О КОНЦЕПЦИИ МОБИЛЬНОЙ ПЛАТФОРМЫ 3D-ВИЗУАЛИЗАЦИЙ

Л. В. Рудикова[1], Г. А. Ломакин[2]

[1]*Гродненский государственный университет имени Янки Купалы*
Гродно, Беларусь
[2]*Белорусский государственный университет информатики и радиоэлектроники*
Минск, Беларусь

e-mail: rudikowa@gmail.com, spellbound.fpmi@gmail.com

Рассматривается общая концепция интегрированной платформы, которая предназначена для создания визуальных приложений посредством доступа с мобильных клиентов. Дается обобщение подходов, используемых при построении 3D-графики, абстрагирование конечного пользователя от таких проблем как поиск контента, написание графического ядра и т. д. Описывается общая архитектура разрабатываемой платформы.

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

ABOUT CONCEPT MOBILE PLATFORM 3D-VISUALIZATION

L. V. Rudikova[1], G. A. Lomakin[2]

[1]*Grodno State Yanka Kupala University*
Grodno, Belarus,
[2]*Belarussian State University of Informatics and Radioelectronics*
Minsk, Belarus

This article describes general concept of integrated platform, which intended for create visual applications through access with mobile clients. Besides, it describes generalization approaches for build 3D-graphic, abstraction user from some problems: content search, writing graphic core and etc. Describes general architecture developing platform.

Keywords: 3D-application; graphic core; framework; content; graphic object; Android; user interface.

ВВЕДЕНИЕ

Экранизация реальных моделей – достаточно сложная задача. Для экранной визуализации синтезированной 3D-модели используются различные алгоритмы, выбор, адаптация и модификация которых зависит от рассматриваемой задачи предметной области.

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

С другой стороны, широкое распространение и развитие мобильных платформ также предоставляет необходимые возможности для разработчиков. Мобильные телефоны уже давно используются не только для разговоров. Они стали выполнять такой широкий спектр компьютерных задач общего профиля, что, вероятнее всего, такие устройства могут стать новым поколением персональных компьютеров.

Для пользователей имеется широкий выбор мобильных телефонов под управлением Android. В отличие от большинства мобильных систем, закрывающих и ограничивающих разработку, а также развертывание сторонних приложений, Android предлагает альтернативу: позволяет писать приложения, использующие весь спектр современного аппаратного обеспечения, и предоставлять 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, – наиболее востребованное направление в развитии компьютерной графики.

МОДУЛЬНАЯ МОДЕЛЬ УНИВЕРСАЛЬНОЙ ПЛАТФОРМЫ

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.

Проанализировав предметную область, функциональность предлагаемой платформы и учитывая модульную структуру при разработке универсальной платформы для создания 3D-приложений, были выделены отдельные блоки-модули (рис. 1).

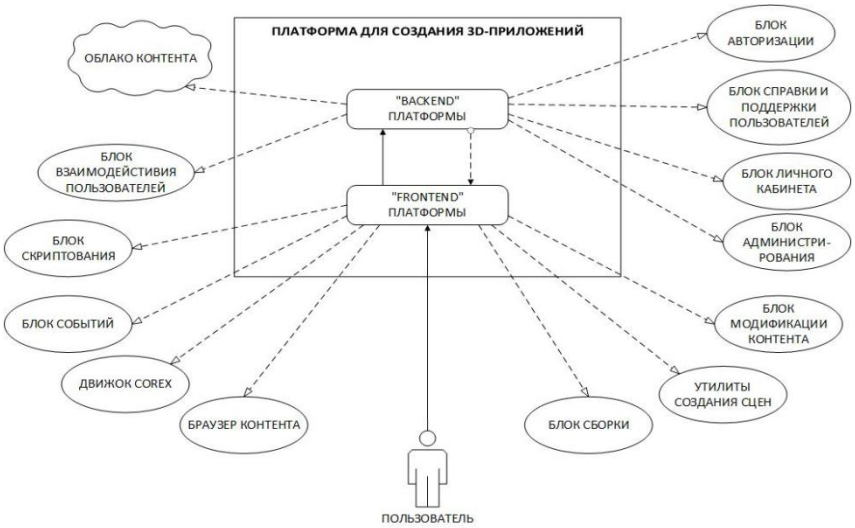

Рис. 1. Блочно-модульная схема платформы

Блок авторизации. Отвечает за авторизацию пользователей в системе. В данном блоке также определяется роль пользователя и назначаются права. Здесь наблюдается частичная интеграция с блоком личного кабинета.

Блок личного кабинета. В данном блоке пользователь может изменять личные данные.

Блок взаимодействия пользователей. Блок включает в себя обсуждение опубликованного контента.

Блок администрирования. В данном блоке предусмотрена настройка контента. Добавление, удаление, редактирование контента. Учет авторских прав.

Облако контента. Содержит весь добавленный пользователями контент в виде текстуры, модели, звука, музыки и сцены.

Блок справки и поддержки пользователей. Блок поддержки пользователей обеспечивает: возможность двухстороннего общения с пользователями приложений портала, анализ пользовательских предпочтений, исправление проявившихся ошибок или недостоверностей. Предлагаемый модуль организован в достаточно простой для восприятия и использования форме. Воспользовавшись центром поддержки, пользователь может отправить на рассмотрение свой вопрос, замечание или предложение. После этого ответственные за определенные типы обращений люди принимаются за решение возникших вопросов.

Блок скриптования. Описание поведения объекта на сцене.

Блок событий. Связан с блоком скриптования. Реагирование и делегирование событий. Обработка системных событий (нехватка памяти, заряд батареи и т. д.).

Движок GOREX. Отрисовка всех графических объектов в режиме реального времени.

Браузер контента. Позволяет получать контент с сервера, а также добавлять локальный контент и размещать его (при необходимости) на сервере.

Утилиты создания сцен. Основной инструмент взаимодействия пользователя с системой позволяет получать доступ как к сцене, так и ко всем остальным компонентам системы (класс-контроллер).

Блок сборки. Упаковка готовой сцены в независимый объект.

Блок модификации контента. Позволяет комбинировать уже существующий контент для создания нового контента.

Для реализации платформы выбраны следующие средства: программный стек Android, виртуальная машина Dalvik, СУБД SQLite, язык программирования Java и среда разработки Eclipse.

ОБЩИЕ КОНЦЕПЦИИ РЕАЛИЗАЦИИ УТИЛИТ ДЛЯ ПЛАТФОРМЫ ВИЗУАЛИЗАЦИИ 3D-ОБЪЕКТОВ

Охарактеризуем также основные утилиты разрабатываемой платформы.

ContentNavigator – утилита, которая позволяет импортировать в движок контент такого типа, как 3D-модели, анимация для них, текстуры, звуки. Кроме того, для каждого вида контента должна быть реализована возможность просмотреть(прослушать), а также изменить базовые свойства. Для 3D-моделей такими свойствами являются масштаб, сдвиг, поворот, материал, тип объекта (статический или нет). Для звука – скорость воспроизведения. Для текстур – сдвиг, масштаб. Также для утилиты ContentNavigator важно иметь базовые редакторы для типов контента. Например, возможность удалять ненужную анимацию у моделей, добавлять Socket в объект. Socket-ом

называется позиция и ориентация в 3D-пространстве относительно модели, в которую может быть добавлена другая модель. Особенность – привязка сокета к скелету модели, таким образом, при анимировании сокет и интегрированная в него модель будут двигаться зависимо друг от друга. Кроме этого, в утилите должен быть редактор материалов, позволяющий комбинировать текстуры, задавать карты нормалей, задавать атрибуты затенения или блеска и другие эффекты.

LandscapeBuilder – утилита для генерации ландшафтов с помощью карты высот. Такая утилита строит по заданным свойствам, таким как размер ландшафта и коэффициент высоты и, собственно, карты высот, mesh-содержащий ландшафт. В утилите также корректно реализован расчет нормалей и текстурных координат для каждой из вершин.

GlobalLight – утилита, которая отвечает за глобальное освещение и эффекты. Содержит настройку эффектов освещения, глубину, вектор направления и силу света, настройка тумана – дальность, цвет, различные postprocess-эффекты, такие как blur, motionblur, HDR и др.

AnimationManager – утилита для управления анимацией объектов; позволяет просматривать каждую анимацию и именовать для дальнейшего использования в скриптах.

LequidManager – утилита для симуляции жидкостей. Содержит настройки материала жидкости, такие как базовый материал, коэффициенты преломления света, прозрачность и искажение цвета света; также возможность задавать симуляцию ветра, а именно: сила ветра, размер волн (зависит от плотности жидкости).

Кроме того, в предлагаемой платформе при реализации графического ядра была добавлена скелетная анимация, т. е. механизм, позволяющий анимировать не каждую вершину объекта, а конкретные части в зависимости от расположения других частей.

Отметим также, что процесс анимации включает следующие этапы.

1. Прохождение по ключам анимации и нахождение двух ключей, между которыми расположено искомое время. Для каждой кости в матрице Release записывается интерполированное значение.

2. После заполнения Release матрицы всех костей обновляется вся иерархия.

3. Домножение Release на матрицу из bind pose и сохранение ее в релизный массив.

4. На уровне шейдеров полученная матрица используется для преобразования вершин и нормалей.

ЗАКЛЮЧЕНИЕ

Предлагаемая универсальная платформа (рис. 2) является альтернативой существующим средам и фреймворкам [1]. В структуре движка заложена архитектура и нейминги, которые позволяют перенести среду на другую платформу или даже язык программирования с минимальными изменениями. Кроме этого, существует возможность расширения предлагаемого продукта сторонними разработчиками и выпуск некоторых плагинов и сборок.

Предлагаемая разработка может быть полезна во многих сферах деятельности, связанных с визуализацией. Например, в образовательных или исследовательских целях – визуализация физических, химических и других процессов; в сфере развлечений и маркетинга – реклама и компьютерные игры, а также в промышленной разработке

программного обеспечения – использование фреймворка для написания коммерческих приложений.

Рис. 2. Примеры загруженных 3D-моделей и сгенерированного ландшафта

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1. Ломакин Г. А., Гусейнова А. С. Общие подходы к разработке интегрированной среды для построения графических приложений // Вест. БГУ. Сер. 1. 2013. С. 54–60.