

СТАНИСЛАВ ГОРНАКОВ



DirectX 9

УРОКИ ПРОГРАММИРОВАНИЯ

НА C++

ПРОГРАММИРОВАНИЕ
ТРЕХМЕРНОЙ ГРАФИКИ

КОМПОНЕНТЫ Direct3D,
DirectInput, DirectMusic
И DirectSound

МАТРИЧНЫЕ
ПРЕОБРАЗОВАНИЯ

ПРОГРАММИРОВАНИЕ
ВЕРШИНЫХ И
ПИКСЕЛЬНЫХ ШЕЙДЕРОВ

СПРАВОЧНИК
ПО DirectX 9 SDK



PRO
ПРОФЕССИОНАЛЬНЫЕ
ПРОГРАММИРОВАНИЕ

+CD

УДК 681.3.068+800.92С++
ББК 32.973.26-018.1
Г67

Горнаков С. Г.

Г67 DirectX 9: Уроки программирования на С++. — СПб.:
БХВ-Петербург, 2004. — 400 с.: ил.
ISBN 5-94157-482-7

Рассмотрено профессиональное программирование трехмерной графики под Windows на языке С++ с использованием библиотеки DirectX 9. Раскрыты возможности компонента Direct3D по выводу трехмерной графики, текстурированию объектов, работе с освещением, вершинными и пиксельными шейдерами и др. Описаны также компоненты DirectInput, DirectMusic и DirectSound. Материал изложен в виде уроков и поможет читателю самостоятельно изучить технологию DirectX, на основе которой создаются профессиональные компьютерные игры. Прилагаемый компакт-диск содержит примеры, рассмотренные в книге.

Для программистов

УДК 681.3.068+800.92С++
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Елена Яковлева</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Иины Тачиной</i>
Оформление обложки	<i>Игоря Цырульников</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 22.10.04.

Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 32,25.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-482-7

© Горнаков С. Г., 2004
© Оформление, издательство "БХВ-Петербург", 2004

Содержание

Введение.....	1
О чем эта книга.....	1
Что вы должны знать.....	3
Какое программное обеспечение мы будем использовать.....	3
Благодарности.....	3
Урок 1. Основы программирования под Windows.....	5
Создаем проект в Visual C++ .NET.....	5
Общая модель сообщений в Windows.....	7
Главная функция <i>WinMain</i>	8
Разбираем класс <i>windowsclass</i>	9
Создаем окно.....	13
Обработываем события в Windows.....	15
Главный обработчик событий.....	16
Итоги урока.....	20
Урок 2. DirectX 9.....	21
Компоненты DirectX 9.....	22
HAL.....	23
COM.....	23
Интерфейс <i>IUnknown</i>	24
Direct3D 9.....	26
Построение сцены в Direct3D.....	27
Интерфейсы Direct3D 9.....	28
Создание указателя на интерфейс.....	29
Установка DirectX 9 SDK.....	30
Итоги урока.....	33
Урок 3. Инициализация Direct3D.....	35
Создаем функцию для инициализации Direct3D.....	36
Обработка ошибок в DirectX.....	47
Рендеринг, или рисуем в окне приложения.....	48
Освобождаем ресурсы, захваченные Direct3D.....	51
Итоги урока.....	56
Урок 4. Рисуем 2D-объект.....	57
Установка формата вершин.....	58
Создание буфера вершин.....	60

Рендеринг объекта.....	66
Рисуем квадрат	74
Итоги урока	76
Урок 5. Матрицы.....	77
Сложение и вычитание матриц	77
Умножение матриц	78
Единичная матрица.....	79
Матрицы в Direct3D	79
Мировая матрица	80
Матрица вида.....	82
Матрица проекции	83
Итоги урока	86
Урок 6. Вывод на экран 3D-объекта	87
Матрицы преобразования	90
Рисуем куб	100
Индексация вершин.....	103
Итоги урока	114
Урок 7. Буфер глубины или Z-буфер.....	115
Связываем стороны куба.....	120
Итоги урока	130
Урок 8. Свет и материал	131
Свет.....	133
Нормаль.....	135
Установка света и материала	139
Итоги урока	150
Урок 9. Текст в Direct3D	151
Создание шрифта	151
Вывод текста на экран.....	156
Полноэкранный режим	158
Итоги урока	169
Урок 10. Текстурирование	171
Загрузка текстуры.....	172
Рендеринг текстурированного объекта.....	174
Итоги урока	188
Урок 11. Мультитекстурирование	189
Цветовые ключи	191
Итоги урока	194

Урок 12. Загрузка X-файлов	195
Итоги урока	210
Урок 13. Вершинные и пиксельные шейдеры	211
Графический конвейер	211
Фиксированный конвейер.....	213
Программируемый конвейер	214
Шейдеры	214
Вершинные шейдеры	216
Архитектура вершинных шейдеров	216
Синтаксис команд	217
Пиксельные шейдеры	227
Архитектура пиксельного шейдера	227
Синтаксис команд	229
Пишем пиксельный шейдер.....	230
Итоги урока	232
Урок 14. Инициализация DirectInput.....	233
Интерфейсы	234
Функции DirectInput8	235
Создание основного интерфейса.....	235
Создание устройства ввода.....	237
Установка формата данных устройства ввода	237
Установка уровня взаимодействия устройства ввода	238
Захват устройства ввода.....	238
Получение данных от устройства ввода	238
Буферизированные данные	239
Итоги урока	239
Урок 15. Работа с клавиатурой.....	241
Создание основного объекта DirectInput8.....	241
Создание устройства клавиатуры	242
Установка формата данных клавиатуры	243
Установка уровня взаимодействия клавиатуры	244
Захват доступа к клавиатуре.....	245
Получение данных с клавиатуры.....	245
Освобождение захваченных ресурсов	250
Итоги урока	251
Урок 16. Мышь	253
Создание устройства	254
Установка формата данных	255

Установка уровня взаимодействия	255
Захват доступа к мыши	256
Получение данных	256
Освобождение захваченных ресурсов	257
Построение класса <i>MyInput</i>	258
Описание функций класса <i>MyInput</i>	259
Итоги урока	262
Урок 17. DirectMusic	263
Интерфейсы DirectMusic	263
Создание приложения	264
Инициализация COM	265
Создание главного интерфейса	265
Создание загрузчика	266
Инициализация аудиосистемы	267
Загрузка файла из каталога	268
Загрузка сегмента в синтезатор и его воспроизведение	270
Освобождение захваченных ресурсов	271
Итоги урока	273
Урок 18. DirectSound	275
Интерфейсы DirectSound	276
Создание основного объекта	277
Установка уровня взаимодействия	277
Запись в созданный буфер	281
Воспроизведение данных	283
Классы	284
Класс <i>CSound</i>	284
Класс <i>CSoundManager</i>	286
Класс <i>CStreamingSound</i>	287
Класс <i>CWaveFile</i>	288
Итоги урока	289
Урок 19. Итоги	291
Создание приложения при помощи мастера	293
Сгенерированные классы	296
Файлы D3DApp.h и D3DApp.cpp	296
Файлы Urok19.h и Urok19.cpp	297
Файл D3DFont.h и D3DFont.cpp	300
Файлы D3DFile.h и D3DFile.cpp	301
Файлы DMUtil.h и DMUtil.cpp	304
Итоги урока	306
Заключение	307

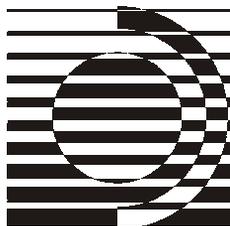
ПРИЛОЖЕНИЯ	309
Приложение 1. Справочная информация	311
DirectX Graphics	312
Интерфейсы DirectX Graphics	312
Функции DirectX Graphics	313
<i>Direct3DCreate9()</i>	313
<i>IDirect3D9::CreateDevice</i>	313
<i>IDirect3D9::GetAdapterDisplayMode</i>	314
<i>IDirect3DDevice9::Clear</i>	314
<i>IDirect3DDevice9::CreateVertexBuffer</i>	315
<i>IDirect3DDevice9::CreateVertexDeclaration</i>	315
<i>IDirect3DDevice9::CreateVertexShader</i>	316
<i>IDirect3DDevice9::DrawPrimitive</i>	316
<i>IDirect3DDevice9::GetDirect3D</i>	316
<i>IDirect3DDevice9::LightEnable</i>	317
<i>IDirect3DDevice9::Present</i>	317
<i>IDirect3DDevice9::SetLight</i>	318
<i>IDirect3DDevice9::SetMaterial</i>	318
<i>IDirect3DDevice9::SetRenderState</i>	318
<i>IDirect3DDevice9::SetStreamSource</i>	318
<i>IDirect3DDevice9::SetTexture</i>	319
<i>IDirect3DDevice9::SetTextureStageState</i>	319
<i>IDirect3DDevice9::SetTransform</i>	320
<i>IDirect3DDevice9::SetVertexDeclaration</i>	320
<i>IDirect3DDevice9::SetVertexShaderConstantF</i>	320
<i>IDirect3DBaseTexture9::SetPrivateData</i>	321
<i>IDirect3DVertexBuffer::Lock</i>	321
<i>ID3DXBaseEffect::SetVertexShader</i>	322
<i>ID3DXBaseMesh::DrawSubset</i>	322
<i>ID3DXFont::DrawText</i>	322
<i>ID3DXSkinInfo::SetFVF</i>	323
<i>D3DXAssembleShader()</i>	323
<i>D3DXAssembleShaderFromFile()</i>	323
<i>D3DXAssembleShaderFromResource()</i>	324
<i>D3DXCleanMesh()</i>	325
<i>D3DXColorModulate()</i>	325
<i>D3DXCompileShader()</i>	326
<i>D3DXCompileShaderFromFile()</i>	326
<i>D3DXCompileShaderFromResource()</i>	327
<i>D3DXComputeNormalMap()</i>	328
<i>D3DXCreateCubeTexture()</i>	329
<i>D3DXCreateCubeTextureFromFile()</i>	329
<i>D3DXCreateFont()</i>	330
<i>D3DXCreateMeshFVF()</i>	330
<i>D3DXCreateSPMesh()</i>	331

<i>D3DXCreateTextureFromFile()</i>	331
<i>D3DXCreateTextureFromFileEx()</i>	331
<i>D3DXDeclaratorFromFVF()</i>	333
<i>D3DXGetImageInfoFromFile()</i>	333
<i>D3DXLoadMeshFromX()</i>	333
<i>D3DXMatrixIdentity()</i>	334
<i>D3DXMatrixInverse()</i>	334
<i>D3DXMatrixLookAtLH()</i>	335
<i>D3DMatrixMultiply()</i>	335
<i>D3DXMatrixMultiplyTranspose()</i>	336
<i>D3DXMatrixPerspectiveFovLH()</i>	336
<i>D3DXMatrixRotationX()</i>	337
<i>D3DXMatrixRotationY()</i>	337
<i>D3DXMatrixRotationZ()</i>	337
<i>D3DXMatrixScaling()</i>	337
<i>D3DXMatrixTranslation()</i>	338
<i>D3DXMatrixTranspose()</i>	338
<i>D3DXSaveTextureToFile()</i>	339
<i>D3DXVec3Normalize()</i>	339
Структуры DirectX Graphics	339
<i>D3DADAPTER_IDENTIFIER9</i>	339
<i>D3DBOX</i>	340
<i>D3DCOLORVALUE</i>	341
<i>D3DDISPLAYMODE</i>	341
<i>D3DLIGHT9</i>	342
<i>D3DMATERIAL9</i>	343
<i>D3DPRESENT_PARAMETERS</i>	344
<i>D3DRANGE</i>	345
<i>D3DRECT</i>	346
<i>D3DVECTOR</i>	346
<i>D3DVERTEXELEMENT9</i>	346
<i>D3DVIEWPORT9</i>	347
<i>D3DXMATERIAL</i>	347
<i>RECT</i>	348
Типы DirectX Graphics	348
<i>D3DBLEND</i>	348
<i>D3DBLENDOP</i>	349
<i>D3DCULL</i>	350
<i>D3DDEVTYPE</i>	350
<i>D3DLIGHTTYPE</i>	351
<i>D3DPOLL</i>	351
<i>D3DPRIMITIVETYPE</i>	352
<i>D3DRENDERSTATETYPE</i>	353
<i>D3DVERTEXBLENDFLAGS</i>	361
<i>D3DZBUFFERTYPE</i>	361

Макросы DirectX Graphics	362
<i>D3DCOLOR_ARGB</i>	362
<i>D3DCOLOR_COLORVALUE</i>	362
<i>D3DCOLOR_RGBA</i>	363
<i>D3DTS_WORLDMATRIX</i>	363
DirectInput	363
Интерфейсы DirectInput	363
Функции DirectInput	364
<i>DirectInput8Create()</i>	364
<i>IDirectInput8::CreateDevice</i>	364
<i>IDirectInput8::EnumDevices</i>	365
<i>IDirectInput8::FindDevice</i>	365
<i>IDirectInput8::GetDeviceStatus</i>	366
<i>IDirectInput8::Initialize</i>	366
<i>IDirectInput8::RunControlPanel</i>	366
<i>IDirectInputDevice8::Acquire</i>	366
<i>IDirectInputDevice8::BuildActionMap</i>	367
<i>IDirectInputDevice8::GetCapabilities</i>	367
<i>IDirectInputDevice8::GetDeviceInfo</i>	367
<i>IDirectInputDevice8::GetDeviceState</i>	367
<i>IDirectInputDevice8::SetDataFormat</i>	368
<i>IDirectInputDevice8::SetCooperativeLevel</i>	368
Структуры DirectInput	369
<i>DIDATAFORMAT</i>	369
<i>DIMOUSESTATE</i>	369
DirectMusic	370
Интерфейсы DirectMusic	370
Функции DirectMusic	370
<i>IDirectMusicLoader8::LoadObjectFromFile</i>	370
<i>IDirectMusicPerformance8::InitAudio</i>	371
<i>IDirectMusicPerformance8::PlaySegmentEx</i>	371
<i>IDirectMusicSegment8::SetRepeats</i>	372
DirectSound	372
Интерфейсы DirectSound	373
Функции DirectSound	373
<i>DirectSoundCreate8()</i>	373
<i>IDirectSound8::CreateSoundBuffer</i>	373
<i>IDirectSound8::SetCooperativeLevel</i>	374
<i>IDirectSoundBuffer8::Lock</i>	374
<i>IDirectSoundBuffer8::Unlock</i>	375
<i>IDirectSoundBuffer8::Play</i>	375
Структуры DirectSound	376
<i>DSBUFFERDESC</i>	376
<i>WAVEFORMATEX</i>	376

Приложение 2. Web-ресурсы	378
Приложение 3. Список использованных источников.....	379
Приложение 4. Описание компакт-диска	380
Предметный указатель	381

Урок 2



DirectX 9

На этом уроке мы познакомимся с библиотекой DirectX 9 SDK, объединяющей в себе ряд компонентов для работы с графикой, звуком, видео, устройствами ввода, а также другими мультимедийными компонентами. В конце урока установим DirectX 9 SDK с компакт-диска, прилагаемого к книге, на ваш компьютер.

Что такое DirectX? Представьте себе ситуацию: вы разрабатываете игру и как разработчик (не говоря уже об издателе) желаете, чтобы игра могла работать на как можно большем количестве компьютеров. Но для этого вам необходимо учесть все мыслимые и немыслимые видео-, звуковые и сетевые карты, рули, мыши, клавиатуры и еще целый "ворох" разного оборудования. Гораздо легче создать определенный стандарт, благодаря которому написание кода, например, для устройств ввода, можно сделать единожды, а работать он будет всегда и везде. К этому и стремится корпорация Microsoft уже много лет, и с каждой новой версией DirectX ей это удается все лучше и лучше.

DirectX — это мультимедийная библиотека, позволяющая напрямую работать с аппаратным обеспечением компьютера в обход традиционных средств платформы Win32. Тесное взаимодействие DirectX и оборудования с драйверами, написанными производителем данного оборудования, дают возможность полностью отвлечься от аппаратной части и сосредоточить свое внимание на создании правильного кода. Вся DirectX 9 делится на компоненты, отвечающие за ту или иную часть работы библиотеки:

- DirectX Graphics;
- DirectXInput;
- DirectXMusic;
- DirectXSound;
- DirectXPlay;
- DirectXShow;
- DirectXSetup.

На рис. 2.1 представлена общая схема взаимодействия DirectX 9 с аппаратным обеспечением компьютера.

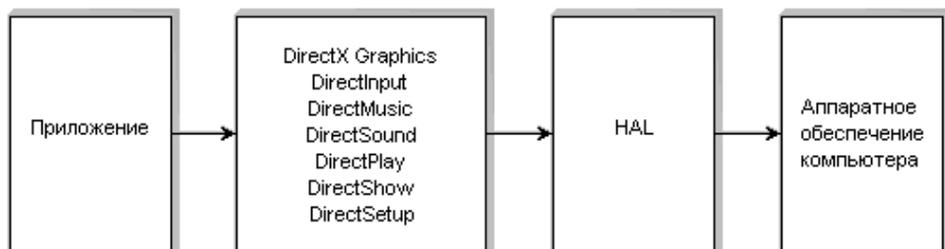


Рис. 2.1. Схема взаимодействия DirectX 9

Компоненты DirectX 9

В этой книге мы познакомимся с четырьмя составляющими DirectX 9 — это Direct3D, входящий в состав DirectX Graphics, DirectInput, DirectMusic и DirectSound. Все перечисленные компоненты необходимы в программировании компьютерных игр. Что касается звука, то вообще достаточно будет DirectMusic, но в целях общего ознакомления рассмотрим некоторые составляющие из DirectSound.

DirectX Graphics в ранних версиях (до седьмой включительно) был разделен на две части: DirectDraw и Direct3D. DirectDraw отвечал за двухмерную графику, тогда как Direct3D за трехмерную. В восьмой версии DirectX DirectDraw был объединен в Direct3D. Произошло это по понятным причинам, настала эра трехмерных игр и удивить кого-то двухмерной графикой уже было невозможно. С выходом DirectX 9 почти ничего не изменилось, хотя конечно возможность к доступу DirectDraw по-прежнему имеется, но вот насколько это актуально? В любом случае мы будем рассматривать только Direct3D, отвечающий за работу с трехмерной графикой.

DirectInput — это компонент, отвечающий за работу со всеми устройствами ввода: клавиатурой, мышью, джойстиком, рулем, шлемом "виртуальной реальности", с устройствами виброотдачи и обратной связи. Данный компонент обеспечивает работу устройств и нужную скорость доступа к приложению за счет прямого взаимодействия программы с аппаратным обеспечением компьютера.

DirectMusic, — как видно из названия, отвечает за работу с музыкой, и в большей степени его задачей является работа с MIDI (Musical Instrument

Digital Interface), но есть возможность проигрывания WAV-файлов. Это очень сложный компонент и одним проигрыванием MIDI- и WAV-файлов он не ограничивается.

DirectSound — эта часть DirectX 9 отвечает за звук и позволяет применять основные, вспомогательные, статистические и потоковые буферы, обеспечивающие всевозможные эффекты звучания. Также может воспроизводить WAV-файлы.

DirectPlay призван обеспечить ваше приложение взаимодействием с локальной сетью или сетью Internet для многопользовательского режима игр. Построен на своем протоколе, обеспечивающем работу со всеми протоколами сети.

DirectShow необходим для работы с потоковым видео, в частности, служит для захвата и воспроизведения видео различных форматов: MPEG (Motion Picture Experts Group), MPEG Audio Layer-3 (всем известный как формат MP3) и, конечно Audio-Video Interleaved (AVI).

DirectSetup — компонент, отвечающий за установку файлов DirectX на компьютер пользователя из вашего приложения. Вы, наверно, сталкивались с ситуацией, когда до или после установки игры на экране появлялось диалоговое окно с предложением установить, например, более позднюю версию DirectX. Как раз за установку и отвечает этот компонент. Также в его обязанности входит автоматический запуск диска с игрой из дисковода.

В итоге, можно сказать, что DirectX 9 — это очень большая библиотека, отвечающая за работу с мультимедийными составляющими компьютера, обеспечивая высокую производительность, и, что самое главное, дает возможность не задумываться об аппаратной части компьютера при написании кода.

HAL

DirectX включает в себя уровень абстракции — *HAL* (Hardware Abstraction Layer — уровень абстракции оборудования), позволяющий настроить работу при любом аппаратном обеспечении.

С помощью уровня абстракции оборудования HAL происходит взаимодействие приложения с оборудованием компьютера, вне зависимости от изготовителя этого оборудования. Это дает возможность единожды написанному коду работать на любом аппаратном обеспечении.

COM

Вся библиотека DirectX построена на основе *COM* (Component Object Model) — модели составных компонентов. Самое главное, это то, что вам не

придется углубляться в сущность COM-технологии, вся работа с COM основана на вызовах соответствующих функций. Однако понимать общие принципы COM необходимо, поэтому немного теории не помешает.

COM — это своего рода спецификация, не зависящая от платформы и языка программирования, определяющая стандарт по использованию и созданию программных компонентов. У вас дома есть несколько розеток и несколько вилок для 220 V, любую вилку вы можете включить в любую розетку — это и есть стандарт, сопровождающийся своей спецификацией. Спецификация COM устанавливает определенный стандарт для разрабатываемой компонентной архитектуры. Программные компоненты состоят из исполняемого кода, представленного в виде динамических библиотек (DLL) и EXE-файлов платформы Windows.

В составе COM имеется API, называемая COM-библиотекой; с ее помощью достигается управление всеми компонентами. Каждый из программных компонентов реализует определенное количество COM-объектов, доступ к которым осуществляется посредством интерфейсов, которые, в свою очередь, состоят из функций. С помощью этих функций и происходит взаимодействие с COM-объектом. Интерфейсы в COM — это строго определенная структура, содержащая массив указателей на функции. Каждый из компонентов DirectX, например Direct3D, включает ряд интерфейсов с набором функций, с помощью которых мы будем получать доступ к COM-объекту. Все интерфейсы в DirectX имеют свое обозначение и начинаются с английской заглавной буквы "I". Далее указывается название COM-объекта, например Direct3D, затем номер версии используемой библиотеки DirectX, например: IDirect3D9. Все интерфейсы строго определены и не могут изменяться, что позволяет иметь доступ к более старым версиям интерфейсов.

COM-объекты и интерфейсы идентифицируются в системе 128-битным глобально-уникальным идентификатором (GUID), или, как говорят, "гидом", назначаемым при разработке компонента. Аббревиатура для записи идентификатора интерфейса существует в виде IID (Interface Identifier — идентификатор интерфейса), а для COM-объекта — CLSID (Class Identifier — идентификатор класса).

Интерфейс *IUnknown*

Все COM-интерфейсы наследуются от интерфейса IUnknown, причем дословный перевод означает "неизвестный". Забавно, не правда ли? Интерфейс IUnknown содержит три функции: `AddRef()`, `Release()` и `QueryInterface()`, а поскольку все интерфейсы наследуются от IUnknown, то все три функции имеются в каждом из COM-интерфейсов.

СОМ-объект содержит счетчик, определяющий количество используемых интерфейсов. С каждым новым задействованным интерфейсом счетчик увеличивается на единицу. С помощью функции `AddRef()` происходит увеличение счетчика определенного СОМ-объекта. Пользоваться нам этой функцией не придется, скажем так, увеличение счетчика ссылок происходит автоматически.

После того как интерфейс отработал, он освобождается, и внутренний счетчик СОМ-объекта уменьшается на единицу. Как только счетчик обнуляется, СОМ-объект выгружает себя из памяти, и в тот же момент все созданные им объекты будут выгружены системой. За уменьшение счетчика ссылок на единицу отвечает функция `Release()`. А вот этой функцией мы будем пользоваться постоянно. Операция, выполняемая функцией, еще называется "освобождением захваченных ресурсов". Далее на уроках мы создадим специальную функцию, которая будет освобождать захваченные ресурсы путем вызова функции `Release()`.

Как уже говорилось, в DirectX 9 нет интерфейса `DirectDraw`, отвечающего за двумерную графику, но вдруг по какой бы то ни было причине, вам захотелось получить доступ к этому интерфейсу. Например, к седьмой версии `DirectDraw7`. Поскольку нам известно, что реализация интерфейсов строго определена для совместимости с более ранними версиями, то такая возможность у вас, конечно, имеется. Для этого существует функция `QueryInterface()`. Если рассмотреть прототип этой функции, то все сразу станет ясно:

```
HRESULT QueryInterface(  
    REFIID iid,  
    void** ppvDest)
```

Функция `QueryInterface()` имеет следующие параметры:

- `iid` — это идентификатор запрашиваемого интерфейса (в соответствии с аббревиатурой IID);
- `ppvDest` — адрес указателя на полученный интерфейс.

Тогда, чтобы получить доступ к интерфейсу `DirectDraw7`, необходимо написать такой код:

```
Указатель_на_интерфейс -> QueryInterface(IID_IDirectDraw7,  
                                           &Указатель_нужной_версии);
```

Здесь, конечно, не хватает нескольких операций, но все станет на свои места по мере прочтения этой книги. При успешном выполнении данной функции вам станет доступен интерфейс `DirectDraw7`.

Direct3D 9

Direct3D — это самый мощный и самый главный компонент DirectX 9, отвечающий за отрисовку трехмерной графики, наложения на объект текстур, за вершинные и пиксельные шейдеры и многое другое. С помощью Direct3D существует возможность встраивать трехмерную графику в приложение для платформы Windows. Имея в своем распоряжении большой набор различных функций, структур, макросов, флагов, разработчик практически отодвинут от аппаратной части компьютера и сосредоточен на программном коде. Сам же программный код, а точнее множество функций трехмерной графики, очень легко встраиваются и взаимодействуют с приложением.

Самой главной составляющей Direct3D является процесс рендеринга (rendering). *Рендеринг* — это визуализация трехмерного объекта на экране монитора. Здесь важно понимать, что любой объект хранится в видео- или системной памяти компьютера, как разобранный на части конструктор, где каждая из частей отвечает за ту или иную составляющую объекта. Например, робот из конструктора изначально состоит из головы, ног, рук, туловища, точно так же и объект разбит на части, состоящие из текстур, света и материала, полигонов и многого другого. Чтобы увидеть целого робота в конструкторе, его нужно собрать из частей, чтобы увидеть объект на экране монитора, он должен пройти через процесс рендеринга, который произведет сборку объекта. Конечно, на самом деле я здесь немного упрощаю общую концепцию, также под объектом подразумевается не только какая-то отдельно взятая трехмерная модель, но и весь окружающий ее мир. В Direct3D это носит название *сцены*.

Весь процесс рендеринга состоит из нескольких механизмов — тесселяции, трансформации, освещения и растеризации. С помощью этих механизмов или, как говорят, *конвейера рендеринга*, достигается визуализация всей сцены на экране.

Тесселяция (tessellation) — это механизм разбиения поверхности на полигоны. Полигон — это определенная площадь в трехмерном пространстве, ограниченная углами. В Direct3D полигоном обычно является треугольник.

Трансформация (transformation) — это преобразование координат объекта с помощью матриц, производящих вращение, перенос, масштабирование и другие различные матричные преобразования объекта.

Освещение (lighting) — прямая аналогия нашему миру, где все предметы имеют цвет, который в свою очередь зависит от источников освещения.

Растеризация (rasterizer) — это самый сложный механизм, который, по сути, и рисует всю сцену на экране монитора, представляя ее на пиксельном уровне, осуществляя текстурирование, Z-буферизацию, затенение, альфа-смешивание, сокрытие невидимых поверхностей и многое другое.

Построение сцены в Direct3D

Прежде чем мы начнем программирование графики средствами Direct3D, хотелось бы в общих чертах коснуться принципа построения сцены, который условно можно разбить на несколько частей.

1. *Создание оконного приложения.* Это то, чем мы занимались на первом уроке. Создав окно, в любом выбранном вами режиме и виде, вы создаете себе рабочую поверхность, куда будете последовательно интегрировать свой код и функции DirectX.
2. *Инициализация Direct3D.* Здесь вы создаете ряд указателей на необходимые вам интерфейсы, инициализируется Direct3D 9, задаются параметры представления и создается устройство Direct3D. Устройство Direct3D — это, если можно так сказать, аппаратная и программная часть вашего компьютера. То есть вы задаете разрешение экрана, частоту смены кадров, узнаете информацию о видеокарте, устанавливаете, как видеоадаптер будет обрабатывать предложенную ему информацию, задаете формат заднего буфера, устанавливаете параметры буфера глубины, вообще все то, что касается настройки аппаратной и программной частей вашего компьютера.
3. *Создание объекта.* Каждый объект, каким бы он большим не был, состоит из полигонов. Обычно это треугольники. Любой треугольник состоит из трех углов или точек. В Direct3D оперируют понятием *вершины*. Вершина задается координатами в пространстве по трем осям X , Y и Z . Зная все это, а также имея представление о тригонометрии, можно построить практически любой объект. Задавая формат вершин в зависимости от того, используете ли вы матричные преобразования для рендеринга объекта или пользуетесь преобразованным форматом вершин, для представления 2D-объекта на экране, вы тем самым создаете объект.
4. *Освещение, материал и текстура.* Реально сцену можно представить на экране, если использовать освещение и материал. Создав различные источники света, например, лампочку, солнце, огонь, и задав материал для объекта, определяющий отражение источника света, вы добавите реализма рисуемой сцене. Ну, а наложив текстуру, например, в виде паркета на пол, вы добьетесь абсолютной идентичности нашему миру.
5. *Рендеринг объекта.* Можно прорисовать сцену на экране, задавая различные значения для всех компонентов конвейера рендеринга.

Конечно, все эти этапы для наглядности разделены на условные составляющие, но общий принцип, я думаю, вам теперь понятен. Сейчас давайте рассмотрим, из каких интерфейсов состоит Direct3D 9 и как создать указатель на необходимый интерфейс. А в конце урока установим DirectX 9 SDK на ваш компьютер.

Интерфейсы Direct3D 9

Direct3D 9 содержит самое большое количество различных интерфейсов отвечающих за текстуры, вершинные и пиксельные шейдеры, буфер вершин, индексный буфер, интерфейс устройства Direct3D 9 и т. д. Все интерфейсы мы рассмотрим не в силах, поэтому разберем только непосредственно встречающиеся в книге:

- `IDirect3D9`;
- `IDirect3DDevice9`;
- `IDirect3DVertexBuffer9`;
- `IDirect3DIndexBuffer9`;
- `IDirect3DTexture9`;
- `IDirect3DVertexShader9`.

`IDirect3D9` — самый главный интерфейс, из него наследуются все остальные интерфейсы. Это первый объект, который вы должны создать, и только потом можно получить доступ ко всем остальным интерфейсам и функциям. Объект интерфейса создается при помощи функции `Direct3DCreate9()`.

`IDirect3DDevice9` — после того как создан объект основного интерфейса `IDirect3D9`, создается интерфейс устройства Direct3D. Как я уже говорил, это, по сути, определенный набор необходимых настроек аппаратной и программной составляющих компьютера, создающих устройство Direct3D, на основе которого рисуется вся сцена. Если мы говорим о графике, то очевидно, что устройство Direct3D будет представлять видеоадаптер. Этот интерфейс всегда создается вторым, с помощью функции `IDirect3D9::CreateDevice`. И, конечно же, данный интерфейс содержит еще ряд функций, осуществляющих операции с устройством Direct3D, также это в полной мере касается и других интерфейсов.

`IDirect3DVertexBuffer9` — интерфейс, с помощью которого создается буфер вершин. Каждый объект состоит из треугольников, заданных точками в пространстве (вершин). Понятно, что весь объект состоит из определенного набора вершин, и для того, чтобы все эти вершины было удобно хранить, создается буфер вершин, являющийся в общем виде массивом данных. Буфер вершин создается при помощи функции `IDirect3DVertexBuffer9::CreateVertexBuffer`.

`IDirect3DIndexBuffer9` — используя этот интерфейс, вы можете создать индексный буфер, позволяющий проиндексировать большое количество вершин объекта. Например, вы имеете прямоугольник, состоящий из двух треугольников, где по крайней мере четыре вершины (по две на каждый из двух углов) имеют одинаковые координаты, т. е. повторяются. С помощью индексного буфера можно проиндексировать вершины и избежать повторения. Может быть, конечно, для треугольника это и не актуально, но для куба индексация

вершин придется кстати. При необходимости индексный буфер можно создать с помощью функции `IDirect3DIndexBuffer9::CreateIndexBuffer`.

`IDirect3DTexture9` — как видно из названия, этот интерфейс отвечает за текстуры. В его состав входит много функций, с помощью которых вы можете накладывать текстуры на объект, одну на другую и т. д.

`IDirect3DVertexShader9` — этот интерфейс, как мне кажется, отвечает за будущее игровой индустрии, по крайней мере, на несколько ближайших лет точно. Дело в том, что механизм трансформации и освещения в более серьезных приложениях потихоньку сдает свои позиции, поскольку имеет свои ограничения в использовании, и ему на смену пришли вершинные шейдеры, улучшающие качество графики. Вершинный шейдер — это программа, написанная на языке, подобном ассемблеру, и интегрированная в создаваемое приложение, позволяющая в реальном времени достичь более качественного уровня графики.

Как уже упоминалось, это далеко не все интерфейсы, а только те, с которыми нам предстоит столкнуться. В состав DirectX 9 еще входит так называемая библиотека утилит DirectX 9. Используя библиотеку, можно загружать объекты, созданные в 3DS MAX, или осуществлять матричные преобразования с объектом. Состав этой библиотеки тоже содержит множество различных интерфейсов, с некоторыми из которых мы познакомимся в этой книге.

Создание указателя на интерфейс

Чтобы создать указатель на интерфейс, нужно просто объявить переменную определенного интерфейса, в которой будет храниться указатель. Существуют, по крайней мере, два способа создания указателя на интерфейс.

Первый способ:

```
IDirect3D9*      pDirect3D;  
IDirect3DDevice9* pDirect3DDevice;
```

Второй способ:

```
LPDIRECT3D9      pDirect3D;  
LPDIRECT3DDEVICE pDirect3DDevice;
```

Оба способа приводят к одному результату. Чтобы было понятно, почему это возможно, приведу спецификацию для интерфейса `IDirect3D9`, которая является показательной и для всех остальных интерфейсов:

```
typedef struct IDirect3D9 *LPDIRECT3D9 *PDIRECT3D9
```

В книге везде используется второй способ. После того как был создан указатель на интерфейс, можно попытаться создать объект. Раз уж мы создали

два указателя на интерфейсы `IDirect3D9` и `IDirect3DDevice9`, давайте пойдем до логического конца и создадим два объекта для обоих интерфейсов:

```
// указатель на интерфейс IDirect3D9
pDirect3D = Direct3DCreate9(D3D_SDK_VERSION);
// указатель на интерфейс IDirect3DDevice9
pDirect3D -> CreateDevice(D3DADAPTER_DEFAULT,
                        D3DDEVTYPE_HAL, hwnd,
                        D3DCREATE_SOFTWARE_VERTEXPROCESSING,
                        &Direct3DParametr, &pDirect3DDevice);
```

Более подробно эти шаги будут разбираться на *уроке 3*, сейчас важно понять общий смысл всех действий, чтобы не напоминать об этом далее. В начале создается указатель на интерфейс, а точнее, переменная, в которой будет храниться указатель на интерфейс. После чего создается объект интерфейса, и указатель на этот созданный объект помещается в переменную, которая получает название — указатель на интерфейс. Поэтому всегда, когда будет упоминаться указатель на интерфейс, в сущности, это переменная, в которой хранится указатель на объект интерфейса.

А теперь давайте установим библиотеку `DirectX 9 SDK`, находящуюся на прилагаемом к книге компакт-диске, на ваш компьютер. Если вы уже это сделали, проверьте, все ли было сделано правильно. Ничего сложного нет, но пара-тройка нюансов имеется.

Установка DirectX 9 SDK

На прилагаемом к книге компакт-диске в папке `DirectX 9 SDK` лежит файл `dx90bsdk.exe` размером 218 Мбайт. На сегодняшний день это самая последняя версия продукта `DirectX 9.0b`, включающая в себя библиотеки для трех языков программирования: `C++`, `C#`, `Visual Basic`.

Установка выполняется следующим образом:

1. Дважды щелкните левой кнопкой мыши на значке файла `dx90bsdk` в папке `DirectX 9 SDK`. Это приведет к распаковке всего SDK на ваш компьютер.
2. После распаковки на экране монитора появится окно **Microsoft DirectX 9.0 SDK - InstallShield Wizard** (рис. 2.2). У вас появится возможность выбора необходимых компонентов. В пункте **DirectX Samples and Source Code** предлагаются к выбору документация, примеры, коды к трем языкам программирования, нас интересует непосредственно `C++`, все остальные языки на ваше усмотрение.