление

Теперь построим интерфейс программы. В верхней части окна будет размещён выпадающий список (компонент *TComboBox*), в котором пользователь вводит выражение (рис. 7.24). При нажатии на клавишу *Enter* выражение вычисляется и его результат выводится

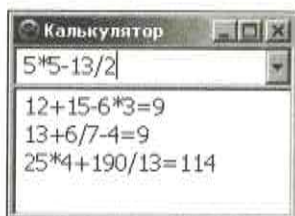


Рис. 7.24

Align, равное alTop. Назовем этот компонент Input (в переводе с англ. — ввод).

Добавляем второй компонент — TMemo (группа Standard), устанавливаем для него выравнивание alClient (заполнить всю свободную область) и имя Answers (в переводе с англ. — ответы). Для того чтобы пользователь не мог менять поле вывода, для компонента Answers устанавливаем логическое свойство ReadOnly (только для чтения), равное True.

Логика работы программы может быть записана в виде псевдокода:

```
если нажата клавиша Enter то
  x:= значение выражения
  добавить результат вычислений в конец поля вывода
если выражения нет в списке то
  добавить его в список
все
все
```

Для перехвата нажатия клавиши Enter будем использовать обработчик OnKeyPress компонента Input. Клавиша Enter имеет код 13, поэтому условие «если нажата клавиша Enter» запишется в виде

```
if Key=#13 then begin
  ...
end;
```

Значение выражения будем вычислять с помощью функции Calc:

```
x:= Calc(Input.Text);
```

Эта функция находится в модуле Model, который нужно подключить, добавив в начало секции **implementation** команду

```
uses Model;
```

в последней строке многострочного редактора текста (компонента TMemo). Список полезен для того, чтобы можно было вернуться к уже введённому ранее выражению и исправить его. Для этого каждое новое выражение будем добавлять в выпадающий список.

Итак, на форму нужно добавить компонент TComboBox (группа Standard). Чтобы прижать его к верху, установим свойство

Компонент TMemo содержит массив строк, которые доступны как свойство-массив Lines. Чтобы добавить к ним новую строку (в конец массива), нужно использовать метод Add (в переводе с англ. — добавить):

```
Answers.Lines.Add (Input.Text+'='+IntToStr(x));
```

Обратите внимание, что результат вычислений переведен в символьный вид с помощью функции IntToStr.

Строки, входящие в выпадающий список, доступны как свойство-массив Items объекта Input. Метод IndexOf служит для поиска строки в списке и возвращает номер найденного элемента (нумерация начинается с нуля) или значение -1, если образец не найден. Поэтому команда добавления в список новой строки выглядит так:

```
i:=Input.Items.IndexOf(Input.Text);
if i<0 then
  Input.Items.Insert(0, Input.Text);
```

Метод Insert добавляет строку в список. На первом месте указывается позиция, в которую добавляется строка (0 — в начало списка). Приведём полностью обработчик OnKeyPress:

```
procedure TForm1.InputKeyPress(Sender: TObject;
  var Key: char);
var x, i: integer;
begin
  if Key=#13 then begin
    x:=Calc(Input.Text);
    Answers.Lines.Add(Input.Text+'='+IntToStr(x));
    i:=Input.Items.IndexOf(Input.Text);
    if i<0 then
      Input.Items.Insert(0, Input.Text)
  end
end;
```

Теперь программу можно запускать и испытывать.

Итак, в этой программе мы разделили модель (данные и средства их обработки) и представление (взаимодействие модели с пользователем), которые разнесены по разным модулям. Это позволяет использовать модуль модели в любых программах, где нужно вычислять арифметические выражения.

Часто к паре «модель — представление» добавляют ещё управляющий блок (контроллер), который, например, обрабатывает ошибки ввода данных. Но при программировании в RAD-средах контроллер и представление, как правило, объединяются вместе — контроль данных происходит в обработчиках событий.



Вопросы и задания

1. Чем хорошо разделение программы на модель и интерфейс? Как это связано с особенностями современного программирования?
2. Что обычно относят к модели, а что — к представлению?
3. Что от чего зависит (и не зависит) в паре «модель — представление»?
4. Приведите свои примеры задач, в которых можно выделить модель и представление. Покажите, что для одной модели можно придумать много разных представлений.
5. Объясните алгоритм вычисления арифметического выражения без скобок.
6. Пусть требуется изменить программу вычисления арифметического выражения так, чтобы она обрабатывала выражения со скобками. Что нужно изменить: модель, интерфейс или и то, и другое?



Подготовьте сообщение

- а) «Зачем нужны шаблоны проектирования?»
- б) «Схема "Модель — представление — контроллер"»



Задачи

1. Измените рассмотренную в параграфе программу так, чтобы она вычисляла выражения с вещественными числами (для перевода вещественных чисел из символьного вида в числовой используйте функцию `StrToFloat`).
2. Добавьте в рассмотренную в параграфе программу обработку ошибок. Подумайте, какие ошибки может сделать пользователь. Какие ошибки могут возникнуть при вычислениях? Как их обработать?
- *3. Измените рассмотренную в параграфе программу так, чтобы она вычисляла выражения со скобками. *Подсказка:* нужно искать последнюю операцию с самым низким приоритетом, стоящую *вне* скобок.

4. Постройте программу «Калькулятор» для выполнения вычислений с целыми числами:



Практические работы к главе 7

- Проект № 1 «Движение на дороге»
 Работа № 62 «Скрытие внутреннего устройства объектов»
 Проект № 2 «Иерархия классов (логические элементы)»
 Работа № 63 «Создание формы в RAD-среде»
 Работа № 64 «Использование компонентов»
 Работа № 65 «Компоненты для ввода и вывода данных»
 Работа № 66 «Разработка компонентов»
 Проект № 3 «Модель и представление»



ЗОР к главе 7 на сайте ФЦИОР (<http://fcior.edu.ru>)



- Объектно-ориентированное программирование
- Основные понятия и принципы ООП
- Этапы объектно-ориентированного программирования
- Объектно-ориентированная модель программирования
- Основные принципы объектно-ориентированного программирования: инкапсуляции и полиморфизма, наследования и переопределения.

Самое важное в главе 7

- Сложность и размеры современных программ таковы, что в их разработке принимает участие множество программистов. Объектно-ориентированное программирование — это метод, позволяющий разбить задачу на части, каждая из которых в максимальной степени независима от других.

- Программа в ООП — это набор объектов, которые обмениваются сообщениями.
- Перед программированием выполняется объектно-ориентированный анализ задачи. На этом этапе выделяются взаимодействующие объекты, определяются их существенные свойства и поведение.
- Любой объект — экземпляр какого-то класса. Классом называют группу объектов, обладающих общими свойствами.
- Объекты не могут «узнать» устройство других объектов (принцип инкапсуляции). При описании класса закрытые поля и методы помещаются в секцию **private**, а общедоступные — в секцию **public**.
- Обмен данными между объектами выполняется с помощью общедоступных свойств и методов, которые составляют интерфейс объектов. Изменение внутреннего устройства объектов (реализации) не влияет на взаимодействие с другими объектами, если не меняется интерфейс.
- Как правило, классы образуют иерархию (многоуровневую структуру). Классы-потомки обладают всеми свойствами и методами классов-предков, к которым добавляются их собственные свойства и методы.
- ООП позволяет обеспечивать высокую скорость и надёжность разработки и модификации больших и сложных программ. В простых задачах применение ООП, как правило, увеличивает длину программы и замедляет её работу.
- Современные программы с графическим интерфейсом основаны на обработке событий, которые вызваны действиями пользователя и поступлением данных из других источников, и могут происходить в любой последовательности.
- Для быстрой разработки программ применяют системы визуального программирования, в которых интерфейс строится без «ручного» написания программного кода. Такие системы, как правило, основаны на ООП.
- В современных программах принято разделять модель (данные и алгоритмы их обработки) и представление (способ ввода исходных значений и вывода результатов).

Глава 8

Компьютерная графика и анимация

§ 57

Основы растровой графики

Как вы знаете из курса 10 класса, растровый рисунок, как мозаика, состоит из отдельных «квадратиков»-пикселей, каждый из которых закрашен своим цветом. Цвет кодируется как набор чисел, например в модели RGB цвет пикселя задаётся тремя значениями — красной (англ. **R** — *red*), зелёной (англ. **G** — *green*) и синей (англ. **B** — *blue*) составляющими.

В этой главе вы узнаете о том, как можно обрабатывать растровые изображения с помощью современных графических редакторов.

Что такое разрешение?

Любое изображение, в конечном счёте, предназначено для просмотра. При выводе на экран или на печать оно должно иметь определённые размеры, поэтому нужно установить связь между шириной и высотой рисунка в пикселях и его размерами (в сантиметрах или других единицах длины) на экране монитора или на бумаге. Эту связь определяет разрешение, т. е. число пикселей на некотором отрезке изображения (по ширине или высоте). По традиции разрешение измеряется в пикселях на дюйм¹ (англ. **ppi** — *pixels per inch*). Чем больше разрешение, тем выше качество изображения, но тем больше места оно занимает в памяти.

Например, пусть мы хотим вывести на экран изображение размером 10 × 15 см. Каковы будут размеры рисунка в пикселях? Обычно стандартным разрешением для изображения на экране считается² 72 ppi или 96 ppi. При разрешении 96 ppi размеры рисунка в пикселях должны быть равны:

высота: $10 \cdot 96 / 2,54 \approx 378$ пикселей;
ширина: $15 \cdot 96 / 2,54 \approx 567$ пикселей.

¹ 1 дюйм = 2,54 см.

² Разрешение 96 ppi, например, соответствует размерам экрана 1280 × 1024 пикселя для монитора с диагональю 17 дюймов.

Конечно, нужно учитывать, что фактическое разрешение экрана может отличаться от 96 ppi, оно зависит от размера монитора и режима работы видеокарты (заданного в её настройках количества пикселей по ширине и высоте экрана).

Теперь предположим, что нам нужно напечатать на бумаге стандартную фотокарточку того же размера (10 × 15 см). Печатающие устройства могут обеспечить значительно более высокое разрешение, чем экран. Для получения отпечатков среднего качества (при котором уже практически незаметно, что изображение состоит из пикселей) требуется разрешение около 300 ppi, а для профессиональных работ — 600 ppi и более (до 2400 ppi). При разрешении 300 ppi размеры рисунка в пикселях будут совсем другие:

высота: $10 \cdot 300 / 2,54 \approx 1181$ пиксель;

ширина: $15 \cdot 300 / 2,54 \approx 1772$ пикселя.

На рисунке 8.1 для сравнения показано одно и то же изображение с разным разрешением.

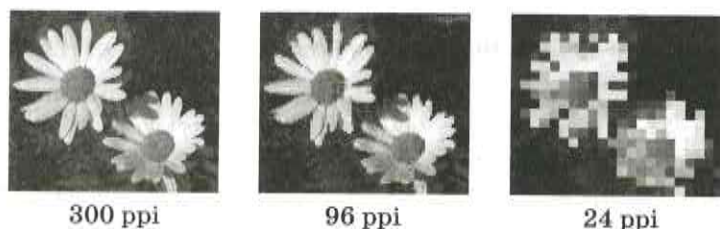


Рис. 8.1

Заметно, что изображение с разрешением 300 ppi смотрится вполне естественно (пиксели не видны), а при разрешении 24 ppi ромашки уже с трудом угадываются в получившемся наборе пикселей.

Для иллюстрации мы будем использовать свободный растровый графический редактор GIMP (www.gimp.org), версии которого существуют для Windows, Linux и macOS. Другие популярные редакторы, например Adobe Photoshop, имеют аналогичные возможности.

Если в редактор GIMP загрузить некоторое изображение, с помощью меню **Изображение — Размер изображения** можно посмотреть и изменить его параметры (рис. 8.2).

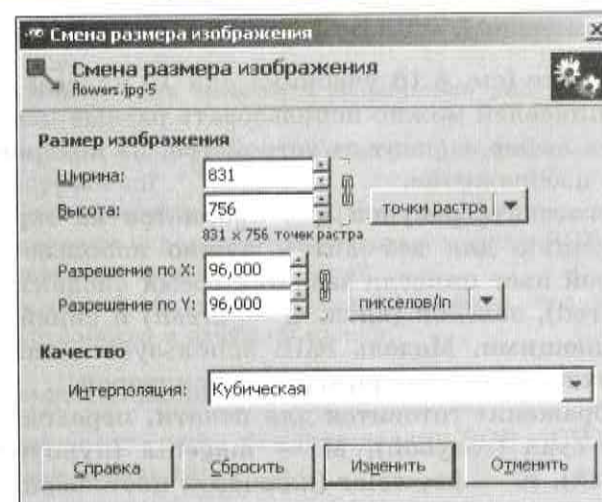


Рис. 8.2

По умолчанию размер изображения задаётся в пикселях (точках растра). Чтобы увидеть размеры отпечатка (в сантиметрах или миллиметрах), в выпадающем списке справа нужно выбрать соответствующую единицу измерения.

Это диалоговое окно позволяет также менять размеры рисунка и его разрешение. Изменяя размер изображения в пикселях, мы искажаем рисунок. Например, пусть при увеличении размера нужно заменить 5 пикселей на 8. В этом случае программе приходится с помощью математических методов перестроить картинку так, чтобы изображение сохранилось как можно лучше. Алгоритм такой обработки задаётся в поле **Интерполяция**¹. Кубическая интерполяция считается одной из лучших, однако при использовании этого метода изображение немного размывается. Чтобы сохранить чёткие границы на рисунке, в списке **Интерполяция** нужно выбрать вариант **Никакая**.

При изменении только разрешения количество пикселей в рисунке остаётся тем же, поэтому никакой потери качества не происходит. Однако размеры отпечатка изменятся: если увеличить разрешение, при печати изображение уменьшится.

¹ Интерполяция — это восстановление промежуточных значений функции.

Цветовые модели

Как вы знаете (см. § 16 учебника для 10 класса) для кодирования цвета пикселей можно использовать разные цветовые модели, причём их выбор зависит от устройства, на которое нужно будет выводить изображение.

Если вы готовите рисунок для просмотра на экране (например, иллюстрацию для веб-сайта), нужно использовать модель RGB, в которой цвет пикселя задается тремя числами — красной (англ. R — red), зелёной (англ. G — green) и синей (англ. B — blue) составляющими. Модель RGB используется для устройств, которые *излучают* свет, например для мониторов.

Если изображение готовится для печати, переходят к модели CMYK: C — cyan (голубой), M — magenta (пурпурный), Y — yellow (жёлтый), K — key color (ключевой цвет, чёрный). Модель CMYK более удобна для описания *отражённого* света, который в этом случае видит человек.

Во всех этих моделях цвет кодируется набором чисел, и только устройство вывода определяет, какой именно цвет мы увидим на самом деле. Поэтому для того, чтобы отпечаток выглядел так же, как изображение на экране, при преобразовании изображения из модели RGB в CMYK нужно использовать **цветовые профили** монитора и принтера, которые определяются с помощью специальных устройств (**калибраторов**). Наибольшими возможностями для подготовки качественных изображений для профессиональной печати обладает программа Adobe Photoshop.

Существуют и другие цветовые модели. Наиболее интересная из них — модель HSV¹ (англ. *Hue* — тон, оттенок; *Saturation* — насыщенность, *Value* — величина), которая ближе всего к естественному восприятию человека. Для кодирования «абсолютного цвета», не зависящего от устройства, применяется модель Lab (англ. *Lightness* — светлота, *a* и *b* — параметры, определяющие тон и насыщенность цвета).



Вопросы и задания

1. Что такое разрешение? В каких единицах оно измеряется?
2. Как выбирать разрешение для вывода изображений на монитор и на печать? Почему для печати требуется более высокое разрешение?

¹ Или HSB (англ. *Hue* — тон, оттенок; *Saturation* — насыщенность, *Brightness* — яркость).

3. От чего зависит фактическое разрешение при выводе изображения на экран?
4. Что произойдёт, если изменить разрешение рисунка, не меняя его размеры в пикселях?
5. Что произойдёт, если изменить размеры рисунка в сантиметрах, сохранив разрешение?
6. Что такое интерполяция? Зачем нужны разные виды интерполяции?
7. В каких случаях используются цветовые модели RGB и CMYK? Какие ещё цветовые модели вы знаете?
8. Вспомните (см. § 16 учебника для 10 класса), что такое цветовой профиль устройства и зачем он нужен.

Подготовьте сообщение

- а) «Преобразование цвета между моделями RGB и CMYK»
- б) «Цветовая модель HSV»
- в) «Цветовая модель Lab»

Задачи

1. Для рисунка размером 200 × 100 пикселей установлено разрешение 300 ppi. Определите его размеры в сантиметрах при выводе на экран и при печати.
2. Определите, каковы должны быть размеры рисунка в пикселях, если нужно напечатать изображение размером 297 мм на 210 мм (формат A4) с разрешением 300 ppi. Сколько мегапикселей (миллионов пикселей) должна содержать чувствительная матрица цифрового фотоаппарата, с помощью которого можно сделать такой снимок?

§ 58

Ввод изображений

Для ввода растровых изображений в компьютер чаще всего применяют цифровые фотоаппараты и сканеры.

Цифровые фотоаппараты

Самый важный элемент цифрового фотоаппарата — это **светочувствительная матрица** — интегральная микросхема, которая состоит из светочувствительных элементов — фотодиодов. Свет, поступивший на фотодиод, преобразуется в электрический сигнал, который с помощью аналого-цифрового преобразователя (АЦП) переводится в числовой код.

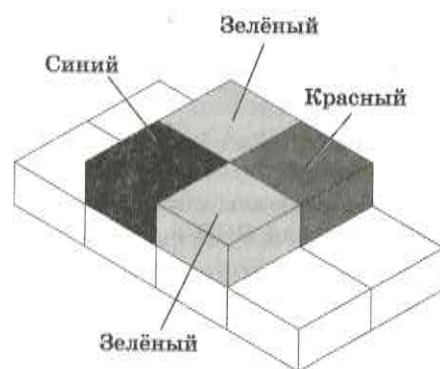


Рис. 8.3

Для того чтобы получить цветное изображение, каждый фотодиод «накрыт» светофильтром определённого цвета (чаще всего используют модель RGB — красные, зелёные и синие светофильтры). Таким образом, фотодиод измеряет яркость только одной составляющей цвета, а остальные компоненты восстанавливаются процессором фотокамеры по соседним пикселям. Во многих фотоаппаратах используют так называемый *фильтр Байера*, состоящий из 25% красных, 25% синих и 50% зелёных фильтров (рис. 8.3 и цветной рисунок на форзаце). Такое соотношение объясняется тем, что человеческий глаз более чувствителен к зелёному цвету.

Цифровые фотоаппараты при съёмке сохраняют изображение на встроенной карте памяти в двух основных форматах: RAW и JPEG. Формат RAW — это необработанные данные (от англ. *raw* — «сырой», «сырьё», необработанный), т. е. коды сигналов, полученных каждым элементом чувствительной матрицы фотоаппарата. Снимок в формате RAW содержит наиболее полные данные, на каждый цветовой канал отводится 12–14 битов (в отличие от 8 битов при обычном RGB-кодировании). Существует более сотни различных RAW-форматов для разных моделей фотоаппаратов.

Если фотоаппарат сохраняет снимки в формате JPEG, «сырые» RAW-коды сразу после съёмки обрабатываются процессором с учётом настроек камеры, выбранных пользователем. Результат этой обработки затем и сохраняется на карте памяти, при этом:

- глубина кодирования уменьшается до 8 битов на канал (потеря информации!);
- изображение сжимается с потерями по алгоритму JPEG, в результате некоторые детали снимка безвозвратно теряются, хотя размер файла значительно уменьшается;
- все исходные данные уничтожаются.

Если камера была неверно настроена, полученное изображение будет низкого качества и восстановить его практически невозможно.

Поэтому профессиональные фотографы практически всегда сохраняют фотографии в формате RAW. Это даёт значительно больше возможностей для коррекции (улучшения) с помощью программного обеспечения. Например, часто удаётся увеличить контраст при плохих условиях съёмки. Для того чтобы получить изображение в одном из компьютерных форматов (JPEG, GIF, TIFF, BMP и др.), «сырые» снимки нужно обработать специальной программой, которая называется RAW-конвертором. Одна из известных программ-конверторов — **LR Adobe Photoshop Lightroom**.

Для загрузки изображений с фотоаппарата в компьютер эти два устройства соединяются кабелем через порт USB. После подключения фотоаппарат обнаруживается как новый съёмный диск. Можно также использовать устройство для чтения карт памяти — **кардридер** (при этом карту памяти нужно вынуть из фотоаппарата).

Сканирование

Сканирование — это ввод изображения в компьютер с помощью сканера. Многие растровые графические редакторы позволяют вводить сканированное изображение с помощью собственного меню. Например, в GIMP для этого нужно выбрать пункт меню **Файл — Создать — Сканер/Камера**. После этого запускается программа, обслуживающая сканер, в которой требуется задать режимы сканирования.

В первую очередь требуется определить **тип изображения**:

- чёрно-белое (двухцветное);
- полутоновое (оттенки серого, от чёрного до белого);
- цветное.

Второй важный параметр — это **разрешение**. Его нужно выбирать с некоторым запасом, учитывая, что отсканированное изображение потом чаще всего обрабатывается. Если изображение сканируется для вывода на экран (с разрешением 72–96 ppi) и его не нужно увеличивать, при сканировании можно выбрать разрешение 150–200 ppi. Если же вы хотите обработать рисунок и потом напечатать его на принтере, нужно выбирать разрешение не менее 300–400 ppi.

Иногда нужно отсканировать текст, чтобы потом подготовить электронную версию документа. Если документ не нужно редактировать и можно сохранить его в виде рисунка, достаточно установить разрешение 150–200 ppi. Если же требуется распознать

текст документа с помощью специальной программы (типа ABBYY FineReader или CuneiForm), нужно выбирать более высокое разрешение — не менее 300 ppi.

Кадрирование

После сканирования, как правило, выполняют **кадрирование**, т. е. выбирают нужные границы изображения, обрезают лишнее. Рассмотрим случай, когда фотография была положена неровно, так что её нужно сначала повернуть, а потом кадрировать (рис. 8.4).



Рис. 8.4

Сначала выполним поворот так, чтобы линия горизонта была строго горизонтальна. Для этого нужно загрузить рисунок в GIMP, выбрать инструмент  **Вращение** и щёлкнуть на рисунке. Появится окно, в котором можно установить нужный угол поворота (рис. 8.5).



Рис. 8.5

При изменении угла в основном окне виден результат применения этой операции. Можно также вращать изображение, схватив его мышью.

Если угол поворота не кратен 90 градусам, растровое изображение искажается. Для построения нового рисунка программа использует математические методы интерполяции, так же как и при изменении его размеров.

Нужно помнить, что при таком вращении границы *холста* (рабочей области рисунка) не меняются. Поэтому если ширина и высота изображения значительно различаются, при вращении на большой угол части рисунка могут оказаться за границами холста и будут обрезаны.

Часто бывает удобно сразу выполнить поворот холста на 90 и 180 градусов. Это можно сделать с помощью меню **Изображение — Преобразования**. Поворот на 90 градусов особенно полезен: так как рисунок вращается вместе с границами, ни одна его часть не будет потеряна. В меню **Изображение — Преобразования** также есть команды, которые строят зеркальное отражение по горизонтали или вертикали.

Теперь выполняем кадрирование¹. Выберем инструмент  **Кадрирование** и выделим прямоугольную область, оставив только нужную часть рисунка. Углы выделенной области можно перетаскивать мышью. При нажатии на клавишу Enter поля будут обрезаны.

Вопросы и задания

1. Какие два основных способа ввода растровых изображений вы знаете?
2. Как вы думаете, когда лучше использовать сканирование, а когда — фотографирование?
3. В каких форматах обычно сохраняют снимки цифровые фотоаппараты?
4. Что такое формат RAW? В чём его преимущества и недостатки по сравнению с форматом JPEG?
5. Что такое RAW-конвертер?
6. Как можно загрузить в компьютер изображения, записанные на карте памяти цифрового фотоаппарата?
7. Что такое сканирование?
8. Какие параметры важны при сканировании?
9. Что такое кадрирование? Зачем оно нужно?

Подготовьте сообщение

- а) «Форматы RAW: за и против»
- б) «Выбор параметров сканирования»

¹ В Adobe Photoshop кадрирование и поворот выполняются сразу (рамку кадра можно вращать).

§ 59



Коррекция фотографий

Качество многих фотографий можно значительно улучшить, устранив дефекты, возникшие из-за неправильной настройки фотокамеры. Кроме того, можно исправить некоторые искажения, которые вносит сам фотоаппарат.

Поскольку растровый рисунок хранится как набор чисел, кодирующих цвета пикселей, коррекция фотографий сводится к математической обработке этих данных с помощью специально разработанных алгоритмов.

Исправление перспективы

Оптическая система фотоаппаратов часто даёт искажения, в результате которых вертикальные линии (например, стены домов) становятся наклонными (рис. 8.6, а).

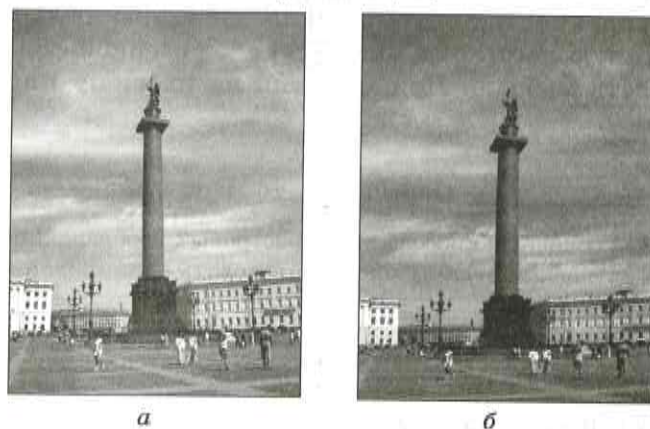


Рис. 8.6

Этот дефект легко исправляется в современных графических редакторах (рис. 8.6, б). В GIMP нужно выделить всё изображение (клавиши Ctrl+A или меню **Выделение – Все**) и выбрать инструмент  **Перспектива**. После этого углы рисунка, выделенные квадратами, надо перетащить так, чтобы выровнять вертикальные линии.

Гистограмма

Очень полезную информацию для оценки изображения даёт **гистограмма** — график специального вида, который показывает распределение пикселей по яркости (рис. 8.7). Высота вертикальных отрезков определяет количество пикселей, имеющих одинаковую яркость, от самых тёмных (слева) до самых светлых (справа).

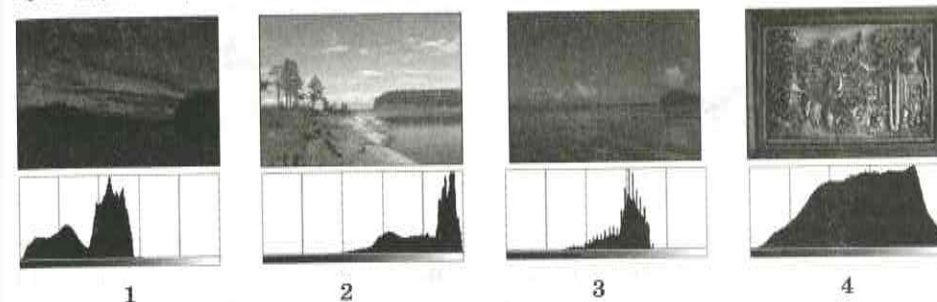


Рис. 8.7

Чтобы увидеть гистограмму загруженного изображения, в редакторе GIMP нужно выбрать пункт меню **Окна – Прикрепляющиеся диалоги – Гистограмма**.

По приведённым на рис. 8.7 гистограммам сразу видно, что:

- фото 1 слишком тёмное, потому что все пиксели сосредоточены в области тёмных тонов (слева);
- фото 2 слишком светлое (нет тёмных тонов);
- фото 3 мало контрастное (есть средние тона, но нет тёмных и светлых);
- фото 4 содержит пиксели всех уровней яркости¹.

Изображениям 1–3 не хватает контраста. Чтобы их улучшить, нужно «растянуть» гистограмму так, чтобы она занимала весь диапазон тонов, от чёрного до белого. Для этого в GIMP используется меню **Цвет – Уровни**². В появившемся окне нужно отрегулировать положение чёрного, серого и белого движков под гистограммой (рис. 8.8).

Все пиксели слева от чёрного движка становятся чёрными. Те, что оказались справа от белого движка, станут белыми. Таким

¹ На этой фотографии — картина «Пристань» художника К. А. Гоголева, вырезанная из дерева.

² В Adobe Photoshop — меню **Изображение – Коррекция – Уровни**.

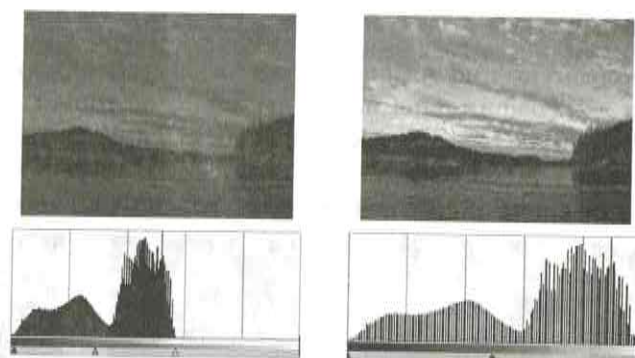


Рис. 8.8

образом, вся область между чёрным и белым движками растянется на весь диапазон. Передвигая серый движок (по умолчанию он находится посередине между чёрным и белым), можно менять контраст средних тонов. При этом в окне изображения мы сразу видим результат (такой эффект называется **предварительным просмотром**, т. е. просмотром результата какой-то операции до её окончательного применения).

После коррекции уровней внешний вид фотографии значительно улучшается, она становится более контрастной, содержит достаточно большое число и светлых, и тёмных пикселей. Однако гистограмма после коррекции не сплошная, а состоит из отдельных полосок (см. рис. 8.8). Это значит, что пикселей с некоторыми значениями яркости нет совсем (подумайте почему).

Заметим, что над гистограммой в окне **Уровни** расположен выпадающий список, в котором можно выбрать для коррекции один цветовой канал: красный, синий или зелёный. Это даёт возможность настраивать каждый канал отдельно.

Для выравнивания уровней можно также использовать окно регулировки яркости и контраста (меню **Цвет – Яркость – Контраст**). Более сложную тоновую коррекцию выполняют с помощью кривых (**Цвет – Кривые**).

Коррекция цвета

В некоторых изображениях явно видно, что какой-то оттенок явно сильнее, чем нужно, и из-за этого листва деревьев может оказаться синей, а красная крыша — зелёной. В этом случае можно применить коррекцию цвета, используя свои знания

о том, какие цвета имеют объекты в действительности. В программе GIMP нужно выбрать пункт верхнего меню **Цвет – Цветовой баланс**.

С помощью окна, изображённого на рис. 8.9, можно отдельно корректировать цвета тёмных участков (теней), пикселей средней яркости (полутонов) и светлых частей.

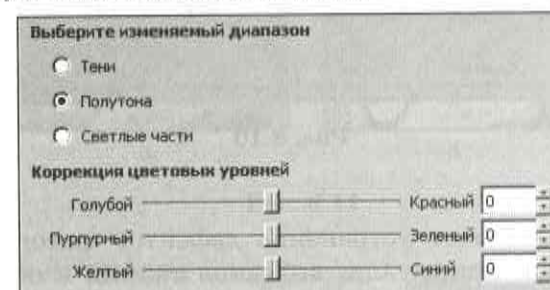


Рис. 8.9

Обратите внимание, что невозможно скорректировать один какой-то цвет, оставив все остальные без изменения. Предположим, что для какого-то пикселя мы уменьшили красную составляющую (в модели RGB). Это автоматически означает, что увеличится относительная доля зелёной и синей составляющих, т. е. усилится голубой канал (англ. *cyan*). Пары красный–голубой, зелёный–пурпурный (фиолетовый) и синий–жёлтый — это так называемые **дополнительные цвета** (увеличение одного из них вызывает уменьшение парного).

Иногда нужно превратить цветное изображение в чёрно-белое (серое), например для публикации в книге. В этом случае информация о цвете будет потеряна и останется только тон (яркость). В редакторе GIMP для этого используется пункт меню **Цвет – Обесцвечивание**.

Эта операция не так проста, поэтому программа предлагает на выбор несколько методов построения «серого» изображения (рис. 8.10).

Например, при выборе варианта **Светимость** тон пикселя вычисляется по формуле $0,21 R + 0,72 G + 0,07 B$, где R, G и B — значения яркостей красного, зелёного и синего каналов. В этой формуле учитывается, что человеческий глаз наиболее чувствителен к зелёному цвету. Нужно выбирать такой метод обесцвечивания, при котором полученное «серое» изображение выглядит, на ваш взгляд, лучше всего.

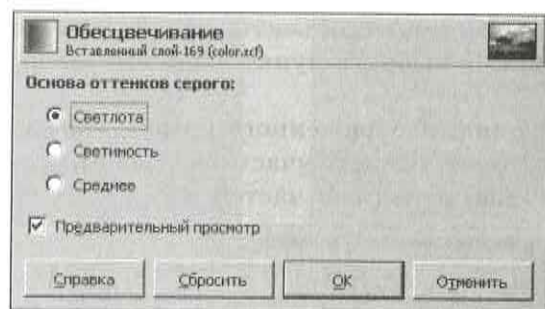


Рис. 8.10

Ретушь

Ретушью называют устранение дефектов фотографий — пятен, царапин, трещин, вуали, дефектов съёмки и обработки. Различают несколько видов ретуши.

Для портретов людей применяют **косметическую ретушь** — устранение дефектов (морщин, родимых пятен, складок кожи) и придание особой выразительности важным частям лица (глазам, бровям, губам). Этот процесс напоминает работу художника-визажиста или гримёра.

При обработке старых фотографий надо восстановить первоначальный вид изображения, внося как можно меньше изменений, поэтому говорят о **реставрации**.

Для устранения художественных дефектов применяют **композиционную ретушь** — кадрирование, удаление лишних элементов, добавление элементов, изменение фона, регулировку освещения.

Ретушь, которой раньше занимались специалисты-ретушёры, — сложное и утомительно занятие. Компьютерная ретушь обладает очень большими возможностями и абсолютно безопасна для оригинального изображения. На рисунке 8.11, а показана исходная фотография: она малоcontrastная, на лице ребёнка есть пятна, в левом верхнем углу видна граница фотокарточки. С помощью приёмов ретуширования можно исправить её (рис. 8.11, б).

Сначала убирают дефект всего изображения — недостаток контраста, это можно сделать с помощью настройки уровней (см. выше).

Для исправления локальных (местных) дефектов используют специальные инструменты редактора GIMP: Штамп, Осветление/Затемнение, Лечащая кисть, Размывание/Резкость.

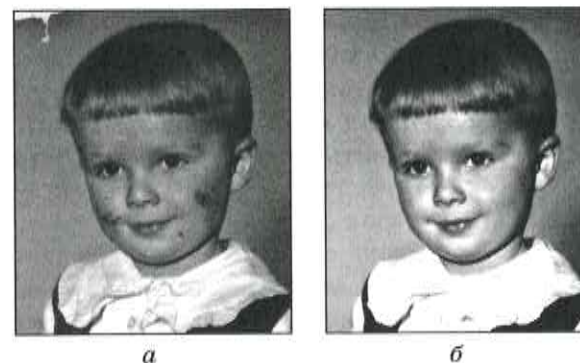


Рис. 8.11

Инструмент **Штамп** переносит изображение с одного участка на другой. Область-источник задаётся щелчком мышью при нажатой клавише **Ctrl**¹. После этого мы как бы рисуем кистью, копируя образец в другое место. В области параметров инструмента (рис. 8.12) можно настроить свойства инструмента, например непрозрачность, форму кисти и её размер (движок **Масштаб**) и др.

Инструмент **Лечащая кисть** работает практически так же, как и **Штамп**, но при переносе изображения учитывает окружающие пиксели, поэтому «впечатывание» получается менее заметным (но изображение немного размывается).

При использовании инструмента **Осветление/Затемнение** кисть при рисовании осветляет или затемняет (в зависимости от режима, выбранного в окне параметров) отдельные области.

Инструмент **Размывание/Резкость** позволяет размыть или повысить резкость участков изображения. Размытие может понадобиться, например, для того, чтобы предметы заднего плана не отвлекали внимание зрителей от главного объекта.

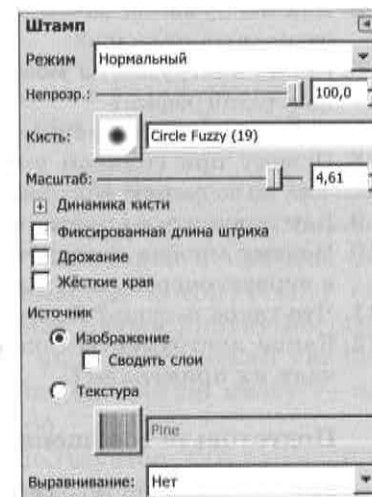


Рис. 8.12

¹ В Adobe Photoshop для этого служит клавиша **Alt**.

Повышение резкости — это некоторая операция с цветом пикселей, которая позволяет чётче выделить границы. Нужно понимать, что значительно увеличить резкость фотографии не удастся, потому что изображение не содержит нужной информации, и программа только пытается её восстановить математическими методами.



Вопросы и задания

1. Почему на многих фотографиях возникает искажение перспективы? Как его убрать?
2. Что такое гистограмма и что она показывает?
3. Как выполняется коррекция уровней?
4. Почему после коррекции уровней гистограмма не сплошная, а состоит из отдельных полосок?
5. Как вы думаете, зачем может понадобиться редактирование уровней отдельных каналов?
6. Какие инструменты можно использовать для коррекции неконтактных фотографий?
7. Как выполняется коррекция цвета?
8. Почему при сильном уменьшении яркости красного цвета фотография приобретает голубой оттенок?
9. Как получить из цветного изображения чёрно-белое?
10. Почему многие алгоритмы обесцвечивания изображений учитывают в первую очередь зелёный цветовой канал?
11. Что такое ретушь? Какие виды ретуши вы знаете?
12. Какие инструменты можно использовать для ретуши? В каких случаях их применяют?



Подготовьте сообщение

- а) «Что такое гистограмма?»
- б) «Коррекция цвета изображения»
- в) «Использование кривых для коррекции фотографий»
- г) «Алгоритмы обесцвечивания изображений»



Задачи

1. Проверьте, как влияет на гистограмму изменение яркости и контраста.
2. Попробуйте обесцветить какое-нибудь изображение несколькими методами и выберите лучший из них (на ваш взгляд).
3. Отсканируйте какую-нибудь старую фотографию с дефектами и реставрируйте её.

§ 60

Работа с областями

Выделение областей

Часто нужно работать не с целым изображением, а только с некоторой областью, так что все пиксели вне этой области не должны изменяться. Для выделения областей в редакторе GIMP используют несколько инструментов.

Простейшие инструменты выделения —  **Прямоугольник** и  **Эллипс**. Если при их использовании удерживать нажатой клавишу Shift, будет выделен соответственно квадрат или окружность, а при нажатой клавише Alt происходит выделение от центра, а не с угла области.

С помощью инструмента  **Лассо** выделяют область, ограниченную ломаной линией (щелчками мышью в узлах ломаной) или вообще произвольную область (обводя её при нажатой левой кнопке мыши).

Инструменты  **Волшебная палочка** и  **Выделение по цвету** применяют для выделения областей одного цвета (или близких цветов, если в параметрах инструмента установлен ненулевой порог чувствительности). Выделение начинается в той точке, где выполнен щелчок мышью. Различие между двумя инструментами состоит в том, что **Волшебная палочка** выделяет только связанную область около начальной точки, а **Выделение по цвету** — одноцветные пиксели по всему изображению.

Инструмент  **Умные ножницы** используется для выделения областей с чёткими, но неровными границами. Пользователь щёлкает мышью в опорных точках, между которыми программа дорисовывает линию выделения, стараясь следовать границе между двумя цветами. Когда выделение закончено, нужно щёлкнуть мышью в середине области.

Если в момент начала выделения области была нажата клавиша Shift, новая область добавляется к уже выделенной, а при нажатой клавише Ctrl¹ — вычитается из выделенной ранее. С помощью такого приёма можно строить сложные области.

Любой пиксель может быть выделен частично, например на 25%. Это значит, что для такого пикселя эффект, применённый к области (например, заливка) будет ослаблен в 4 раза. Частичное

¹ В редакторе Adobe PhotoShop для этой цели используется клавиша Alt.

выделение получается при использовании двух параметров: сглаживания (англ. *antialiasing*) и растушёвки (они задаются в окне параметров выбранного инструмента). Выделим три одинаковых круга тремя способами (рис. 8.13):

- 1) без сглаживания и растушёвки;
- 2) со сглаживанием;
- 3) с растушёвкой.

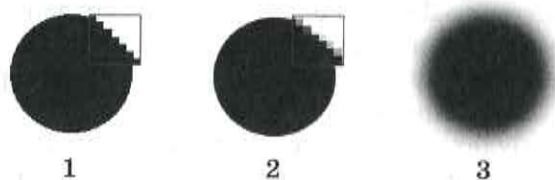


Рис. 8.13

Каждую из этих областей зальём чёрным цветом (пункт меню **Правка — Залить цветом переднего плана**).

В первом случае видим резкую границу в виде «лесенки»: пиксели или выделены на 100% (они стали чёрными), или вообще не выделены (остались белыми). Во втором случае граница сглаживается за счёт частично выделенных (серых) пикселей. При использовании растушёвки (случай 3) граница размыта.

В меню **Выделение** собраны команды для работы с областью (увеличение, уменьшение, растушёвка и т. д.).

Быстрая маска

Маска — это «накладка», которая скрывает часть объекта. В графических редакторах маска позволяет защитить от изменений некоторые части изображения. Например, когда мы выделяем область, создаётся маска, которая защищает всё, что не выделено.

В простейшем случае каждый пиксель изображения может быть полностью выделен (открыт для изменений) или невыделен (защищён). Кроме того, пиксель может быть частично выделен — есть 256 ступеней выделения, которые кодируются числами от 0 (чёрный цвет, защищённый пиксель) до 255 (белый цвет, полностью выделен, открыт). Таким образом, маска может быть закодирована как чёрно-белое полутоновое изображение.

При выделении сложных областей, которые нужно не раз редактировать, удобно использовать режим **Быстрая маска**. В этом режиме маску можно редактировать с помощью инструментов рисования. Для перехода в режим быстрой маски в редакторе GIMP используют клавиши Shift+Q. Вся закрытая (невыделенная) часть рисунка заливается полупрозрачным красным цветом, а выделенная полностью прозрачна. Рисуя (например, карандашом или кистью) чёрным цветом, мы скрываем область (удаляем из выделения), а при рисовании белым цветом открываем её. Если выбрана кисть с мягкими границами, некоторые пиксели будут выделены частично. В режиме быстрой маски часто используется инструмент  **Градиент**, с помощью которого можно получить плавный переход от полного выделения к невыделенной части (маска плавно меняется от белого до чёрного цвета).

Исправление «эффекта красных глаз»

При фотосъёмке со вспышкой нередко возникает так называемый «эффект красных глаз» — лучи вспышки отражаются от глазного дна глаз человека (оно имеет красный цвет) и попадают в объектив фотокамеры. Это особенно заметно при съёмке в тёмном помещении, когда зрачки глаз расширены. На самом деле, без вспышки мы видим тёмный зрачок, который и нужно восстановить.

Существуют различные методы устранения «эффекта красных глаз». Один из них — обесцвечивание (до серого цвета) и затем тонирование, при котором мы как бы смотрим на чёрно-белый рисунок через прозрачное цветное стекло. Цвет этого «стекла» в программе GIMP задаётся с помощью меню **Цвет — Тонирование**: в диалоговом окне можно с помощью ползунков выбрать оттенок (**Тон**), насыщенность и светлоту (яркость) цвета.

На рисунке 8.14 показан один из вариантов установки движков, при котором выделенный зрачок становится тёмным.

Кроме того, можно использовать фильтры — алгоритмы автоматической обработки изображений (см. следующий параграф). Например, в редакторе GIMP в меню **Фильтры** есть фильтр **Улучшение — Удалить эффект красных глаз**, который автоматически выполняет тонирование красных участков в выделенной области.

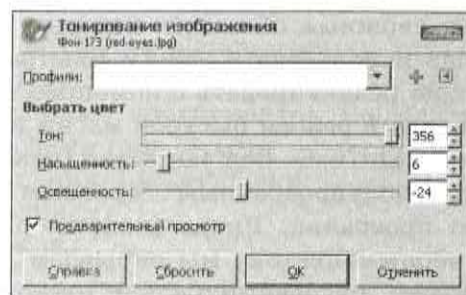


Рис. 8.14

? Вопросы и задания

1. Зачем нужно выделять области? Какие инструменты для этого используют?
2. Какие способы выделения и редактирования сложных областей вы знаете?
3. Что такое частичное выделение пикселя?
4. Что такое сглаживание? В каких случаях без сглаживания получается граница области в виде «лесенки», а в каких — нет?
5. Что такое растушёвка? Чем она отличается от сглаживания?
6. Подумайте, для каких целей можно использовать сильную растушёвку области выделения.
7. Что такое быстрая маска и зачем она нужна?
8. Объясните, почему возникает «эффект красных глаз» и как его исправить.

✓ Задача

Попробуйте исправить «эффект красных глаз» на какой-нибудь фотографии.

§ 61 Фильтры

Что такое фильтры?

Фильтр — это алгоритм автоматической обработки пикселей изображения, который применяется ко всему изображению или к выделенной области.

Фильтры используют некоторые математические преобразования (иногда достаточно сложные), в результате которых цвет

(код) каждого пикселя изменяется с учётом кодов соседних пикселей. Большинство фильтров настраивается с помощью диалоговых окон, в которых можно изменять параметры и сразу видеть получаемый результат.

Если эффект оказался слишком слабым и действие фильтра надо повторить, достаточно выбрать первый сверху пункт меню **Фильтр** (в нём будет название последнего применённого фильтра) или нажать клавиши Ctrl+F.

Наоборот, если действие фильтра надо ослабить, можно выбрать пункт меню **Правка — Ослабить**. В этом случае итоговое изображение строится как результат наложения отфильтрованного рисунка (с заданным уровнем непрозрачности) и исходного. Степень непрозрачности устанавливается в диалоговом окне.

Фильтры можно (очень условно) разделить на 2 группы: фильтры для коррекции изображений и фильтры для художественной обработки.

Современные графические редакторы не только содержат большой набор фильтров, но и позволяют пользователю создавать и подключать свои собственные фильтры. Их также называют **плагинами** (от англ. *plug-in* — независимый программный модуль, подключаемый к программе). Плагины для редактора GIMP обычно пишутся на языках Script-Fu или Python.

Фильтры для коррекции изображений

Нередко изображение получается нечётким (размытым) из-за того, что автофокус фотоаппарата неверно определил объект, интересующий фотографа, или в момент съёмки камера немного сдвинулась. Для того чтобы повысить резкость, применяют специальные фильтры из группы **Улучшение: Нерезкая маска** и **Повышение резкости**. При этом нужно понимать, что эти фильтры не позволяют восстановить потерянную информацию, поэтому в результате их применения мелкие детали не появятся. Во многих случаях результат можно значительно улучшить, однако при сильном размытии качественное изображение получить не удастся.

Достаточно часто применяются и фильтры группы **Размывание**. Они выполняют обратное действие — снижают резкость. При обработке фотографий это может понадобиться, например, для того, чтобы не отвлекать внимание зрителя на второстепенные детали заднего плана, оставив чётким только центральный объект. Из этой группы наиболее популярны фильтры **Гауссово размыва-**

ние (равномерное) и Размывание движением (позволяет создать эффект движения в определённом направлении).

Художественные фильтры

Художественные фильтры предназначены для имитации каких-то эффектов. На рисунке 8.15 показана фотография цветка и результат применения к нему нескольких фильтров редактора GIMP.

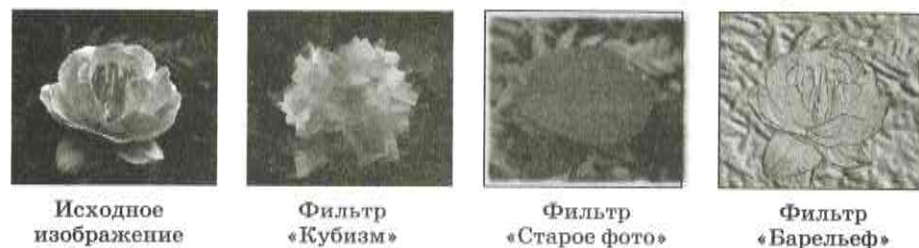


Рис. 8.15

Изучать многочисленные художественные фильтры лучше всего на практике, экспериментируя с их настройками.



Вопросы и задания

1. Что такое фильтр? Как можно применить фильтр к части изображения?
2. Как усилить или ослабить действие фильтра?
3. Можно ли подключить к графическому редактору GIMP свой фильтр?

§ 62

Многослойные изображения

Зачем нужны слои?

Пусть нам нужно поместить нарисованного человечка на некоторый фон так, как показано на рис. 8.16. Скорее всего, сделать рисунок с первого раза не удастся, и его придётся не раз переделывать, чтобы получился хороший результат.



Рис. 8.16

Как вы знаете, растровый рисунок — это просто множество пикселей, каждый из которых имеет свой цвет, независимый от других. Представим себе, что после того, как человечек был нарисован, мы захотели передвинуть его в другое место. К сожалению, сделать это весьма непросто, потому что при рисовании мы меняем цвет пикселей фона и таким образом уничтожаем существующую информацию, которую потом невозможно восстановить.

Такая же проблема возникает при добавлении надписей на рисунок — текст надписи очень сложно изменить, если изображение под ней испорчено.

Как же избежать необратимых изменений? Человек видит в рисунке не пиксели, а знакомые ему объекты: скалы, берег, воду, тело человечка, футболку, шорты, кепку. Поэтому нужно попытаться каждый объект рисовать отдельно, так, чтобы его можно было изменять независимо от других.

Представим себе, что над фоновым рисунком расположено несколько стёкол, на каждом из которых нанесено изображение какого-то объекта (рис. 8.17).

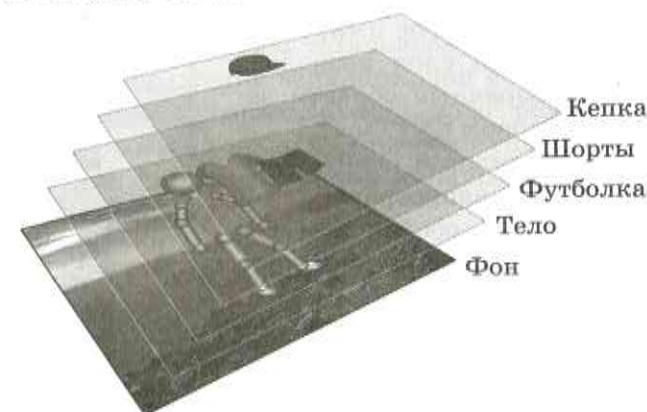


Рис. 8.17

Фактически мы разбили весь рисунок на отдельные слои (англ. *layers*), каждый из которых можно изменять и перемещать независимо от других. Однако если посмотреть на эту стопку сверху, мы увидим полный рисунок. Через прозрачные области верхних слоев видны изображения на нижних слоях. Этот принцип широко используется в графических редакторах, такие изображения называются многослойными. Применяя многослойные изображения, можно, например, составить портрет человека по

описанию («фоторобот») или «примерить» ему новую одежду или причёску.

Не все форматы графических файлов поддерживают слои. Например, популярные форматы BMP, JPEG, GIF и PNG могут хранить только однослойные («плоские») рисунки. Для записи многослойных изображений чаще всего используют форматы PSD (редактор Adobe Photoshop) или XCF (редактор GIMP). Нужно учитывать, что при этом в файле фактически хранится несколько отдельных изображений, поэтому его объём значительно возрастает.

Работа со слоями

Для работы со слоями в редакторе GIMP предназначено специальное окно **Слои** (рис. 8.18). Если этого окна нет на экране, вызвать его можно с помощью меню **Окна – Прикрепляющиеся диалоги – Слои** или комбинации клавиш **Ctrl+L**.

Каждый слой имеет свое имя, двойной щелчок на имени слоя позволяет изменить его. Прозрачные области залиты клетчатым узором.

Обычно самый нижний слой — фоновый, он полностью непрозрачен. При необходимости можно обойтись и без фона, например если нужно получить изображение с прозрачными областями.

Рисование происходит на активном (текущем) слое, который выделен в списке слоев. Остальные слои при этом не изменяются. Выбрать активный слой можно щелчком мыши в списке слоев.

Кнопки  и  позволяют переместить активный слой выше или ниже по списку (изменить порядок слоев). То же самое можно сделать с помощью мыши, перетащив слой в нужное место списка.

Если установить флажок **Запереть**, все прозрачные области активного слоя будут сохранены (рисовать можно только там, где уже что-то нарисовано).

Для того чтобы переместить всё изображение слоя, применяют инструмент  **Перемещение**. В зависимости от настроек мож-

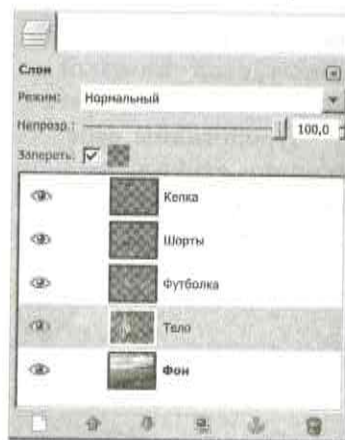


Рис. 8.18

но перемещать активный слой или слой, выбранный щелчком мыши на поле рисунка.

С помощью кнопки  можно создать новый пустой слой выше активного, а кнопка  создаёт копию активного слоя (например, чтобы сохранить исходное изображение при экспериментах).

Слой можно временно отключить, щелкнув на значке  в соответствующей строке. Щелкнув в этом же месте повторно, мы вновь сделаем слой видимым. Для того чтобы совсем удалить текущий слой, нужно щелкнуть на кнопке  или перетащить слой из списка на эту кнопку.

Слой можно сделать полупрозрачным: для этого движок **Непрозрачность** сдвигается влево. Справа от этого движка показывается непрозрачность слоя в процентах.

Список **Режим** определяет, как слой влияет на изображение, полученное с нижних слоев. Пусть, для простоты, рисунок содержит два слоя. Тогда цвет пикселя итогового изображения вычисляется по некоторому алгоритму на основе цветов соответствующих пикселей этих двух слоев. Этот алгоритм и определяется в списке **Режим**. По умолчанию установлен режим **Нормальный** — это значит, что изображение верхнего слоя полностью перекрывает нижний (с учётом прозрачности). Другие режимы позволяют, например, перекрывать только тёмные или только светлые области, затемнять или осветлять рисунок, находить «разность» (различие двух рисунков).

Иногда требуется связать несколько слоев так, чтобы они перемещались вместе. Для этого напротив каждого слоя нужно щелкнуть мышью между значком  и уменьшенным изображением слоя. В этом месте появится изображение участка цепи (рис. 8.19).

Слои можно объединять, т. е. делать из двух или нескольких слоев новый слой. Нужно учитывать, что, если изображения на этих слоях перекрываются, разделить их будет практически невозможно. Существуют три варианта объединения:

- объединение текущего слоя с предыдущим (нижележащим);
- объединение всех видимых слоев;
- сведение изображения (объединение всех слоев в один фоновый слой).

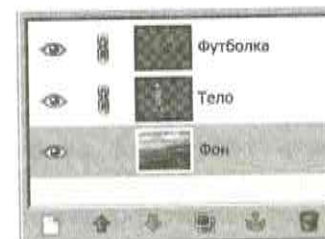


Рис. 8.19

В редакторе GIMP эти операции выполняются с помощью контекстного меню слоя, которое появляется при щелчке правой кнопкой мыши на нужном слое в окне **Слои**.

Текстовые слои

В простых графических редакторах текст, размещённый на поле рисунка, «встраивается» в изображение и сразу становится набором пикселей. Такой текст нельзя редактировать, перемещать и т. п. В более совершенных программах надписи хранятся на отдельных слоях. Большинство используемых сейчас компьютерных шрифтов — *векторные*, в них буквы задаются узловыми точками и соединяющими их отрезками или кривыми. Это позволяет многократно изменять содержание и оформление текста (в том числе гарнитуру и размер шрифта), не теряя качество изображения. Например, можно легко исправить опечатку, замеченную не сразу.

В редакторе GIMP инструмент **Текст** обозначается кнопкой . После его выбора нужно щёлкнуть мышью в том месте, где будет левый верхний угол надписи и ввести текст в появившемся окне (рис. 8.20). Свойства текста настраиваются на панели свойств инструмента в нижней части Панели инструментов.

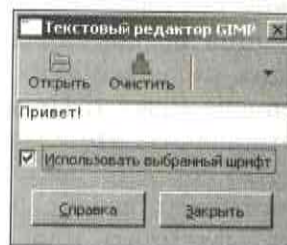


Рис. 8.20

В окне **Слои** появляется новый слой с текстом, причём вместо уменьшенного изображения содержимого вы увидите значок . Это означает, что текст в этом слое хранится в векторном формате. Можно превратить текстовый слой в обычный (растровый), выбрав в меню пункт **Слои — Удалить текстовую информацию**. После этого текст будет редактироваться только как точечный рисунок.

Маска слоя

При создании сложных изображений желаемый результат практически никогда не получается с первого раза. Поэтому важно применять по возможности неразрушающие методы обработки, при которых информация не теряется и всегда можно вернуться к исходным данным. Один из таких приёмов — маска слоя, которая позволяет сделать часть изображения на слое невидимой (или полупрозрачной), ничего не удаляя.

Маска слоя — это полутоновое («серое») изображение, связанное с данным слоем. Чёрные области в маске закрывают рисунок на слое, а белые открывают. Серые тона — это частично открытые (полупрозрачные) области. Фактически маска слоя позволяет установить разную прозрачность для разных участков слоя. При этом рисунок на слое полностью сохраняется, что позволяет вернуться к исходному варианту в случае неудачи.

Чтобы добавить или удалить маску слоя, используют команду **Добавить маску слоя** из контекстного меню окна **Слои**. Если у слоя есть маска, в списке слоёв появляется второй значок: . Если выделен левый значок, вы меняете изображение слоя, а если правый — маску слоя. При редактировании маски используются только оттенки серого цвета. Чтобы увидеть маску, в контекстном меню слоя нужно выбрать пункт **Показать маску**. Действие маски можно временно отключить с помощью команды **Скрыть маску** из контекстного меню.

Вопросы и задания

1. Что такое «неразрушающие методы обработки»?
2. Зачем используют слои? Что это даёт?
3. Какие форматы файлов используют для хранения многослойных изображений?
4. Какие операции можно выполнять со слоями?
5. Что такое фоновый слой?
6. Как обозначаются прозрачные области слоя?
7. Какие свойства слоя определяются с помощью выпадающего списка **Режим**?
8. Каким образом можно объединить слои?
9. Какие особенности имеют текстовые слои?
10. Что такое маска слоя? Зачем она нужна?

§ 63 Каналы

Цветовые каналы

При RGB-кодировании с глубиной 24 бита на пиксель цвет каждого пикселя хранится как набор из трёх чисел (от 0 до 255), обозначающих яркости красной, зелёной и синей составляющих. Возьмём одну из трех цветовых составляющих (например, крас-

ную) и построим новое изображение, в котором каждому пикселю соответствует код от 0 до 255. Будем считать, что 0 обозначает чёрный цвет, 255 — белый, а промежуточные значения — оттенки серого цвета. Тогда получается, что мы построили полутоновое («серое») изображение, которое совпадает по размерам с исходным рисунком и показывает «вклад» выбранного цвета. Такое вспомогательное изображение называют каналом.



Канал — это чёрно-белое полутоновое изображение, которое показывает степень влияния какого-то эффекта на изображение.

На рисунке 8.21, а (и цветном рис. на форзаце) изображён светофор со всеми зажжёнными лампами, а на рис. 8.21, б–г — цветовые каналы модели RGB, показывающие влияние красного, зелёного и синего цветов.

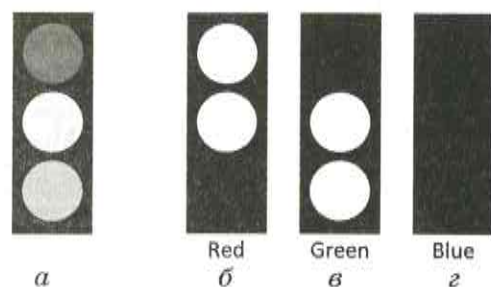


Рис. 8.21

Очевидно, что для области красного цвета будет активным только красный канал, в котором эта область закрашена белым цветом. Для области зелёного цвета также «открыт» только один канал — зелёный. Жёлтый цвет получается «сложением» красного и зелёного, поэтому в обоих этих каналах жёлтой области соответствует белый цвет. Синего цвета на рисунке нет вообще, поэтому синий канал залит чёрным цветом.

В редакторе GIMP цветовые каналы модели RGB можно увидеть в окне Каналы — рис. 8.22 (меню **Окна** — **Прикрепляющиеся диалоги** — **Каналы**). Любой канал можно отключить, щёлкнув на значке  в соответствующей строке (повторный щелчок в этом месте снова включает канал).



Рис. 8.22

По умолчанию выделены (синим фоном) все три цветовых канала, т. е. все инструменты рисования и коррекции применяются к ним одновременно. При желании можно выделить какой-то один канал и редактировать только его.

У изображений с прозрачными и полупрозрачными областями появляется четвёртый канал — так называемый **альфа-канал**, определяющий прозрачность (рис. 8.23). Чёрные пиксели альфа-канала определяют прозрачное изображение, белые — полностью непрозрачное. Чтобы построить такое изображение, нужно выбрать команду **Добавить альфа-канал** из контекстного меню окна **Слои**. При этом фоновый слой превращается в обычный слой, на нём теперь можно делать прозрачные участки (например, удалив выделенную область или используя инструмент  **Ластик**).

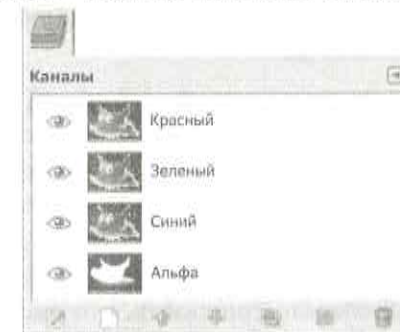


Рис. 8.23

Редактор GIMP не позволяет редактировать изображение в других цветовых моделях (например, в модели CMYK, которая используется при печати). Однако с помощью меню **Цвет** — **Составляющие** — **Разобрать** можно построить новое многослойное полутоновое изображение, в котором каждый слой будет представлять собой канал выбранной цветовой модели. После этого можно отдельно редактировать каждый канал и затем обратно «собрать» полное изображение с помощью меню **Цвет** — **Составляющие** — **Собрать**. Например, чтобы избежать искажения цветов при повышении резкости изображения, рекомендуется применять эту операцию только к каналу яркости (V-каналу) в модели HSV.

Сохранение выделенной области

Когда вы выделяете какую-то область рисунка, создаётся маска — временный канал, в котором чёрный цвет обозначает невыделенные пиксели, а белый — выделенные. Эту область можно запомнить как новый канал (**Выделение** — **Сохранить в канале**).

Каналы, полученные в результате сохранения выделенных областей (на рис. 8.24 — канал **Цветок**), располагаются в нижней части окна **Каналы** (ниже цветовых каналов). Чтобы снова превратить такой канал в выделение, нужно выделить его в списке и щёлкнуть по кнопке



Рис. 8.24



Вопросы и задания

1. Что такое канал?
2. Что такое альфа-канал?
3. Можно ли сказать, что маска — это канал? Ответ обоснуйте.
4. Сколько ступеней выделения можно применять, если использовать 16 битов на канал?
5. Как можно сделать изображение с прозрачными областями?
6. Как разбить изображение на отдельные каналы в цветовых моделях CMYK и HSV?
7. Как собрать изображение из отдельных каналов?
8. Как сохранять выделенные области в каналах и использовать их повторно?



Подготовьте сообщение

- а) «Редактирование изображений в модели CMYK»
- б) «Редактирование изображений в модели HSV»
- в) «Редактирование изображений в модели Lab»



Задача

Разбейте какое-нибудь размытое изображение на составляющие модели HSV, примените операцию повышения резкости к каналу V и соберите изображение заново. Сравните результат с тем, что получается при применении той же операции повышения резкости ко всем каналам.

§ 64

Иллюстрации для веб-сайтов

Иллюстрации для веб-сайтов должны быть сохранены только в тех графических форматах, которые умеют отображать браузеры:

- **JPEG** (англ. *Joint Photographic Experts Group* — объединённая группа фотографов-экспертов, файлы с расширением jpg или jpeg);
- **GIF** (англ. *Graphics Interchange Format* — формат для обмена изображениями, файлы с расширением gif);
- **PNG** (англ. *Portable Network Graphics* — переносимые сетевые изображения, файлы с расширением png).

Все эти форматы предназначены только для «плоских» (однослойных) изображений, поэтому все слои многослойных изображений при сохранении приходится сводить в один слой. При этом теряется информация (снова разделить слои практически невозможно), поэтому рекомендуется всегда оставлять на всякий случай исходные многослойные файлы (в форматах PSD или XCF).

В форматах JPEG, GIF и PNG используется сжатие данных для того, чтобы уменьшить объём файлов и таким образом ускорить загрузку веб-страницы.

При сохранении рисунков в формате JPEG можно регулировать степень сжатия. Если увеличивать степень сжатия, то качество рисунка ухудшается, и наоборот. Поэтому при сохранении нужно выбирать максимальную степень сжатия, при которой изображение выглядит приемлемо.

При выборе формата для конкретного изображения нужно учитывать, что:

- в формате JPEG можно хранить только полноцветные изображения (с глубиной цвета 24 бита на пиксель); для уменьшения объёма файла используется сжатие с потерями, которое приводит к размытию границ объектов, появлению пятен вокруг них и одноцветных квадратов размером 8×8 пикселей (это так называемые *артефакты* алгоритма сжатия JPEG);
- формат JPEG не поддерживает прозрачность;
- в формате GIF можно хранить только изображения с палитрой (от 2 до 256 цветов);

- анимация поддерживается только в формате GIF;
- в форматах GIF и PNG используется сжатие без потерь (рисунок при сжатии не искажается); для уменьшения объёма файла можно уменьшать количество цветов в палитре;
- полупрозрачные изображения можно сохранять только в формате PNG, где для каждого пикселя может храниться дополнительный байт, задающий прозрачность (альфа-канал).

Напомним, что файлы всех форматов кроме основной информационной части, в которой кодируется цвет каждого пикселя, содержат так называемый *заголовок* — служебную информацию: размеры рисунка, цветовую палитру и т. п.

В таблице на рис. 8.25 показаны изображения, полученные при сохранении одного и того же рисунка в разных форматах, а также указаны области применения каждого формата.

Формат	Примеры			Применение
JPEG				Фотографии, непрозрачные изображения с размытыми границами объектов и плавными переходами цветов
GIF				Рисунки с небольшим количеством цветов; мелкие рисунки с чёткими границами; рисунки с прозрачными областями; анимация
PNG				Высококачественные изображения, рисунки с прозрачными и полупрозрачными областями

Рис. 8.25

Чтобы при сохранении рисунка вручную задать количество используемых цветов, необходимо преобразовать его к **индексированному режиму**, т. е. закодировать с палитрой. Для этого используется пункт меню **Изображение — Режимы — Индексированное**. Если исходное изображение имеет глубину цвета 24 бита на пиксель (режим RGB), при таком преобразовании происходит потеря информации о цвете. Поэтому лучше работать с копией рисунка, сохранив полноцветный оригинал в отдельном файле. Отметим, что формат PNG обеспечивает лучшее сжатие, чем GIF.

Как видно из таблицы на рис. 8.25, кодирование с палитрой (форматы GIF и PNG) плохо подходит для хранения изображений с плавными переходами цветов (**градиентами**). Это связано с тем, что уменьшается количество используемых цветов и переходы становятся более резкими. Чтобы несколько улучшить результат, при переходе к индексированному изображению можно включить **размывание цвета** (англ. *dithering* — растривание). Суть этого подхода состоит в том, что области, залитые в исходном рисунке «отброшенными» цветами, строятся как мозаика из пикселей тех цветов, которые остались в палитре. На рисунке 8.26 показан результат применения размывания при использовании 8-цветной палитры (редактор GIMP, размывание Флойда–Стейнберга). Объём файла увеличился с 1184 до 1664 байтов из-за того, что алгоритм сжатия LZW, используемый в формате GIF, хуже сжимает разноцветные области, чем строки одного цвета.



Рис. 8.26

Вопросы и задания

1. Какие форматы используются для хранения изображений на веб-страницах?
2. Опишите особенности и области применения каждого из форматов.
3. Определите лучший формат для сохранения:
 - а) фотографии;
 - б) изображения с чёткими границами областей;
 - в) изображения с прозрачными областями;
 - г) изображения с полупрозрачными областями;
 - д) анимации.
4. Что такое индексированное изображение?
5. Что происходит при переходе от режима RGB к индексированному изображению? Как выбрать количество цветов в палитре?
6. Что такое размывание цвета и зачем оно используется?



Подготовьте сообщение

«Оптимизация изображений для веб-страниц»

§ 65

Анимация



Анимация (англ. *animation* — одушевление) — это «оживление» изображения. При анимации несколько рисунков (кадров) сменяют друг друга через заданные промежутки времени.

Для создания анимации в графических редакторах обычно используются многослойные изображения, каждый слой соответствует одному кадру. В простейшем случае слои просто сменяют друг друга: сначала виден только первый слой, через заданное время он скрывается и показывается второй слой и т. д. Такой метод называется **заменой** (англ. *replace*) — рис. 8.27.

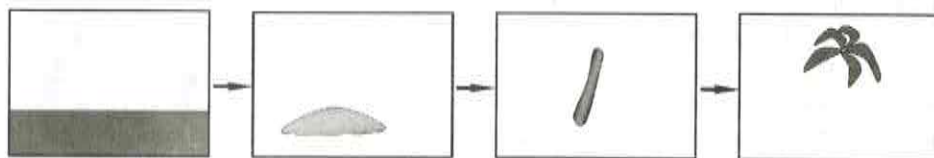


Рис. 8.27

Можно использовать и другой подход: новый слой «накладывается» на существующее изображение — это метод **объединения** (англ. *combine*). Для того же набора слоёв, что и в предыдущем примере, порядок анимации такого типа показан на рис. 8.28.

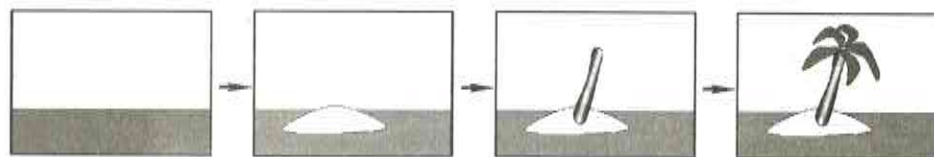


Рис. 8.28

Тип анимации для каждого слоя устанавливается отдельно, поэтому в одной анимации можно совмещать оба метода.

Анимация строится за два шага:

- 1) готовятся изображения для всех слоёв;
- 2) устанавливается время задержки и вид анимации (для каждого слоя отдельно).

Чтобы готовую анимацию можно было просматривать без графического редактора (например, на веб-странице), нужно сохранить её в файле формата GIF (т. е. **экспортировать** — записать в другом формате с преобразованием).

В редакторе GIMP при сохранении файла с несколькими слоями в формате GIF появляется диалоговое окно, в котором нужно выбрать вариант **Сохранить как анимацию** и нажать кнопку **Экспорт** (рис. 8.29).

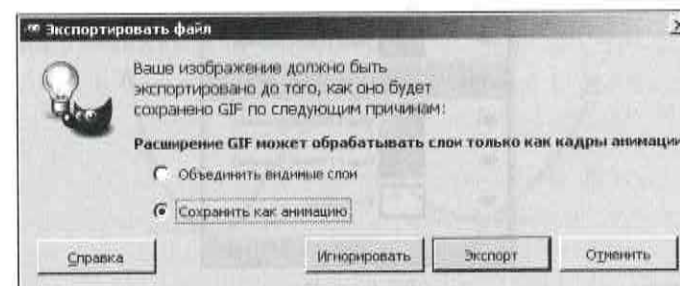


Рис. 8.29

После этого в следующем окне нужно настроить параметры сохранения анимации (рис. 8.30).

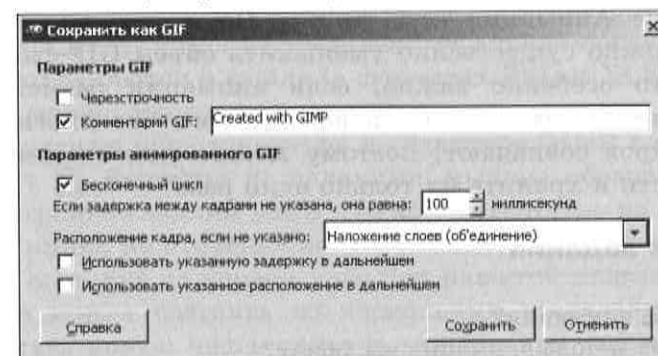


Рис. 8.30

Если установлен флажок **Бесконечный цикл**, то анимация будет повторяться бесконечно, иначе она выполнится только один

раз. Здесь же выбирается задержка между кадрами и вид анимации (список **Расположение кадра**).

Если теперь открыть сохранённый таким образом GIF-файл с анимацией, в названиях слоёв-кадров мы увидим в скобках время задержки (например, **100ms** обозначает 100 миллисекунд) и метод анимации (**combine** или **replace** — рис. 8.31). Изменив названия слоёв, можно задать задержку и метод анимации для каждого слоя отдельно.

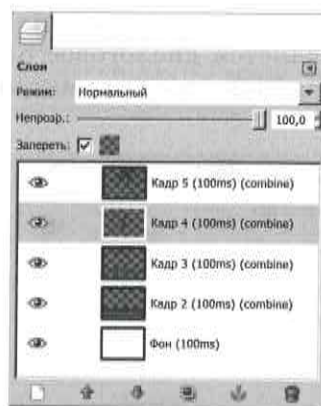


Рис. 8.31

Чтобы просмотреть анимацию, не выходя из редактора GIMP, нужно выбрать фильтр **Анимация – Воспроизвести** в меню **Фильтры**.

В группе **Анимация** есть фильтр **Оптимизация**, с помощью которого можно существенно уменьшить объём GIF-файла с анимацией. Это особенно важно, если анимация размещается на веб-странице. Оптимизация основана на том, что многие пиксели разных кадров совпадают, поэтому можно удалить все повторяющиеся части и хранить их только один раз.



Вопросы и задания

1. Что такое анимация?
2. Какие два метода анимации вы знаете?
3. Что такое экспортирование документа? Зачем оно применяется?
4. Как в редакторе GIMP установить время задержки и способ анимации для каждого кадра отдельно?
5. На чем основана оптимизация файлов с анимацией?



Подготовьте сообщение

«Анимация на веб-страницах: за и против»

§ 66 Контур

Современные растровые графические редакторы (Adobe Photoshop, GIMP) могут работать с векторными изображениями — **контурами** (англ. *path*). Контур хранится в памяти не как набор пикселей, а как *кривая Безье*, которая задаётся опорными точками (на рис. 8.32 — точки **A**, **B**, **B**, **Г** и **Д**) и координатами «рычагов» (управляющих линий), связанных с каждой точкой (см. § 16 из учебника для 10 класса).

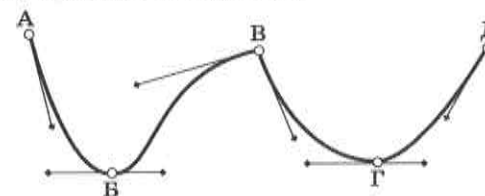


Рис. 8.32

Если оба управляющих рычага находятся на одной прямой, получается сглаженный узел (узлы **B** и **Г** на рисунке), если нет, то угловой узел (например, узел **B**).

С помощью контуров удобно выделять области рисунка, которые должны иметь строго определённые «правильные» границы. Контур сохраняется в файле (в форматах PSD и XCF), после создания их можно многократно изменять.

Для создания нового контура в редакторе GIMP надо выбрать инструмент  **Контур** и щёлчками мышью обозначить узлы. Чтобы построить гладкий узел, нужно «вытащить» из него управляющую линию, не отпуская левую кнопку мыши. Контур можно замкнуть, щёлкнув на первом узле при нажатой клавише Ctrl.

Контур может состоять из нескольких несвязанных частей. Чтобы начать новую часть контура, нужно при добавлении узла нажать клавишу Shift. Чтобы объединить две части, нужно выделить конечный узел одной из них и при нажатой клавише Ctrl щёлкнуть на узле, с которым его нужно соединить.

После создания контур можно изменять, перетаскивая узлы и управляющие рычаги. Если при этом удерживать клавишу Shift, оба рычага будут расположены на одной прямой и получается гладкий узел.

Новые узлы добавляются на соединяющие линии при нажатой клавише Ctrl. Для удаления узлов и соединяющих линий нужно удерживать одновременно Ctrl и Shift.

Чтобы выделить одновременно несколько узлов, нужно удерживать клавишу Shift. При нажатой клавише Alt контур можно перемещать как одно целое.

Флажок **Многоугольник** на панели свойств инструмента **Контур** позволяет работать с многоугольниками, состоящими только из прямых сегментов.

Новый контур появляется в окне **Контур** — рис. 8.33 (меню **Окна — Прикрепляющиеся диалоги — Контур**). Это окно аналогично окну **Слои**, с которым вы уже знакомы. Поскольку контуры в GIMP предназначены, главным образом, для выделения областей, основные операции — это преобразование контура в выделенную область (кнопка ) и наоборот (кнопка ). Когда область с растушёванной границей преобразуется в контур, растушёвка пропадает, граница становится резкой.

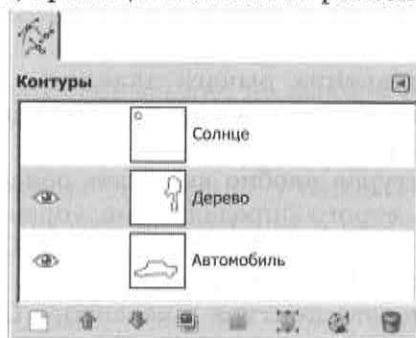


Рис. 8.33

Кроме того, можно расположить текст вдоль контура. В редакторе GIMP для этого надо выделить контур в окне **Контур**, добавить новый текст (текстовый слой) с помощью инструмента **Текст**  и щёлкнуть по кнопке **Текст по контуру** в окне параметров инструмента **Текст**. При этом строится новый контур по форме букв, размещённых вдоль контура. Затем можно залить эту область цветом (например, с помощью меню **Правка — Залить цветом переднего плана**), а текстовый слой удалить, он больше не

нужен. Конечно, редактировать такой текст уже нельзя, потому что он теперь хранится как растровая картинка.

Контуры, созданные в редакторе GIMP, можно сохранить на диске в виде файлов формата SVG (англ. *Scalable Vector Graphics* — масштабируемые векторные изображения). Для этого используют команду **Экспортировать контур** из контекстного меню **Контур**, которое можно вызвать щелчком правой кнопкой мыши в списке контуров или с помощью кнопки . Вместе с тем контуры, созданные в других программах (например, в Inkscape) и сохранённые в формате SVG, можно загружать в GIMP с помощью команды **Импортировать контур** того же меню.

Вопросы и задания

1. Что такое контур? Из каких элементов состоит контур?
2. Как регулируется угол наклона касательной в узлах контура?
3. Что такое гладкий узел, угловой узел?
4. В каких форматах контуры сохраняются вместе с основным изображением?
5. Как можно редактировать контур?
6. Как превратить контур в выделение, и наоборот?
7. Как вы думаете, почему при преобразовании выделенной области в контур пропадает растушёвка? Как сохранить область с растушёвкой для повторного использования? Приведите примеры.
8. Как расположить текст вдоль контура? Почему после этого текст нельзя редактировать?

Подготовьте сообщение

«Использование контуров в практических задачах»

Практические работы к главе 8

- Работа № 67 «Ввод и кадрирование изображений»
- Работа № 68 «Коррекция фотографий»
- Работа № 69 «Работа с областями»
- Работа № 70 «Быстрая маска. Фильтры»
- Работа № 71 «Многослойные изображения»
- Работа № 72 «Маска слоя»
- Работа № 73 «Каналы»
- Работа № 74 «Иллюстрации для веб-сайтов»
- Работа № 75 «GIF-анимация»
- Работа № 76 «Контур»

ЭОР к главе 8 на сайте ФЦИОР (<http://fcior.edu.ru>)

- Растровые редакторы
- Размещение графики на интернет-странице

Самое важное в главе 8

- Растровое изображение — это набор пикселей. Цвет пикселя задаётся в виде числового кода.
- Качество растрового изображения определяется разрешением и глубиной цвета. Чем выше разрешение и глубина цвета, тем лучше качество, но больше объем файла.
- Файл, в котором хранится растровое изображение, содержит не только коды пикселей, но и служебную информацию: размеры рисунка, цветовую палитру и др.
- Редактирование растрового изображения — это изменение кодов, задающих цвета пикселей, с помощью математических операций.
- Ретушь — это исправление дефектов изображения.
- Фильтр — это алгоритм автоматической обработки пикселей изображения, который применяется ко всему изображению или к выделенной области. Различают фильтры для коррекции изображений и художественные фильтры.
- В многослойных документах на каждом слое может строиться отдельное изображение, которое редактируется независимо от других. Каждый слой может разными способами накладываться на изображение, полученное с предыдущих слоёв.
- Канал — это чёрно-белое полутоновое изображение, которое показывает степень влияния какого-то эффекта на изображение. Например, степень прозрачности пикселей изображения записывается в так называемый альфа-канал.
- Анимация — это быстрая смена отдельных изображений, создающая у человека иллюзию движения. Анимация строится на основе многослойных изображений.
- Современные графические форматы позволяют хранить векторные и растровые элементы изображения в одном файле.

Глава 9 Трёхмерная графика

§ 67 Введение

Что такое трёхмерная графика?

Раньше, говоря о компьютерной графике, мы имели в виду двумерные («плоские») изображения. Невозможно «повернуть» автомобиль, изображённый на таком рисунке, и посмотреть на него с другой стороны. В то же время реальный автомобиль — это трёхмерный объект, поэтому при решении многих задач его «плоской» модели (рисунка, фотографии) недостаточно.

Трёхмерная графика (3D-графика, от англ. *3-Dimensions* — 3 измерения) — это раздел компьютерной графики, который занимается созданием моделей и изображений трёхмерных объектов.

В программах для работы с 3D-графикой строятся трёхмерные (пространственные) модели объектов, в которых каждая точка имеет три координаты (а не две, как на «плоском» рисунке). Затем пользователь может выбрать в пространстве точку наблюдения и получить плоское изображение, т. е. построить проекцию трёхмерной сцены на плоскость. Многие программы позволяют создавать анимацию, показывающую движение трёхмерных объектов в пространстве.

3D-модели применяются не только для построения двумерных изображений. Их используют для различных вычислений, например для расчёта прочности деталей. В последние годы активно разрабатываются 3D-принтеры, которые позволяют методом послойной печати построить объёмный физический объект (чаще всего из пластика) по его трёхмерной модели.

Перечислим важнейшие области применения трёхмерной графики:

- компьютерное проектирование машин и механизмов (САПР — системы автоматизированного проектирования¹);
- компьютерные тренажёры и обучающие программы;
- построение трёхмерных моделей в науке, промышленности, медицине;
- дизайн зданий и интерьера (внутренней обстановки);
- компьютерные эффекты в кино и телевидении; существуют даже полнометражные фильмы, которые полностью созданы с помощью трёхмерной графики и анимации;
- телевизионная реклама;
- интерактивные игры.

Создание изображений с помощью 3D-графики включает несколько этапов:

- 1) **моделирование** — создание трёхмерных объектов, персонажей;
- 2) **текстурирование** (раскраска) — наложение на модели рисунков (текстур), которые имитируют реальный материал (дерево, мрамор, металл, кожу и пр.);
- 3) **освещение** — установка и настройка источников света;
- 4) **анимация** — описание изменения объектов во времени (изменение положения, углов поворота, свойств);
- 5) **съёмка** — установка камер (выбор точек съёмки), перемещение камер по сцене;
- 6) **рендеринг** (визуализация) — построение фотореалистичного изображения или анимации.

Среди профессиональных программ 3D-моделирования наиболее популярны продукты фирмы Autodesk (www.autodesk.com):  3ds MAX,  Maya и  AutoCAD, а также программа  Cinema4D фирмы MAXON (www.maxon.net). Мы будем использовать для иллюстраций свободно распространяемый пакет  Blender (www.blender.org), версии которого существуют для операционных систем Windows, Linux, macOS и других.

Для работы с программами трёхмерной графики нужен компьютер с мощным процессором и большим объёмом оперативной и дисковой памяти. Построение качественных фотореалистичных изображений (которые выглядят как фотографии) занимает огромное время, иногда несколько часов расчётов на один кадр.

¹ В английском языке — CAD-системы (CAD — Computer Aided Design, проектирование с помощью компьютера).

Во многих программах есть возможность **сетевого рендеринга**, когда для расчёта изображения используются мощности нескольких компьютеров, объединённых в сеть (англ. *render farm* — рендер-фермы).

Проекции

Хотя программы трёхмерной графики предназначены для создания трёхмерных моделей объектов, пользователь видит только плоское (двухмерное) изображение на мониторе или бумажном отпечатке, т. е. **проекцию**. На рисунке 9.1 показаны четыре проекции модели головы обезьянки Сюзанны (объект **Monkey**), которая включена в набор стандартных объектов программы Blender. Вы видите три стандартные проекции этой модели (виды спереди, сверху и справа) и одну произвольную проекцию (проекцию пользователя).



Рис. 9.1

Программа Blender позволяет видеть четыре проекции одновременно или оставить только одну проекцию пользователя, которая занимает всю рабочую область. Для переключения между этими режимами используется комбинация клавиш Ctrl+Alt+Q.

Обычно работают с одним видом, который занимает всю рабочую часть окна. Для быстрого перехода к стандартным проекциям (видам спереди, сверху, справа и др.) используется меню **Вид (View)** или дополнительная цифровая клавиатура (англ. *numpad*), расположенная в правой части стандартной клавиатуры (рис. 9.2).



Рис. 9.2

Далее для обозначения этих клавиш будем использовать «приставку» *Num*, например *Num1* обозначает клавишу «1» на дополнительной цифровой клавиатуре.

Существует два типа проекций: **перспективные** и **ортогональные** (их также называют **прямоугольные** или **ортографические**). На рисунке 9.3 показаны перспективная и ортогональная проекции куба.

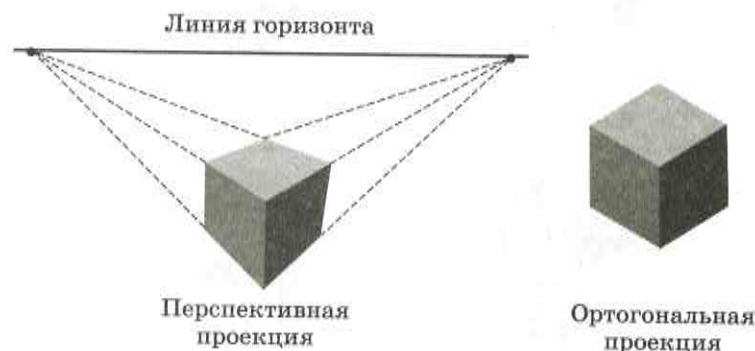


Рис. 9.3

Наш взгляд привык к **перспективе**: удалённые предметы кажутся меньше по размеру, параллельные линии «сходятся» в бесконечной точке (вспомните, как выглядит уходящее вдаль шоссе). Однако при трёхмерном моделировании такие проекции не совсем удобны, потому что искажают форму и размеры объектов.

Для **ортогональной проекции** всё по-другому: размеры не зависят от расстояния до предмета, а параллельные грани остаются параллельными и на проекции. Ортогональные проекции очень полезны, потому что делают трёхмерную сцену проще и позволяют оценить истинные размеры объектов.

В редакторе Blender тип проекции показывается в левом верхнем углу рабочего окна. Например, надпись **Top Ortho** означает «вид сверху» (англ. *top view*), ортогональная проекция (англ. *orthographic*). Надпись **Front Persp** означает «вид спереди» (англ. *front view*), перспективная проекция (англ. *perspective*).

Чтобы переключиться с ортогональной проекции на перспективную или наоборот, нужно нажать кнопку «5» на дополнительной клавиатуре (*Num5*).

Вращая колёсико мыши, пользователь может уменьшать и увеличивать масштаб изображения в окне, над которым находится курсор мыши (размеры самого объекта при этом не меняются). Для вращения произвольной проекции нужно перемещать мышь при нажатой средней кнопке (колёсике). Нажав одновременно колёсико мыши и клавишу Shift, можно перемещать изображение в окне, не поворачивая его.

Для вращения и перемещения удобно использовать клавиши-стрелки на дополнительной цифровой клавиатуре (*Num2*, *Num4*, *Num6* и *Num8*): в обычном режиме они вращают сцену, а при нажатой клавише Ctrl перемещают точку наблюдения.

Вопросы и задания

1. Как строится двумерное изображение трёхмерной модели?
2. В каких задачах необходимо использование трёхмерных моделей?
3. Как вы думаете, в каком виде хранится в памяти информация о трёхмерных объектах?
4. На каких этапах создания изображений в программах 3D-моделирования используется векторная и растровая графика?
5. Объясните, что такое моделирование, текстурирование, рендеринг.
6. Вспомните, что такое свободное программное обеспечение. В чём его достоинства и недостатки?
7. Что такое кроссплатформенное программное обеспечение? Относится ли программа Blender к этому типу программ?
8. Объясните, почему для работы с программами трёхмерной графики требуются мощные компьютеры.
9. Что такое проекции? Зачем они нужны?
10. Чем отличаются перспективные и ортогональные проекции? Когда их удобно использовать?

Подготовьте сообщение

«Программы для 3D-моделирования»



Задача

Загрузите в программу любую трёхмерную модель и научитесь ориентироваться в трёхмерном пространстве: переходить от одной проекции к другой, менять точку наблюдения, наблюдать разные стороны объектов.

§ 68

Работа с объектами

Примитивы

Построение трёхмерных моделей обычно начинается с **примитивов** — простейших объектов. К ним относятся плоскость (точнее, её прямоугольная часть), куб, сфера, цилиндр, конус, тор и некоторые другие (рис. 9.4).



Рис. 9.4

При создании объекта ему автоматически присваивается имя. Например, первый куб будет называться *Cube*, второй — *Cube.001* и т. д. Это имя можно изменить на панели свойств объекта, которая расположена в правой части окна программы Blender (рис. 9.5).

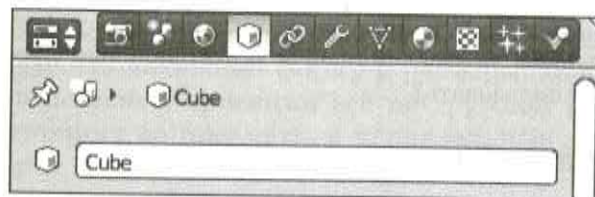


Рис. 9.5

Выделение объектов

Для того чтобы работать с объектом, например изменять его свойства, предварительно нужно выделить его. В программе Blender для этого используется *правая* кнопка мыши (а не левая, как в других программах).

При нажатой клавише Shift можно выделить правой кнопкой мыши несколько объектов одновременно. Если повторно щёлкнуть на объекте, выделение снимается.

С помощью клавиши *A* (от англ. *all* — всё) снимается выделение со всех объектов, а если ни один объект не был выделен, все они выделяются.

После нажатия клавиши *B* (от англ. *border select* — выделить рамкой) можно с помощью мыши обвести прямоугольной рамкой все объекты, которые нужно выделить. Если объект хотя бы частично попал в рамку, он будет выделен. Кроме того, можно нажать левую кнопку мыши и, удерживая нажатой клавишу Ctrl, «обвести» все нужные объекты произвольным контуром.

Для мелких объектов удобно использовать **круговое выделение**. Этот инструмент включается при нажатии клавиши *C*: появляется окружность, размер которой регулируется колёсиком мыши. Затем щелчком (или «протаскиванием») левой кнопкой мыши выделяются все элементы, попадающие внутрь окружности. Если в этом режиме случайно выделен лишний объект, выделение можно снять, нажав на колёсико мыши.

Существуют и другие, более сложные способы выделения объектов, которые доступны через меню **Select**.

С помощью клавиши *Num.* (точка на цифровой клавиатуре) можно приблизить выделенный объект, а клавиша *Num/* временно скрывает остальные объекты (кроме выделенных).

Преобразования объектов

Любой объект можно перемещать, вращать и масштабировать (изменять размеры). Эти операции называются **преобразованиями** объектов или трансформациями (англ. *transformations*).

В опорной точке (англ. *origin* — начало координат) выделенного объекта появляется так называемый манипулятор с тремя стрелками, параллельными осям координат (рис. 9.6). Ось *Z* (синяя стрелка в Blender) на-

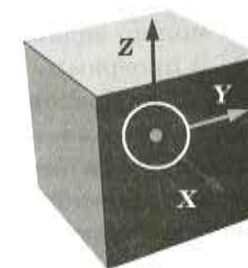


Рис. 9.6

правлена вверх, а оси X (красная стрелка в Blender) и Y (зелёная стрелка в Blender) находятся в горизонтальной плоскости.

За эти стрелки объект можно перетаскивать левой кнопкой мыши, меняя его положение только по одной выбранной оси. Центральный круг дает возможность произвольно перемещать объект в плоскости проекции.

Объекты можно вращать как в плоскости проекции (нажав клавишу R , от англ. *rotate* — вращать), так и вокруг одной выбранной оси (для этого после нажатия клавиши R нужно нажать клавишу с названием оси — X , Y или Z). Можно использовать специальный манипулятор вращения, позволяющий поворачивать объект в нужное положение за выделенные части окружностей (рис. 9.7)



Рис. 9.7

Чтобы изменить размеры объекта, нужно нажать клавишу S (от англ. *scale* — изменить масштаб) и перемещать мышью. Чтобы изменять размер только по одной из осей, надо после нажатия клавиши S нажать клавишу с названием оси. Манипулятор изменения размеров содержит небольшие кубики на концах указателей осей (см. рис. 9.7). Перетаскивая один из них, можно менять соответствующий размер.

Координаты, углы поворота и размеры объектов можно задавать с клавиатуры в числовой форме. В программе Blender для этого используют панель **Преобразование** (Transform) — рис. 9.8, которая появляется при нажатии клавиши N . Числовые свойства можно также изменять с помощью мыши (перетаскивая влево и вправо стрелки рядом со значениями).

Копия выделенного объекта создается с помощью клавиш $\text{Shift}+D$. Затем объект-копию пе-

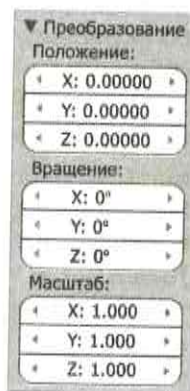


Рис. 9.8

ремещают мышью в нужное место и фиксируют щелчком левой кнопки. Если предварительно нажать клавишу X , Y или Z , объект-копия будет перемещаться только вдоль указанной оси.

Удалить выделенный объект со сцены можно с помощью клавиши **Delete**.

Системы координат

По умолчанию используется **глобальная** («мировая») **система координат**, начало которой находится в точке с координатами $(0,0,0)$ пространства сцены, она не зависит от положения объекта. Иногда бывает удобно перейти к **локальной системе координат**, которая связана с объектом. Её центр находится в опорной точке объекта, а оси меняют направление при его вращении, показывая, где у объекта «верх» (локальная ось $Z_{\text{лок}}$), «право» (локальная ось $X_{\text{лок}}$) и т. д. На рисунке 9.9 показаны направления осей глобальной и локальной систем координат для повернутого конуса.

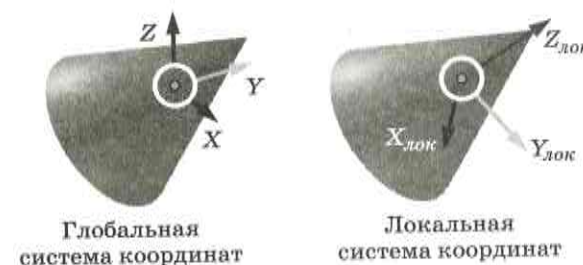


Рис. 9.9

В программе Blender можно выбрать нужную в данный момент систему координат с помощью выпадающего списка в нижней части рабочего окна. Именно эти оси будут использоваться при перемещении, вращении и изменении размеров объекта.

Слой

Трёхмерная сцена в Blender может состоять из нескольких слоёв. В рабочем окне видны только те объекты, которые расположены на активном слое. Активный слой можно выбрать щелчком мышью на специальной панели в нижней части рабочего окна (рис. 9.10).



Рис. 9.10

Всего можно использовать 20 слоёв, в данном случае (см. рис. 9.10) какие-то объекты есть на слоях 1, 2 и 4. Активный слой — слой 2, его клетка выделена тёмно-серым фоном. Щёлкая мышью на клетках панели при нажатой клавише Shift, можно сделать активными несколько слоёв, чтобы показать все связанные с ними объекты.

Объект может принадлежать не только одному, но и нескольким слоям. Для этого их нужно выделить в аналогичном элементе управления на панели свойств объекта. Здесь же можно перевести объект с одного слоя на другой. Для этих операций можно также использовать всплывающее окно, которое появляется при нажатии клавиши *M*.

Связывание объектов

Представим себе, что мы построили трёхмерную модель стола, на котором размещены тарелки, чашки и столовые приборы. Теперь нужно передвинуть стол в другое место сцены и немного развернуть его. При этом точно так же должны переместиться и все объекты, стоящие на столе. Чтобы проще решить эту задачу, нужно «привязать» к столу все стоящие на нем предметы, т. е. на панели свойств установить для них так называемый **родительский объект** (англ. *parent* — родитель). Можно поступить иначе: выделить, удерживая клавишу Shift, все нужные объекты (тарелки, чашки и т. д.), последним выделить родительский объект (стол) и нажать клавиши *Ctrl+P*.

После этого при всех преобразованиях объекта-родителя (перемещении, вращении, масштабировании) те же самые преобразования применяются и к объекту-потомку. В то же время объект-потомок по-прежнему можно перемещать независимо от родителя, т. е. мы можем свободно двигать чашку по столу.

Обратите внимание на отличие группы в Blender от аналогичных средств в большинстве программ, работающих с графикой: объекты в такой связанной группе неравноправны — среди них есть один родительский объект и несколько потомков.

Всю структуру сцены (иерархию объектов) можно посмотреть в специальном окне **Структура проекта (Outliner)** — рис. 9.11. Здесь видно, например, что объект *Cube* — родительский для объекта *Cube.001*.

Справа от имени объекта в окне **Структура проекта** показаны три значка: щелчок мышью на изображении глаза скрывает объект; если щёлкнуть на значке , то объект будет нельзя выделить (иногда это не нужно), а щелчком на значке  можно запретить показ объекта при рендеринге.



Рис. 9.11

Вопросы и задания

1. Что такое примитивы? Зачем они нужны?
2. Как вы думаете, зачем каждому объекту сцены присваивается уникальное имя?
3. Как выделить одновременно несколько объектов?
4. Какие преобразования объектов вы знаете?
5. Как можно применить преобразования только по одной оси?
6. Что такое манипуляторы? Как их использовать?
7. Какие системы координат применяются при трёхмерном моделировании? Чем они различаются и когда используются?
8. Зачем нужны слои?
9. В каких случаях удобно использовать связь объектов «родитель — потомок»?

Задача

В программе трёхмерного моделирования научитесь создавать различные типы примитивов и применять к ним преобразования.

§ 69

Сеточные модели

Как строятся объекты?

По умолчанию объекты в Blender изображаются как объёмные твердые тела (англ. *solid* — сплошной). При этом не сразу понятно, как же программа может быстро перестраивать изображение

при изменении точки наблюдения. Для того чтобы понять внутреннее устройство объектов, нужно с помощью списка  переключиться в другой режим, который называется **Каркас** (англ. *wireframe*). Мы увидим, что каркас куба включает (рис. 9.12):

- 8 вершин: *A, B, C, D, E, F, G* и *H*;
- 12 рёбер, соединяющих вершины: *AB, AD, BC, CD, EF, EH, FG, GH, AE, BF, CG* и *DH*.

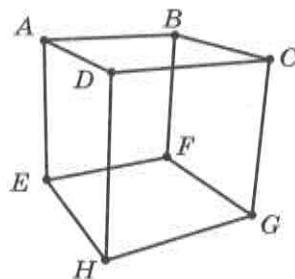


Рис. 9.12

Вершина (англ. *vertex*) — это точка в трёхмерном пространстве, которая задается тремя координатами. **Ребро** (англ. *edge*) — это отрезок, соединяющий две вершины. Рёбра ограничивают **грани** (англ. *face*) — участки поверхностей. У куба 6 граней: *ABCD, EFGH, ABFE, CDHG, ADHE* и *BCGF*.

Такие модели называются **сеточными** (англ. *mesh*), потому что они представляют собой поверхности, которые строятся на сетке из рёбер. Чаще всего используются треугольные и четырёхугольные грани, однако последние версии Blender могут работать с гранями, состоящими из большего числа рёбер, — так называемыми *N*-угольниками или **полигонами**. Поэтому часто применяют выражение «*полигональная модель*» (от англ. *polygon* — многоугольник, полигон). Сфера и другие криволинейные поверхности тоже строятся из плоских граней, но их значительно больше, чем у куба.

Трёхмерные модели хранятся в векторном формате, для построения поверхности достаточно запомнить координаты вершин каркаса. По ним можно рассчитать координаты всех точек рёбер и граней. Каждая грань обрабатывается отдельно, поэтому чем больше граней, тем большее время требуется для расчётов.

Редактирование сетки

Для изменения положения элементов сетки нужно перейти в режим редактирования (англ. *Edit mode*). В программе Blender для этого используется клавиша *Tab*.

В режиме редактирования можно работать с вершинами, рёбрами и гранями. Нужный тип объектов выбирается с помощью показанного на рис. 9.13 элемента управления, который расположен в нижней части рабочего окна, или с помощью всплывающего меню, которое вызывается нажатием клавиш *Ctrl+Tab*. На рисунке 9.13 первая кнопка выделена тёмным фоном, это означает, что включён режим работы с вершинами.



Рис. 9.13

При нажатой клавише *Shift* можно включить несколько режимов выделения сразу, например выделять рёбра и грани.

Для выделения элементов сетки используются те же методы, что и при выделении объектов. Затем их можно перемещать, вращать и масштабировать. Очень удобно использовать *круговое выделение* (см. § 67).

По умолчанию при работе с рёбрами и гранями выделяются не только видимые, но и невидимые элементы, расположенные на задней поверхности объекта. Чтобы этого не происходило, нужно ограничить выделение только видимыми элементами, щёлкнув на кнопке  в нижней части рабочего окна.

В режиме работы с вершинами щелчок левой кнопкой мыши при нажатой клавише *Ctrl* создаёт новую вершину, которая соединяется с уже выделенной вершиной. Если выделить две вершины (используя клавишу *Shift*) и нажать клавишу *F*, между ними строится новое ребро.

Чтобы создать новую грань, нужно выделить все вершины замкнутого многоугольника и нажать клавишу *F*.

Для доступа к другим операциям с элементами сеточной модели можно использовать всплывающие меню, которые появляются при нажатии клавиш *Ctrl+V* (меню для вершин), *Ctrl+E* (меню для рёбер) и *Ctrl+F* (меню для граней).

В Blender существует особый режим **пропорционального редактирования** (англ. *proportional editing*). В этом режиме переме-

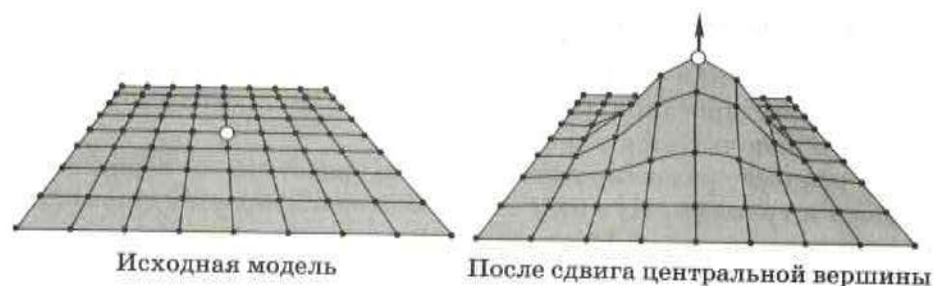


Рис. 9.14

щаемая вершина, ребро или грань увлекает за собой соседние. В примере, показанном на рис. 9.14, перемещалась вверх только центральная вершина сетки.

Деление рёбер и граней

Часто требуется разделить ребро или грань на несколько частей. Для этой цели проще всего использовать инструмент **Подразделить** (Subdivide), который делит выделенные рёбра или грани на несколько равных частей. Примеры его использования показаны на рис. 9.15.

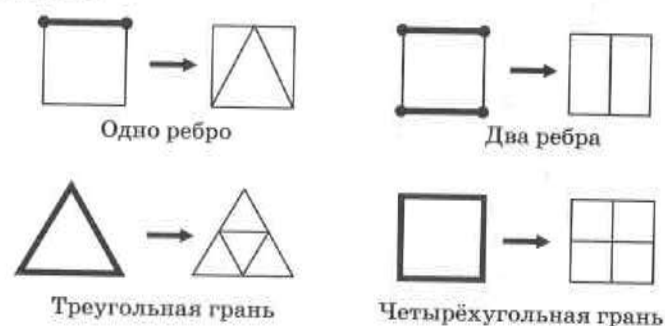


Рис. 9.15

Существует также инструмент **Нож** (Knife), который позволяет «разрезать» выделенные рёбра. Для этого нужно при нажатой клавише **K** нажать и не отпускать левую кнопку мыши, после чего курсор становится похожим на нож. Теперь остаётся провести мышью через точки деления рёбер. Если нажимать **Shift+K** вместо **K**, рёбра, через которые проходит нож, делятся ровно пополам.

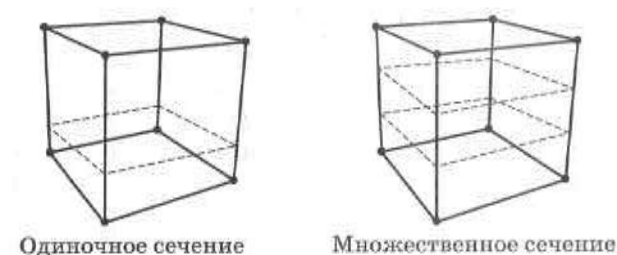


Рис. 9.16

Инструмент **Разрезать петлёй со сдвигом** (Loop Cut and Slide), который вызывается нажатием клавиш **Ctrl+R**, позволяет расцезать грани по контуру вокруг объекта и сдвинуть сечение в нужное место (рис. 9.16). Колёсиком мыши можно увеличивать число сечений.

Выдавливание

Один из самых полезных инструментов при работе с сеточными моделями — **Выдавить участок** (Extrude). Выдавливание состоит в том, что выбранная грань перемещается вдоль нормали (перпендикуляра к этой грани) и вокруг неё создаются новые грани. На рисунке 9.17 показаны два типа выдавливания центральной верхней грани куба.



Рис. 9.17

Для того чтобы выполнить выдавливание, нужно выделить грань, нажать клавишу **E** и мышью переместить грань в нужное положение.

Выдавливание можно применять также к выделенным рёбрам и вершинам. Чтобы они перемещались только по одной оси, нужно после клавиши **E** нажать клавишу с названием этой оси (**X**, **Y** или **Z**).

Сглаживание

Вы уже знаете, что модель любой поверхности в программах трёхмерного моделирования строится из отдельных плоских граней. Однако во многих случаях реальный объект моделирования — гладкий, и на картинке нужно получить сглаженную поверхность (рис. 9.18).



Рис. 9.18

Для сглаживания стыков между гранями используют инструмент **Сгладить** (Smooth), который может применяться как ко всему объекту, так и к отдельным граням. При этом важно, что геометрия объекта (количество граней, способ разбивки на грани) не изменяется. Обратите внимание на контур объекта — он остался угловатым. Программа выполняет сглаживание («скругление») граней только при выводе изображения на экран, используя специальные алгоритмы затенения.

Есть и другой способ сглаживания — дополнительная разбивка на более мелкие грани, мы рассмотрим его в следующем параграфе.



Вопросы и задания

1. Что такое сеточная модель? Из каких элементов она состоит?
2. Подумайте, какими достоинствами и недостатками обладают сеточные модели.
3. В каком формате (растровом или векторном) хранится информация о сеточной модели?
4. Как связано количество граней модели и время расчёта изображения сцены? Какие рекомендации вы можете дать в связи с этим?
5. Расскажите о приёмах, которые можно использовать для редактирования сетки.
6. Что такое выдавливание?
7. Зачем нужно сглаживание?



Задачи

1. Исследуйте сеточную модель куба и научитесь изменять её.
2. Исследуйте сеточные модели двух типов сфер, которые в Blender называются **UV-сфера** (UV-sphere) и **Икосаэдр** (Icosphere). Чем они различаются?

§ 70
Модификаторы

Что это такое?

Модификатор — это преобразование объекта, которое выполняется автоматически при выводе проекции на экран или построении готового изображения (*рендеринге*). При этом геометрия объекта не меняется, т. е. это неразрушающая операция (действие модификатора всегда можно отменить). Как правило, модификатор имеет настройки, которые можно менять в диалоговом режиме.

Для любого объекта можно использовать несколько модификаторов. Они действуют последовательно, т. е. первый модификатор (верхний в списке) «работает» с исходной сеточной моделью, второй — с результатом работы первого и т. д. Все применённые модификаторы образуют **стек модификаторов**. В программе Blender вершина стека (последний применяемый модификатор) находится в конце списка. Порядок применения модификаторов можно изменять.

Каждый раз, когда пользователь изменяет вид сцены (например, поворачивая или перемещая объекты), для построения проекции модификаторы применяются заново к изменённой сеточной модели, а это требует значительного времени и ресурсов компьютера. Однако исходная сеточная модель остаётся довольно простой, и её легко редактировать.

Когда модель полностью готова, можно раз и навсегда применить модификатор, т. е. внести соответствующие изменения в сеточную модель. Нужно учитывать, что во многих случаях (например, при сглаживании) новая модель будет содержать значительно большее число граней, занимать больше места на диске и требовать больше времени для рендеринга. Кроме того, редактировать её будет намного сложнее.

Далее мы познакомимся с некоторыми часто используемыми модификаторами программы Blender. Про остальные вы можете узнать в справочной системе.

Сглаживание

В природе редко встречаются чёткие и ровные углы. Поэтому для сеточных моделей животных, растений, людей, сказочных персонажей обязательно используют сглаживание.

В предыдущем параграфе уже рассматривался один вариант сглаживания — с помощью инструмента **Сгладить** (Smooth). Подобную операцию можно выполнить с помощью модификатора **Подразделение поверхности** (Subsurf, англ. *subdivision surface* — разбиение поверхностей).

У модификатора **Подразделение поверхности** есть настройки, с помощью которых можно регулировать степень сглаживания (рис. 9.19).



Рис. 9.19

Симметрия

При моделировании симметричных объектов, например мордочки животного, хочется автоматически поддерживать одинаковую форму левой и правой частей. Для этого удобно применять модификатор **Отражение** (Mirror).

Основной объект в примере на рис. 9.20 — это правая половина модели головы обезьянки Сюанны, которая включена в Blender как тестовый объект¹. При использовании модификатора **Отражение** все изменения, которые выполняются с правой частью, автоматически применяются и к левой.

Если применить модификатор, щёлкнув на кнопке **Применить** (Apply), будет построена новая симметричная сеточная модель,

¹ В других программах в качестве тестового объекта используют чайник.



Рис. 9.20

в которой левая и правая части независимы друг от друга (т. е. при изменении одной половины вторая уже не будет изменяться).

Логические операции

С помощью модификатора **Логический** (англ. *boolean*) можно строить объединение, пересечение и «разность» двух объектов. На рисунке 9.21 показаны четыре возможные «логические» операции, которые можно применить к кубу и сфере.

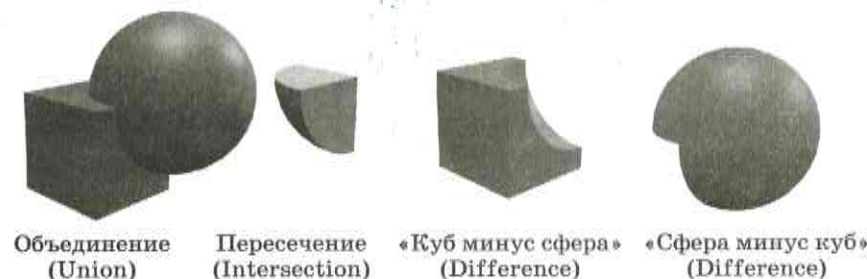


Рис. 9.21

Легко заметить, что объединение и пересечение можно сопоставить логическим операциям «ИЛИ» и «И», которые вы изучали в 10 классе.

Модификатор применяется к одному объекту, а второй указывается в параметрах модификатора в поле **Объект** (Object). На рисунке 9.22 показаны настройки модификатора **Логический** для случая, когда нужно построить «разность» сферы (объект с именем **Sphere**) и куба (объект **Cube**). Список **Операция** (Operation) позволяет выбрать нужную операцию (здесь — операция **Разность**

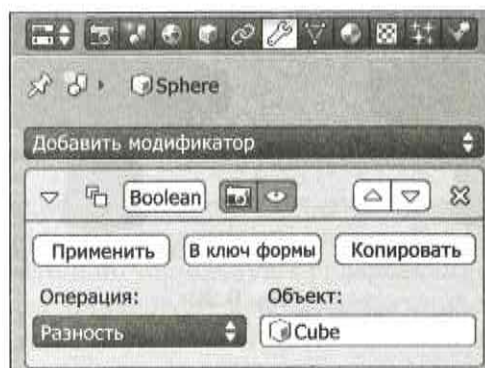


Рис. 9.22

(Difference)). При этом куб никак не меняется, а вместо сферы, к которой применён модификатор, появляется объект «сфера минус куб», т. е. часть шара, которая не входит в куб.

Если после этого сдвинуть куб, полученный объект-«разность» изменится, потому что он каждый раз строится заново с учётом текущего положения сферы и куба. Чтобы новый объект стал независимым, нужно применить модификатор с помощью кнопки **Применить** (Apply). Теперь изменения куба не будут на него влиять.

Сетка объекта, полученного в результате логической операции, значительно усложняется в местах стыковки исходных тел, поэтому потом её довольно сложно редактировать.

Массив

Модификатор **Массив** (Array) позволяет создать несколько копий (**клонов**) основного объекта, которые смещены друг относительно друга на одинаковое расстояние по осям X, Y и Z. Например, так можно из одной модели солдата построить модель целого взвода, стоящего в колонну или шеренгу. Если в том же случае повторно использовать модификатор **Массив** и задать смещение по другой оси, мы сможем построить солдат в несколько колонн.

Копии (клоны) сохраняют связь с основным объектом, при любом его изменении клоны также меняются. Если применить модификатор **Массив** (с помощью кнопки **Применить**), связь копий с исходным объектом разрывается, после этого их можно редактировать независимо друг от друга.

Деформация

Для изменения формы объектов часто удобен модификатор **Решётка** (Lattice). Исходный объект (на рис. 9.23 — сфера) помещается внутрь специальной «клетки» — объекта **Решётка**. Объект **Решётка** — это вспомогательная сетка, которая не показывается при рендеринге, т. е. не влияет на итоговую картинку.

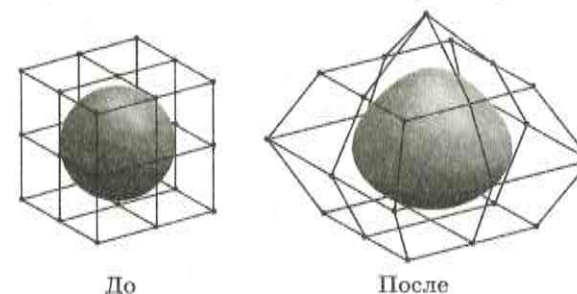


Рис. 9.23

Затем к исходному объекту применяется модификатор **Решётка**. В результате объект и «клетка» оказываются связанными, так что при изменении формы клетки (например, при перемещении её вершин и ребер) меняется и форма основного объекта.

С помощью решётки удобно менять форму объекта при анимации, не меняя его сеточную модель. Например, мяч при отскоке от пола немного сплющивается, а потом опять принимает нормальную форму.

Вопросы и задания

1. Что такое модификатор?
2. Почему модификаторы — это неразрушающий метод редактирования?
3. Что означает «применить модификатор»? Какие достоинства и недостатки имеет этот приём?
4. Что такое стек модификаторов? Как влияет расположение модификаторов в стеке на окончательный результат?
5. Опишите действия модификаторов, рассмотренных в тексте параграфа.

Задачи

1. Научитесь использовать модификаторы, рассмотренные в тексте параграфа.
- *2. Найдите документацию по другим модификаторам и научитесь их применять.

§ 71 Кривые

Основные понятия

Кривые используются в программах трёхмерной графики как вспомогательные векторные объекты, например для того, чтобы задать форму тела вращения или поперечное сечение трубы. С помощью кривых определяют траектории движения объектов при анимации, искривляют текстовые строки, моделируют провода и нитки. Замкнутую кривую называют **контуром**.

В программе Blender можно строить два типа кривых — **кривые Безье** и **кривые NURBS** (англ. *Non Uniform Rational B-Splines* — неравномерные рациональные В-сплайны). В этом параграфе мы рассмотрим только кривые Безье как самые простые. Как вы знаете, они состоят из **узлов** (опорных точек) и **сегментов** (соединяющих их линий). Кривизну сегментов определяют направляющие линии (касательные) в каждом узле, положение которых можно регулировать с помощью «рычагов», их концы обозначены на рис. 9.24 белыми кружками.

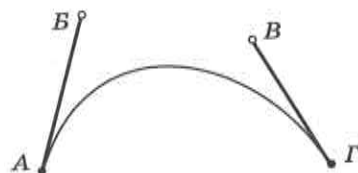


Рис. 9.24

Для построения каждого сегмента используются только 4 точки, показанные на рис. 9.24: два узла, определяющие сегмент (A и G), и концы рычагов (B и B).

Кривая на этом участке выходит из точки A в направлении точки B, затем изгибается к точке B и, наконец, входит по касательной в точку G.

В программе Blender различаются четыре типа узлов (рис. 9.25):

- **векторные узлы** (англ. *vector* — векторный), предназначенные для рисования ломаных (касательные направлены в соседнюю точку);
- **гладкие узлы** (англ. *aligned* — выровненный), в которых обе направляющие лежат на одной прямой;

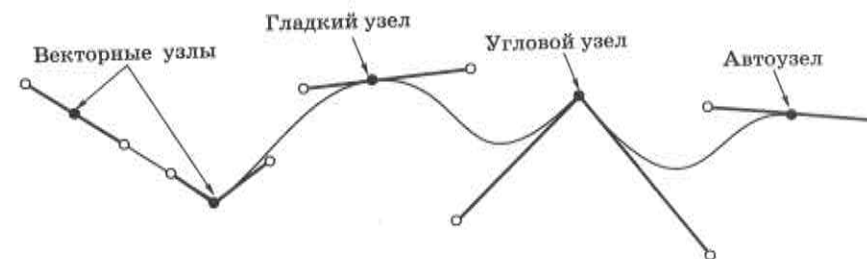


Рис. 9.25

- **угловые, или свободные узлы** (англ. *free* — свободный), в которых каждую направляющую можно настраивать независимо от другой;
- **автоузлы** (англ. *auto* — автоматический) — гладкие узлы, в которых положение направляющих выбирается программой.

При создании **кривой Безье** в Blender (меню **Добавить — Кривая Безье** (Add — Curve — Bezier)) строится стандартная кривая с двумя гладкими узлами, находящаяся в плоскости XY. Кривую в целом можно редактировать как обычный объект (перемещать, вращать, изменять размеры).

Для настройки отдельных узлов нужно перейти в режим редактирования (клавиша Tab), в котором можно перемещать сами узлы и их управляющие рычаги. Создать новый узел между двумя выделенными узлами можно с помощью всплывающего меню **Специальные — Подразделить** (Specials — Subdivide), которое появляется при нажатии клавиши W. Меню для выбора типа узла вызывается клавишей V.

Кривые часто используют как вспомогательные объекты для построения более сложных фигур, поэтому при рендеринге их не видно. Однако когда мы моделируем с помощью кривой например, кабель, трубу, нитку и т. п., она должна присутствовать на готовом изображении. Для этого нужно изменить её свойства особым образом:

- сделать кривую трёхмерной (включить режим 3D);
- в списке **Заполнение** (Fill) выбрать вариант **Полностью** (Full); другие варианты позволяют построить половину или четверть трубы;
- применить **Скос** (Bevel).

После этого будет построена трубка, её диаметр определяется глубиной скоса (параметр **Глубина** (Depth)), а гладкость поверхности — параметром **Разрешение** (Resolution). Если нужно увеличить толщину стенок, к такой трубке можно применить модификатор **Объёмность** (Solidify).

Пластины

При нажатии клавиш **Alt+C** кривая замыкается (или размыкается, если она была замкнута). Контур (замкнутую кривую) можно превратить в пластинку, которая будет показана при рендеринге картинки. Для этого в свойствах объекта нужно установить режим **2D** (англ. *2-dimensions*, двумерная, или плоская кривая) и ненулевой параметр **Выдавить** (Extrude), который определяет толщину пластинки. Параметр **Скос** задаёт фаску, которая применяется ко всем граням (рис. 9.26).



Рис. 9.26

Эти операции фактически представляют собой модификаторы, применяемые к контуру. Это значит, что контур по-прежнему можно свободно редактировать. Чтобы сделать из такого объекта сеточную модель, нужно нажать клавиши **Alt+C** и выбрать во всплывающем меню команду **Полисетка из кривой** (Mesh from Curve). Если теперь выделить объект и перейти в режим редактирования узлов, мы увидим «обычную» сетку из узлов, рёбер и граней.

Профили

Часто нужно смоделировать кабель, трубу или брусок, имеющий заданный профиль сечения. В программах трёхмерной графики для этого обычно используются две кривые: одна задаёт профиль сечения, а вторая — путь. На рисунке 9.27 показаны кривые, с помощью которых построена модель балки с сечением в форме тавра.

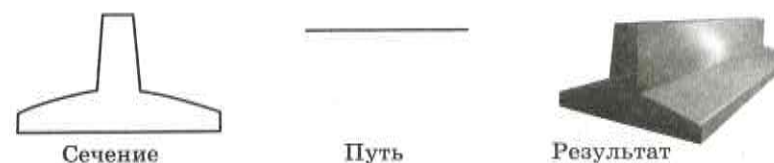


Рис. 9.27

Для того чтобы связать два контура в программе Blender, нужно выделить контур-путь и перейти на страницу данных объекта (Object Data). Затем в поле **Форма скоса** (Bevel object) из списка существующих контуров выбирается имя контура, задающего профиль. В некоторых программах, например в 3ds Max, такой приём называется «лофтинг» (англ. *lofting*).

Объект, который мы видим, строится только при выводе на экран. В памяти он хранится как две кривые, а не как сеточная модель. Поэтому можно как угодно изменять форму пути и профиля, но нельзя работать с отдельными вершинами, рёбрами и гранями.

К такому объекту можно применять модификаторы, например **Подразделить** (Subsurf) для получения гладкой поверхности. Можно также преобразовать его в сеточную модель с помощью клавиш **Alt+C**.

Тела вращения

В жизни нас окружает множество объектов, которые могут быть построены как тела вращения: тарелки, стаканы, бокалы, вазы и т. п. Для их моделирования также можно использовать профили, но в данном случае контур, определяющий путь, — это окружность. На рисунке 9.28 показано, как построить трёхмерную модель тарелки.

При создании профиля на окружности оказывается опорная точка кривой, определяющей сечение (начало локальных координат).

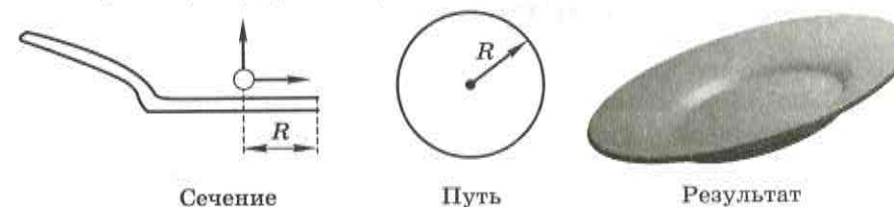


Рис. 9.28

нат), поэтому эту точку нужно размещать на расстоянии радиуса окружности от края кривой (см. рис. 9.28). Для перемещения опорной точки можно щелчком мышью установить курсор в нужное место и выбрать в меню под проекцией команду **Объект – Преобразования – Опорную точку к 3D курсору** (Object – Transform – Origin to 3D Cursor).

Отметим, что в программе Blender существуют и другие способы построения тел вращения, например инструмент **Spin** (англ. *spin* — вращение), применяемый к сеточным моделям.



Вопросы и задания

1. Зачем используются кривые в программах трёхмерного моделирования?
2. Из каких элементов состоит кривая Безье? Как изменять форму такой кривой?
3. Какие типы узлов используются при построении кривых Безье? Как изменить тип узла?
4. Как сделать видимую нить или трубу с помощью кривых?
5. Как создать пластину?
6. Как используются кривые для моделирования объектов с заданным профилем сечения?
7. Что нужно сделать, чтобы можно было изменять положение отдельных вершин такой модели?
8. Как смоделировать тело вращения?



Задачи

1. Постройте трёхмерную модель логотипа программы Blender, используя описанный метод создания пластин.
2. Постройте трёхмерную модель балки с сечением в форме двутавра (рисунок справа).
3. Постройте трёхмерную модель чашки или вазы.



§ 72

Материалы и текстуры

В природе нет объектов, которые были бы абсолютно гладкими и ровно покрашены в один серый цвет. Сцена не будет смотреться реалистично, пока мы не применим **материалы**, т. е. не зададим какие-то свойства, по которым можно отличить дерево, металл, мрамор, кирпич, песок. Без использования материалов невозможно сделать тела блестящими, прозрачными, светящимися и т. п.

Отражение света

Для того, чтобы разобраться со свойствами материалов, нужно понять, как мы видим окружающие предметы. Какой-то источник света (солнце, лампочка и т. п.) излучает световые волны, которые попадают на объекты. При этом часть волн поглощается материалом, а остальные отражаются от поверхности. Глаз воспринимает попадающие в него отражённые световые волны, свойства которых определяют видимый цвет предмета. Например, если источник излучает «белый» свет (включающий волны всех частот видимого светового диапазона), а мы видим зелёную поверхность, это означает, что материал поглощает все волны, кроме тех, которые соответствуют зелёному цвету.

Из курса физики вам известен закон отражения света, согласно которому угол отражения равен углу падения. Такое отражение называется **зеркальным** (англ. *specular*), при этом предполагается, что поверхность идеально ровная (рис. 9.29). Однако на самом деле любой материал имеет шероховатости, различающиеся по размеру. Поэтому лучи света отражаются от большинства предметов во всех направлениях, такое отражение называется **рассеянным**, или **диффузным** (англ. *diffuse*). Именно диффузное отражение определяет цвет объекта, который мы видим (см. рис. 9.29).

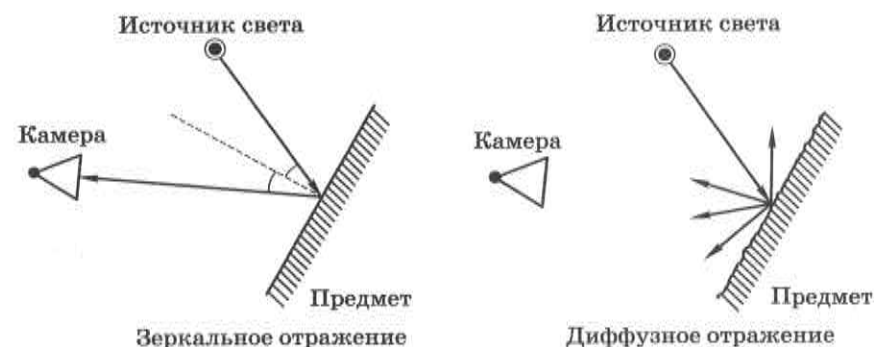


Рис. 9.29

Тип отражения зависит от степени шероховатости. Если размеры неровностей соизмеримы с длиной световой волны, происходит зеркальное отражение. Если неровности значительно больше длины волны, происходит рассеяние света (диффузия) и в глаз (или в съёмочную камеру) попадают лучи, отражённые от всех точек поверхности.

Простые материалы

Для объектов трёхмерной сцены можно построить сколько угодно различных материалов, выбирая для каждого нужные свойства. Материалы можно использовать повторно, т. е. материал одного объекта можно применять к другим объектам.

При настройке материала в первую очередь задаются два цвета — цвет диффузного (рассеянного) отражения и цвет зеркального отражения (цвет бликов). В программе Blender для изменения свойств материала используется страница свойств **Материал** (Material). В окне **Предпросмотр** (Preview) показан один из тестовых объектов, к которому применён выбранный материал (рис. 9.30).

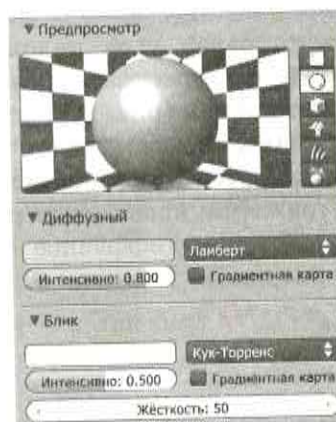


Рис. 9.30

Цвета для диффузного и рассеянного отражения задаются на панелях **Диффузный** (Diffuse) и **Блик** (Specular) соответственно, параметр **Интенсивно** (Intensity) определяет яркость цвета. В правой части каждой панели можно выбрать один из **шейдеров** (англ. *shader*) — так называют алгоритмы, с помощью которых рассчитывается цвет каждой точки изображения. По умолчанию в Blender используются алгоритмы **Ламберт** (Lambert) для диффузного отражения и **Кук-Торренс** (CookTorr) для зеркального. Включив флажок **Градиентная карта** (Ramp), можно задать **градиент** — плавный переход между двумя или несколькими цветами. Параметр **Жёсткость** (Hardness) определяет размытость бликов, чем он больше, тем более резкая граница у блика.

С помощью панели **Прозрачность** (Transparency) можно сделать материал полупрозрачным. Такой материал пропускает часть падающих на него лучей света, например красное стекло пропускает только красные лучи, а остальные поглощает.

Панель **Отражение** (Mirror) позволяет получить на предмете отражения окружающих объектов.

Многокомпонентные материалы

Разным граням (полигонам) одного объекта можно присвоить разные материалы. В этом случае объект должен быть связан с несколькими материалами, список которых находится на странице свойств **Материал** (Material). На рисунке 9.31 показан случай, когда для объекта используются три материала с именами **Red**, **Green** и **Blue**. В данном примере выбран и редактируется материал **Red**.



Рис. 9.31

Если перейти в режим редактирования (когда можно работать с отдельными гранями), появляются три кнопки:

- **Присвоить** (Assign) — присвоить выбранный материал выделенным граням;
- **Выделение** (Select) — выделить грани, для которых установлен выбранный материал;
- **Снять выделение** (Deselect) — отменить выделение граней, для которых установлен выбранный материал.

Текстуры

При создании фотореалистичных изображений часто используются не простые одноцветные материалы, а так называемые **текстуры** — точечные (растровые) изображения, которые накладываются на поверхность для изменения окраски или имитации рельефа (рис. 9.32).

Текстура обычно относится к какому-то материалу, причем с каждым материалом можно связать несколько текстур (так же, как с одним объектом можно связать несколько материалов).



Рисунок на сфере

Имитация рельефа

Рис. 9.32

Текстуры можно разделить на два типа: готовые изображения и так называемые **процедурные текстуры**, которые строятся по различным математическим алгоритмам. В окне свойств есть страница **Текстуры** (Texture), где для материала, выбранного на странице **Материал** (Material), можно задать одну или несколько текстур.

При создании новой текстуры по умолчанию выбирается тип **Облака** (Clouds). Это одна из стандартных процедурных текстур. Чтобы загрузить текстуру из файла на диске, нужно выбрать тип **Изображение или фильм** (Image or Movie), а затем, щёлкнув на кнопке **Открыть** (Open), выбрать нужный файл на диске.

В простейшем случае для наложения рисунка на объект текстура загружается в цветовой канал **Диффузный** (Diffuse), который определяет «нормальный» цвет объекта. Кроме того, текстуры можно использовать для канала **Блик** (Specular), альфа-канала (прозрачность), рельефа. Эти режимы задаются на панели **Влияние** (Influence), где нужно отметить параметры, на которые влияет текстура. Степень воздействия текстуры можно регулировать, например смешивать установленный для материала цвет и текстуру в некоторой пропорции.

Для того чтобы наложить рисунок на поверхность, каждой точке этой поверхности нужно сопоставить определённый пиксель рисунка. Это фактически означает переход к другой системе координат. Способ такого преобразования координат задаётся на панели **Отображение** (Mapping). В списке **Координаты** (Coordinates) по умолчанию установлен вариант **Сгенерировать** (Generated), т. е.

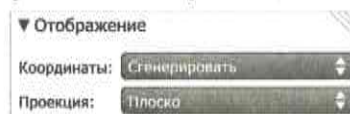


Рис. 9.33

построить автоматически (рис. 9.33). В списке **Проекция** (англ. *Projection*) выбирается форма тела: **Плоское** (Flat), **Куб** (Cube), **Сфера** (Sphere) или **Трубка** (Tube).

UV-проекция

Пользователь может вручную определить, как именно точки рисунка будут проецироваться на поверхность. Для этого применяется так называемая **UV-проекция**, или **UV-развёртка** (UV unwrap).

Чтобы вручную задать способ наложения текстуры на грани объекта, используется специальная система координат **UV**, которая похожа на стандартную прямоугольную систему **XY** на плоскости. Оси **U** и **V** аналогичны осям **X** и **Y**, но относятся не к трёхмерной модели, а к текстуре (рис. 9.34). Построить **UV-проекцию** означает сопоставить каждой точке (x, y, z) поверхности объекта некоторую точку текстуры (u, v).

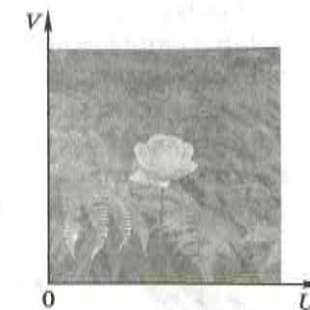


Рис. 9.34

Для каждой грани можно настраивать **UV-проекцию** отдельно с помощью окна **Редактор UV-изображений** (UV/Image Editor). Для этого в окне трёхмерной проекции (**3D View**) надо перейти в режим редактирования сетки (клавиша Tab) и выделить нужную грань. Затем следует выбрать пункт меню **Полисетка – UV-развёртка – Развернуть** (Mesh – UV Unwrap – Unwrap), и в окне редактора UV-проекций появляется плоская сетка, повторяющая форму грани (рис. 9.35).

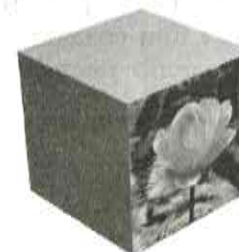
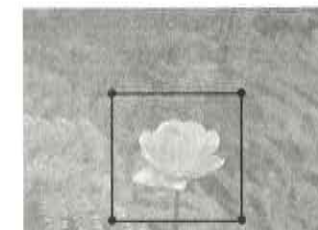


Рисунок на грани куба



UV-проекция

Рис. 9.35

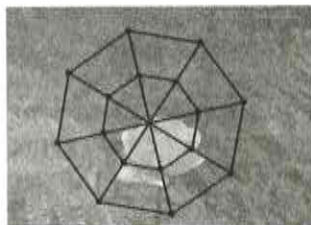
Положение вершин и рёбер такой сетки можно изменять так же, как и положение вершин и рёбер трёхмерной сеточной модели в режиме редактирования. С помощью этого приёма можно, например, вырезать из одной и той же текстуры разные рисунки для каждой грани.

Для того чтобы при построении изображения использовались координаты, заданные пользователем, а не построенные автоматически, в списке **Координаты** (Coordinates) на панели **Отображение** (Mapping) нужно выбрать вариант **UV** вместо **Сгенерировать** (Generated).

Если граней много, возникают некоторые сложности. В этом случае удобно выделить в режиме редактирования сразу несколько граней объекта, тогда в окне редактора UV-проекций будет показана вся соответствующая им сетка (рис. 9.36). Сетку и её части можно перемещать по текстуре (клавиша *G*), вращать (клавиша *R*) и масштабировать (клавиша *S*).



Рисунок на зонтике



UV-проекция

Рис. 9.36

Программа Blender позволяет построить развёртку поверхности сложного объекта, указав, где нужно сделать разрезы. После этого на такой развёртке можно рисовать текстуру для готовой модели. Такой приём широко используется при создании персонажей в играх: по готовой развёртке модели рисуют текстуру (кожу, волосы, шкуру, одежду и т. д.).



Вопросы и задания

1. Расскажите о различиях диффузного и зеркального отражения света. Почему чаще всего нужно учитывать оба варианта?
2. Приведите примеры материалов, у которых практически нет диффузного или зеркального отражения.
3. Что такое шейдер? Зачем нужны разные типы шейдеров?
4. Как регулируется размытость бликов?
5. Что такое многокомпонентные материалы? Зачем они используются?
6. Что такое текстуры? Зачем они используются?
7. Что такое процедурные текстуры? В чём их достоинства и недостатки?
8. Что означает выражение «UV-проекция»?
9. Как наносят текстуру на сложные объекты?

Задачи

1. Попробуйте применять различные шейдеры для одного и того же материала и посмотрите, как меняется внешний вид объекта.
2. Примените различные текстуры к объектам разной формы (кубу, сфере, цилиндру).
3. Создайте плоскость, разбейте её на 4 клетки и присвойте каждой клетке свой цвет.
4. Нарисуйте 6 разных изображений на одном рисунке и назначьте их разным граням куба с помощью UV-развёртки.
5. Постройте модель зонтика и нанесите на него рисунок с помощью UV-развёртки.

§ 73

Рендеринг

Рендеринг — это построение готового изображения: проекции трёхмерной сцены на плоскость с учётом материалов, текстур, освещённости, свойств внешней среды и т. п.

Для этого нужно после подготовки трёхмерной модели:

- расставить и настроить источники света;
- установить камеру, которая будет «снимать» сцену, и настроить её свойства;
- определить свойства внешней среды (цвет неба, туман и т. п.);
- выполнить рендеринг (нажать клавишу *F12*);
- сохранить готовое изображение с помощью клавиши *F3*.

Результат рендеринга появляется в окне **Редактор UV-изображений** (UV/Image Editor). В этом окне есть так называемые **слоты** (англ. *slot* — позиция, ячейка), в каждом из которых можно хранить одно изображение. Это позволяет запомнить несколько результатов рендеринга, полученных в разных условиях, а затем сравнить их.

Источники света

Источники света в Blender называются **лампами** (англ. *lamp*). Существует несколько типов ламп, различающихся по своим свойствам.

По умолчанию вновь созданная сцена содержит источник типа **Точка** (Point) — точечный источник света, лучи от которого расходятся во все стороны (радиально) — рис. 9.37.



Рис. 9.37

Освещённость зависит от расстояния между источником и поверхностью (согласно законам физики, она обратно пропорциональна квадрату этого расстояния). На рисунке 9.38 показаны элементы, позволяющие настраивать основные свойства лампы.

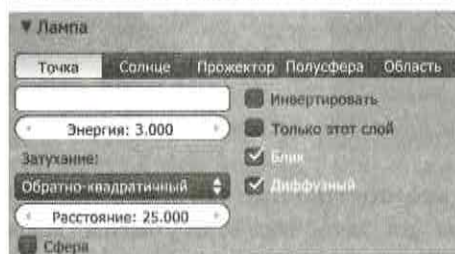


Рис. 9.38

В верхней части панели **Лампа** (Lamp) находятся кнопки для выбора типа источника света, который можно в любой момент изменить.

Параметр **Энергия** (Energy) определяет силу света. Чуть выше расположено поле выбора цвета. По умолчанию цвет лампы белый, однако в жизни крайне редко встречаются источники, излучающие белый свет (т. е. волны сразу всех длин светового диапазона). Для изменения цвета нужно щёлкнуть мышью в этом поле и выбрать нужный цвет из палитры.

Группа элементов **Затухание** (Falloff) определяет затухание света в зависимости от расстояния. Установленный по умолчанию метод **Обратно-квадратичный** (Inverse Square) лучше всего соответствует законам физики.

Параметр **Расстояние** (Distance) определяет расстояние (в условных единицах), на котором интенсивность света уменьшается в два раза. Если отметить флажок **Сфера** (Sphere), за пределами этого расстояния объекты не будут освещаться вообще.

Отключив флажок **Блик** (Specular), получаем источник, не дающий зеркального отражения. Флажок **Диффузный** (Diffuse) отключает рассеянное отражение света. Если отметить флажок **Только этот слой** (This layer only), источник не будет освещать объекты, которые находятся на других слоях. При включённом флажке **Инвертировать** (Negative) лампа не освещает, а затемняет поверхности.

Теперь рассмотрим другие типы ламп. Их основные параметры аналогичны настройкам лампы типа **Точка** (Point).

Солнце (Sun) — источник, моделирующий направленный солнечный свет (рис. 9.39). Поскольку **Солнце** находится от нас на очень большом расстоянии, солнечные лучи можно считать параллельными.



Рис. 9.39

Освещённость в этом случае не зависит от расположения лампы на сцене (в том числе от расстояния от лампы до объекта), а зависит только от направления лучей. Поэтому лампу типа **Солнце** можно ставить в любом месте сцены.

Полусфера (Hemi, от англ. *hemisphere* — полусфера) — источник, моделирующий рассеянный свет от большой полусферы (рис. 9.40). Такой световой поток содержит лучи разных направлений, поэтому освещение получается значительно мягче, чем при солнечном свете. Источники типа **Полусфера** можно использовать для подсветки теневых частей объектов. При освещении объекта таким источником падающей тени от объекта не будет.

Так же как и для лампы типа **Солнце**, освещённость не зависит от расположения лампы на сцене (в том числе от расстояния

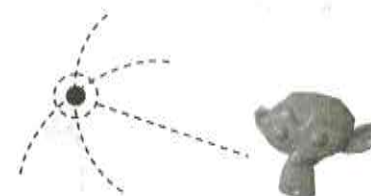


Рис. 9.40

от лампы до объекта), а зависит только от направления лучей. Такую лампу тоже можно ставить в любом месте сцены.

Область (Area) — это источник направленного света от прямоугольной площадки (рис. 9.41). Освещённость зависит от расстояния до источника и угла падения луча на поверхность. Его можно использовать, например, для создания подсветки от экрана телевизора.



Рис. 9.41

Прожектор (Spot) — это источник, который даёт направленный свет в пределах конуса (рис. 9.42). Освещённость зависит от расстояния до источника и угла падения луча на поверхность. Форма пятна может быть круглой или прямоугольной. Световой конус можно сделать видимым, используя эффект «гало» (англ. *halo* — ореол, сияние).



Рис. 9.42

Камеры

Камеры (рис. 9.43) — это специальные объекты, которые позволяют посмотреть на сцену с разных точек, как через видоискатель фотоаппарата или видеокамеры.



Рис. 9.43

По умолчанию на сцене находится одна камера. Если камера выделена, её можно перемещать (клавиша *G*) и вращать (клавиша *R*), как и любой объект. Нажав клавишу *0* на цифровой клавиатуре (*Num0*), можно переключиться на вид с камеры, при этом часть сцены, не попавшая в «поле зрения» камеры, затемняется.

Если включён вид с камеры, удобно настраивать его в «режиме полёта», который включается при нажатии клавиш *Shift+F*. Колёсико мыши приближает и удаляет камеру от объекта, а движение мыши поворачивает её в соответствующем направлении. Настроив вид, нужно завершить процедуру щелчком левой кнопкой мыши.

Ещё один вариант — получить нужный вид в окне проекции и затем поставить камеру в найденную таким образом точку наблюдения, нажав клавиши *Ctrl+Alt+Num0*.

Параметры камеры задаются на странице свойств **Камера** (Camera). В режиме **Перспективный** (Perspective) камера снимает изображение с учётом перспективы, а в режиме **Ортогональный** (Orthographic) строит ортогональную проекцию (рис. 9.44).

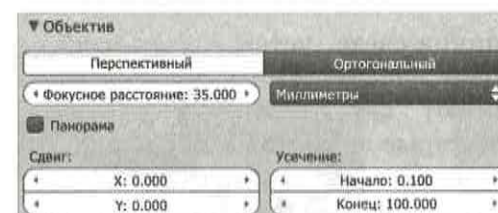


Рис. 9.44

Параметр **Фокусное расстояние** (Angle) соответствует фокусному расстоянию объектива фотоаппарата. По умолчанию используется фокус 35 мм, который примерно соответствует углу зрения человека.

Флажок **Панорама** (Panorama) предназначен для съёмки панорамных сцен.

С помощью полей группы **Сдвиг** (Shift) можно сдвинуть «поле зрения» камеры, не меняя её положение на сцене.

Параметры группы **Усечение** (Clipping) определяют область видимости камеры. Все объекты, находящиеся ближе расстояния **Начало** (Start) и дальше расстояния **Конец** (End), камера «не видит», и на итоговой картинке их не будет.

На сцене можно использовать несколько камер, переключаясь между ними. Чтобы поместить на сцену новую камеру, нужно вы-

брать пункт меню **Добавить — Камера** (Add — Camera). Клавиши **Ctrl+Num0** делают выделенную камеру активной, т. е. при рендеринге будет построено изображение именно с этой камеры.

Камеру можно «привязать» к какому-то объекту сцены, т. е. сделать так, чтобы она была всё время направлена на этот объект. Для этого на странице свойств  **Ограничения объекта** (Object Constraints) можно добавить ограничение **Слежение** (англ. *Track To*, от *track* — следовать, сопровождать¹), указав в качестве объекта-цели тот объект, на который нужно направить камеру. После этого при любых перемещениях камера будет всегда направлена на этот объект. Такая связь сохранится и при анимации — камера будет следить за объектом, если он движется.

Если в точке, куда должна смотреть камера, нет никакого объекта, можно добавить на сцену пустой объект (меню **Добавить — Пустышка** (Add — Empty)). Пустой объект можно передвигать, так же как и другие объекты, но он не отображается на сцене при рендеринге.

Внешняя среда

Кроме источников света, установленных на сцене, на итоговое изображение влияют параметры внешней среды, которые задаются на странице свойств  **Мир** (World) — рис. 9.45). По умолчанию для сцены используется серый фон, который можно изменить с помощью поля **Цвет горизонта** (Horizon Color). **Цвет окружения** (Ambient Color) задаёт цвет теней (по умолчанию чёрный).

Можно сделать градиентный фон — переход между двумя цветами, один из которых — цвет горизонта, а второй — **Цвет зенита** (Zenith Color). Для этого необходимо отметить флажок **Смесь неба** (Blend Sky). При включённом флажке **Реальное небо** (Real Sky) фон зависит от точки установки камеры: на уровне горизонта цвет совпадает с цветом горизонта, а выше и ниже горизонта переходит в цвет зенита. Если включить флажок **Псевдонебо** (Paper Sky), цвет горизонта всегда будет располагаться по центру камеры, независимо от того, как она расположена.

¹ В других программах это ограничение называется **Look At** («смотреть на»).

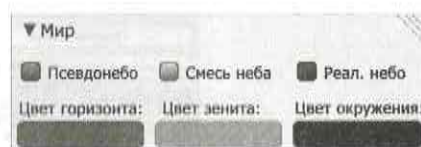


Рис. 9.45

В качестве фона можно установить растровый рисунок. Для этого нужно отменить выделение всех объектов на сцене и добавить новую текстуру на странице свойств  **Текстуры** (Texture).

Кроме того, на странице свойств **Мир** есть панели **Туман** (Mist) и **Звёзды** (Stars), с помощью которых можно создать эффекты тумана и звёзд на небе.

Параметры рендеринга

Перед тем как выполнять рендеринг (нажав на клавишу **F12**), нужно определить, какое именно изображение вам требуется. Для этого на странице свойств  **Рендер** (Render) задаются следующие параметры:

- **разрешение** (англ. *resolution*) — размеры получаемой картинки в пикселях (по умолчанию 1920 × 1080 пикселей);
- **масштаб** в процентах¹; при увеличении масштаба время расчёта сцены также увеличивается; для предварительного просмотра обычно достаточно установить масштаб 25% (рис. 9.46);



Рис. 9.46

- **сглаживание острых граней** (англ. *anti-aliasing*) (рис. 9.47);
- **тип изображения**:
 - BW — чёрно-белое (от англ. *black and white*);
 - RGB — цветное (цветовые каналы: Red — красный, Green — зелёный и Blue — синий);
 - RGBA — цветное с альфа-каналом, определяющим степень прозрачности; нужно помнить, что альфа-канал можно сохранять только в некоторых форматах (например, в PNG);

¹ При уменьшении масштаба программа уменьшает размер получаемого изображения, т. е. при масштабе 10% размер картинки при стандартных настройках будет 192 × 108 пикселей.

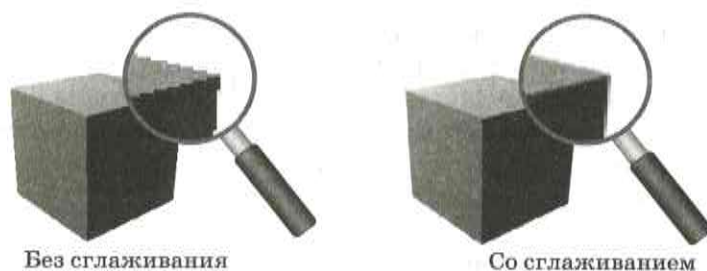


Рис. 9.47

- **формат файла** для сохранения (по умолчанию — PNG, часто выбирают другой популярный формат — JPEG);
- **дополнительную информацию** (англ. *stamp* — штамп), которая «впечатывается» в картинку: дату и время, название файла, время рендеринга и др.

Тени

В реальном мире все предметы отбрасывают тени. В программах трёхмерного моделирования для построения теней используются **алгоритмы трассировки лучей**. Это значит, что программа «запускает» большое количество лучей света от источника и просчитывает эффект, который даёт каждый луч при проходе через среду и отражении от граней, встречающихся на его пути. Такие расчёты требуют значительных вычислительных ресурсов, поэтому по умолчанию тени не строятся.

Чтобы включить построение теней, нужно настроить источники освещения и параметры рендеринга. Прежде всего, в свойствах источника света на панели **Тень** (Shadow), показанной на рис. 9.48, надо включить режим **Трассировка теней** (Ray Shadow).



Рис. 9.48

Цвет тени можно настроить с помощью поля в левой части панели (по умолчанию цвет теней чёрный). При включённом флажке **Только этот слой** (This layer only) лампа будет создавать тени только у тех объектов, которые находятся на том же слое, что и сама лампа. Флажок **Только для теней** (Only Shadow) позволяет сделать лампу, которая не освещает объекты, а только создаёт тени.

В реальности редко бывают чёткие тени с резкими краями. Поэтому используют смягчение теней, которое задаётся параметром **Размер мягкого освещения** (Soft Size). Увеличение параметра **Сэмплов** (Samples) позволяет сделать более равномерную и качественную тень, но это требует дополнительного времени при расчёте.

Кроме того, на странице параметров рендеринга (Render) на панели **Тень** (Shadow) нужно отметить флажок **Трассировка лучей** (Ray Tracing).

Вопросы и задания

1. Что такое рендеринг? Что влияет на результат рендеринга?
2. Какие типы ламп есть в Blender? Зачем они используются?
3. Что такое камера?
4. Подумайте, зачем может понадобиться использовать несколько камер.
5. Почему по умолчанию для камеры установлено фокусное расстояние 35 мм?
6. Когда, на ваш взгляд, имеет смысл использовать ортогональную проекцию?
7. Как сделать, чтобы камера всегда была направлена в какую-то точку сцены? А если там нет никакого объекта?
8. Какие свойства внешней среды можно настраивать в Blender?
9. Что такое качество рендеринга? Как оно связано со временем рендеринга?
10. Объясните, почему при рендеринге качественных изображений необходимо сглаживание граней (anti-aliasing).
11. Что такое трассировка лучей? Почему эта операция очень трудоёмкая?

Задачи

1. Загрузите в программу какую-нибудь трёхмерную сцену и попробуйте менять освещение, устанавливая дополнительные лампы и настраивая их свойства.

2. Установите для лампы ограничение **Слежение** (Track To) и проверьте, что будет происходить при движении лампы.
3. Включите трассировку лучей и сравните результаты рендеринга с тенями и без них.

§ 74

Анимация

Анимация объектов

Анимация — это быстрая смена изображений, которые называются **кадрами** (англ. *frames*). Если кадры сменяют друг друга достаточно часто (не реже, чем 16–18 раз в секунду), человеческий глаз воспринимает это как непрерывное движение.

Для работы с кадрами в Blender используется окно **Линия времени** (Timeline). По умолчанию на шкале показаны первые 250 кадров (рис. 9.49). **Курсор** (зелёная линия, перемещается мышью), показывает текущий кадр, который изображается в окне трёхмерной проекции.



Рис. 9.49

Под шкалой размещаются поля, в которых записаны номера начального (англ. *start*), конечного (англ. *end*) и текущего кадров анимации (рис. 9.50).



Рис. 9.50

Эти три значения можно ввести с клавиатуры или изменить с помощью стрелок по сторонам полей. Здесь же расположена кнопочная панель управления: кнопки для просмотра (⏮ — вперёд, ⏭ — назад) и перехода между ключевыми кадрами (рис. 9.51).



Рис. 9.51

Создание анимации в программах трёхмерной графики очень похоже на ручное рисование мультфильмов: сначала строятся так называемые **ключевые кадры**, в которых положение и свойства объектов точно задаются. Затем программа автоматически достраивает оставшиеся (промежуточные) кадры.

Для анимации движения объекта нужно:

- 1) установить курсор на временной шкале на выбранный кадр;
- 2) задать положение объекта для этого кадра;
- 3) вставить ключевой кадр, нажав клавишу **I**;
- 4) повторить эти действия для всех ключевых кадров.

При добавлении ключевого кадра появляется меню, в котором требуется выбрать тип нового кадра. Он зависит от того, какие изменения происходят с объектом: изменение положения (**Location**), вращение (**Rotation**), масштабирование (**Scaling**) или их комбинации. Ключевые кадры обозначаются на временной шкале жёлтыми линиями. Чтобы удалить ключевой кадр, нужно сделать его текущим (установить на него курсор) и нажать клавиши **Alt+I**.

В программе Blender можно анимировать не только перемещение объекта, но и изменение любого свойства, например цвета. Для вставки ключевого кадра изменения цвета нужно нажать правую кнопку мыши на поле установки цвета и выбрать из контекстного меню пункт **Вставить ключевые кадры** (Insert keyframes).

Для каждого свойства, которое участвует в анимации, устанавливаются свои ключевые кадры. Например, для поворота объекта могут быть выбраны ключевые кадры 10, 20 и 100, а для изменения цвета — кадры 1, 20, 50 и 70.

Кнопка  справа от кнопочной панели предназначена для включения режима автоматической записи: при каждом изменении свойств объекта на место текущего кадра вставляется ключевой кадр.

Анимация в окне проекции запускается с помощью клавиш **Alt+A** (вперед) или **Shift+Alt+A** (назад). Повторное нажатие останавливает анимацию на текущем кадре. Остановить анимацию и вернуться к начальному кадру можно с помощью клавиши **Esc**.

Редактор кривых

Когда установлены ключевые кадры, все изменяемые свойства объекта в промежуточных кадрах рассчитываются автоматически. По умолчанию программа выполняет плавное изменение па-

параметров от одного значения к другому. Иногда нужно изменить такое поведение, например сделать переход, который резко начинается и плавно заканчивается, или скачок значения. Для этого в Blender есть возможность ручной настройки переходов между ключевыми кадрами.

Изменение любого параметра (например, координаты или угла поворота) в зависимости от номера кадра можно изобразить на графике. Кривая на рис. 9.52 показывает изменение угла поворота вокруг оси X, рассчитанное программой.

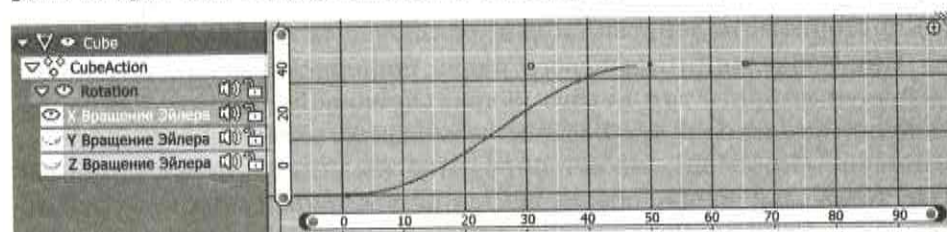


Рис. 9.52

Это окно называется **Редактор кривых (Graph Editor)**. В левой части перечислены все параметры, которые изменяются в ходе анимации. В данном случае изменяются только углы поворота объекта **Cube** (ключевые кадры типа **Rotation**). Тип перехода (постоянное значение, линейный, плавный) можно выбрать для каждого узла из всплывающего меню, которое появляется при нажатии клавиши **T**.

С помощью флажков можно отключать любую из кривых, например на рис. 9.52 показано только изменение угла поворота вокруг оси X, остальные линии скрыты.

Кривую можно редактировать так же, как и обычный контур. Узлы кривой расположены в ключевых кадрах (на рисунке 9.52 это кадры 1 и 50). У выделенного узла показаны рычаги, с помощью которых можно менять кривизну линии. Узлы перемещаются с помощью правой кнопки мыши (щелчок левой кнопкой заканчивает перемещение) или клавиши **G**. Ключевые кадры (узловые точки) каждой кривой можно устанавливать независимо от других кривых.

Щелчок на значке с изображением глаза отключает изменение соответствующего параметра при анимации, а с помощью значка «замок» можно заблокировать кривую (защитить её от изменений).

Простая анимация сеточных моделей

Выше мы говорили только об изменении свойств объектов в целом. При создании анимационных фильмов необходимо уметь перемещать части сеточной модели, например, для того, чтобы персонаж моргнул глазами. Для этого используются **ключи формы**, или **ключевые формы** (англ. *shape keys*), — так называются некоторые заранее заданные положения сеточной модели, между которыми выполняется переход. Есть одно важное ограничение: при создании таких ключевых форм нельзя менять геометрию модели, т. е. удалять и добавлять вершины, рёбра и грани.

Сначала нужно создать сами ключевые формы и определить положение узлов сетки для каждой из них (в Blender для этого используется панель **Ключи формы (Shape keys)** на странице свойств сеточной модели **Данные объекта (Object Data)**). Например, для анимации улыбающегося рта нужно построить, по крайней мере, две ключевые формы: рот без улыбки (основная форма, которая называется **Basis**) и рот с улыбкой (назовем её *Smile*) (рис. 9.53).

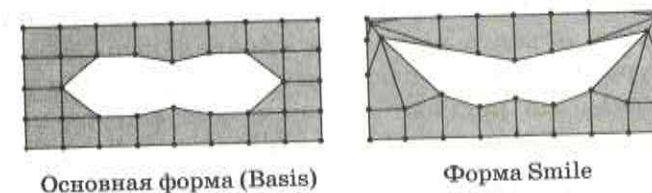


Рис. 9.53

Единственное различие этих форм в том, что вершины формы *Smile* передвинуты в другое положение.

В программе Blender удобнее всего использовать окно **Редактора ключей формы (ShapeKey Editor)**, в левой части которого перечислены все ключевые формы и показаны коэффициенты (от 0 до 1), определяющие степень влияния каждой формы на текущий кадр (рис. 9.54).

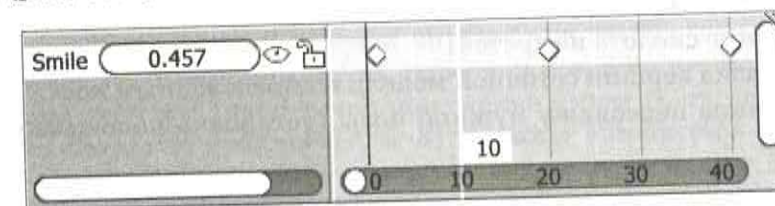


Рис. 9.54

Ключевые кадры в правой части обозначены маркерами-ромбами. Чтобы добавить новый ключевой кадр, достаточно установить на него курсор (светлую линию) и изменить значение параметра в левой части. Ключевые кадры можно перемещать (клавиша *G*) и масштабировать (регулировать интервал между кадрами, клавиша *S*). Для удаления ключевого кадра нужно выделить соответствующий ему маркер-ромб и нажать клавишу *Delete*.

Можно использовать не один, а несколько каналов анимации, при этом ключевые кадры каждого канала задаются независимо от других каналов. Форма кривой изменения параметра настраивается в окне Редактора кривых.

Арматура

Анимация с помощью ключевых форм хороша тогда, когда нужно изменить положение небольшого числа вершин сеточной модели. Во многих случаях, например при повороте шеи или сустава руки персонажа, необходимо передвинуть сотни вершин. В этом случае используют другой подход, суть которого состоит в том, что внутрь объекта вставляют специальные объекты («кости», «арматуру»), которые играют роль скелета¹. При рендеринге кости не видны. На рисунке 9.55 показана фигура шахматного короля с арматурой в двух положениях.

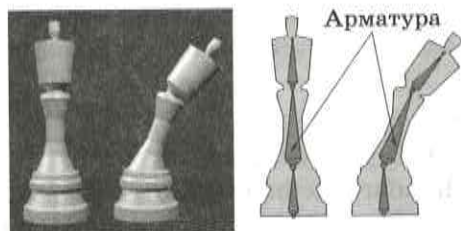


Рис. 9.55

Обычно выделяют три этапа моделирования персонажа с использованием арматуры:

- 1) создание скелета из костей;
- 2) привязка вершин сеточной модели к определённым костям;
- 3) придание персонажу нужной позы (установка положения костей).

¹ В английском языке эта процедура называется *rigging* (от англ. *rig* — оснастка).

На втором этапе арматуру, которая, как правило, состоит из нескольких связанных костей, нужно сделать родительским объектом для объекта-оболочки, установив между ними связь (клавиши *Ctrl+P*). В отличие от простой связи «объект — объект», когда преобразования объекта-родителя применяются ко всем потомкам, здесь устанавливается особая связь «арматура — оболочка».

Программа автоматически определяет, какие именно вершины сеточной модели оболочки попадают в «зону влияния» каждой из костей и будут перемещаться вслед за ней. Существует специальный режим **Оболочка** (*Envelope*), в котором можно увидеть и редактировать эти зоны влияния. Режим **Оболочка** включается на панели свойств арматуры (**Данные объекта**). При необходимости можно вручную назначить ведущую кость для каждой вершины. Для этого вершины объединяются в группы, названия которых должны совпадать с названиями объектов-костей.

Таким образом, для того чтобы изменить форму объекта, достаточно изменить положение костей. Вслед за костями переместятся и все связанные с ними вершины сеточной модели, а их могут быть тысячи!

Прямая и обратная кинематика

На рисунке 9.56 показана арматура, которую можно использовать для моделирования руки человека. Кости *A*, *B* и *B* управляют соответственно плечом, предплечьем и кистью персонажа.



Рис. 9.56

Обычно в модели эти кости связаны отношениями «родитель — потомок»: кость *A* — это родитель для *B*, а *B* — родитель для *B*. Перемещение родителя приводит к перемещению всех потомков, т. е. при перемещении кости *A* кости *B* и *B* перемещаются вслед за ней. Такая связь называется **прямой кинематикой** (англ. *forward kinematics*).

В некоторых случаях прямая кинематика затрудняет построение анимации. Например, пусть персонажу нужно взять что-то в руку. При этом мы знаем точно положение кисти *B*, тогда как положения остальных костей (*A* и *B*) должны измениться соответственно, чтобы сохранилась связь в цепочке костей. Это значит, что перемещение кости *B* приводит к согласованному перемещению костей *A* и *B*. Такая связь называется **обратной кинематикой**.

(англ. *inverse kinematics*), потому что движение передается в обратную сторону.

Для построения связи «обратная кинематика» на кость *B* нужно наложить ограничение **Обратная кинематика** (*Inverse kinematics*). В Blender для этого используется специальная страница свойств  **Ограничения кости** (*Bone constraints*), которая открывается в режиме просмотра **Режим позы** (*Pose mode*). В параметрах этого ограничения нужно указать длину цепочки костей, на которую действует ограничение (параметр *Chain length*, длина цепи). В рассмотренном случае длина цепочки равна 3 (кости *A*, *B* и *C*).

Физические явления

Современные программы трёхмерной графики содержат средства для моделирования физических процессов. Это значит, что при построении анимации тела перемещаются с учётом законов физики: дым поднимается вверх или стелется по ветру, вода капает или стекает вниз, тела сталкиваются между собой и отскакивают, ткань полощется на ветру и т. п. В этих моделях для создания реалистичной анимации используются достаточно сложные математические уравнения, их решение требует большого объёма вычислений и мощного компьютера.

В программе Blender для этой цели используют две панели свойств:  **Частицы** (*Particles*) и  **Физика** (*Physics*). С их помощью можно моделировать:

- системы частиц (дым, огонь, волосы, траву);
- жидкости;
- столкновения тел;
- силовые поля, например ветер и магнитное поле;
- ткань.

Математические модели, описывающие эти явления, уже заложены в программу. Изменяя их параметры, можно получать самые разнообразные эффекты.

Система частиц — это модель объекта, который не имеет чётких границ, например пара, огня, дыма, дождя, снега и т. п. Для таких объектов невозможно построить сеточную модель, поэтому они моделируются как поток маленьких частиц, каждая из которых имеет скорость, цвет, направление движения. При движении частиц учитываются свойства внешней среды — сила тяготения (гравитация), ветер и т. п. Для настройки системы частиц

используется множество параметров, меняя которые, можно строить модели различных явлений (дым, огонь и т. д.).

В Blender есть особый тип системы частиц — «волосы» (англ. *hair*). Они могут использоваться при моделировании волос человека, шерсти животных, а также травы (рис. 9.57).



Рис. 9.57

Жидкость (англ. *fluid*) моделируется как поверхность с большим числом граней, причём она может состоять из многих отдельных частей (капель). Для того чтобы построить анимацию жидкости, для каждого кадра положение всех вершин рассчитывается на основе предыдущего кадра и физических свойств жидкости (вязкости). Такой пересчёт иногда называют «выпечкой» (англ. *bake*). Он занимает длительное время, которое зависит от установленного разрешения (англ. *resolution*) и количества кадров анимации. При этом сеточные модели для каждого кадра создаются на диске в каталоге для временных файлов. При повторном просмотре анимации они уже не пересчитываются заново, а загружаются с диска (этот подход называют *кэшированием*). Однако если вы измените какой-либо параметр, всю анимацию придётся просчитывать заново с самого начала.

Ткань (англ. *cloth*) — это тоже сложная поверхность. В отличие от жидкости ткань создаётся и разбивается на грани вручную (например, используется плоскость, разбитая на части с помощью инструмента **Подразделить** (*Subdivide*)).

При построении анимации программа рассчитывает для каждого кадра положение всех вершин сеточной модели (падающей ткани) с учётом свойств выбранного типа ткани (хлопок, шёлк, кожа и др.). Чем мельче грани модели, тем точнее моделирование, но и больше время расчёта положения ткани в каждом кадре. Для объектов, которые должны задерживать ткань, нужно установить свойство «столкновение» (англ. *collision*). Примеры