

РАЗРАБОТКА ГРАФИЧЕСКОЙ БИБЛИОТЕКИ ДЛЯ СОЗДАНИЯ ПРИЛОЖЕНИЙ НА БАЗЕ *DIRECTX*

Филиал «Протвино» университета «Дубна»
Кафедра информационных технологий

В работе описывается создание графической библиотеки для приложений на базе *DirectX*, представлены реализованные возможности, рассматриваются примеры использования этой библиотеки.

Для облегчения создания современных игр активно применяют игровые движки — набор библиотек, обеспечивающих визуализацию графики, звука, основных технологий, используемых в игре, интерфейса, физики и т. д.

Графическая библиотека — часть игрового движка, задачей которой является визуализация двумерной или трёхмерной компьютерной графики. Современные графические библиотеки визуализируют графику с помощью графических процессоров.

Разрабатываемая библиотека является первым этапом в создании игры, а именно — в создании игрового движка для неё. Но, в отличие от физических библиотек, библиотек интерфейса и т. п., которые разрабатываются под конкретные жанры, графическая библиотека может успешно использоваться в различных игровых жанрах.

Графические библиотеки являются «оболочками» над определенными *API*, самыми популярными из которых являются *DirectX* и *OpenGL*.

API (*Application programming interface*) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах. Используется программистами для написания всевозможных приложений.

DirectX (от англ. *direct* — прямо, напрямую) — это набор *API*, разработанных для решения задач, связанных с программированием под *Microsoft Windows*. Наиболее широко используется при написании компьютерных игр, некоторых научных пакетов, программ создания 3D-графики. Пакет средств разработки *DirectX* под *Microsoft Windows* бесплатно доступен на сайте *Microsoft*.

OpenGL (*Open Graphics Library* — открытая графическая библиотека, графический *API*) — определяет независимый от языка программирования программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику. Включает более 250 функций для рисования сложных трёхмерных сцен из простых примитивов. Используется при создании компьютерных игр, САПР (Система Автоматизированного Проектирования), виртуальной реальности, визуализации в научных исследованиях.

В целом функциональности данных *API* приблизительно равны, как и их производительность. В качестве оболочки для создаваемой библиотеки была выбрана *DirectX* ввиду наличия подробной документации с примерами.

Основной целью работы было создание такой библиотеки, которая позволит сосредоточиться непосредственно на разработке самой игры, а не на работе с *DirectX SDK* (инициализацией графических ресурсов и т. п.).

SDK (от англ. *software development kit*) — комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, игровых консолей, операционных систем и прочих платформ.

Создаваемая графическая библиотека имеет следующие, необходимые для разработки 3D-игр, возможности:

- Инициализация устройства (видеокарты, а также заднего буфера (в него происходит рисование) и буфера глубины (позволяет предметам, которые ближе к камере, скрывать те, которые находятся за ними)).
- Создание окна, в которое будет выводиться изображение.
- Инициализация ресурсов (вершинных, пиксельных и константных шейдеров).
- Загрузка ресурсов (например, текстур) и 3D-моделей в программу.
- Вывод текста.
- Реализация освещения.

- Аппаратная тесселяция, то есть разбиение каждого полигона в 3D-модели на несколько, — это увеличивает качество её детализации.
- Реализация теней.

В качестве примеров прототипов некоторых функций библиотеки можно привести следующие:

- *HRESULTInitDevice (UINTwidth, UINTheight, boolWindowed)* — функция должна вызываться в начале программы. Она инициализирует основные ресурсы программы (окно для вывода изображения, основные буферы и устройство). В параметрах передается высота и ширина экрана, а также — оконное оно или нет.

- *HRESULTCleanup* — производит очистку ресурсов и должна вызываться перед завершением работы приложения.

- *HRESULTLoadSDKMesh(LPWSTRname)* — функция производит загрузку 3D-модели в программу. В качестве параметра передается имя модели.

- *HRESULTLoadPS(LPWSTRname, constchar_tfunk, PixelShader** ps)* — функция производит загрузку пиксельного шейдера в программу. В качестве параметров передаются название файла с шейдерами, имя функции, являющейся точкой входа, и указатель на сам шейдер, в который он будет загружен. В библиотеке есть прототип и для загрузки вершинных шейдеров.

- *HRESULTInitCBuf(Buffer** buf,inti)* — данная функция инициализирует константный буфер. В параметрах передается указатель на сам буфер и требуемый размер. В библиотеке также есть аналогичные функции для инициализации других типов буферов, например, вершинного или индексного.

Примеры, использующие библиотеку (рис. 1 и рис. 2):

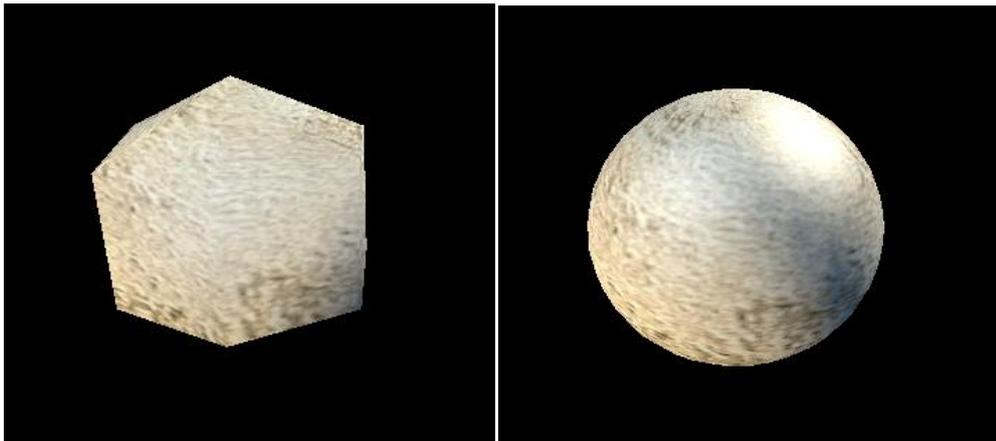


Рис. 1 Использование аппаратной тесселяции на простой 3D-модели (справа — без тесселяции, слева — с тесселяцией)

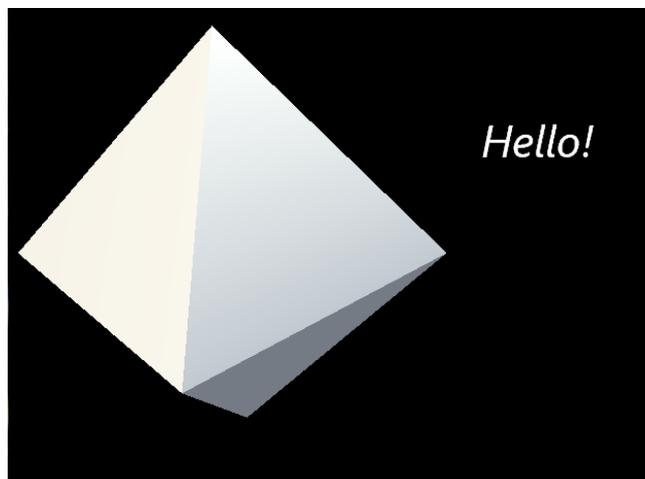


Рис. 2 Вывод на экран примитива и текста «Hello!»

Текст на рис. 2 является динамическим, то есть его можно изменять во время работы приложения.

Благодаря библиотеке, в данных примерах реализованы:

- Инициализация всех графических ресурсов.
- Загрузка самой модели и текстуры.
- Создание направленного освещения.
- Аппаратная тесселяция модели.
- Создание динамического текста.
- Вывод изображения на экран.

В дальнейшем, разрабатываемая библиотека будет расширяться, в нее планируется добавить возможности загрузки моделей и текстур различных форматов (в данный момент есть только один тип), поддержку анимаций.

Ссылки на Интернет-ресурсы:

1. <http://ru.wikipedia.org/wiki/DirectX>
2. <http://ru.wikipedia.org/wiki/OpenGL>
3. http://ru.wikipedia.org/wiki/Игровой_движок

