О РАЗРАБОТКЕ ИНТЕГРИРОВАННОЙ СРЕДЫ ДЛЯ СОЗДАНИЯ 3D-ПРИЛОЖЕНИЙ

Г. А. Ломакин

Гродненский государственный университет имени Я. Купалы
Гродно, Беларусь
E-mail: spellbound.fpmi@gmail.com

Рассматриваются общие подходы к разработке интегрированной среды для создания 3D-приложений. Разработано графическое ядро и набор инструментов для работы с ним, набор классов и интерфейсов для интеграции ядра в конкретные разработки на платформе NET-приложений. Область применения – направление, связанное с визуализацией сложных графических объектов.

Ключевые слова: 3D-приложение, графическое ядро, фреймворк, контент, графический объект, Android, графический интерфейс пользователя.

Введение

В настоящее время значительный научный и практический интерес представляют задачи создания виртуальных реальностей, которые отождествляют реальные прототипы. Как правило, подход к решению такого рода задач требует интерактивной экранизации.

Экранизация реальных моделей является достаточно сложной задачей. Для экранной визуализации синтезированной 3D-модели применяются различные алгоритмы, выбор, адаптация и модификация которых зависит от рассматриваемой задачи предметной области. Более того, чрезмерная трудоемкость создания объектов реального мира приводит к необходимости разработки специальных моделей и новых методов экранизации для таких объектов.

Отметим, что при современном развитии графических систем и технологий актуален подход, связанный с переводом визуализации 3D-объектов на новый уровень. Разработка 3D-приложений характеризуется определенным набором ограничений, что сказывается на результатах визуализации.

Наиболее часто разработчики 3D-приложений обращаются к двум основным библиотекам 3D-графики: DirectX и OpenGL. Однако работа с ними вызывает трудности, связанные с написанием многих строк кода, использованием API и указателей. Часть проблем по реализации 3D-объектов можно решить с использованием XNA framework, используя соответствующие классы. Однако XNA не позволяет полностью устранить проблемы при разработке 3D-приложений, так как указанный framework является NET реализацией DirectX.

С другой стороны, широкое распространение и развитие мобильных платформ также предоставляет необходимые возможности для разработчиков, выполняя такой широкий спектр компьютерных задач общего профиля. На сегодня для пользователей имеется широкий выбор мобильных телефонов под управлением ОС Android, которая предлагает неплохую альтернативу: позволяет писать приложения, использующие весь спектр современного аппаратного обеспечения, предоставляет новые возможности по разработке и использованию инструментальных средств.

Таким образом, разработка среды, которая предоставит пользователю возможность быстрого создания 3D-объектов, используя современные интернет- и мобильные технологии, востребованное направление в развитии компьютерной графики.

Краткая характеристика интегрированной среды

Следует отметить, что существует небольшое количество доступных сред для создания интерактивных 3D-визуализаций, к недостаткам которых можно отнести отсутствие хранилища различного 3D-контента в облаке и возможностей, предоставляющих пользователю высокоуровневый подход для работы с 3D-пространством и различными объектами.

Можно определить следующие основные требования, связанные с разработкой: определение конкретных рамок и цели создаваемого графического ядра; обеспечение доступа к контенту; реализация набора программ-утилит для использования контента и графического ядра под различные платформы; реализация фреймворка с высоким уровнем абстракций; обеспечение широкого выбора различных алгоритмов для рендеринга изображения.

Предлагаемая интерактивная среда состоит из нескольких компонентов: облака для хранения контента в различных категориях; графического ядра, реализованного с использованием OpenGL и DirectX; набора утилит для персонального компьютера; приложения для мобильной операционной системы Android; фреймворка, обеспечивающего доступ ко всем возможностям ядра.

Таким образом, главными особенностями веб-среды являются: открытое ядро на OpenGL и DirectX; набор классов и интерфейсов для рендеринга примитивов, набор базовых шейдеров для реализации различных эффектов на графическом конвейере; утилиты для построения визуализаций на платформах PC и Android; синхронизация контента клиентского приложения с сервером.

Рассмотрим основные аспекты, связанные с архитектурной и программной реализацией, а также – с основными конструктивными особенностями предлагаемой интегрированной среды разработки 3D-приложений.

Архитектура интегрированной среды

Предлагаемая интегрированная среда абстрагирует пользователя от промежуточных уровней создания графики и дает возможность напрямую создавать графические сцены посредством GUI, в том числе обеспечивая пользователя шаблонным контентом [1].

Были реализованы следующие модули [2]: ядро для построения сложных графических сцен, предоставляющее открытое API; графическое ядро на OpenGL для ОС Windows 7 и OpenGL ES 2.0 для ОС Android; набор утилит, использующих ядро для манипулирования графическими процессами с помощью GUI; сервер с контентом для предоставления пользователю возможности использования шаблонного наполнения, а так же загрузки пользователем произвольного контента.

В общем виде архитектура интегрированной среды представлена на рис. 1.

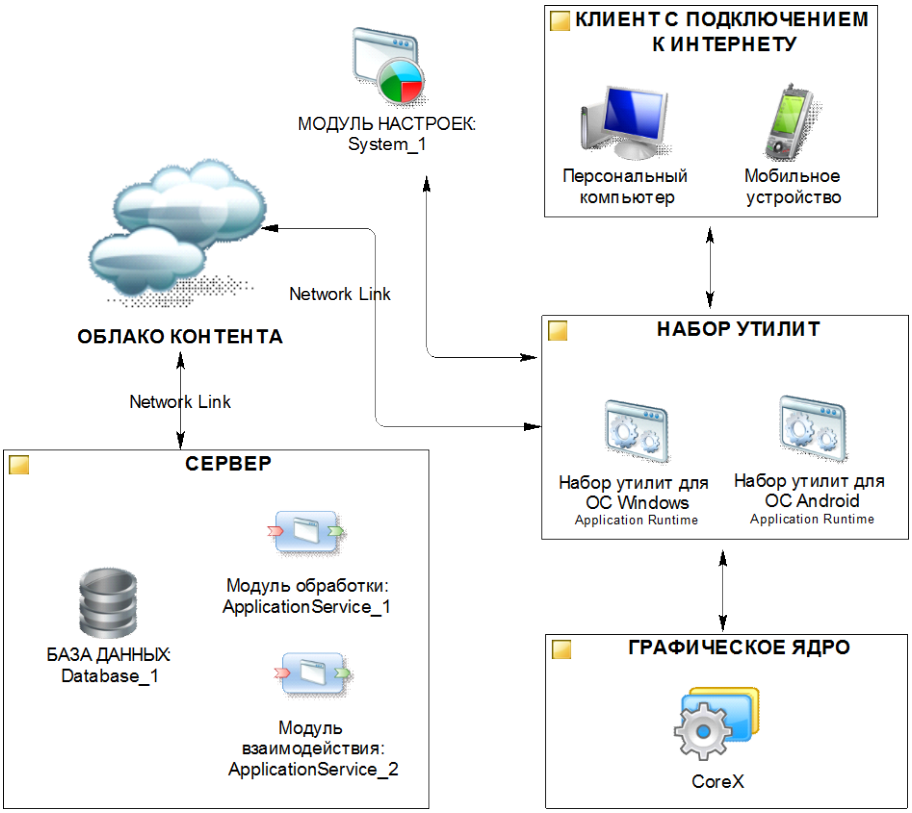

Рис. 1. Архитектура интегрированной веб-среды для построения графических приложений

Интерфейс взаимодействия пользователя с ядром

Для визуализации сложных трехмерных объектов реального и виртуального мира используется разработанное графическое ядро CoreX, базирующееся на XNA framework, которое включает Content Pipeline. Это упрощает работу с контентом и, при необходимости, позволяет расширить список поддерживаемых типов.

Отметим, что графические объекты представляются в качестве классов. Класс каждого графического объекта описывает его основные свойства, а также методы для отображения объекта в пространстве и обновления его состояния. Каждый класс связывается с определенным контентом, в нем же определяется, как данный контент будет использоваться.

На базе ядра CoreX реализован и графический интерфейс пользователя (GUI), позволяющий производить интеграцию различных элементов управления в сцене и манипулировать происходящим на ней. Кроме графической составляющей в CoreX присутствует звук. Графическое ядро CoreX поддерживает DirectX 10, а также шейдерную модель версии 4.0. Реализованы эффекты per-Pixel lighting, bump mapping, parallax mapping, shadow mapping, shadow volumes, multisampling.

Реализованное ядро предоставляет API для построения приложений. Разработанное API позволяет строить и изменять сцену, добавлять новые объекты для рендеринга и изменять их параметры (положение в пространстве, коэффициенты масштабирования, углы наклона относительно осей, текстуры, шейдер и др.). Кроме этого есть возможность применения глобальных эффектов ко всей сцене (таких как глобальное освещение/затенение, motion blur, HDR и др.).

Пользователь взаимодействует с ядром при помощи универсальной утилиты. Утилита состоит из двух основных частей. Первая часть отвечает за контент и объекты, использующие этот контент, за свойства и различные характеристики этих объектов. Вторая часть отвечает за настройку ядра CoreX: детализацию, качество текстур, используемые технологии и др. После конфигурации, все данные передаются на ядро, после чего запускается визуализация.

При задании параметров объектов имеется возможность предварительного просмотра и взаимодействия с объектами и их свойствами посредством работы в окне предварительного просмотра. Таким образом, пользователь всегда видит, где он размещает объект, как он расположен относительно других объектов и др.

Серверная часть

Для предоставления пользователю шаблонных ресурсов разработано дистанционное хранилище контента. Пользователю доступна возможность загрузки на сервер ресурсов для общего использования. Доступ к серверу осуществляется как из утилит, так и с помощью веб-интерфейса, включая мобильные платформы.

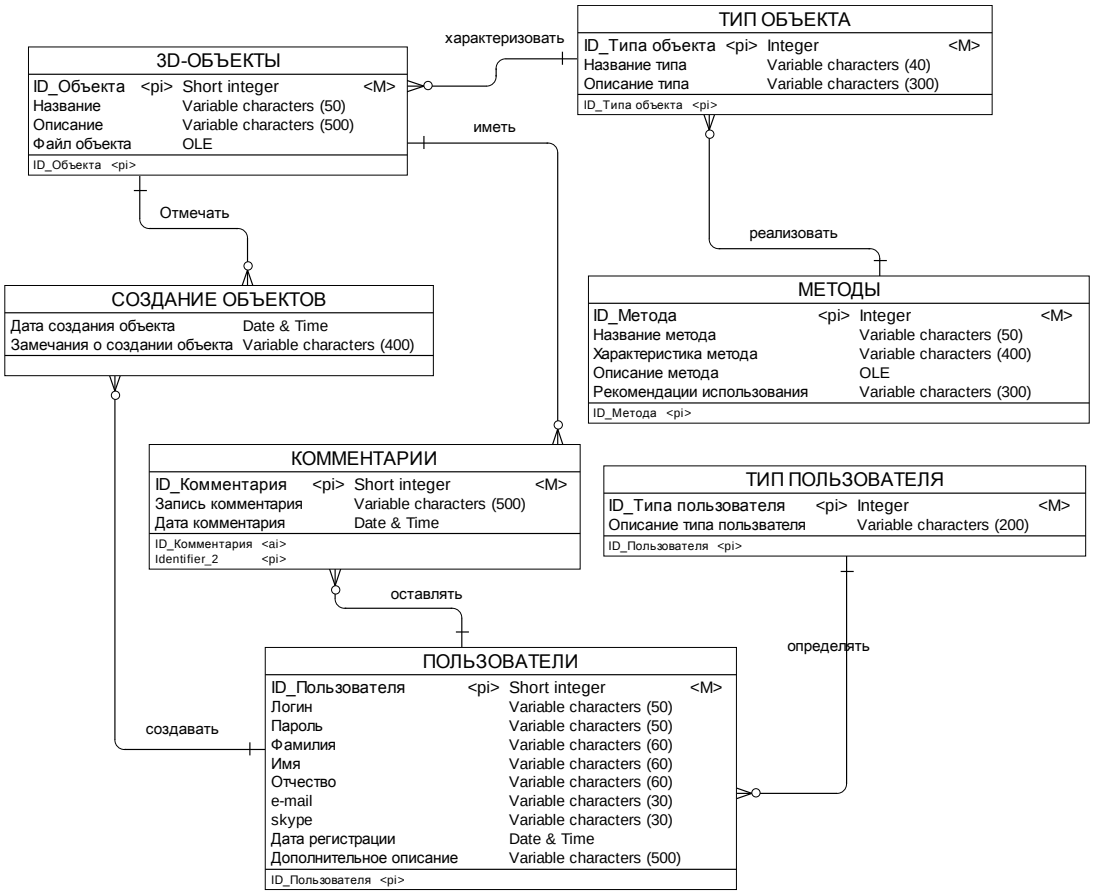

Рис. 2. Фрагмент концептуальной модели базы данных, предназначенный для создания и хранения 3D-объектов

На рис. 2 представлен фрагмент модели базы данных, предназначенный для создания и хранения 3D-объектов. Каждый объект, который помещается в базу, харак-

теризуется типом и методом его реализации. Пользователи, которые добавляют или же используют соответствующий объект, могут добавить необходимые комментарии по поводу использования, методов реализации, характеристик объекта. В зависимости от типа пользователям назначаются соответствующие привилегии по работе с 3D-объектами. Отметим, что в модели данных поддерживается декларативным ссылочная целостность.

Основные выводы

Таким образом, разработана интегрированная среда, которая предоставляет пользователям возможности по формированию сцены, приданию различным объектам требуемых физических свойств, а также заданию необходимых взаимодействий объектов. Все взаимодействия в среде осуществляются посредством графического интерфейса.

Дальнейшая разработка системы предполагает добавление нового контента на сервер, расширение графических возможностей, перенос системы на другие платформы (например, Unix-системы). Несомненно, разработка такого рода найдет широкое применение как в научных, так в игровых и промышленных визуализациях.

Библиографические ссылки

1. *Мэрдок Келли Л.* Autodesk 3ds Max 2009. 3D Studio max. Библия пользователя = Autodesk 3ds Max 2009 Bible. 3D Studio max. М. : Диалектика, 2009.
2. *Ломакин Г. А.* Об одном подходе к построению графических приложений // Наука-2011 : сб науч. ст. В 2 ч. Ч. 2. Минск : БГУ, 2011. С. 89–91.