Список литературы:

1. Налоги и налогообложение, учебное пособие 2005г. (Сычева Е. И., Шувалова Е. Б.)
2. Налоговый Кодекс Российской Федерации
3. Федеральная Налоговая Служба, раздел статистики и аналитики - http://www.nalog.ru/nal_statistik/

РАЗРАБОТКА ГРАФИЧЕСКОЙ БИБЛИОТЕКИ ДЛЯ УПРОЩЕНИЯ СОЗДАНИЯ ИГР

Автор: Евтушенко Ян, студент II курса.
Руководитель: Кульман Татьяна Николаевна, доцент кафедры Информационных технологий, к.т.н.
Образовательное учреждение: ГОУ Московской области Международный университет природы, общества и человека «Дубна», филиал «Протвино», г. Протвино.

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

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

Разрабатываемая библиотека является первым этапом в создании игры в жанре RPG, а именно первым этапом 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, виртуальной реальности, визуализации в научных исследованиях.

В целом функциональность данных API приблизительно равна, как и 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:

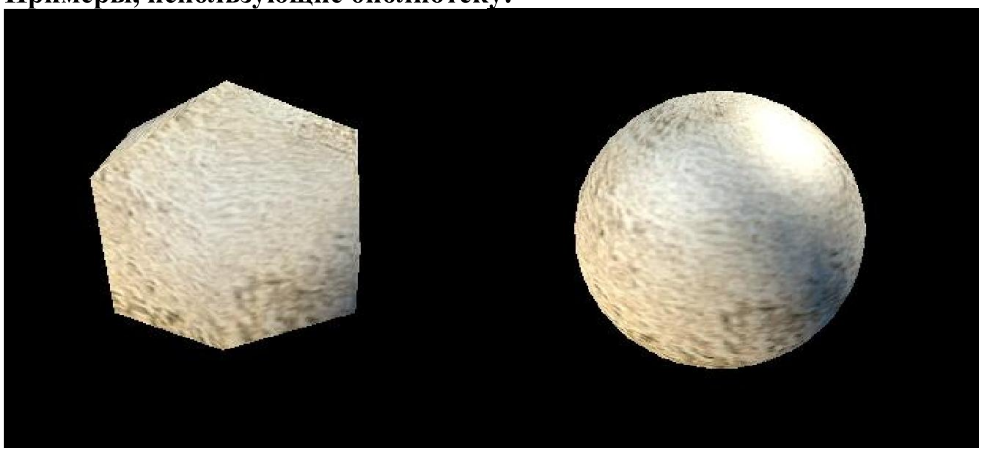
- Инициализация устройства (Видеокарты, а также заднего буфера (в него происходит рисование) и буфера глубины (позволяет предметам, которые ближе к камере скрывать те, которые находятся за ними)).
- Создание окна, в которое будет выводиться изображение.
- Инициализация ресурсов (Вершинных, пиксельных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

- Реализация освещения.
- Аппаратная тесселяция, т.е. разбиение каждого полигона в 3D-модели на несколько, что увеличивает качество её детализации.
- Реализация теней.

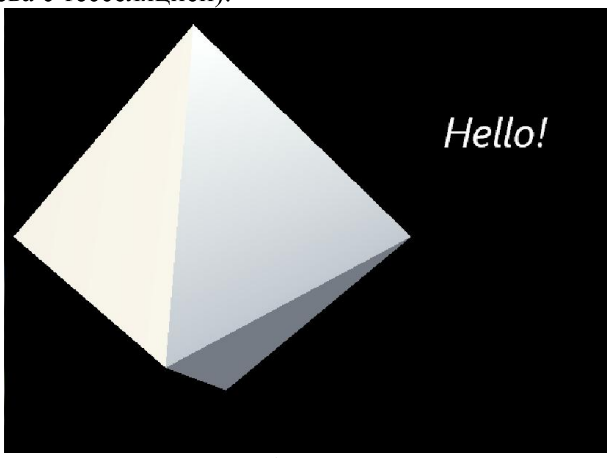
В качестве примеров прототипов некоторых функций библиотеки можно привести следующие:

- HRESULT InitDevice(UINT width, UINT height, bool Windowed) – Данная 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.
- HRESULT Cleanup() – производит очистку ресурсов и должна вызываться перед завершением работы приложения.
- HRESULT LoadSDKMesh(LPWSTR name) – Данная функция производит загрузку 3D-модели в программу. В качестве параметра передается имя модели.
- HRESULT LoadPS(LPWSTR name, const char_t funk, PixelShader** ps) – Данная 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.
- HRESULT InitCBuf(Buffer** buf,int i) – 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:



Пример использования аппаратной тесселяции на простой 3D-модели (Справа без тесселяции, слева с тесселяцией).



На данном изображении показан пример, выводящий на экран примитив и текст “Hello!”. Данный текст является динамическим, т.е. 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 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/%D0%98%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA

РАЗРАБОТКА ПРОГРАММЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ В УСЛОВИЯХ ПОЛНОЙ ОПРЕДЕЛЕННОСТИ

Автор: Кондратов Олег Сергеевич студент 4 курса

Руководитель: Черникова Лилия Валентиновна, преподаватель специальных дисциплин.

Образовательное учреждение: ГБОУ СПО МО Серпуховский технический колледж, г.Серпухов

Develop programs to support decision making under conditions of full certainty

Целью работы является разработка программы поддержки принятия решения в условиях полной определенности методом аддитивной оптимизации. Математические модели исследуемых явлений или процессов могут быть представлены таблицами, элементами которых являются значения частных критериев эффективности функционирования системы, вычисленные для каждой из сравниваемых стратегий при строго заданных внешних условиях. Принятие решений может производится по одному (однокритериальные задачи) или по нескольким критериям (многокритериальные задачи). Выбор решения задачи по одному критерию, как правило, не вызывает затруднений. Что же касается выбора решения по нескольким критериям, то тут применяется метод аддитивной оптимизации.

Пусть $f_i(a_{ij}) = \sum_{j=1}^{n} \gamma_j * a_{ij}, \ i = 1, m$ определяет аддитивный критерий оптимальности, где величины γ_j являются весовыми коэффициентами, которые определяют в количественной форме степень предпочтения (важность) j-го критерия оптимальности по сравнению с другими критериями. При этом более важному критерию приписывается больший вес, при этом: $\sum_{i=1}^{n} \gamma_j = 1, \ \gamma_j \geq 0, \ j = 1, n$

Обобщенная функция цели может быть использована для свертывания частных критериев оптимальности , если:

1) частные (локальные) критерии количественно соизмеримы по важности, т.е. каждому из них можно поставить в соответствие некоторое число γ_j , которое численно характеризует его важность по отношению к другим критериям;

2) частные критерии являются однородными (имеют одинаковую размерность).

Если локальные критерии неоднородны, то требуется провести их нормализацию – приведения к единому, безразмерному масштабу измерения. Один из вариантов нормализации рассмотрен ниже:

1. сначала определяется максимум каждого локального критерия, т.е.:

$a_j^+ = \max a_{ij}, \qquad i = 1, m$

2. выделяется группа критериев $a_j, i = 1, l$, которые максимизируются при решении задачи, и группа критериев $a_j, j = l + 1, n$, которые минимизируются при решении задачи;

3. если исходная задача на максимум, то для критериев, которые максимизируются, нормализованные критерии определяются по формулам:

$$A_{ij} = \frac{A_{ij}}{A_j^+}, j = 1, l,$$