МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент
генеральный директор
ООО «СТ»

_____ Дербенцев К.С.

“___”_____ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,
д.ф.-м.н., профессор

_____ Л.Б. Соколинский

“___”_____ 2018 г.

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ВИЗУАЛИЗАЦИИ ПРОЕКТОВ СТРОИТЕЛЬНОЙ КОМПАНИИ С ИСПОЛЬЗОВАНИЕМ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2018.115-035.ВКР

Научный руководитель
д.ф.-м.н., доцент
_____ Г.И. Радченко

Автор работы,
студент группы КЭ-402
_____ Е.В. Мовчан

Ученый секретарь
(нормоконтролер)

_____ О.Н. Иванова

“___”_____ 2018 г.

Челябинск-2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 4
1. ОБЗОР ИСТОЧНИКОВ ПО ТЕМЕ РАБОТЫ 6
1.1. Инструменты для работы с дополненной реальностью 6
1.1.1. Tango 7
1.1.2. Vuforia 6
1.1.3. ARCore 8
1.1.4. ARKit 9
1.2. Обзор существующих проектов с использованием AR 10
2. АНАЛИЗ ТРЕБОВАНИЙ К ПРИЛОЖЕНИЮ 13
2.1. Функциональные требования 13
2.2. Нефункциональные требования 13
2.3. Диаграмма вариантов использования 14
3. АРХИТЕКТУРА СИСТЕМЫ 16
3.1. Описание компонентов, составляющих систему 16
3.2. Диаграмма классов анализа 17
3.3. Диаграмма классов 17
3.4. Диаграмма последовательности 19
4. РЕАЛИЗАЦИЯ 21
4.1. Структура проекта 21
4.2. Используемые технологии и библиотеки 21
4.3. Реализация графического интерфейса приложения 22
4.4. Реализация приложения 24
5. ТЕСТИРОВАНИЕ 26
ЗАКЛЮЧЕНИЕ 29
СПИСОК ЛИТЕРАТУРЫ 30

ВВЕДЕНИЕ

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Дополненная реальность (Augmented Reality, AR) — результат введения в поле восприятия любых сенсорных данных с целью дополнения сведений об окружении и улучшения восприятия информации AR.

Окклюзия — ситуация, в которой два объекта расположены на одной линии и один объект, расположенный ближе к виртуальной камере, частично или полностью закрывает видимость другого объекта.

Пользователь — человек, использующий приложение для предварительного просмотра еще не построившегося строительного проекта.

АКТУАЛЬНОСТЬ

В настоящее время технологии виртуальной реальности и инструменты дополненной реальности становятся все лучше и более оптимизированными. По мере того, как доступность дополненной реальности увеличивается, находится все меньше и меньше причин, почему каждый архитектор, инженер и подрядчик не должны использовать какую-либо форму дополненной реальности для воплощения своих проектов в жизнь.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Целью данной работы является разработка Android-приложения, которое помогает дизайнерам визуализировать свои проекты, используя дополненную реальность, чтобы подробно посматривать созданные архитектурные проекты.

Для разработки приложения необходимо решить следующие задачи:

1) выполнить анализ предметной области;
2) изучить существующие платформы и проекты, связанные с дополненной реальностью;
3) спроектировать архитектуру приложения;
4) реализовать мобильное приложение;
5) провести тестирование.

Разрабатываемое приложение позволяет пользователю загрузить модель, перемещать модель в пространстве и изменять ее размеры. Должна присутствовать возможность загружать несколько объектов и впоследствии выбирать объект, который требуется изменить.

СОДЕРЖАНИЕ РАБОТЫ

Первая глава «Анализ предметной области» содержит постановку задачи и обзор аналогичных проектов.

Вторая глава «Проектирование системы» содержит описание и анализ требований к разрабатываемому мобильному приложению, описание архитектуры приложения.

Третья глава «Реализация системы» описывает подробности реализации мобильного приложения для платформы Android.

Четвертая глава «Тестирование системы» посвящена результатам тестирования мобильного приложения для платформы Android.

В заключении описываются основные результаты, полученные при выполнении дипломной работы.

1. ОБЗОР ИСТОЧНИКОВ ПО ТЕМЕ РАБОТЫ

В данном разделе рассмотрены инструменты для работы с дополненной реальностью и существующие проекты.

1.1. Инструменты для работы с дополненной реальностью

Дополненная реальность позволяет пользователю увидеть реальный мир с наложенными виртуальными объектами на объекты реального мира. AR дополняет реальность, а не полностью заменяет ее. Для пользователя это выглядит таким образом, что виртуальные и реальные объекты сосуществуют в одном и том же пространстве [13].

Одной из важных проблем использования дополненной реальности в мобильных устройствах, низкая мощность телефонов и планшетов в отличии от стационарных компьютеров и ноутбуков [2].

Рассмотрим наиболее популярные инструменты для работы с дополненной реальностью.

1.1.2. Vuforia

Vuforia — это платформа дополненной реальности и инструментарий разработчика программного обеспечения дополненной реальности для мобильных устройств, разработанные компании Qualcomm. Vuforia использует технологии компьютерного зрения, а также отслеживания плоских изображений и простых объемных реальных объект [10].

Возможность определения мишеней позволяет разработчикам располагать и ориентировать виртуальные объекты, такие, как 3D-модели и медиаконтент. Виртуальный объект ориентируется на реальном образе так, чтобы точка зрения наблюдателя относилась к ним одинаковым образом для достижения главного эффекта — ощущения, что виртуальный объект является частью реального мира.

Мишень — это некий реальный объект, зная который, приложение расставляет виртуальные объекты в нужных местах и соответствующих пропорциях.

Vuforia предоставляет следующие типы мишеней:

1) image targets — базовый вид мишеней, представляющий собой обычное изображение, например, обложку журнала, фотографию или афишу нового фильма. Изображение выполняет роль своего рода двумерного штрих-кода, только без черно-белых регионов. Хорошими мишенями являются те изображения, в которых много контрастных деталей. Именно на этих деталях и строится опорная матрица для последующего распознания мишеней;

2) simple 3D targets — мишени в виде прямоугольных параллелепипедов, включая куб. Такой мишенью может служить любая коробка, состоящая из шести плоскостей;

3) cylinder targets — данный вид мишеней, представляет собой усеченный конус с возможностью задавать диаметры оснований. Для того, чтобы создать такую мишень требуется не только указать диаметры оснований и высоту, но также добавить три изображения — по одному для каждого из двух оснований и еще одну для боковой поверхности;

4) frame markers — мишень в виде специально подготовленной рамки с изображением. В такую рамку можно поместить любое изображение. Данный вид мишеней отлично подходит в случае, если изображение не было достаточно детализировано и создать из него image target не получилось;

5) word targets — мишень в виде текста. На данный момент поддерживается только латиница.

1.1.1. Tango

Tango — платформа для работы с дополненной реальностью, разработанная компанией Google. Технология применяет компьютерное зрение для мобильных устройств, таких как смартфоны и планшетные компьютеры, для определения своего положения относительно окружающего мира без использования GPS или других внешних сигналов. Это позволяет разработчикам приложений создавать системы, включающие в себя внутреннюю

навигацию, 3D-картографирование, распознавание окружающей среды и использование дополненной реальности [7].

Платформа Tango подразделяется на три технологии:

1) отслеживание движений (Motion tracking) — технология, позволяющая устройству вычислять позицию, ориентацию и движение в трехмерном пространстве. Устройство, используя акселерометр и гироскоп, выдает координаты и вектор движения;

2) изучение области (Area Learning) — технология, позволяющая устройству использовать визуальные подсказки для распознавания мира вокруг, самостоятельно исправляя ошибки в отслеживании движений и переопределяя себя в тех местах, где оно уже было. Устройство распознает место и корректирует движения на основе этого;

3) восприятие глубины (Depth Perception) — технология, позволяющая устройству получать облако точек для текущей области прстранства. Для построения облака точек сенсоры определяют расстояние до предметов, размеры и поверхности окружающей среде.

Платформа Tango поддерживается только на устройствах, оснащенных специальными сенсорами [7].

1.1.3. ARCore

ARCore — это платформа Google для создания приложений дополненной реальности. ARCore позволяет поддерживать такие платформы как Android и iOS [3].

ARCore использует три ключевых возможности для интеграции виртуального контента с реальным миром:

1) light estimation — технология для обнаружения информации об освещении окружающей среды и определения средней интенсивности и коррекции цвета для данного изображения камеры. Полученная информация позволяет освещать виртуальные объекты в тех же условиях, что и окружающая среда, увеличивая ощущение реализма.

2) motion tracking — технология использующая процесс, называемый параллельной одометрией и сопоставлением, для чтобы понять, где находится устройство относительно мира вокруг него. ARCore обнаруживает визуально различные точки в захваченном изображении камеры, называемые feature point (особенные точки), и использует эти точки для вычисления изменения местоположения. Визуальная информация объединяется с измерениями из инерционного измерительного устройства для оценки положения и ориентации камеры относительно мира во времени;

3) environmental understanding — технология для поиска кластеров feature point, которые лежат на общих горизонтальных или вертикальных поверхностях. ARCore также определяет границу каждой плоскости и передает эту информацию приложению;

В данный момент времени ARCore поддерживает только несколько устройств с версией Android 7.0 и выше.

ARCore поддерживает различные средства разработки такие как UnrealEngine, Unity и AndroidStudio. Благодаря различным плагинам и SDK существует возможность разработки под различные операционные системы, а также при web-разработке.

1.1.4. ARKit

ARKit — это мобильная платформа AR для разработки приложений дополненной реальности на iOS.

ARKit сочетает в себе систему отслеживания движения устройства, захват сцены камеры, расширенную обработку сцены и отображения объектов, чтобы упростить задачу создания приложений с дополненной реальностью [4].

Благодаря ARKit SDK порог вхождения в эту технологию стал значительно ниже. поддерживают эту технологию.

ARKit будет работать только на устройствах с процессором A9 и A10, под операционной системой iOS

1.2. Обзор существующих проектов с использованием AR

По мере того как дополненная реальность набирает популярность, создается все больше и больше проектов использующих дополненную реальность в различных сферах. В данном разделе будут рассмотрены наиболее популярные приложения с использованием дополненной реальности.

1.2.1 URBASEE Future

URBASEE Future — проект французской компании Artefacto, основанной двумя преподавателями-архитекторами. Приложение позволяет просматривать проекты в режиме реального времени в масштабе один к одному с помощью смартфона или планшета. Для стабильной работы приложения требуется хорошее соединение с сервером для загрузки модели и стабильный сигнал GPS для установки объекта. Для работы с приложением пользователю требуется загрузить макет здания на сервер и привязать его к геолокации. Когда пользователь окажется на выбранном им месте, программа синхронизируется с сервером и отобразит проект в масштабе один к одному [9].

Значительными недостатком приложения является необходимость стабильного интернета для загрузки макета и отсутствие окклюзии.

К минусам данного приложения можно отнести следующее:

- требуется стабильное подключение к серверу для загрузки модели с сервера;
- требуется стабильное подключение к сети GPS для получения координат устройства;
- отсутствие окклюзии, реальные объекты находящиеся ближе к камере чем виртуальные не отображаются на экране.

К плюсам данного приложения можно отнести следующее:

- Не требуется дополнительная настройка местоположения объекта, координаты вычисляются с помощью GPS;
- Возможность редактирования собственных моделей на сервере.

1.2.2. GAMMA BIM

GAMMA BIM — это проект немецкой компании Formitas. Приложение визуализирует проекты строительной компании на будущее место постройки. Данное приложение разработано с использованием платформы Tango. Для работы с приложением пользователю требуется загрузить макет здания на сервер. Когда пользователь окажется на месте постройки здания он может загрузить объект в масштабе один к одному [5].

Основным минусом данного приложения можно отнести требование стабильного подключения к серверу для загрузки модели.

К плюсам данного приложения можно отнести следующее:

- точность позиционирования модели;
- требуется стабильное подключение к серверу;
- стабильное определение окружающей поверхности с помощью облака точек.

1.2.3. Roomle 3D

Приложение Roomle 3D помогает пользователю выбрать мебель из каталога и с помощью технологии дополненной реальности установить в реальном мире. Roomle 3D поддерживает множество платформ и такие технологии как ARCore, ARKit и Tango [6].

Основным минусом данного приложения является отсутствие возможности загрузки собственной модели строительной компании с устройства и или сервера;

К плюсам данного приложения можно отнести следующее:

- точность позиционирования модели;
- поддержка различных платформ и различных операционных систем.

Вывод

Из рассмотренных проектов наиболее актуальным и новым на сегодняшний день является GAMMA BIM. Поскольку GAMMA BIM позволяет

загружать собственную модель здания без сигнала GPS, используется платформа Tango. Используя данную платформу, устройство точно позиционирует себя в пространстве и обеспечивает стабильную работу.

Обзор аналогов показал, что в первую очередь стоит обратить внимание на возможность приложения загружать объект независимо от наличия интернет подключения, а также на управление и трансформацию макета непосредственно через устройство. Пользователь должен без подключения к серверу загружать макеты и перемещать их так, как сочтет нужным. Приложение должно обладать интуитивно понятным интерфейсом.

В разрабатываемом приложении должна быть высокая точность позиционирования объектов на местности.

Кроме того, приложение должно обладать удобным и интуитивно понятным интерфейсом и дизайном, обеспечивающим комфортную работу с приложением.

2. АНАЛИЗ ТРЕБОВАНИЙ К ПРИЛОЖЕНИЮ

Создаваемое приложение предназначено приложения для визуализации проектов строительной компании с использованием дополненной реальности. Анализ предметной области позволил определить следующие функциональные требования.

2.1. Функциональные требования

Разрабатываемое приложение для визуализации строительных объектов должна удовлетворять следующим функциональным требованиям:

1) приложение должно предоставлять пользователю возможность загружать объекты различных форматов;

2) приложение должно использовать технологию дополненной реальности;

3) приложение должно обеспечивать стабильную работу и определение позиции устройства в пространстве;

4) приложение должно позволять пользователю загружать объекты без подключения к интернету;

5) приложение должно предоставлять пользователю возможность управлять объектом, изменять его положение в пространстве и угол поворота.

2.2. Нефункциональные требования

В результате анализа функциональных требований и обзора аналогичных проектов, были сформулированы следующие нефункциональные требования:

1) приложение должно быть написано на языке C# под платформу Android;

2) приложения должно быть разработано на платформе Tango;

3) приложения должно быть разработано с помощью межплатформенной среды разработки компьютерных игр Unity.

2.3. Диаграмма вариантов использования

В ходе анализа проектируемого приложения были выявлены основные варианты использования, которые представлены на рис. 1.

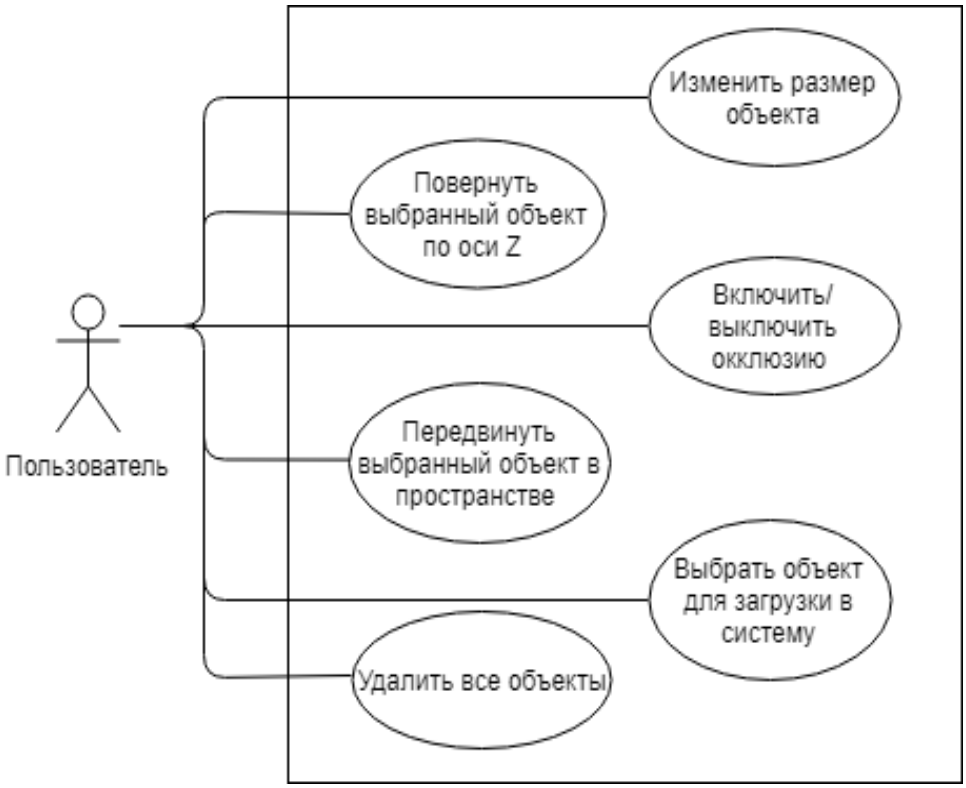

Рис. 1. Диаграмма вариантов использования

В системе определено действующее лицо: пользователь. Пользователь системы имеет следующие варианты использования:

1) повернуть выбранный объект по оси Z — поворачивает загруженный объект для правильной установки его в реальном мире;

2) удалить все объекты — удаляет все загруженные модели со сцены;

3) включить/выключить окклюзию — включает или отключает возможность показа объектов реального мира, которые находятся ближе к камере чем виртуальная загруженная модель;

4) изменить размер объекта — изменяет размер загруженного объекта для правильного отображения в пространстве. Данная возможность предусмотрена по причине того, что различные дизайнеры-архитекторы могут использовать различные размеры в 3D-редакторах;

5) передвинуть выбранный объект в пространстве — перемещает загруженный объект для корректировки местоположения и правильной установки в реально мире;

6) выбрать объект для загрузки в систему — загружает и устанавливает выбранную модель на плоскость, на которую указывает камера устройства.

Вывод

Были сформированы функциональные и нефункциональные требования. На основе требований была спроектирована диаграмма вариантов использования.

3. АРХИТЕКТУРА СИСТЕМЫ

Приложение для визуализации проектов строительной компании с использованием дополненной реальности в самом общем виде состоит из:

- класса загрузки объектов, использующего Tango SDK для работы с дополненной реальностью; [7]
- библиотеки TriLib, использующаяся для загрузки модели с устройства пользователя; [8]
- пользовательского интерфейса.

3.1. Описание компонентов, составляющих систему

Разберем отдельные компоненты, составляющие приложение.

1. Пользовательский интерфейс (UI) — оболочка приложения для удобного взаимодействия с пользователем.

2. Класс SceneManager — управляющий класс, предназначенный для управления пользовательским интерфейсом. Открывает дополнительные кнопки и окна в пользовательском интерфейсе для расширенного управления объектом.

3. Модель строительной компании (Model) — класс-сущность, набор данных, представленных в виде текстур и файла набора вершин и путей текстур. Файл загружается на устройство пользователем из различных 3D-редакторов.

4. Класс ModelEditor — управляющий класс, предназначенный для установки модели, загруженной с помощью библиотеки TriLib, в сцену проекта. Данный класс вызывает функцию загрузки объекта и текстур в программу, после чего получает координаты точки, на которую указывает камера, и устанавливает новый объект на сцену с полученными координатами.

5. Библиотека TriLib — управляющий класс, получает из класса ModelEditor путь к загружаемой модели, загружает модель с устройства. Из загруженной модели получает список текстур и загружает их. Собирает полученные данные в объект, который в последствии возвращает в ModelEditor.

3.2. Диаграмма классов анализа

Диаграмма классов анализа позволяет показать, как взаимодействуют между собой основные компоненты приложения. На рис. 2 показаны пять классов: один класс-граница (UI), один класс-сущности (SceneManager, ModelEditor, TriLib) и три управляющих класса (Model).

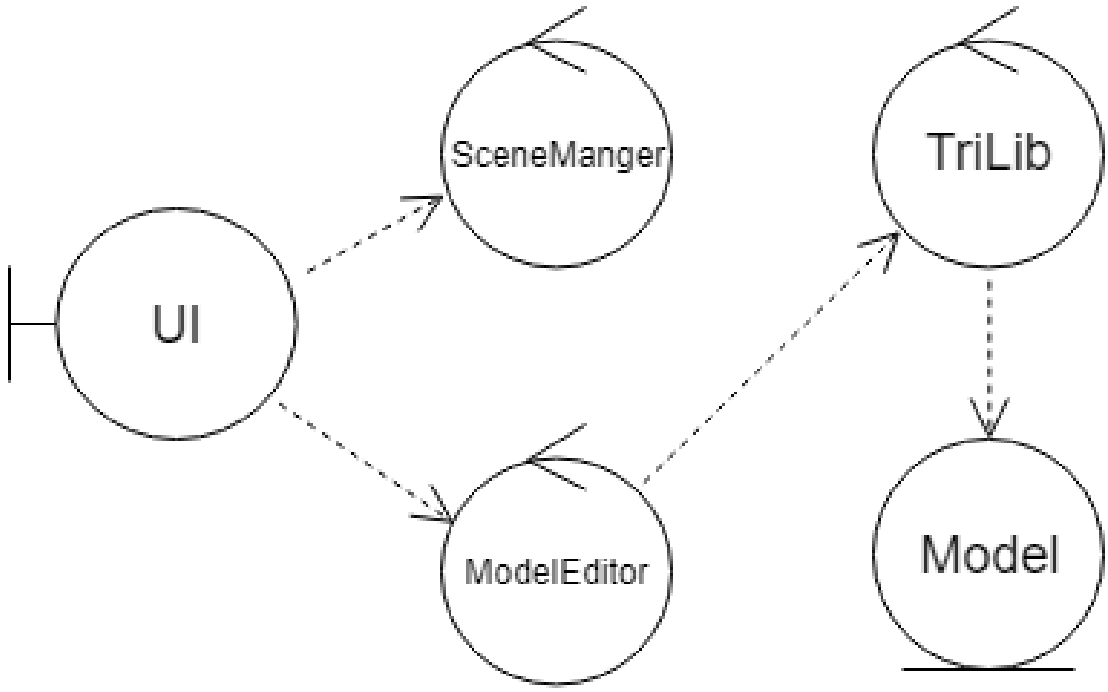

Рис. 2. Диаграмма классов анализа

В данной диаграмме классов анализа показаны основные взаимодействия компонентов. Класс-управления SceneManager изменяет параметры класса UI. Класс-управления ModelEditor использует библиотеку TriLib для загрузки модели, которую впоследствии устанавливает на сцене по координатам, полученным с камеры. В классе-управления App создается экземпляр класса Model.

3.3. Диаграмма классов

Диаграмма классов демонстрирует классы системы, их атрибуты, методы и взаимосвязи между ними. На рис. 3 представлены основные классы, используемые при проектировании и реализации.

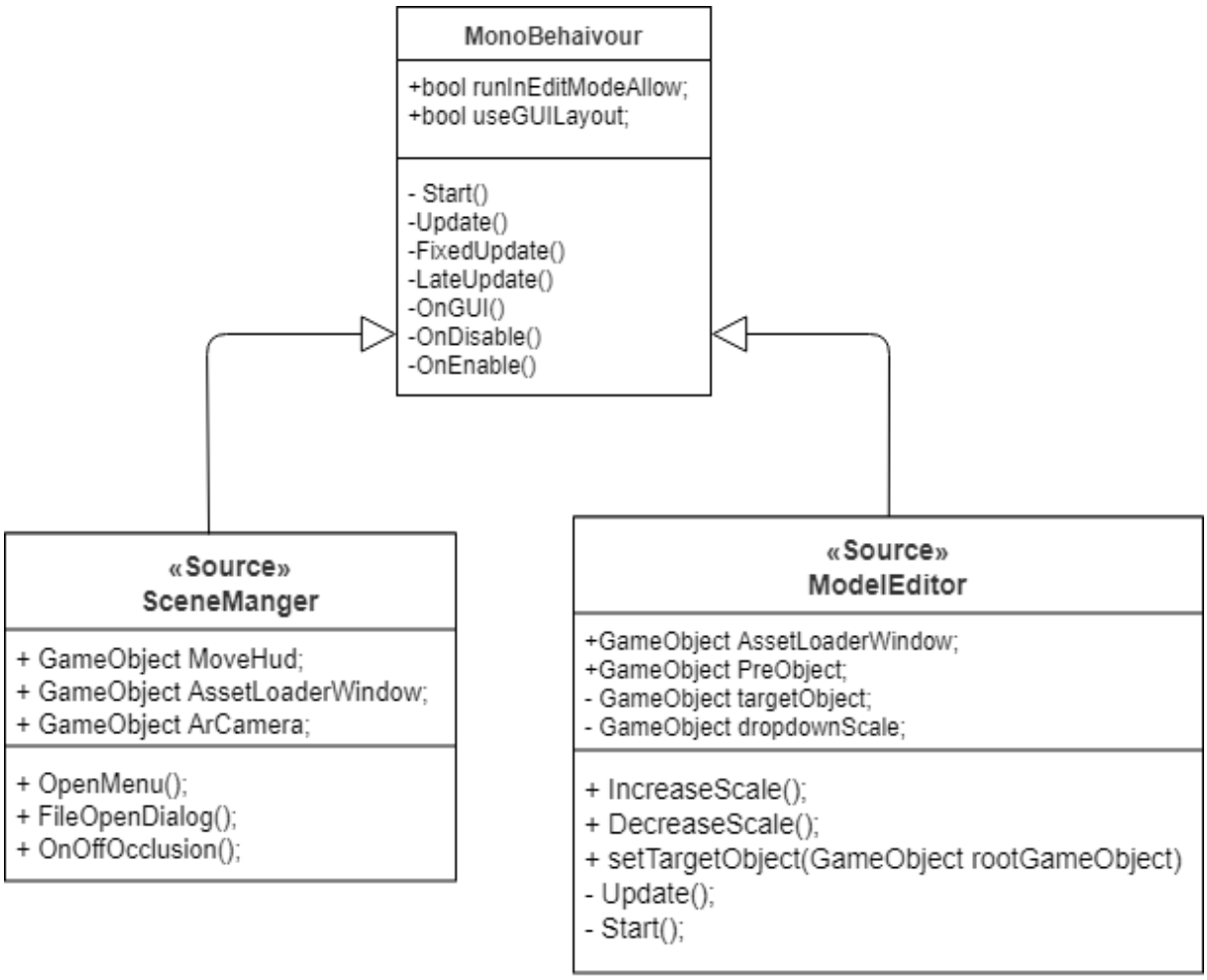

Рис. 3. Диаграмма классов

Класс *MonoBehaviour* — это базовый класс, от которого наследуются все классы (скрипты).

Класс *ModelEditor* отвечает за загрузку и управление характеристиками объекта. В методах IncreaseScale() и DecreaseScale() изменяется размер модели. Метод SetTargetObject() вызывается в классе библиотеки TrliLib и передает загруженный объект в качестве параметра rootGameObject. В методе Update() проверяется и нажатие клавиш управления моделью и изменяется позиция объекта targetObject.

Класс *SceneManager* отвечает за управлением дополнительно меню и открытием проводника для выбора модели на устройстве. Для включения и выключения окклюзии вызывается метод OnOffOcclusion(), в данном методе изменяется параметр объекта ArCamera.

3.4. Диаграмма последовательности

Диаграмма последовательности показывает жизненный цикл каких-либо объектов и их взаимодействия. На рис. 4 продемонстрирован жизненный цикл разрабатываемого приложения.

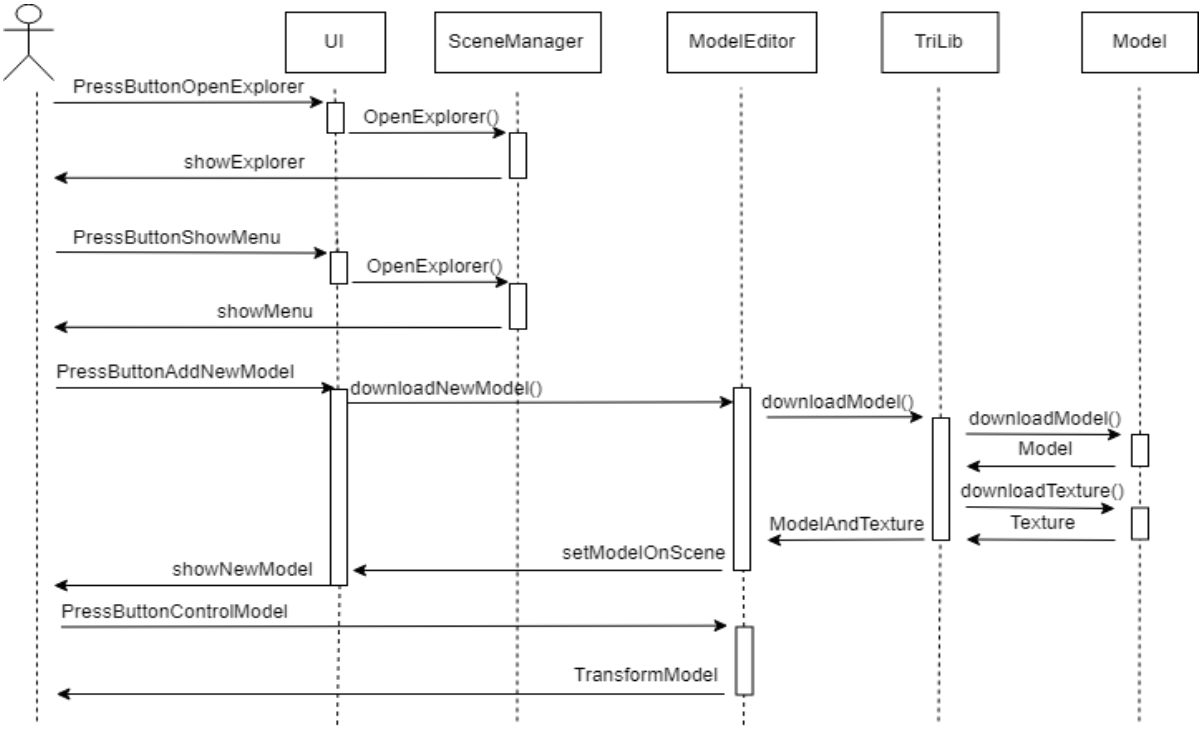

Рис. 4. Диаграмма последовательности

Данная диаграмма иллюстрирует то, что пользователь может различными способами взаимодействовать с приложением:

1) PressButtonOpenExplorer — описывает нажатие пользователем на кнопку открытия дополнительного меню. После того как UI получит сигнал о том, что кнопка была нажата будет вызван метод OpenExplorer() класса SceneManager. Класс SceneManager открывает дополнительное окно для просмотра файлов находящихся на устройстве.

2) PressButtonShowMenu — описывает нажатие пользователем на кнопку открытия дополнительного меню. После того как UI получит сигнал о том, что кнопка была нажата будет вызван метод OpenExplorer() класса SceneMangaer. Класс SceneMangaer открывает дополнительное меню для управления объектом.

3) PressButtonAddNewModel — описывает нажатие пользователем на кнопку загрузки модели с устройства. После того, как UI получит сигнал о том, что кнопка была нажата будет вызван метод downloadNewModel() класса ModelEditor. ModelEditor вызывает метод downloadModel() библиотеки TriLib. После загрузки модели будет получен список текстур, которые будут собраны в один объект. Загруженная модель будет отправлены в класс ModelEditor. Класс ModelEditor высчитывает координаты нового объекта и устанавливает в данный координаты новый объект.

4) PressButtonControlModel — описывает нажатие пользователем на кнопки управления моделью. После того, как UI получит сигнал о том, что кнопка была нажата будет вызван один из методов класса SceneManager. В зависимости от нажатой кнопки будут изменены координаты объекта или размер.

Вывод

В соответствии с требованиями были спроектированы архитектура и дизайн приложения под платформу Android.

4. РЕАЛИЗАЦИЯ

В данном разделе будет рассмотрена реализация всех компонентов, составляющих приложение и используемые технологии и библиотеки.

4.1. Структура проекта

Разработка велась на языке программирования C# [14, 15] в среде разработки MonoDevelop. Приложение состоит из одной сцены. Иерархия объектов представлена на рис. 5. Объекты условно можно разделить на две категории: отвечающие за общую логику программы (объект Controller) и графический интерфейс (объект Canvas и вложенные объекты).

Рис. 5. Иерархия объектов

Вложенность объектов определяет точку отсчета их координат: объект верхнего уровня за точку отсчета принимает центр игровой сцены, а вложенный объект — координаты объекта более высокого уровня.

4.2. Используемые технологии и библиотеки

Система была реализована на высокоуровневом языке программирования C# [14, 15]. Для реализации системы использовались следующие технологии и библиотеки:

1) Tango — платформа компьютерного зрения для мобильных устройств, разработанная группой инженеров ATAP (Advanced Technology and Projects) [7];

2) TriLib — это библиотека для загрузки моделей в Unity, которая позволяет пользователю загружать различные форматы 3D-моделей в редакторе Unity или во время исполнения приложения [8];

3) Unity — инструмент для разработки кроссплатформенных 2D и 3D игр, главной особенностью является среда разработки, которая позволяет работать с графическим представлением, аудио и видео компонентами и физикой игрового приложения без кодирования [1, 11, 12].

4.3. Реализация графического интерфейса приложения

Графический интерфейс игрового приложения разработан с помощью инструмента Unity.UI. К игровому объекту Canvas подключен компонент Canvas, а к вложенным объектам — Canvas Renderer. Это позволяет реализовать тонкую настройку положения и масштабирования элементов графического интерфейса в зависимости от размеров и соотношения сторон экрана.

Основное меню, показанное на рис. 6 состоит из двух элементов: кнопки открытия проводника Open и кнопки для показа или сокрытия дополнительного меню, в виде шестеренки.

Рис. 6. Основное меню

Графический интерфейс разделен на две части: основное меню и дополнительный скрытый интерфейс.

Кнопка Open вызывает метод OpenExplorer(), открывающий проводник для выбора загружаемого пользователем файла, для открытия дополнительного меню вызывается метод ShowMenu(). Открытый проводник показан на рис. 7.

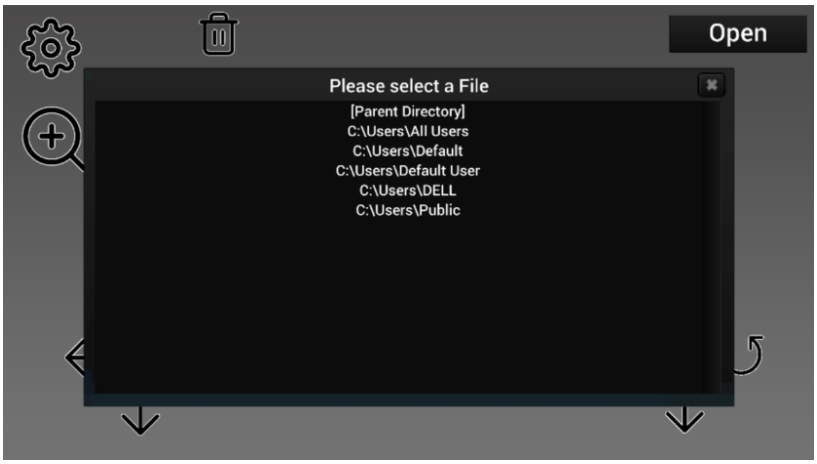

Рис. 7. Проводник для выбора модели

Дополнительное меню управления, представленное на рис. 8, содержит кнопки для изменения координат модели, размеров модели, кнопку включения и выключения окклюзии, выполненную с использованием компонента UI.Toggle, кнопку удаления всех объектов и также выпадающий список, содержащий в себе список загруженных объектов, реализованный с использованием UI.Dropdown.

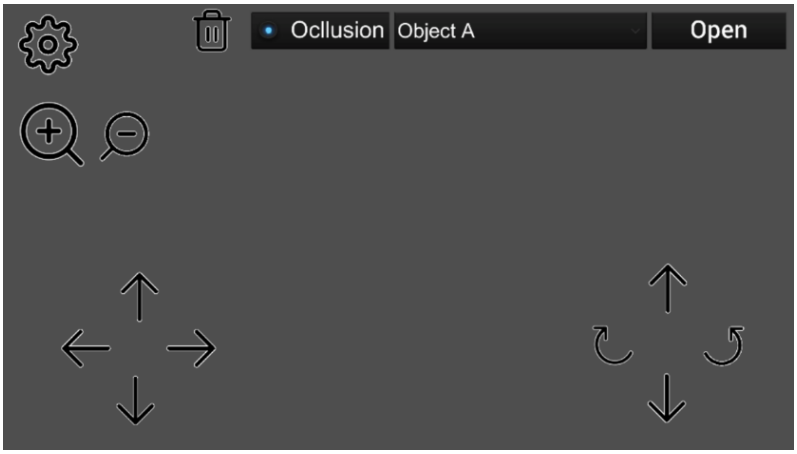

Рис. 8. Дополнительное меню управления

4.4. Реализация приложения

При нажатии пользователем на файл в проводнике, с помощью библиотеки TriLib загруженный файл передается в класс ModelEditor в виде обьекта класса GameObject как параметр в функции setTargetObject(GameObject rootGameObject). Листинг данного метода приведен на рис. 9.

```
public void setTargetObject(GameObject rootGameObject)
{
      targetObject = rootGameObject;

      listObject.Add (targetObject);

      m_Dropdown.options.Clear ();
      foreach (GameObject c in listObject)
      {
            m_Dropdown.options.Add (new Dropdown.OptionData()
{text=c.name});
      }

      Camera cam = Camera.main;
      Vector3 planeCenter;
      Plane plane;
      if (!m_pointCloud.FindPlane(cam, new Vector2(Screen.width/2,
Screen.height/2), out planeCenter, out plane))
      {
            Debug.Log("cannot find plane.");
            return;
      }
      targetObject.transform.position = planeCenter;
}
```

Рис. 9. Листинг метода setTargetObject

В представленном методе setTargetObject загруженный объект устанавливается как текущий, который будет изменяться при использовании кнопок дополнительного меню. После чего мы добавляем новый загруженный объект в List всех загруженных моделей, очищаем выезжающий список Dropdown и добавляем все элементы списка в Dropdown.

Для установки объекта в нужное нам место, мы определяем новые координаты объекта с помощью облака точек m_pointCloud и функции FindPlane для поиска плоскости, на которую направленна камера в текущий момент.

Вывод

На основе спроектированной архитектуры было реализовано приложение для платформы Android.

5. ТЕСТИРОВАНИЕ

Для проверки работоспособности игрового приложения было проведено функциональное тестирование.

Функциональное тестирование — это тестирование ПО в целях проверки реализуемости функциональных требований, т. е. способности ПО в определенных условиях решать задачи, нужные пользователям.

Тест № 1. Загрузка модели пользователем.

Входные данные: пользователь выбирает модель на устройстве и устанавливает ее для просмотра.

Ожидаемый результат: установка модели на сцене, добавление модели в выпадающий список.

Полученный результат: совпадает с ожидаемым. Полученный результат изображен на рис. 10.

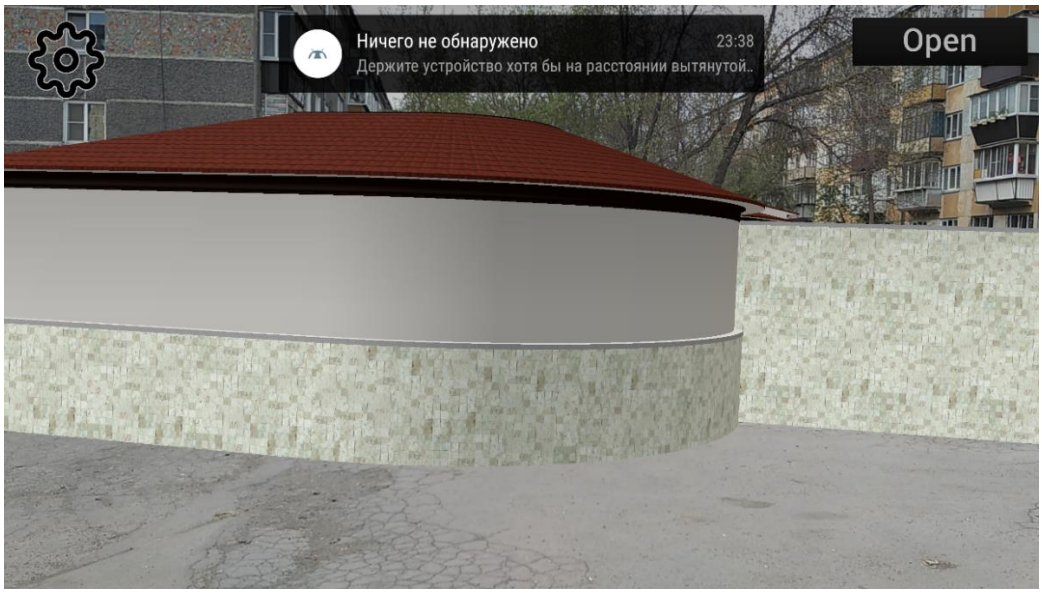

Рис. 10. Загруженная при тестировании модель

Тест пройден успешно.

Тест № 2. Изменение положения загруженного объекта в пространстве.

Входные данные: то же, что и в тесте № 1, после чего пользователь изменяет координаты модели.

Ожидаемый результат: модель перемещается в пространстве.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест № 3. Изменение размеров загруженного объекта.

Входные данные: то же, что и в тесте № 1, после чего пользователь изменяет размеры модели.

Ожидаемый результат: модель изменяется в размерах.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест № 4. Изменение угла поворота загруженного объекта.

Входные данные: то же, что и в тесте № 1, после чего пользователь изменяет угол поворота модели.

Ожидаемый результат: модель вращается в выбранную пользователем сторону.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест № 5. Включение окклюзии.

Входные данные: то же, что и в тесте № 1, после чего пользователь включает окклюзию.

Ожидаемый результат: материальные объекты находящиеся ближе к камере не закрываются отображением виртуального объекта.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест № 6. Выключение окклюзии.

Входные данные: то же, что и в тесте № 1, после чего пользователь отключает окклюзию.

Ожидаемый результат: материальные объекты находящиеся ближе к камере закрываются отображением виртуального объекта.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест № 7. Удаление всех загруженных моделей.

Входные данные: загрузка двух объектов, что и в тесте № 1, после чего удаление всех объектов нажатием на кнопку корзины.

Ожидаемый результат: все объекты удаляются из списка и со сцены.

Полученный результат: совпадает с ожидаемым. Полученный результат изображен на рис. 11.

Рис. 11. Измененная модель при тестировании № 3

Вывод

В результате было выполнено функциональное тестирование. Все тесты были выполнены успешно.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были решены следующие задачи:

1) произведен анализ предметной области проекта;
2) спроектировано приложение;
3) реализовано приложение;
4) произведено тестирование.

СПИСОК ЛИТЕРАТУРЫ

1 Kim S.L., Suk H.J., Kang J.H. Using unity 3d to facilitate mobile augmented reality game development. //In Internet of Things, IEEE World Forum 2014. – P. 21–26.

2 Occlusion in augmented reality [Electronic resource] URL: https://hackernoon.com/why-is-occlusion-in-augmented-reality-so-hard-7bc8041607f9 (date of access: 21.05.2018).

3 Official Site ARCore. [Electronic resource] URL: https://developers.google.com/ar/discover/ (date of access: 21.05.2018).

4 Official Site ARKit. [Electronic resource] URL: https://developer.apple.com/arkit/ (date of access: 21.05.2018).

5 Official Site GAMMA BIM. [Electronic resource] URL: https://gamma-app.de/ (date of access: 21.05.2018).

6 Official Site Roomle 3D. [Electronic resource] URL: https://www.roomle.com/ (date of access 21.05.2018).

7 Official Site Tango. [Electronic resource] URL: http://doc.sdk.tango.me/ (date of access: 21.05.2018).

8 Official Site TriLib. [Electronic resource] URL: https://ricardoreis.net/trilib/manual/html/ (date of access: 21.05.2018).

9 Official Site URBASEE Future. [Electronic resource] URL: https://urbasee.com/ (date of access: 21.05.2018).

10 Official Site Vuforia [Electronic resource] URL: https://www.vuforia.com/ (date of access: 21.05.2018).

11 Unity Scripting API. [Electronic resource] URL: https://docs.unity3d.com/ScriptReference/ (date of access: 21.05.2018).

12 Unity User Manual. [Electronic resource] URL: https://docs.unity3d.com/Manual/index.html/ (date of access: 21.05.2018).

13 Van Krevelen D.W.F., Poelman R., A survey of augmented reality technologies, applications and limitations //Int. J. Virtual Reality,2010 – Vol.9. – P. 1–20. (date of access: 21.05.2018).

14 Руководство по программированию на C#. [Электронный ресурс] URL: https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx (дата обращения: 21.05.2018).

15 Шилдт Г. C# 4.0. Полное руководство. — М.: Издательский дом «Вильямс», 2015. — 1056 с.