

ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

**институт Принтмедиа и информационных технологий
кафедра Информатики и информационных технологий**

направление подготовки 09.03.01 «Информатика и вычислительная техника»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

БАКАЛАВРСКАЯ РАБОТА

Тема: «Разработка приложения для городского ориентирования с элементами дополненной реальности»

Исполнитель Власова Алина Олеговна _____
(Фамилия И.О.) (Подпись)

Руководитель Попов Дмитрий Иванович д.т.н., профессор _____
(Фамилия И.О., степень, звание) (Подпись)

Консультанты 1. _____
(Фамилия И.О., степень, звание) (Подпись)

2. _____
(Фамилия И.О., степень, звание) (Подпись)

Нормоконтроль Попов Дмитрий Иванович д.т.н., профессор _____
(Фамилия И.О., степень, звание) (Подпись)

Антиплагиат Попов Дмитрий Иванович д.т.н., профессор _____
(Фамилия И.О., степень, звание) (Подпись)

«ДОПУЩЕНО К ЗАЩИТЕ»

Зав. кафедрой: д.т.н., проф. Д.И. Попов _____
(Подпись)

Москва

2017



ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

**институт Принтмедиа и информационных технологий
кафедра Информатики и информационных технологий**

направление подготовки 09.03.01 «Информатика и вычислительная техника»

УТВЕРЖДАЮ

Зав. кафедрой, д.т.н. проф. Д.И.Попов

« ____ » _____ 2016 г. _____

(Подпись)

ЗАДАНИЕ НА БАКАЛАВРСКУЮ РАБОТУ

Студент Власова Алина Олеговна группа ДЦасБ-4-1

(Фамилия, Имя, Отчество)

Тема «Разработка приложения для городского ориентирования с элементами дополненной реальности»

Утверждена приказом по Университету №169 от «27» февраля 2017 г.

1. Срок представления работы к защите «6» июня 2017 г.

2. Исходные данные для выполнения работы:

Консультации с дипломным руководителем, литературные и интернет источники

3. Содержание бакалаврской работы: Аннотация, реферат, введение, 1, 2 и 3 главы, заключение, список литературы, приложения 1, 2 и 3

4. Перечень графического материала рисунки, таблицы, схемы

5. Консультанты по разделам:

наименование раздела

(подпись, Фамилия И.О., ученая степень, звание)

наименование раздела

(подпись, Фамилия И.О., ученая степень, звание)

6. Дата выдачи задания: «15» ноября 2016 г.

7. Руководитель:

_____ / _____ /
(Подпись)

8. Задание к исполнению принял:

_____ / _____ /
(Подпись)



ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

**институт Принтмедиа и информационных технологий
кафедра Информатики и информационных технологий**

направление подготовки 09.03.01 «Информатика и вычислительная техника»

КАЛЕНДАРНЫЙ ПЛАН

Студент Власова Алина Олеговна группа ДЦасБ-4-1

(Фамилия, Имя, Отчество)

**Тема ВКР: Разработка приложения для городского ориентирования с
элементами дополненной реальности**

№ п/п	Наименование этапа ВКР	Срок выполнения этапа	Примечание
1.	Поиск и исследование литературы по теме дипломной работы	15.12.16-17.12.16	Выполнено
2.	Исследование предметной области	19.12.16-22.12.16	Выполнено
3.	Выполнение аналитической части	25.12.16-24.01.17	Выполнено
4.	Настройка ПО	27.01.17-05.02.17	Выполнено
5.	Создание 3D моделей	10.02.17-20.03.17	Выполнено
6.	Разработка приложение на платформе Unity3D	25.03.17-20.05.17	Выполнено

Руководитель Попов Дмитрий Иванович д.т.н., профессор

(Фамилия И.О., степень, звание)

(Подпись)

Заведующий кафедрой _____ /Д.И. Попов/

(Подпись)

« ____ » _____ 2016 г

АННОТАЦИЯ

Информационные технологии уже прочно вошли в жизнь современного человека. Невозможно представить себе ни одну отрасль жизнедеятельности, в которой бы они активно не использовались. Благодаря информационным технологиям человечество сделало большой скачок вперед в своем развитии. Несмотря на то, что информационные технологии являются эффективным инструментом развития, они тоже развиваются. Появляются различные инновации, несущее в себе все новые и новые перспективы. Одна из них – это технология дополненной реальности. Она сразу стала востребована во многих областях, как профессиональной деятельности человека, так и в культурной, образовательной и социальной сферах.

Проанализировав рынок AR-приложений для различных областей, было отмечено, что в некоторых из них, например, туризм, данная технология внедряется недостаточно активно. Это приводит к явному противоречию, ведь наглядность, образность, яркость приложений дополненной реальности смогли бы оптимизировать и расширить возможности туристического бизнеса. Именно этим обусловлен выбор и актуальность темы данной работы.

Целью дипломной работы является создание мобильного приложения для городского ориентирования по конкретному туристическому кластеру, которое поможет путешественникам выбрать наиболее интересные для изучения объекты и эргономично рассчитать время для их посещения.

ABSTRACT

Information technology has already firmly established itself in the life of modern man. It is impossible to imagine any branch of life activity, in which they would not be actively used. Thanks to information technology, humanity has made a big leap forward in its development. Despite the fact that information technology is an optimal development, they are also developing. Various innovations carry new perspectives. One of them is a technology of augmented reality. It immediately became in demand in many areas, both professional activities, and in the cultural, educational and social spheres.

Analyzing the AR-application market for various areas, for example, in some of them, for example, tourism, this technology is not implemented actively. This leads to a clear contradiction, because the visibility, imagery, brightness of applications of augmented reality, allowing to optimize and expand the possibilities of the tourist business. It is this that determines the choice and relevance of the topic of this work.

The purpose of the thesis is to create a mobile application for urban orientation on a specific tourist cluster that will help travelers choose the most interesting objects for study and ergonomically calculate the time for their visit.

РЕФЕРАТ

Цель работы: создание мобильного приложения для городского ориентирования по конкретному туристическому кластеру, которое поможет путешественникам выбрать наиболее интересные для изучения объекты и эргономично использовать время для их посещения.

Задачи:

- анализ области исследования;
- исследование требований, методов и алгоритмов решения поставленной цели;
- разработка структуры программного обеспечения;
- создание опытного образца программного обеспечения.

Первая глава включает в себя описание и анализ предметной области. Во второй главе рассматриваются технологии, выбранные для реализации данного проекта. В завершающей части рассматриваются технологии создания 3D моделей, меток. Описание создания мобильного приложения с элементами дополненной реальности на платформе Android. Объем пояснительной записки составляет 96 страниц. Количество рисунков – 31, таблиц – 4, листингов – 8, приложений – 4.

Ключевые слова: AR-технологии, дополненная реальность, 3D модель, Unity3D, приложение дополненной реальности.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
1. ОПИСАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	11
1.1. История возникновения и развития технологии дополненной реальности. 11	
1.2. Классификация систем дополненной реальности	13
1.3. Программное обеспечение и алгоритмы	16
1.4. Сферы применения AR-приложений	18
1.5 Вывод.....	23
2. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ ПРОГРАММНЫХ ПРОДУКТОВ И СИСТЕМ	24
2.1. Анализ и выбор фреймворка.....	24
2.2. Vuforia SDK	26
2.3. Анализ и выбор движка для разработки приложения	30
2.4. Unity3D	32
2.5. Графический 3D редактор. Blender	34
2.6. Объект исследования	36
2.7. Вывод к главе 2.....	39
3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ	41
3.1. Создание 3D моделей.....	41
3.1.1. Установка и настройка ПО для создания 3D моделей	41
3.1.2. Инструменты проектирования 3D моделей	42
3.2. Vuforia. Создание меток и маркеров	46
3.3 Этапы разработки приложения на Unity3D.....	49
3.2.1. Установка и настройка ПО	50
3.2.2. Создание сцены. Изучение интерфейса. Импорт активов	51
3.2.3. Создание сцены и мишеней	56
3.2.4. Разработка пользовательского интерфейса	59
3.2.5. Уменьшение размера файла сборки. Build Settings	67
3.3. Вывод к главе 3.....	70
ЗАКЛЮЧЕНИЕ	71
СПИСОК ЛИТЕРАТУРЫ.....	73
ПРИЛОЖЕНИЕ А	75

ПРИЛОЖЕНИЕ Б.....	80
ПРИЛОЖЕНИЕ В	85
ПРИЛОЖЕНИЕ Г	87
ПРИЛОЖЕНИЕ Д.....	9595

ВВЕДЕНИЕ

Основными направлениями разработки программного обеспечения в настоящее время являются "облачные" технологии, технологии обработки больших массивов данных и приложения для мобильных устройств. Наряду с этим, одним из перспективных направлений развития современных IT-технологий сегодня является технология дополненной реальности (Augmented reality, AR). Тема дополненной реальности живет в умах людей с тех пор, как об этом впервые упомянули писатели-фантасты. Сегодня мы как никогда близки к воплощению этой мечты. Дополненная реальность является интеграцией цифровой информации с окружающей средой пользователя в режиме реального времени. В отличие от виртуальной реальности, что создает совершенно искусственную среду, дополненная реальность использует существующую среду и накладывает новую информацию поверх неё. Дополненная реальность меняет наш взгляд на мир - или, по крайней мере, как пользователи видят мир.

В течение последних нескольких лет различные компании пытались создать устройства, которые давали бы возможность пользователям погрузиться в мир дополненной реальности. В 2013 году Google представил Google Glass – очки дополненной реальности. Через небольшой проектор на экране пользователя отображалась необходимая информация, управление которой происходило при помощи голосовых команд. Приложение Star Walk, при наведении пользователем камеры на ночное небо, позволяет увидеть название звезд и планет. Некоторые приложения для планшетов и телефонов работают с другими объектами. Disney Research разработал AR-раскраску, в которой вы, используя всю свою фантазию, раскрашиваете различных персонажей. Приложение использует доступ к камере на устройстве, чтобы определить персонажа и создать его 3D модель на экране.

Дипломный проект представляет собой мобильное приложение для городского ориентирования с элементами дополненной реальности. Основная

цель, которая стоит перед разработчиком – сделать приложение максимально информативным, красочным и запоминающимся, чтобы пользователь выбрал представленный в приложении объект. В качестве объекта для разработки приложения по городскому ориентированию предлагается один из туристических центров Подмосковья – древний город Коломна. В городе находятся большое количество памятников архитектуры и истории, разноплановых музеев и мест отдыха. Приложение нацелено помочь туристам выбрать для посещения интересные для него объекты и наиболее эргономично рассчитать время их посещения. В окне приложения, при наведении смартфона на бумажный носитель (карту города Коломна) «вырастают» 3D объекты трех различных категорий: достопримечательности, места отдыха и события. Пользователь может выбрать как объекты какой-либо одной категории, так и запросить показать все категории сразу. Основной принцип работы приложения. При наведении камеры смартфона на карту происходит считывание маркеров. Вместо простого знака, обозначающего объект на карте, появляется 3D модель, показывающая приближенную к реальности копию объекта. Если пользователь хочет узнать какую-либо дополнительную информацию, ему достаточно коснуться экрана в зоне расположения 3D объекта. В результате появляется страничка с краткой информацией и мультимедией.

Рассмотрим один пример возможного использования данного приложения. Культовым местом посещения туристических групп в Коломне является Коломенский Кремль. После экскурсии в Кремле у туристов есть немного свободного времени для посещения иных музеев на свой выбор. Данное приложение поможет им быстро и удобно найти ближайший заинтересовавший их объект, например, музей Пастилы, Калачная, место для отдыха или фотосессии на красивом фоне.

Для достижения поставленной цели необходимо решение следующих задач:

- обзор существующих решений;

- исследование требований, методов и алгоритмов решения поставленной задачи;
- разработка структуры программного обеспечения;
- создание опытного образца программного обеспечения.

Таким образом, мобильное приложение для городского ориентирования с элементами дополненной реальности будет способствовать информационному обогащению и более рациональному использованию свободного времени человека, которого в современном мире так не хватает для отдыха, развития и «общения» с окружающим миром.

1. ОПИСАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. История возникновения и развития технологии дополненной реальности

Дополненная реальность была впервые достигнута, в некоторой степени, кинематографистом Мортонем Хейлигем в 1957 году. Он изобрел Sensorama – устройство, которое использовало визуальные эффекты, звуки, вибрация и запах на зрителя (рис. 1.1). Конечно, это не контролировалось компьютером, но это был первый опыт внедрения дополненной реальности.

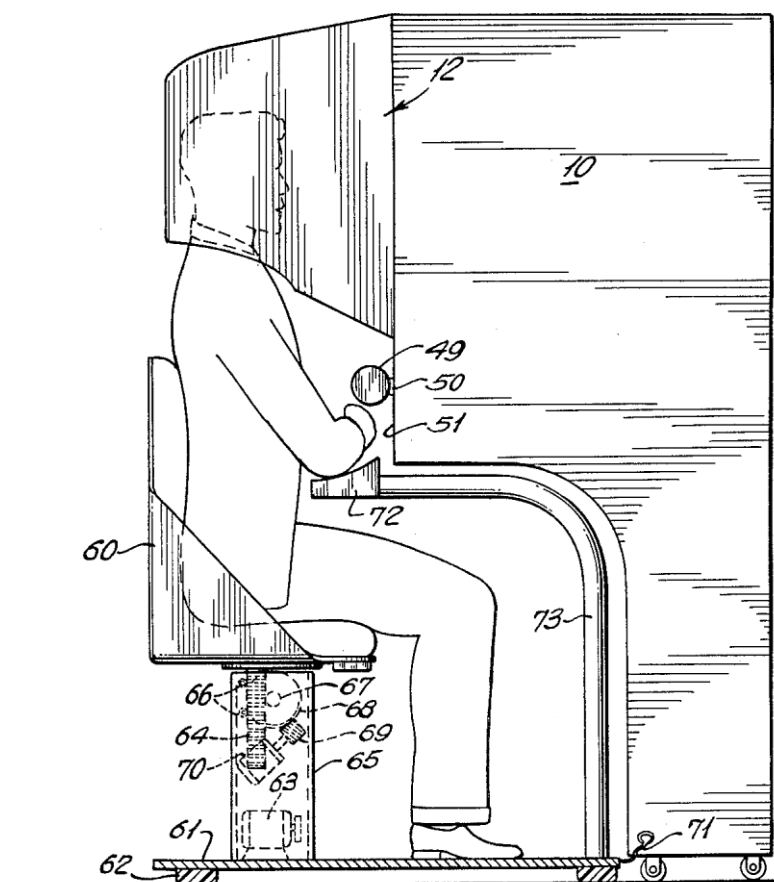


Рисунок 1.1 – Первое устройство дополненной реальности.

Дополненная реальность может и является новой технологией для некоторых, но этот термин, на самом деле, был использован уже в 1990-х годах. Создание термина принадлежит профессору Тому Коделле из Boeing Computer Services в Сиэтле. Он искал новый способ, чтобы помочь процессу производства и начал использовать некоторые технологии виртуальной реальности. В результате он создал сложное программное обеспечение, которое

имело возможность наложения виртуальных объектов в процессе строительства на места их расположения. 1999 заслуживает особого внимания: в этот год увидела свет 'ARToolKit' - библиотека с открытым исходным кодом для разработки программ с элементами дополненной реальности. Несколько лет спустя, эра смартфонов возглавила новую волну развития AR, в этот раз с особым наплывом мобильных приложений для широкого круга потребителей [13].

Рональд Азума (Ronald Azuma) выделил ряд признаков, которыми должна обладать расширенная реальность:

- 1) комбинирование реального и виртуального мира;
- 2) интерактивность;
- 3) трехмерное представление объектов.

Схема среды дополненной реальности представлена на рисунке 1.2.



Рисунок 1.2 – Схема среды дополненной реальности.

Чтобы понять, что представляет из себя технология AugmentedReality, необходимо рассмотреть эту технологию со всех сторон, включая причины её появления, характеристики и требования для работы, сферы применения, плюсы и минусы использования AR в различных отраслях.

Начало 90-х годов можно обозначить как период миниатюризации компьютеров. Следовательно, технологии виртуальной и дополненной реальности начинают применяться шире. Появляется новое направление – разработка переносной электроники и мобильных компьютерных систем. В качестве примера можно привести коммерчески успешные устройства типа

AppleNewtonMessagePad (1993 год) или PalmPilot (1997 год), созданные для упрощения выполнения повседневных задач (создание и контроль расписания дня, дистанционное общение, обмен данными и так далее). В 2000 году появилась первая видеоигра с использованием технологий дополненной реальности, а к 2008 году широкое применение смартфонов среди обычных пользователей сделало возможным распространение данных технологий.

Если раньше данная технология применялась в военной промышленности, компьютерных играх, то сейчас дополненная реальность проникает практически во все сферы деятельности человека: медицину, образование, архитектуру, рекламу и так далее. Дополненная реальность вызывает такой интерес не только потому, что для этого используются новые технологии, но также дополненная реальность обещает помочь людям преодолеть современную информационную перегрузку[16].

1.2. Классификация систем дополненной реальности

Человек получает представление об окружающем пространстве с помощью большого набора органов чувств. Система дополненной реальности, являясь посредником между человеком и реальностью, должна создавать сигнал для одного из таких органов. Таким образом, по типу представления информации системы дополненной реальности бывают:

1. Визуальные. В их основе лежит зрительное восприятие человека. Задача таких систем – создать изображение, которое будет использовано человеком. Поскольку изображение для человека является более информативным и понятным, такой вид систем является более распространенным.

2. Аудио. Такие системы ориентированы на слуховое восприятие. Чаще всего такие системы используются в навигации. Например, они выдают специальные сигналы, когда человек достигает определенного места. Возможно

использование стереоскопического эффекта, позволяющего человеку идти в нужном направлении, ориентируясь на источник звука.

3. Аудиовизуальные. Это комбинация двух предыдущих типов, однако, аудиоинформация в них имеет лишь вспомогательный характер.

Системы дополненной реальности всегда нуждаются в информации, получаемой из окружающей среды. Именно на основе этих данных строятся виртуальные объекты. Каждая из таких систем обладает определенным набором сенсоров – устройств, позволяющих собирать информацию из окружающей среды: звуковые и электромагнитные колебания, ускорение и т.д. Для классификации имеет смысл разделять сенсоры не по типам регистрируемых физических величин, а по их назначению, поскольку сходные по своей природе сигналы могут нести различную информацию. По типу сенсоров можно выделить следующие системы:

1. Геопозиционные. Ориентируются, прежде всего, на сигналы систем позиционирования GPS или ГЛОНАСС. В дополнение к приемникам таких сигналов геопозиционные системы могут использовать компас и акселерометр для определения угла поворота относительно вертикали и азимута.

2. Оптические. Такие системы обрабатывают изображение, полученное с камеры, которые могут перемещаться вместе с системой или вне зависимости от нее.

Системы дополненной реальности можно различать по степени взаимодействия с пользователем. В некоторых системах пользователь играет пассивную роль, он лишь наблюдает за реакцией системы на изменения в окружающей среде. Другие же системы требуют активного вмешательства пользователя – он может управлять как работой самой системы для достижения результатов, так и изменять виртуальные объекты. По этому признаку системы делятся на:

1. Автономные. Они не требуют вмешательства пользователя. Задача таких систем сводится к предоставлению информации об объектах. Например, подобные системы могут анализировать объекты, находящиеся в поле зрения

человека и выдавать справочную информацию о них. Также системы такого типа используются в медицине. Например, система Gait Aid для людей с нарушениями опорно-двигательного аппарата. Она путем использования виртуальных объектов предоставляет мозгу дополнительную информацию, помогающую координировать движения.

2. Интерактивные. Такие системы основаны на взаимодействии с пользователем. На различные действия пользователь получает различный ответ. В подобных системах необходимо устройство ввода информации. В качестве такого устройства может применяться сенсорный экран мобильного телефона, планшет или специальный манипулятор. Выбор устройств ввода зависит от специфики системы. В случае простых действий с виртуальным объектом, достаточно простого указывающего устройства. Если же необходима имитация каких-либо реальных процессов и выполнения сложных манипуляций с объектами используются специальные манипуляторы, имеющие различное количество степеней свободы.

Интерактивность выражается в различной степени. Бывают системы, позволяющие пользователю активно изменять виртуальную среду. Обычно это системы-симуляторы каких-либо реальных действий. Они используются в случае, когда использование реальных объектов невозможно, например, специализированные медицинские тренажеры, позволяющие начинающим врачам отрабатывать необходимые навыки.

Существуют другие системы, где пользователю не нужно изменять виртуальную среду. Вместо этого пользователь выбирает, какие виртуальные объекты он хочет увидеть. Пользователь также имеет возможность манипулировать виртуальными объектами, но не на уровне структуры, а на уровне отображения, т.е. применять, например, аффинные преобразования типа поворота, перемещения и т.д. К данной группе можно отнести различные архитектурные системы, позволяющие увидеть, как впишется в реально существующую обстановку новое сооружение или его часть, а также навигационные и геоинформационные системы. Подобные системы могут

показывать части объектов интереса, скрытые другими постройками, дополнительную информацию о выбранных объектах и т.д.

По степени мобильности системы дополненной реальности можно классифицировать как:

1. Стационарные. Системы этого типа предназначены для работы в фиксированном месте; перемещение таких систем означает частичную или полную приостановку их работоспособности.

2. Мобильные. Системы этого типа могут без труда перемещаться; зачастую такое перемещение и лежит в основе выполняемой ими функции.

Принадлежность к тому или иному типу определяется функциями системы. Так, симулятор хирургического стола не должен быть мобильным, поскольку его задача – воссоздать для человека специальные условия, максимально приближенных к реальным. В то же время навигационная система должна быть как можно более мобильной, чтобы она могла перемещаться вместе с транспортным средством или человеком, не создавая дополнительных расходов на ее перемещение [21].

1.3. Программное обеспечение и алгоритмы

Программное обеспечение должно получить реальные координаты с камеры. Этот процесс называется регистрацией изображения, который использует различные методы компьютерного зрения. Многие методы компьютерного зрения дополненной реальности унаследованы от визуальной одометрии. Обычно эти методы состоят из двух частей. Первый этап может использовать методы обнаружения, например, как функцию обнаружение угла, обнаружение двоичных объектов, обнаружение края и / или другие методы обработки изображений. Второй этап восстанавливает реальный мир системы координат из данных, полученных на первом этапе. Некоторые методы предполагают наличие объектов с известной геометрией. В некоторых из этих случаев 3D структура сцены должна быть предварительно рассчитана.

Математические методы, используемые на второй стадии, включают проективную геометрию, геометрическую алгебру, представление вращения с экспоненциальным отображением. Augmented Reality Markup Language (ARML) является стандартом данных, разработанным в рамках Open Geospatial Consortium (OGC), который состоит из грамматики XML для описания местоположения и внешнего вида виртуальных объектов в сцене, а также ECMAScript привязки для обеспечения динамического доступа к свойствам виртуальных объектов. Для обеспечения быстрого развития дополненной реальности применяют некоторые комплекты разработки программного обеспечения (SDK). Некоторые из хорошо известной AR SDKs предлагаются Metaio, Vuforia, Wikitude и Layar.

Дополненная реальность достигается с помощью различных технологических инноваций; они могут быть реализованы по отдельности или в сочетании друг с другом. Чтобы создать дополненную реальность необходимо:

- Общие аппаратные компоненты – процессор, дисплей, датчики и устройства ввода. Как правило, смартфон содержит процессор, дисплей, акселерометры, GPS, камеру, микрофон и т.д., и содержит все необходимое оборудование.
- Дисплеи – в то время как монитор вполне способен отображать данные AR существует и другие системы, такие как оптические проекционные системы, дисплеи головного монтажа, очки, контактные линзы, виртуальные дисплеи сетчатки, EyeTap (это устройство, которое изменяет лучи света, захваченные из окружающей среды) пространственной дополненной реальности (РС - который использует обычные методы проекции в качестве замены для отображения любого вида) и портативных дисплеев.
- Датчики и устройство ввода включают в себя – GPS, гироскопы, акселерометры, компасы, RFID, беспроводные датчики,

сенсорное распознавание речи, отслеживание глаз и периферийные устройства.

- Программное обеспечение – большинство развития для АРА будет в разработке дальнейшего программного обеспечения, чтобы воспользоваться аппаратными возможностями. Существует уже дополненная реальность Markup Language (ARML), которая используется для стандартизации грамматики XML для виртуальной реальности [17].

1.4. Сферы применения AR-приложений

1. Археология

AR может быть использован для помощи в археологических исследованиях, путем расширения современного ландшафта, что позволяет археологам сформулировать выводы о размещении тех или иных объектов. AR-приложение дает возможность для пользователей восстанавливать руины, здания, или даже пейзажи, как они ранее существовали.

2. Архитектура

AR может помочь в визуализации строительных проектов. Компьютерные генерируемые изображения могут быть наложены в реальной жизни на место зрения строящегося здания построено. AR также могут быть использованы в рабочем пространстве архитектора, рендеринг анимированной 3D визуализации и чертежей. Архитектура достопримечательность может быть усовершенствована с AR-приложением, позволяющем пользователям видеть внешний вид здания через его стенку, рассматривая все его предметы интерьера и расположение.

3. Искусство

AR-технология помогает инвалидам создавать произведения искусства с помощью отслеживания движения глаз, чтобы перевести движение глаз

пользователя в рисунки на экране. Элемент может быть сконструирован таким образом, что при сканировании с помощью AR-устройства он отображает дополнительные объекты и слои информации, которые не были видны в реальном мире. В 2013 году L'Oreal использовали CrowdOptic технологию для создания дополненной реальности на седьмом ежегодном фестивале Luminato в Торонто, Канада.

4. Коммерция

AR может улучшить предварительные просмотры продуктов, так как позволяет клиенту посмотреть, что находится внутри упаковки продукта, не открывая её. AR также может быть использован в качестве помощи при выборе продуктов из каталога или через киоск. Отсканированные изображения продуктов можно активировать в виде дополнительного контента. AR используется для интеграции печатных и видео - маркетинг. Печатные рекламные материалы могут быть разработаны с учетом определенных «триггерных» образов, которые при сканировании с помощью устройства с поддержкой AR с помощью распознавания изображения, активизируют видео версию рекламного материала.

5. Строительство

С непрерывным улучшением точности GPS, предприятия могут использовать дополненную реальность для визуализации географической привязки моделей строительных площадок, подземных сооружений, кабелей и труб с помощью мобильных устройств.

6. Образование

Приложения дополненной реальности могут усовершенствовать стандартную учебную программу. Текст, графика, видео и аудио могут быть наложены в реальную картинку. Учебники учебные материалы могут содержать внедренные «маркеры», которые, при сканировании с помощью

устройства AR, производят дополнительную информацию для студента, вынесенную в мультимедийном формате.

Учащиеся могут заниматься на компьютере в интерактивном режиме моделированием исторических событий, изучением и обучением деталей каждой значительной площади в месте проведения соревнований. AR может помочь студентам в понимании химии, позволяя им визуализировать пространственную структуру молекулы и взаимодействовать с её виртуальной моделью, которая появляется в изображении камеры. Технология дополненной реальности позволяет также обучать посредством дистанционного взаимодействия, в котором студенты и преподаватели могут разделить общую виртуальную среду обучения, населенную виртуальными объектами и учебными.

7. Повседневное использование

Начиная с 1970-х годов, Стив Манн разрабатывает технологии, предназначенные для повседневного использования, то есть «горизонтальной» во всех приложениях, а не для конкретного «вертикального» рынка. Приложение «EyeTap Digital Eye Glass», которое дает возможность управления динамическим диапазоном (HDR зрения), например, уменьшение электрической дуги, глядя на сварочную горелку.

8. Компьютерные игры

Дополненная реальность позволяет игрокам испытать цифровую игру в реальных условиях. За последние 10 лет было много усовершенствований, что приводит к улучшению обнаружения движения.

9. Промышленный дизайн

AR может помочь промышленным дизайнерам испытать конструкцию и эксплуатацию продукта до его завершения. Volkswagen использует AR для

сравнения расчетных и фактических показателей краш-теста. AR может быть использован для визуализации и изменения структуры кузова автомобиля и расположение двигателя. AR также может быть использован для сравнения цифровых макетов с физическими макетами для обнаружения расхождений между ними.

10. Медицина

Дополненная реальность может обеспечить хирурга информацией, которая изначально скрыта, например, показатель скорости сердечных сокращений, кровяное давление, состояние органов пациента и т.д. AR может использоваться, чтобы позволить врачу «заглянуть» внутрь пациента путем комбинирования нескольких источников, таких как рентгены с другим видами изображений. AR может улучшить просмотр плода в утробе матери.

11. Военная сфера

В бою AR может служить сетевой системой связи, которая выдает полезные данные с поле боя на очки солдата в режиме реального времени. С точки зрения солдата, люди и различные объекты могут быть отмечены специальными индикаторами, чтобы предупредить о потенциальной опасности. Виртуальные карты и камеры с видом на 360 ° также могут быть полезны при навигации солдата и перспективы на поле боя. Данные могут быть переданы военным лидерам в командный центр дистанционного.

12. Навигация

AR может увеличить эффективность навигационных устройств. Информация может отображаться на лобовом стекле автомобиля, например, сведения о погоде, рельеф местности, дорожные условия и информацию о дорожном движении, а также, предупреждения о потенциальной опасности на пути. На борту морских судов AR может непрерывно

контролировать важную информацию, такую как: заголовок судна и скорость при движении по всему мосту или выполнение других задач.

13. Спорт

Самый банальный пример — нанесение на картинку трансляции футбольного матча некоторых дополнительных линий: линии, показывающей, был игрок в положении «вне игры» или нет; линии с указанием расстояния до ворот, до стенки и т. д. Для зрителя у телевизора это наглядно и удобно. Аналогично дополненная реальность может быть полезна и в других игровых видах спорта. Возьмем еще к примеру бег. На реальную картинку, показывающую соревнования восьми спортсменов, накладываем виртуальную фигуру — тень девятого бегуна. Бег этого спортсмена-тени синхронизирован с графиком бега рекордсмена мира. Телезритель же наглядно видит на беговой дорожке состязание не только бегунов между собой, но и их состязание с рекордсменом. Интересно и зрелищно. Вместо фигуры рекордсмена может быть представлена тень бегуна, показавшего лучшее на текущий момент время в других забегах. В Португалии создан бильярдный тренажер дополненной реальности. Над бильярдным столом размещена камера, которая передает в автоматизированную систему управления (АСУ) данные о положении игрока, кия, бильярдных шаров. Лазерное оборудование проецирует на стол траекторию движения шаров. Таким образом, бильярдист, видя на столе траекторию, может скорректировать свой прицел. Промахнуться почти нереально. Это хороший тренажер для начинающих.

14. Туризм

Приложения дополненной реальности могут помочь пользователям при движении, обеспечивая в реальное время информацией о его месте нахождения, включая замечания, сделанные предыдущими посетителями сайта. AR приложения позволяют туристам испытать моделирование исторических

событий, места и объекты, вставляя их в их реальный вид ландшафта. AR-приложение может также представлять информацию о местоположении с помощью аудио, видео и другой мультимедии.

15. Перевод

AR системы могут интерпретировать иностранный текст в дополненном представлении пользователя, повторно отображая текст на языке пользователя. Разговорные слова иностранного языка могут отображаться в режиме пользователя, как напечатанные субтитры [16].

1.5 Вывод

Обзор существующих приложений показал основные тенденции в развитии технологии дополненной реальности. Эта технология в наше время очень бурно развивается и может найти применение во многих областях.

Анализ приложений показал, что сейчас они имеют, в основном, либо развлекательный, либо узконаправленный характер. Основное внимание в таких приложениях уделяется их возможностям при взаимодействии с окружающей средой и пользователем.

Для создания функционирующей системы необходима достаточно мощная платформа, которой может быть современное мобильное устройство, в виду их широкой распространенности и постоянно растущих возможностей.

2. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ ПРОГРАММНЫХ ПРОДУКТОВ И СИСТЕМ

2.1. Анализ и выбор фреймворка

OpenCV – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Java, Ruby, Matlab, Lua и других языков. Может свободно использоваться в академических и коммерческих целях – распространяется в условиях лицензии BSD.

Vuforia SDK – это программное обеспечение для мобильных устройств, которое позволяет создавать приложения дополненной реальности. Оно использует технологию компьютерного зрения для того, чтобы распознавать и отслеживать плоские изображения и 3D-модели в реальном мире, когда они рассматриваются через камеру мобильного устройства. Положение и ориентация виртуального объекта отслеживается в реальном времени, так что точки зрения зрителя на объект соотносятся с их точкой зрения на изображение, так что кажется, что виртуальный объект является частью реальной сцены мира.

Vuforia SDK поддерживает различные 2D и 3D целевые типов, включая безмаркерные цели. Дополнительные возможности SDK включает локализованные обнаружения окклюзии с помощью виртуальных кнопок, изображений выполнения целевого отбора, а также возможность создать и изменить целевые наборы программно во время выполнения.

Vuforia обеспечивает API для C++, Java, Objective-C, и .Net языков. Есть расширение игрового движка Unity. Таким образом, SDK поддерживает как родные для IOS и Android языки, так и одновременно позволяет разрабатывать приложения дополненной реальности в Unity, которые легко переносимым на обе платформы. Приложений, разработанные с использованием совместимы с широким спектром мобильных устройств, включая iPhone, iPad, Android и телефонов, и планшетов под управлением ОС Android [15].

ARToolkit – это библиотека компьютерного слежения для создания приложений с дополненной реальностью. Для этого он использует возможности видео слежения, расчет реального положения и ориентации камеры по отношению к квадратному физическому маркеру в режиме реального времени. Когда реальное положение камеры известно виртуальная камера может быть расположена в той же точке и 3D модель накладывается на реальный маркер. Так ARToolKit решает две ключевые проблемы в дополненной реальности; отслеживания точки зрения и виртуального взаимодействия объектов.

ARToolKit был первоначально разработан Hirokazu Kato в 1999 году и был выпущен в университете Лаборатория НИТ Вашингтоне. В настоящее время он функционирует как проект с открытым исходным размещенные на SourceForge с коммерческой лицензии доступны ARToolWorks. ARToolKit очень широко используется (библиотека с более чем 160000 загрузок).

Metaio SDK – готовая библиотека для создания мобильных приложений дополненной реальности. Использует OpenGL использует SLAM методы для более точной работы

String – библиотека для создания мобильных приложений ориентирована на iOS устройства [15].

Сравнение альтернатив и критериев фреймворка приведено в таблице 1.

Таблица 1 – Таблица альтернатив и критериев.

Критерии Альтернативы	Кроссплатформенность	Наличие документации и литературы	Поддержка различных языков программирования	Тип лицензии
OpenCV	iOS, PS, Android, Linux	Много	C/C++, Python, Java, Ruby, Matlab, Lua	Boost Software License
Vuforia	iOS, Android, Unity	Достаточно	C++, Java, Objective-C, .Net	Free Commercial SDK option

Продолжение Таблицы 1 – Таблица альтернатив и критериев.

Критерии Альтернативы	Кроссплатформенность	Наличие документации и литературы	Поддержка различных языков программирования	Тип лицензии
ARToolkit	iOS, Android, Unity	Много	Java, Objective-C	Free Commercial SDK option
Metaio SDK	iOS, Android, Unity, WEB	Мало	Java, Objective-C, .Net	Free Commercial SDK option
String	iOS, Unity	Мало	Objective-C, .Net	Free Commercial SDK option

Обзор альтернатив с учетом наиболее важных критериев показал, что самым оптимальным Фреймворком для разработки системы является Vuforia. Vuforia – бесплатная библиотека, постоянно модернизирующаяся, позволяет создавать кроссплатформенные приложения с помощью специального игрового движка Unity.

2.2. Vuforia SDK

Компании Qualcomm Vuforia дополненной реальности (AR) SDK представляет собой твердый SDK с надежными вариантами развития. Они предоставляют услугу распознавания target для обработки изображений. Кроме того, вы можете загружать ваши target с помощью target manager. Vuforia также поддерживает Unity игровой движок и Native SDK для Android и IOS. Расширение Vuforia AR позволяет разработчикам создавать AR-приложения с помощью Unity. Оба стандарта Unity и Unity Pro совместимы с расширением AR.

Ядром платформы является библиотека QCAR, написанная на C++ и прячущая за свои интерфейсы практически все от определения мишени (target) и заканчивая рендерингом картинки. Помимо библиотеки, платформа включает в себя: iOS и Android Vuforia SDK для разработчиков; Target Manager – систему для создания и управления мишенями, а также набор web сервисов, в которые можно внести хранение мишеней и еще разного рода функционал. Платформа развивается очень динамично и новые части добавляются довольно часто.

Одним из плюсов SDK Vuforia является то, что у него очень простой процесс установки. Компания предоставляет информацию, которая поможет вам настроить среду разработки и установить соответствующий SDK. Есть различные версии SDK, включая расширение Unity, Android Native SDK и IOS Native SDK.

SDK Vuforia предоставляет два варианта для размещения и управления вашими таргетами. Вы можете использовать либо target manager или Web Service API. Target manager представляет собой интернет-инструмент для создания и управления базами данных как устройств, так и облачных баз данных. Эти два типа базы данных позволяют выбрать, как ваши мишени будут размещены и развернуты в вашем приложении. API Vuforia Web Service работает на основе API REST, что позволяет интегрировать существующие системы управления контентом для управления и обновления ваших target images без необходимости обновлять свои приложения. Vuforia рекомендует использовать API веб-службы, если у вас есть большое количество target для управления.

AR SDK Vuforia – это решение корпоративного класса, который может быть использован любым разработчиком для создания приложения дополненной реальности.

Информация, представленная на сайте Хабр: «Чтобы что-то добавить в реальный мир, нужно в этом мире “зацепиться” за какой-либо объект. Именно для этой цели и служат мишени. Другими словами, мишень – это некий

реальный объект, зная который наше приложение может расставить виртуальные объекты в нужных местах и соответствующих пропорциях.

Vuforia предоставляет богатый выбор мишеней (рис. 2.1):

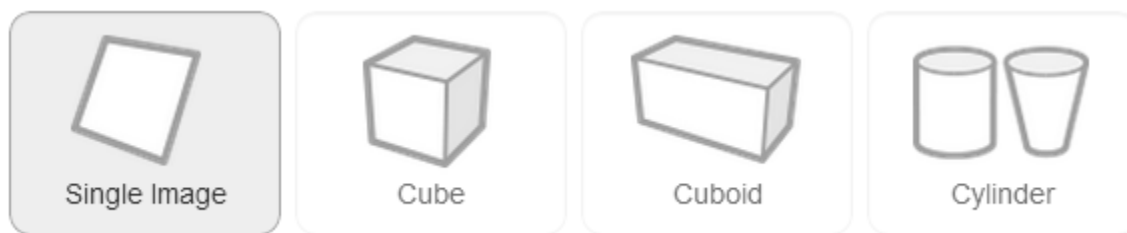


Рисунок 2.1 – Виды мишеней в Qualcomm Vuforia.

- Image targets – базовый вид мишеней, представляющий собой обычную картинку, например, обложку журнала, фотографию или афишу нового фильма. Картинка выполняет роль своего рода двумерного штрих-кода, только без черно-белых регионов. По ней мы можем определить, какая именно картинка попала в объектив камеры, а также её расположение в пространстве и масштаб. Стоит сказать, что не любая картинка подойдет для создания мишени. Хорошими мишенями являются те, в которых много контрастных деталей. Именно на этих деталях и строится опорная матрица для последующего распознавания мишеней.
- Simple 3D targets (Cube and Cuboid) – это мишени в виде прямоугольных параллелепипедов (включая куб). Например, такой мишенью может служить упаковка из-под сухих завтраков, спичечный коробок или только что купленная настольная игра. Как и любая коробка, такая мишень состоит из шести плоскостей, и чтобы создать её, нам понадобится шесть картинок для каждой из них. Если вы когда-нибудь склеивали кубик из бумаги – то быстро справитесь с задачей.
- Cylinder targets – этот вид мишеней, несмотря на название, представляет собой усеченный конус с возможностью задавать диаметры оснований. Конечно если выбрать одинаковые диаметры, то получатся как раз цилиндр, но все же это частный случай. Для

того, чтобы создать такую мишень нам понадобится не только указать диаметры оснований и высоту, но также добавить три картинки – по одной для каждого из двух оснований, и еще одну для боковой поверхности.

- **Frame markers** – эта мишень в виде специально подготовленной рамки, которая уже куда более походит на штрих-код. В такую рамку можно поместить любую картинку. Данный вид мишеней отлично подходит в случае, если картинка не была достаточно детализирована и создать из нее толковую *image target* не получилось.
- **Text (word targets)**. В библиотеку *Vuforia* встроено еще и распознавание текста, поэтому любое слово или их сочетание может являться мишенью. На данный момент поддерживается только латиница, но и это уже очень интересно. Только представьте насколько интересным может стать изучение иностранных слов.

В зависимости от количества мишеней, необходимых для вашего приложения, вы можете либо хранить их в так называемой *Device Database*, всегда иметь к ним доступ и распознавать их непосредственно на самом устройстве, либо переложить часть этой нагрузки на *Cloud Databases* – сервис из набора *Vuforia Web Services*, предназначенный для хранения мишеней и определения их на основании присланных с устройства данных. Оба подхода имеют свои плюсы и минусы, но определившись с целями вашего приложения выбрать будет не так сложно.

Помимо этого, мишени могут быть созданы непосредственно из самого приложения используя картинку, полученную с камеры. Стоит также отметить, что *QCAR* не требует попадания всей мишени в кадр. Для того, чтобы её распознать и использовать достаточно будет лишь части. Это очень важно и позволяет рассматривать всю «магию» вашего приложения под самыми разными углами». [8]

Следует также отметить, что, не смотря на свободное распространение данного SDK среди разработчиков все внутренние реализации скрыты и являются коммерческой тайной.

2.3. Анализ и выбор движка для разработки приложения

Unity3D – это кросс-платформенный игровой движок, разработанный Unity Technologies, который в основном используется для разработки видеоигр и моделирования для компьютеров, консолей и мобильных устройств.

Project Anarchy — новый бесплатный игровой движок от Navok, который вполне может конкурировать с Unity3D. Project Anarchy, также, как и Unity, включает два типа лицензии: Free и Pro. На данный момент Project Anarchy не публикует цены на Pro версию.

Сравнение лицензий игровых движков приведено в таблице 2.

Таблица 2 – Сравнение лицензии.

Unity3D	Project Anarchy
Unity3D (цена: Free) Для дистрибьюции на десктоп, мобильные устройства и web	Project Anarchy (цена: Free) Только для дистрибьюции на мобильные устройства
	Project Anarchy PC Exporter (цена: 499\$) Расширение Free версии для дистрибьюции на PC
Unity (цена: 1500 или подписка 75\$/месяц)+дополнительные платные плагины Для дистрибьюции на десктоп, мобильные устройства, консоли и web	Project Anarchy Pro (цена: неизвестна) Для дистрибьюции на десктоп, мобильные устройства и консоли

Сравнение поддерживаемых платформ игровыми движками приведено в таблице 3.

Таблица 3 – Сравнение поддерживаемых платформ.

Платформа	Unity3D (v. 4.2)	Project Anarchy (v2013.2)
Windows Desktop	да	да (Pro)
Mac OS	да	да (Pro)
Linux (Ubuntu)	да	нет
Web	да	нет
iOS	да	да
Android ARM7	да	да
Android x86	нет (анонсировано)	да
Tizen	нет (анонсировано)	да
Windows Phone 8	да	да (Pro)
Blackberry	да	нет (анонсировано)
Консоли (ps3/xbox/wii/psvita)	да (Pro)	да (Pro)

Project Anarchy использует C++ для native разработки и Lua для скриптования в редакторе. Unity использует Mono и поддерживает только managed разработку на C#, JavaScript# and Boo. Для обоих движков можно писать плагины на native коде.

Сравнение платформ для разработки приложения на игровых движках представлено в таблице 4.

Таблица 4 – Сравнение платформ для разработки.

Платформа	Unity3D (v. 4.2)	Project Anarchy (v2013.2)
Windows Desktop	да	да
Mac OS	да	да
Linux (Ubuntu)	да	нет

Project Anarchy пока, увы, не может быть конкурентом Unity для разработки приложений. Project Anarchy более сложный, не такой гибкий и простой в освоении как Unity. В нем нет подобия Unity Store с кучей плагинов и различных пакетов. Также недостатком Project Anarchy является большая зависимость от платформы Windows. Все её средства разработки завязаны на Windows. Unity имеет более кроссплатформенную архитектуру. Также пока неизвестна цена за Pro версию Project Anarchy [11].

2.4. Unity3D

Unity – это инструмент для разработки двух- и трёхмерных приложений и игр, работающий под операционными системами Windows, Linux и OS X. Созданные с помощью Unity приложения работают под операционными системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а также на игровых приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One и MotionParallax3D дисплеях (устройства для воспроизведения виртуальных голограмм), например, Nettlebox. Есть возможность создавать приложения для запуска в браузерах с помощью специального подключаемого модуля Unity (Unity Web Player), а также с помощью реализации технологии WebGL. Ранее была экспериментальная поддержка реализации проектов в рамках модуля Adobe Flash Player, но позже команда разработчиков Unity приняла сложное решение по отказу от этого.

Unity охватывает весь 3D вид окружающей среды, подходит для прокладки уровней, создание меню, создания анимацию, написание сценариев и организации проектов. Пользовательский интерфейс хорошо организован и панели могут быть полностью настроены с помощью перетаскивания. Панель Сцена представляет собой 3D-окно просмотра, где вы можете физически организовать активы, перемещая их вокруг в 3D пространстве (рис. 2.2).

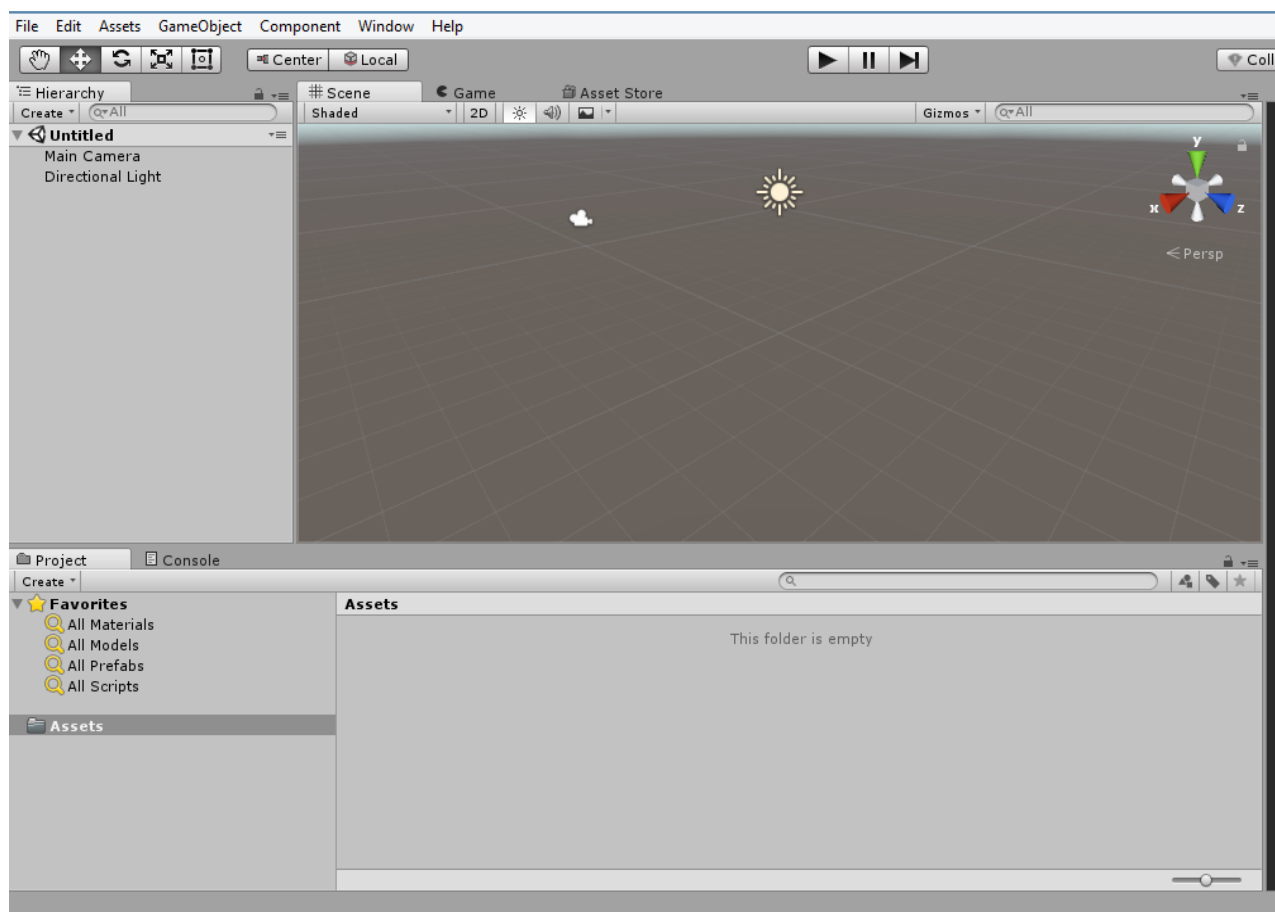


Рисунок 2.2 – Рабочая панель проекта.

Коллайдер — это область пространства, при взаимодействии с которой выполняются те или иные скрипты, действия. Чтобы лучше понять, что это такое, представьте, что вы подходите к автоматически открывающимся дверям универмага. За пару метров от дверей, срабатывает датчик и двери распахиваются перед вами, но если вы пройдете за три метра, то ничего не произойдет. Как раз та зона, в которой действует датчик движения, отвечающий за открывание дверей, и есть коллайдер. Вы входите в зону —

двери открываются, выходите из зоны — и датчик уже никак не будет реагировать на ваше присутствие [11].

Unity использует большое количество ресурсов. Они включают в себя 3D-модели, материалы, текстуры, звук, скрипты и шрифты и так далее. За исключением нескольких простых объектов, таких как кубы и сферы, Unity не может фактически создать большую часть этих активов. Вместо этого они должны быть созданы извне с помощью 3D-приложений моделирования и инструменты рисования, а затем импортировать в Unity.

К счастью, импорт активов Unity является надежным и умным. Традиционно, 3D движки, как правило, были привередливы и очень часто о том, какие файлы вы даете им, заставляя разработчиков тщательно конвертировать все свои файлы. Но только не Unity. Он будет принимать все популярные форматы файлов, включая 3D Maya, 3D Studio Max, материалов и текстур нетронутыми. Unity также поддерживает все распространенные форматы файлов изображений, включая PNG, JPEG, TIFF и даже слоистых PSD-файлов непосредственно из Photoshop. Когда дело доходит до аудио, Unity поддерживает WAV и AIF, идеально подходят для звуковых эффектов, а также MP3 и OGG для музыки.

Unity имеет Asset магазин, где вы можете приобрести 3D модели, персонаж, текстуры, звуковые эффекты, музыку, инструменты и даже сценарии. Unity Asset магазин быстро стал неоценимым ресурсом для разработчиков игр [11].

2.5. Графический 3D редактор. Blender

Blender является профессиональным свободным и с открытым исходным кодом 3D редактор компьютерной графики. Программный продукт используется для создания анимационных фильмов, визуальные эффекты, искусство, 3D печатных моделей, интерактивных 3D-приложений и видеоигр. Возможности Blender включают в себя 3D-моделирование, текстурирование,

растровая графика редактирование, такелаж и свежевание и многое другое. Кроме того, он имеет встроенный игровой движок [6].

Существование этой программы лишний раз доказывает, что бесплатное еще не означает плохое. Создание приложения, предназначенного для трехмерного моделирования и анимации, — это невероятно трудная задача. А если к этому добавить, что за такую титаническую работу разработчик не получит никакого вознаграждения, кроме благодарности 3D-художников, то может показаться, что создание бесплатного профессионального редактора трехмерной графики невозможно (рис. 2.3).

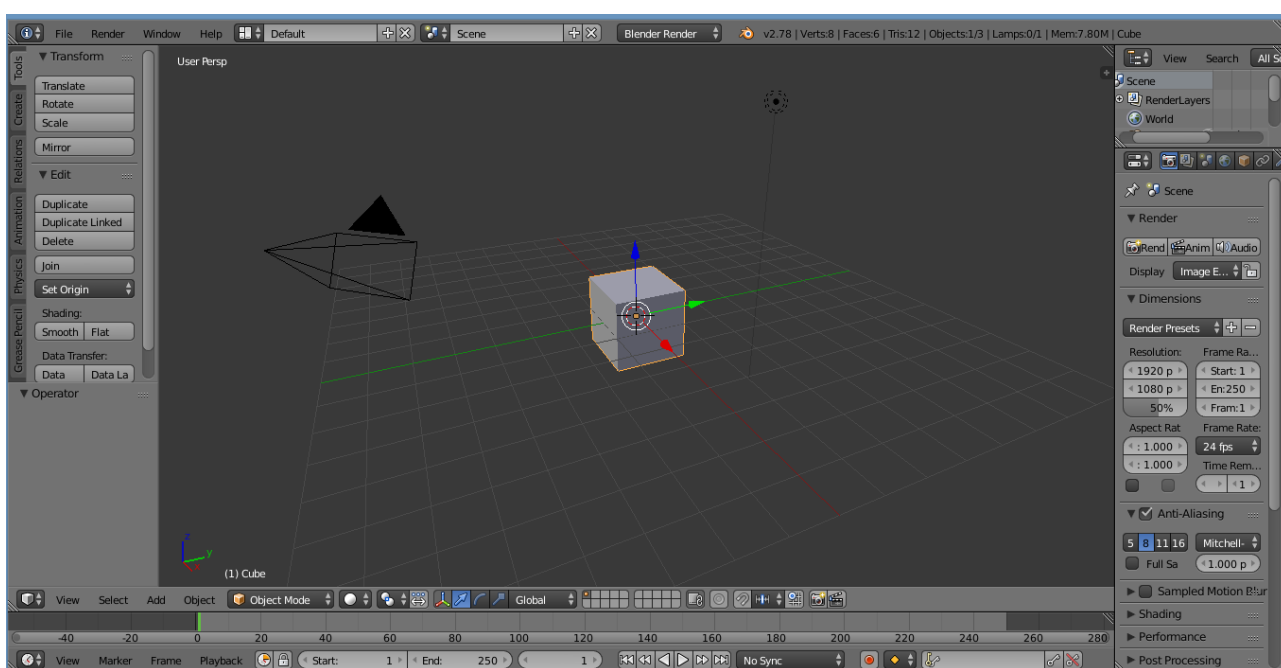


Рисунок 2.3 – Рабочая панель Blender.

Тем не менее проект Blender не только появился на свет, но и активно развивается, не уступая ни в чем коммерческим аналогам. Секрет успеха кроется в том, что к работе над Blender может присоединиться любой желающий. Многие из инструментов, которые появились в этой программе, были добавлены совершенно разными людьми, создававшими те или иные функции для решения своих задач. Однако справедливости ради стоит заметить, что Blender начинался как коммерческий проект, но позже был закрыт и возрожден уже с открытыми исходными кодами. По сравнению с коммерческими разработками размер этого редактора совершенно мизерный —

всего несколько десятков мегабайт. Одно из самых главных преимуществ программы — кроссплатформенность. Blender одинаково хорошо и стабильно работает в Linux и Windows. Кроме того, программа может функционировать даже на ПК с очень слабыми конфигурациями, вплоть до нетбуков. Минимальные требования к системе более чем скромные: процессор с одним ядром, работающий на частоте 1 ГГц, оперативная память 512 Мбайт и видеокарта с поддержкой Open GL и объемом памяти не ниже 64 Мбайт. Программа включает в себя большой арсенал средств для создания трехмерной графики. Так, в Blender можно оперировать системами частиц, контролировать веса отдельных частиц при текстурировании, применять направляющие при анимации и использовать внешние силы, например, ветер [3].

2.6. Объект исследования

В качестве объекта для разработки приложения по городскому ориентированию предлагается один из туристических центров Подмосковья — древний город Коломна. В городе находятся большое количество памятников архитектуры и истории, разноплановых музеев и мест отдыха. Приложение нацелено помочь туристам выбрать для посещения интересные для него объекты и наиболее эргономично рассчитать время их посещения.

Первым этапом разработки мобильного приложения для городского ориентирования с элементами дополненной реальности являлся анализ и выбор объектов, которые будут созданы в графическом 3D редакторе.

Для создания мобильного приложения требовалось: 4 объекта исторической важности; 3 объекта, являющиеся популярным местом отдыха, будь то ресторан или гостиница; 3 объекта, которые отображали бы ближайшие события, проводимые в городе Коломна.

В качестве объектов для первой категории были выбраны следующие достопримечательности.

1. Коломенский Кремль

Многие едут в Коломну, чтобы увидеть прославленный Кремль. Могучий Коломенский Кремль (рис. 2.4), построенный по оценкам историков примерно в начале 15 века, хорошо сохранился до наших дней и помогает оказавшимся здесь туристам ощутить незабываемое путешествие во времени. Осмотр Коломенского Кремля может смело занять полдня, поскольку, кроме самых защитных стен и башен крепости, здесь много уникальных архитектурных памятников.



Рисунок 2.4 – Коломенский Кремль.

Так, на территории кремля находится Успенский кафедральный собор, Ново-Голутвинский женский монастырь, Воскресенская церковь, Тихвинский храм, Успенский Брусенский женский монастырь, а также Крестовоздвиженская церковь.

2. Богоявленский Старо-Голутвин монастырь

Богоявленский Старо-Голутвин монастырь – одна из самых посещаемых религиозных достопримечательностей Коломны (рис. 2.5). Возведение обители связывают с именем преподобного Сергия Радонежского, который по летописным источникам руководил строительством будущего монастыря. Точную дату возникновения Богоявленского Старо-Голутвин монастыря ученые не могут назвать до сих пор, но считается, что обитель отсчитывает свою историю с 14 века. За это время монастырские сооружения были

обновлены и сейчас представляют собой красивейший архитектурный ансамбль.

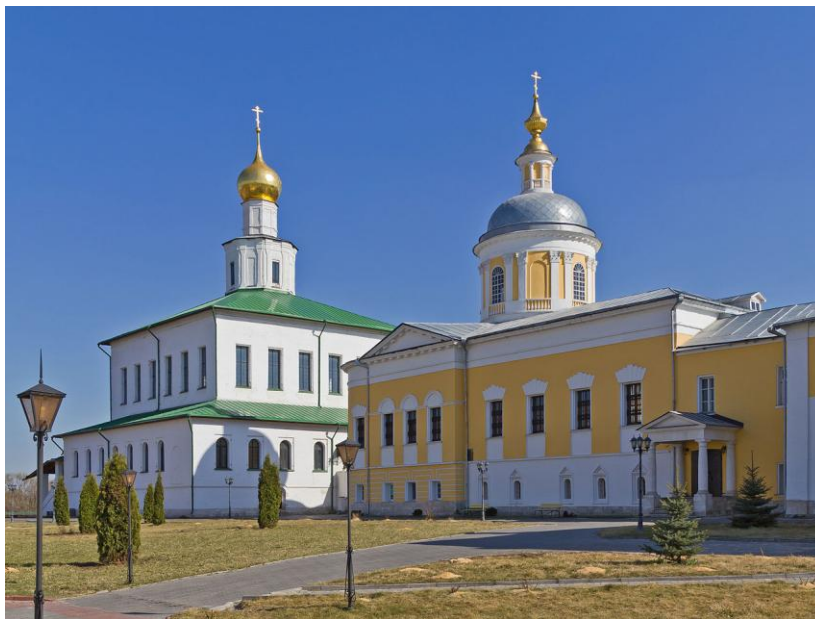


Рисунок 2.5 – Богоявленский Старо-Голутвин мужской монастырь.

3. Музей Пастилы

Музей "Коломенская пастила" (рис. 2.6) представляет собой новый тип музея: сохраняющего и предьявляющего нематериальное наследие, нечто эфемерное и ускользающее — вкусы, запахи, культуру бытования, манеру говорить.

Этот музей притягивает своей уютной атмосферой, тщательно и с любовью подобранным интерьером, интересно рассказанными и даже инсценированными историями и конечно, диковинным вкусом пастилы, такой разнообразной и не похожей на стандартные, всем нам известные белые брусочки из магазина.



Рисунок 2.6 – Музей Коломенской Пастилы.

4. Усадьба купцов Лажечниковых

Усадьба купцов Лажечниковых (рис. 2.7) – это исторический памятник середины 18 века. Иван Иванович Лажечников – писатель, получивший широкую известность как автор русского исторического романа [14].



Рисунок 2.7 – Усадьба купцов Лажечниковых.

2.7. Вывод к главе 2

Во второй главе были рассмотрены технологии и программное обеспечение, которые использовались в процессе разработки приложения для городского ориентирования с элементами дополненной реальности.

- Фреймворк Vuforia SDK;
- Кроссплатформенный движок Unity3D;
- Графический 3D редактор Blender.

Так же была рассмотрена область исследования проекта с представлением примеров используемого материала.

3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

При разработке мобильного приложения для городского ориентирования с элементами дополненной реальности, было установлено и настроено программное обеспечение: 3D редактор компьютерной графики Blender; Vuforia SDK и игровой движок Unity3D.

3.1. Создание 3D моделей

3.1.1. Установка и настройка ПО для создания 3D моделей

Blender – это профессиональный 3D редактор компьютерной графики с открытым исходным кодом.

Данное ПО можно скачать на официальном сайте разработчика: <https://www.blender.org/download/>.

Установка программы очень быстрая, удобная и понятная. Для этого нужно:

1. Дать согласие на правила пользования ПО;
2. Выбрать место установки;
3. И нажать кнопку «Install».

Установка занимает не более двух минут и по окончании установки перед вами появится финальное окно загрузки (рис. 3.1) [20].

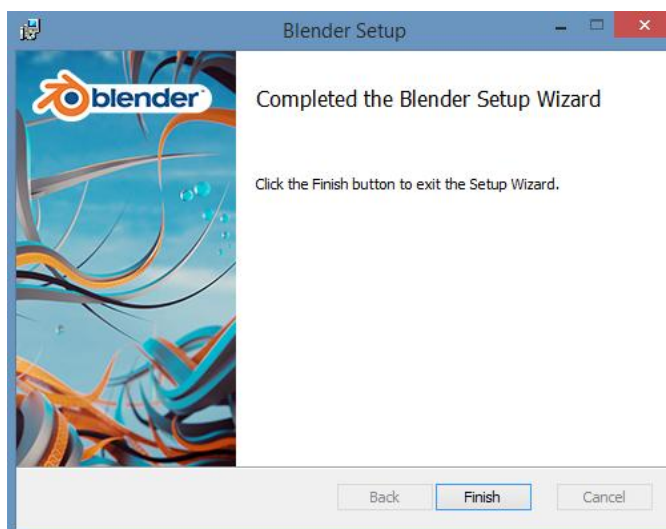


Рисунок 3.1 – Финальное окно загрузки Blender.

3.1.2. Инструменты проектирования 3D моделей

Blender получил свою популярность благодаря быстрой работе в нем с помощью большого количества горячих клавиш. Те из них, которые были использованы при создании проекта чаще всего, рассмотрим ниже:

- Клавиша [TAB] – переключение между режимом редактирования (редактирование вершин) и Объектным режимом (рис. 3.2).
- Клавиша «O» – запускает режим пропорционального редактирования вершин, пока мы находимся в режиме редактирования.
- Клавиша «A» – в режиме редактирования данная клавиша позволяет быстро выделить все вершины, при повторном нажатии происходит обратная реакция.
- [Space Bar] – вызывает меню инструментов, в котором мы можем выбрать добавление таких меш-объектов, как камера, лампа и т.д.

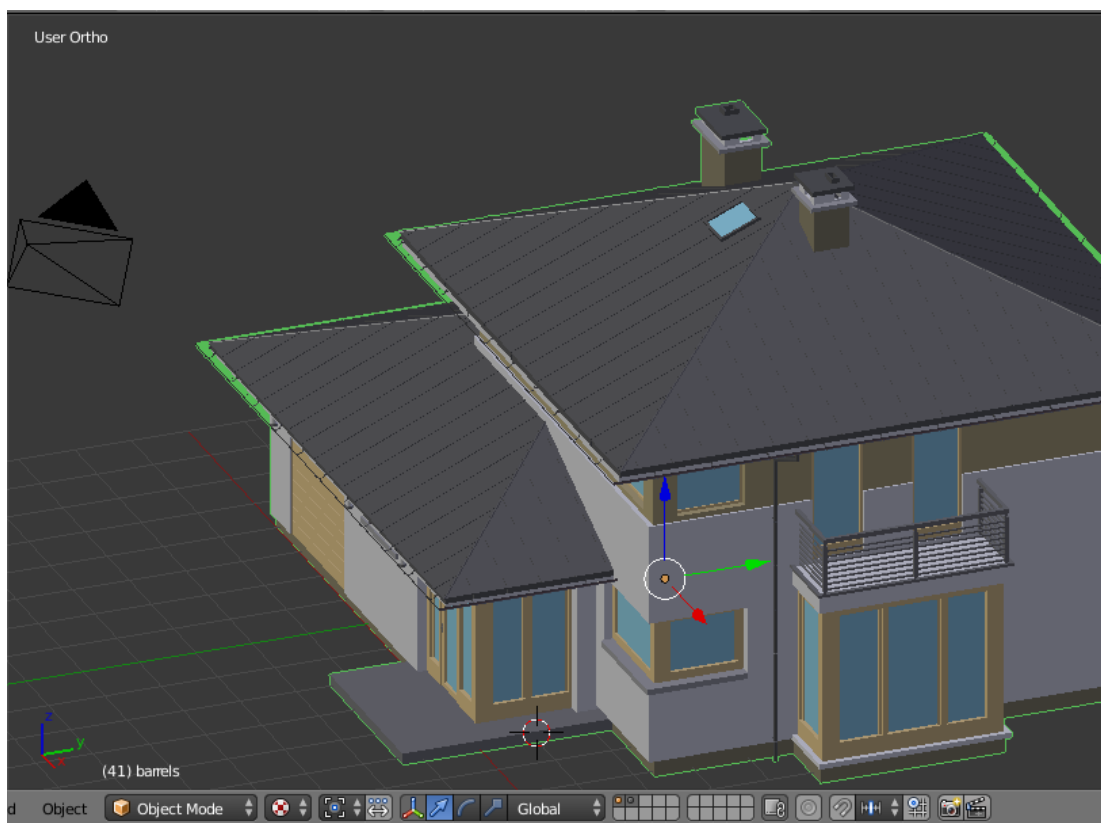


Рисунок 3.2 – Использование клавиши [TAB]. Объектный режим.

- Действия мышью – левая клавиша предназначена для манипулирования, правая – для выделения, средняя – для вращения просмотра и изменения

фокусного расстояния. При удерживании Shift и одновременно средней кнопки мыши можно передвигать 3d проекцию экрана.

- Клавиша «R» – вращение выделенных объектов или вершин.
- Клавиша «S» – масштабирование выделенных объектов или вершин.
- Клавиша «G» – перемещение выделенных объектов или вершин.
- Клавиша «U» – в объектном режиме вызывает меню единого использования (для создания уникальных материалов, анимации) (рис. 3.3).

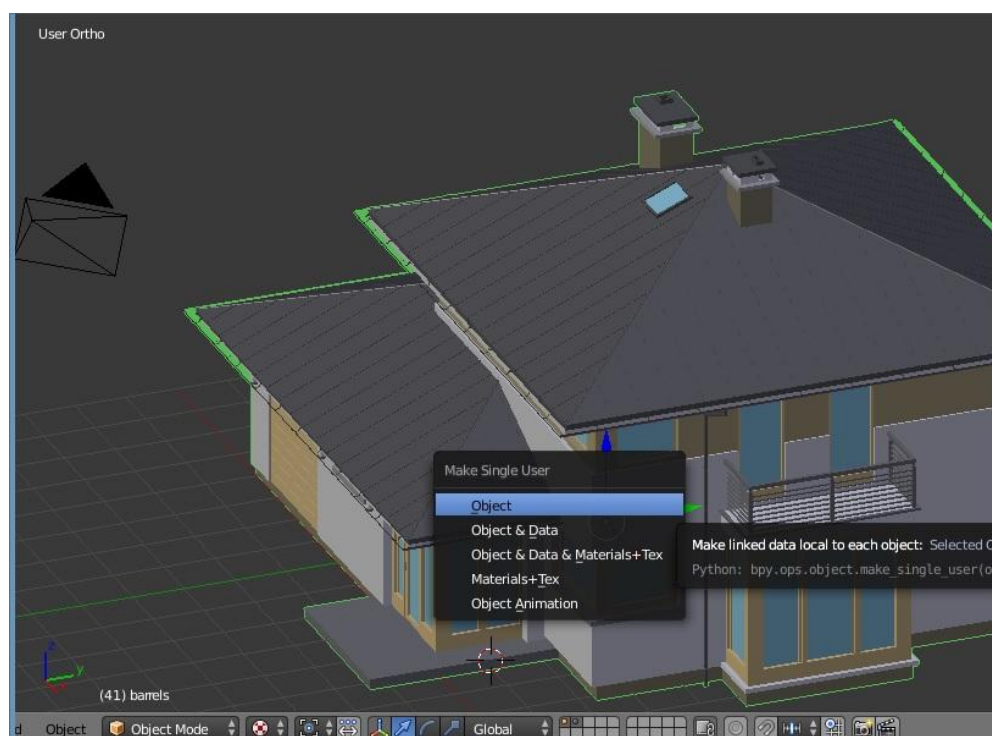


Рисунок 3.3 – Использование клавиши «U». Объектный режим.

- Клавиша «N» – вызывает информационное окно выделенного объекта (данные в окне подлежат изменению) (рис. 3.4).
- Комбинация [Ctrl+J] – объединение выбранных объектов в одно целое.
- Клавиша «E» – в режиме редактирования позволяет экструдировать (вытягивать) выделенные вершины.
- Клавиша «X» или Delete – удаление выбранного объекта, вершины или грани.

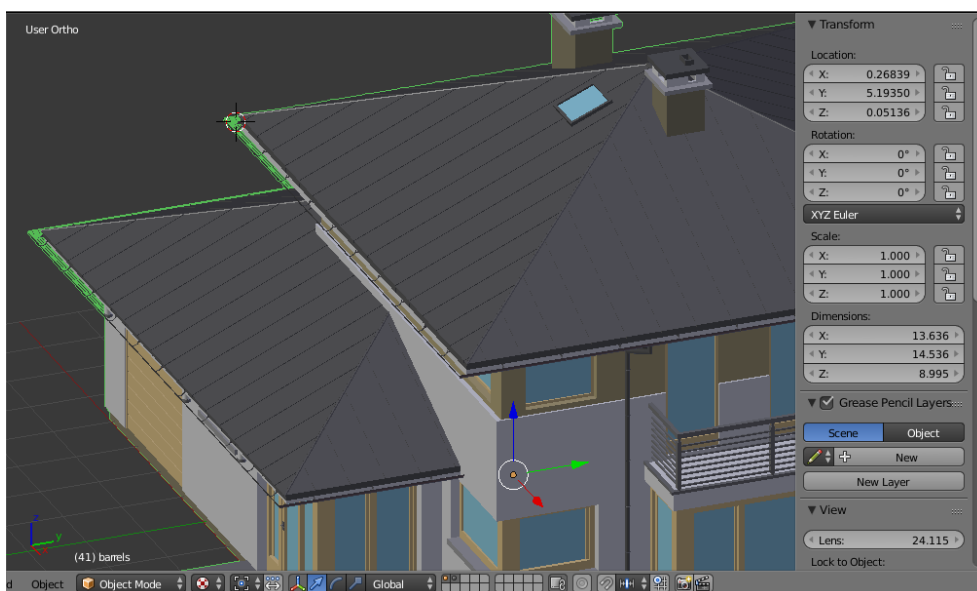


Рисунок 3.4 – Использование клавиши «N». Информационное окно в правой части экрана.

В Blender имеются различные виды окон. Всего их 3: верхнее окно – Панель Инструментов; среднее – 3D Окно (рис. 3.5); нижнее – Окно Кнопок. Мы можем изменять тип любого из приведенных ранее окон. Самые распространенные и интересные типы окон:

- File/Image Browsers (Броузер Файлов/Картинок) - Появляется автоматически при сохранении файла или картинки;
- Buttons Window (Окно Кнопок) - Различные Опции и Настройки Объекта;
- User Preferences (Пользовательские Настройки) - Меню и опции;
- Video Sequence Editor (Редактор Видео Последовательности) - Объединение клипов и эффектов;
- UV/Image Editor (Редактор Изображений/UV) - Для разработки текстур, в частности для игр;
- Action Editor (Редактор Действий) - Для Анимации;
- Ipo Curve Editor (Редактор Ipo Кривых) - Так же используется для анимации;
- 3D View (3D Просмотр) - Для просмотра и работы с моделями.

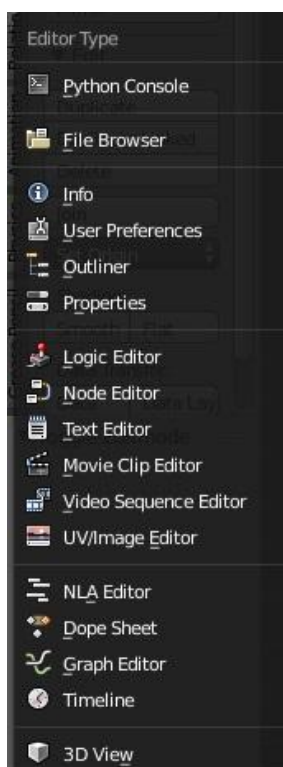


Рисунок 3.5 – 3D Окно.

Для того, чтобы сохранить ваш проект в формате .blend, нужно использовать функцию «Pack Data» (Упаковка Данных), которая находится в меню «File» (рис. 3.6). Для того, чтобы размер файла оставался небольшим, текстуры, аудио и другие файлы автоматические не сохраняются в файле .blend. При открытии файла происходит поиск текстур и звуков по определенным ссылкам. Если файлы найти не удастся, то и присутствовать в сцене они не будут. При «Упаковке Данных» файлы текстур и аудио будут включены в файл .blend. Теперь все файлы всегда будут на месте, но при этом увеличится размер файла. Если упаковка файлов прошла успешно, то в верхней части экрана появится изображение коробки. Так же существует и обратная операция – «Unpack Data» [4].

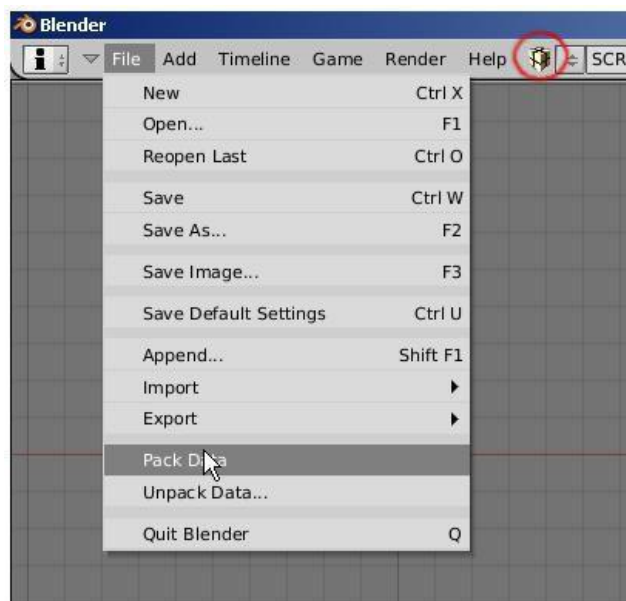


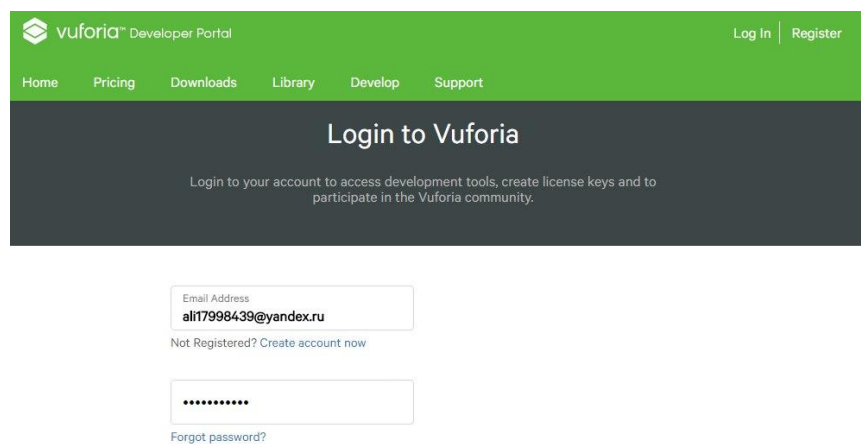
Рисунок 3.6 – Упаковка Данных.

3.2. Vuforia. Создание меток и маркеров

AR SDK Vuforia – это решение корпоративного класса, который может быть использован любым разработчиком для создания приложения дополненной реальности.

Чтобы что-то добавить в реальный мир, нужно в этом мире “зацепиться” за какой-либо объект. Именно для этой цели и служат мишени. Другими словами, мишень – это некий реальный объект, зная который наше приложение может расставить виртуальные объекты в нужных местах и соответствующих пропорциях. Именно AR SDK Vuforia мы будем использовать для создания таргетов (мишеней).

Для того, чтобы приступить к созданию маркеров, нужно зарегистрироваться или войти на сайт разработчика (рис. 3.7).

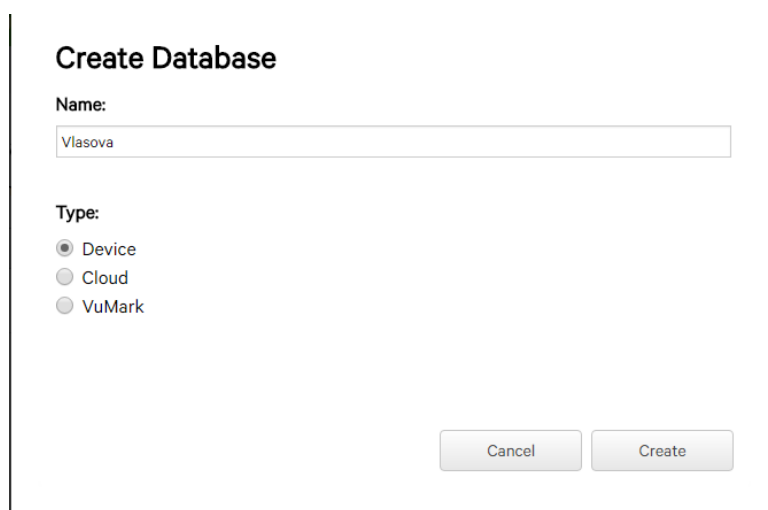


The screenshot shows the Vuforia Developer Portal login interface. At the top, there is a green navigation bar with the Vuforia logo and the text "vuforia™ Developer Portal". On the right side of this bar are links for "Log In" and "Register". Below the navigation bar is a dark grey header with the text "Login to Vuforia". Underneath, a smaller line of text reads: "Login to your account to access development tools, create license keys and to participate in the Vuforia community." The main content area contains two input fields: the first is labeled "Email Address" and contains the text "ali17998439@yandex.ru"; below it is a link that says "Not Registered? Create account now"; the second is a password field with masked characters "....."; below it is a link that says "Forgot password?".

Рисунок 3.7 – Окно авторизации пользователя.

Для того, чтобы создать таргет или ключ, нужно перейти во вкладку «Developer». В данном окне расположены базы, в которых можно добавлять, удалять или изменять данные. Главной «мишенью» в проекте является карта города Коломна. Именно ее фотографию нужно загрузить в базу. Создается это следующим образом:

1. Нажать на кнопку «Add Database» (Создать Базу);
2. В появившемся окне нужно дать имя базе и выбрать устройство, для которых будет в дальнейшем использоваться эта база (рис. 3.8);
3. Нажимаем «Create»;



The screenshot shows the "Create Database" form. The title "Create Database" is at the top. Below it is a "Name:" label followed by a text input field containing the text "Vlasova". Underneath is a "Type:" label followed by three radio button options: "Device" (which is selected), "Cloud", and "VuMark". At the bottom right of the form are two buttons: "Cancel" and "Create".

Рисунок 3.8 – Создание Базы в Vuforia.

4. После создания базы в нее нужно добавить таргет. Нажимаем на кнопку «Add Target»;

5. В появившемся окне выбираем тип мишени, загружаем файл, задаем его размер и имя, нажимаем «Add»;

В окне информации о созданной мишени (рис. 3.9) есть значение «Augmentable» (показатель степени распознаваемости мишени). Выражается в количестве звезд (от 0 до 5). Чем больше присвоено звезд данному таргету, тем быстрее и эффективнее будет распознавание.

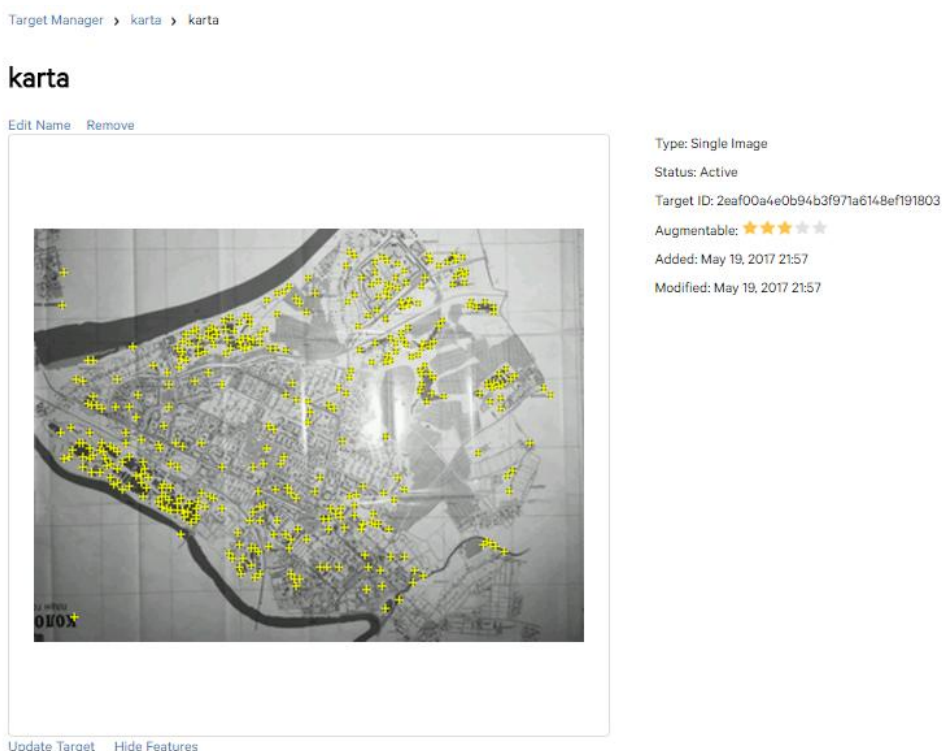


Рисунок 3.9 – Информационное окно.

Для того, чтобы в дальнейшем использовать данный таргет в своем приложении, базу нужно скачать. Во вкладке созданной базы нужно нажать на кнопку «Download Database (All)». Скачается файл типа Unity package file, который потом нужно будет импортировать в проект Unity.

В зависимости от количества мишеней, необходимых для приложения, вы можете либо хранить их в так называемой Device Database, всегда иметь к ним доступ и распознавать их непосредственно на самом устройстве, либо переложить часть этой нагрузки на Cloud Databases – сервис из набора Vuforia Web Services, предназначенный для хранения мишеней и определения их на основании присланных с устройства данных. Оба подхода имеют свои плюсы и

минусы, но определившись с целями вашего приложения выбрать будет не так сложно.

Для каждого нового проекта Unity3D нужен свой, отличающийся от остальных, уникальный ключ. Создается он так же в Vuforia. Переходим во вкладку Develop/License Manager и нажимаем кнопку «Add License Key». В появившемся окне выбираем устройство и выбираем имя. В результате Vuforia выдает нам сгенерированный ключ (рис. 3.10).

License Manager > karta

karta [Edit Name](#) [Delete License Key](#)

License Key Usage

Please copy the license key below into your app

```
AfKGl2D/////AAAAGVhjkJpmYUwwpnkHSrkKjWxb1eglcw/QiRLN
FbtheFUXsoMomFtmr1IzGpAu/naMIFR+xvUdYlkNc3bFZjr90Sh1
q23mPkFldg7VHJXpV+XF3nTR9ewU9aNu5o8R9/q52a8xY0UdnXZZ
lCM3S/Q9syZV4iJlVYX1leYgBH2/fgCYqw75hXJFe0aE8dyHoe2J
+bJHeT5ZaNnpXEK7tKqgH1BEE2uSaGhNY5Av/mVWPktYsFZ1kNMu
OSQIogb3yRSmriCTA3BrtsZKOMwMzpYYgW16HW1Nct9iuJc8pxeg
2U063NEMGBTfseUoiW8GQ7hvNeduXdTAmTg9ixDtb14x4e/uzOgT
bhnEAWnDz24wwTXP
```

Type: Develop
Status: Active
Created: May 19, 2017 21:58

History:
License Created - May 19, 2017 21:58

Рисунок 3.10 – Создание лицензионного ключа.

Лицензионный ключ должен быть добавлен в поле App License Key в VuforiaConfiguration. Данный актив доступен на панели инспектора ARCamera через Open Configuration Vuforia. Можно также искать через VuforiaConfiguration на панели Project.

3.3 Этапы разработки приложения на Unity3D.

Скачать данное программное обеспечение можно по ссылке:
<https://unity3d.com/ru/get-unity/download>.

3.2.1. Установка и настройка ПО

В Unity с версии 5.0 и далее, совместно скачивается Assistant – небольшая программа (около 1 МБ), позволяет выбрать, какие компоненты редактора Unity вы хотите загрузить и установить (рис. 3.11). Если вы не уверены, какие компоненты вы хотите установить, оставьте настройки по умолчанию и нажмите «Продолжить», следуя инструкциям программы установки [19].

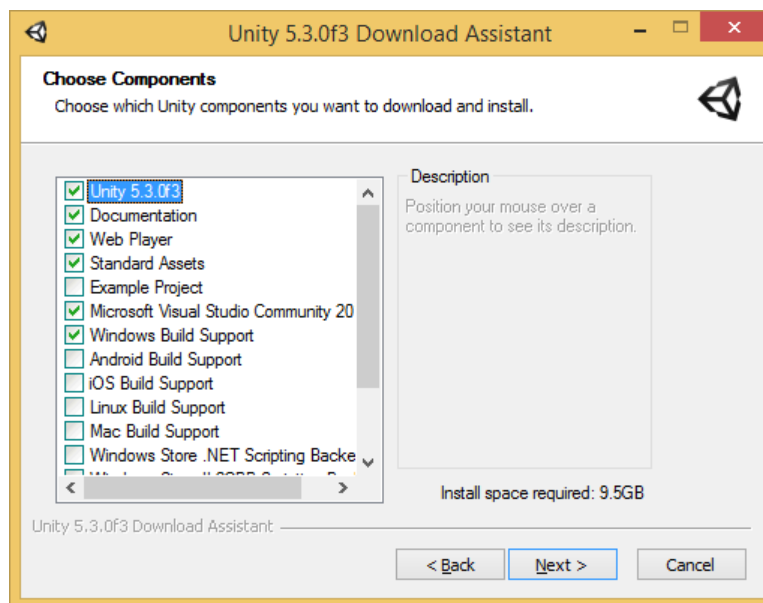


Рисунок 3.11 – Установка Unity3D.

Рабочий процесс для добавления Targets объектов в проект Vuforia Unity очень похож на типы изображений Target.

Первый шаг – это импортировать расширение в Vuforia. Расширение будет обеспечивать предварительно заполненную и сконфигурированную сцену Unity для распознавания объектов.

Чтобы импортировать расширение либо выберите Asset / Import Package в редакторе Unity или дважды щелкните файл расширения в вашей файловой системе.

Префабы, используемые для распознавания объектов можно найти в Assets / Vuforia / Prefabs внутри панели проекта.

Добавление базы данных устройства, содержащего целевой объект:

1. Создание базы данных устройств в Target Manager, который включает в себя целевой объект, который хотим использовать.

2. Загрузить эту базу данных в формате * .unitypackage.
3. Либо импортировать .unitypackage из Asset > Import Package . в редакторе Unity, либо дважды щелкнув файл пакета в файловой системе. База данных будет добавлен в Streaming Assets / QCAR

Примечание: с введением ObjectTarget мы также вводим новый тип конфигурационного файла, который будет скопирован в редактор / QCAR [19].

3.2.2. Создание сцены. Изучение интерфейса. Импорт активов

На Начальном экране создания проекта можно дать название, выбрать опции, а также указать местоположение нового проекта (рис. 3.12).

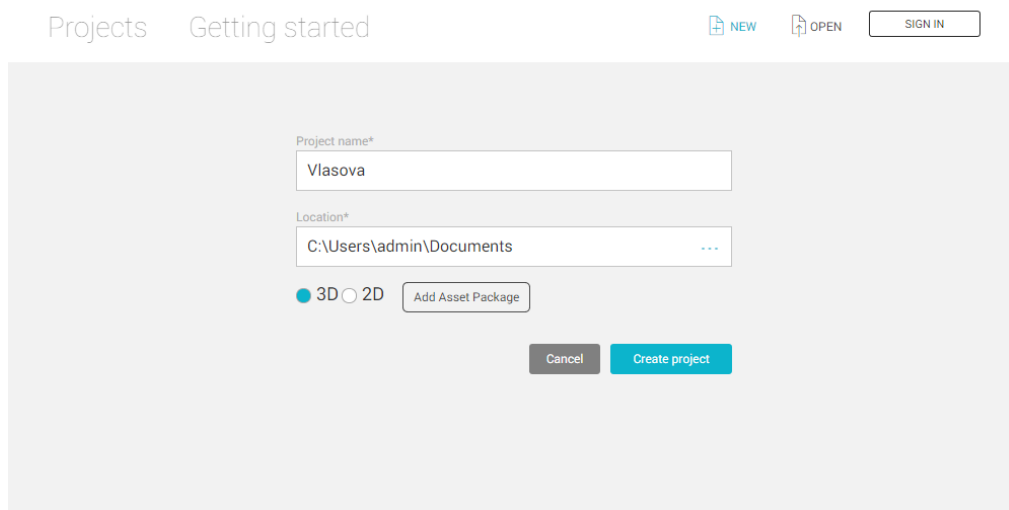


Рисунок 3.12 – Экран создания проекта Unity.

Чтобы создать новый проект:

1. По умолчанию имя проекта New Unity, но вы можете изменить его на то, что вы хотите. Введите имя, которым вы хотите назвать свой проект в нужное поле.
2. По умолчанию расположение в вашей домашней папке на вашем компьютере, но вы можете изменить его. Нажмите на три синие точки «...».
3. Затем, в Finder или File Explorer, выберите папку проекта, который вы хотите сохранить свой новый проект, и выберите «Выбрать».

4. Выберите тип 3D или 2D для вашего проекта. По умолчанию 3D красного цвета, чтобы показать, что она выбрана. (Если вы не знаете, что выбрать, оставьте его в 3D, вы сможете изменить эту настройку позже.)
5. Существует возможность выбора пакетов Asset. Пакеты активов предварительно загружены, такие как изображения, стили, эффекты освещения, и в игре управлении персонажем, среди многих других полезных игр, создавая инструментов и контента. Пакеты активов, предлагаемых здесь бесплатно, в комплекте с Unity, который вы можете использовать, чтобы начать работу над проектом. Если вы не хотите импортировать эти пакетные активы сейчас, или не уверены, просто игнорируйте эту опцию. Вы можете добавить эти активы и многие другие позже с помощью редактора Unity. Или, если вы хотите импортировать эти пакетные активы сейчас, выберите пакеты активов ..., чтобы отобразить список доступных активов, выберете те, которые вы хотите, а затем нажмите на Done.
6. Теперь выберите «Create project».

Главное окно редактора состоит из нескольких вкладок, называемые Виды (Views). В Unity есть несколько типов видов - все они предназначены для конкретных целей. Это означает, что внешний вид редактора может отличаться от одного проекта к другому, и одного разработчика к другому, в зависимости от личных предпочтений и типа работы, которую вы делаете. Расположение по умолчанию окон дает практический доступ к наиболее распространенным окнам. Вы можете идентифицировать их, по имени на вкладке. Наиболее распространенные и полезные окна показаны на рисунке 3.13.

- The Project Window (Окно проекта) отображает библиотеку активов, которые доступны для использования в вашем проекте. При импорте активов в свой проект, они появляются здесь.
- The Scene View (Сцена) позволяет визуальнo перемещаться и редактировать сцены. Вид сцены может показывать 3D или 2D перспективу, в зависимости от типа проекта.

- The Hierarchy Window (Иерархия Окон) представляет собой иерархическое текстовое представление каждого объекта в сцене. Каждый элемент в сцене имеет запись в иерархии, так что два окна неразрывно связаны между собой. Иерархия раскрывает структуру, как объекты присоединены друг к другу.
- The Inspector Window (Окно Инспектора) позволяет просматривать и редактировать все свойства выбранного объекта. Поскольку различные типы объектов имеют разные наборы свойств, расположение и содержание окна инспектора будут меняться.
- The Toolbar (Панель инструментов) обеспечивает доступ к наиболее важным рабочим функциям. Слева она содержит основные инструменты для манипулирования вида сцены и объектов в ней. В центре кнопки воспроизведения, паузы и шага управления. Кнопки справа дают вам доступ к Unity Cloud Services, затем меню видимости слоя, и, наконец, меню макет редактор (который предоставляет некоторые альтернативные макеты для окон редактора, и позволяет сохранить свой собственный макеты) [19].

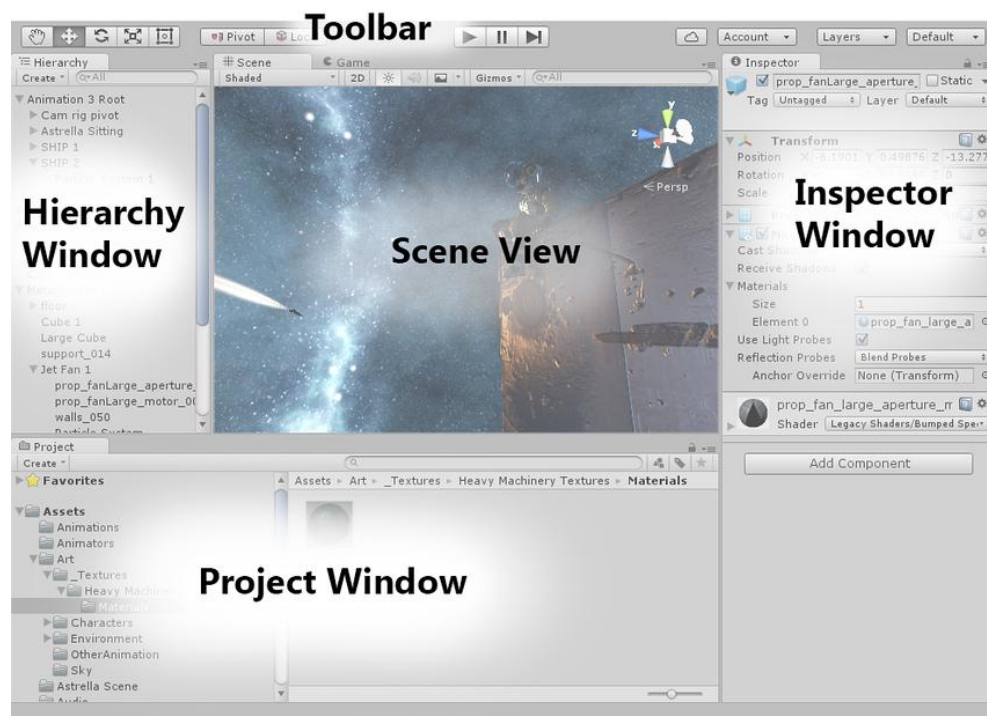


Рисунок 3.13 – Главное окно редактора Unity.

Asset является представлением любого элемента, который может быть использован в вашей игре или проекте. Asset может поступать из файла, созданного вне Unity, такие как 3D-модель, аудиофайл, изображение, или любые другие типы файлов, которые поддерживают Unity. Есть также некоторые виды asset, которые могут быть созданы в рамках Unity, такие как контроллер аниматор, аудиомикшер или рендер текстуры.

Для многих распространенных форматов, вы можете сохранить исходный файл непосредственно в папку Assets вашего проекта и Unity будет в состоянии прочитать его. Unity будет заменен при сохранении новых изменений в файл и повторно импортировать по мере необходимости. При создании проекта в Unity, вы создаете папку проекта - который содержит следующие подкаталоги (рис. 3.14):

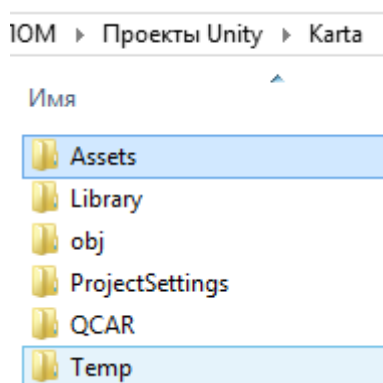


Рисунок 3.14 – Основная структура проекта в Unity.

Assets – это папка, где вы можете сохранить или скопировать файлы, которые вы хотите использовать в вашем проекте.

Содержимое окна проекта в Unity показывает элементы в вашей папке Assets. Так, что, если вы хотите сохранить или скопировать файл в папку Assets, он будет импортирован и станет видимым в окне Project.

В папке Assets есть .meta файлы, перечисленные в файловой системе, но они не отображаются в окне проекта Unity. Unity создает эти файлы для каждого актива и папок, но они скрыты по умолчанию, так, что вы не можете увидеть их. Они содержат важную информацию о том, как актив используется в проекте, и они должны оставаться с файлом активом в одном месте, так что

если вы переместите или переименуете файл актива, то необходимо переместить или переименовать метафайл соответственно.

Самый простой способ безопасно переместить или переименовать свои активы, это всегда делать это внутри папки проекта Unity. Таким образом, Unity будет автоматически перемещать или переименовывать соответствующий метафайл.

Каждый тип актива, который поддерживает Unity имеет множество настроек импорта, которые влияют на то, как представляется актив или ведет себя. Чтобы просмотреть параметры импорта актива, выберите актив в Project View. Параметры импорта для данного актива будут отображаться в инспекторе (Inspector) (рис. 3.15). Параметры, которые отображаются, будут варьироваться в зависимости от типа актива, который был выбран.

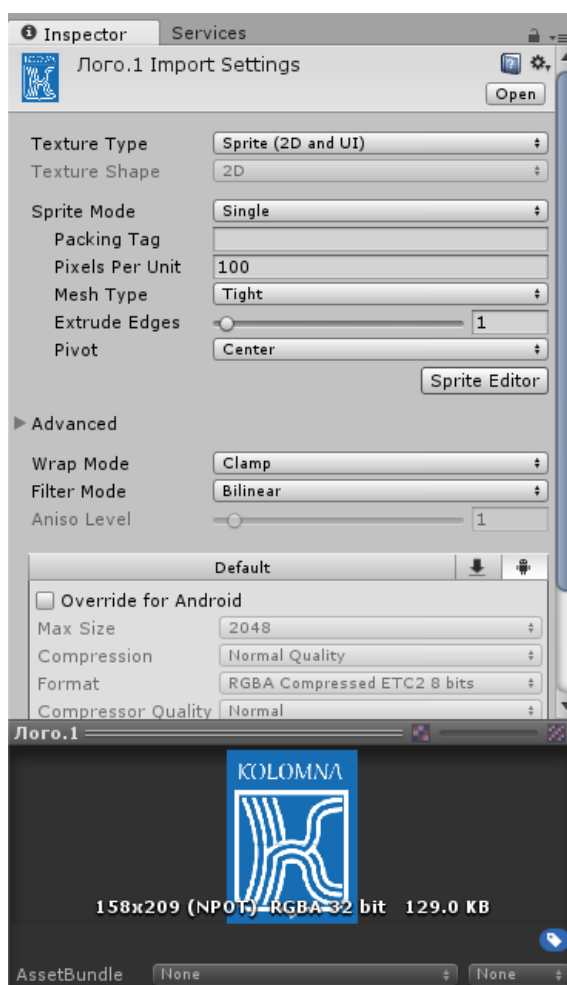


Рисунок 3.15 – Окно инспектора для редактирования активов.

Например, параметры импорта для изображения позволят вам выбрать будет ли оно импортировано в качестве текстуры, 2D-спрайтов или обычной картинки. Параметры импорта для файла FBX позволяют регулировать масштаб, создавать нормали или координаты и т.д [19].

3.2.3. Создание сцены и мишеней

Сцены содержат объекты приложения. Они могут быть использованы для создания главного меню, отдельных уровней и т.д. При создании нового проекта Unity, перед вами откроется проект с новой пустой сценой. Это безымянные и несохраненные сцены. Сцена будет пустой для объектов default кроме: камеры, или AR-камеры и направленный свет, в зависимости от того, начались ли вы проект в режиме 2D или 3D.

GameObject является наиболее важным понятием в редакторе Unity. Каждый объект в игре является GameObject. Это означает, что все, что вы можете создать в сцене должно быть GameObject. Однако, GameObject сам по себе ничего делать не может. Ему нужно задать определенные свойства, прежде чем он станет средой, специальным эффектом и т.д.

Для того, чтобы начать создавать сцену для AR-приложения, нужно сделать замену камеры. По умолчанию в сцене находится Main Camera, нам же понадобится ARCamera. Находится она в окне проекта во вкладке Vuforia/Prefabs. Теперь в сцене есть камера, при помощи которой мы будем видеть всю магию дополненной реальности.

После замены камеры в сцену можно добавлять таргеты. Находятся они в той же самой вкладке, где и камера и называются ImageTarget. Для добавления мишени в сцену ее достаточно перетащить левой кнопки мыши в сцену проекта. В приложении для городского ориентирования их будет 6: центральный и главный таргет – это карта города Коломна, остальные 5 таргетов служат для помощи в ориентировании. Они представляют из себя

фотографии мест в городе с названием улиц, расположенных недалеко от какой-либо достопримечательности.

Для того, чтобы мишень работала, во вкладке Inspector для каждого ImageTarget нужно выбрать базу, заранее скаченную с Vuforia SDK (рис. 3.16). В пункте Image Target Behaviour (Script) нажимаем на выпадающее меню с вариантами баз в подпункте Database и Image Target.

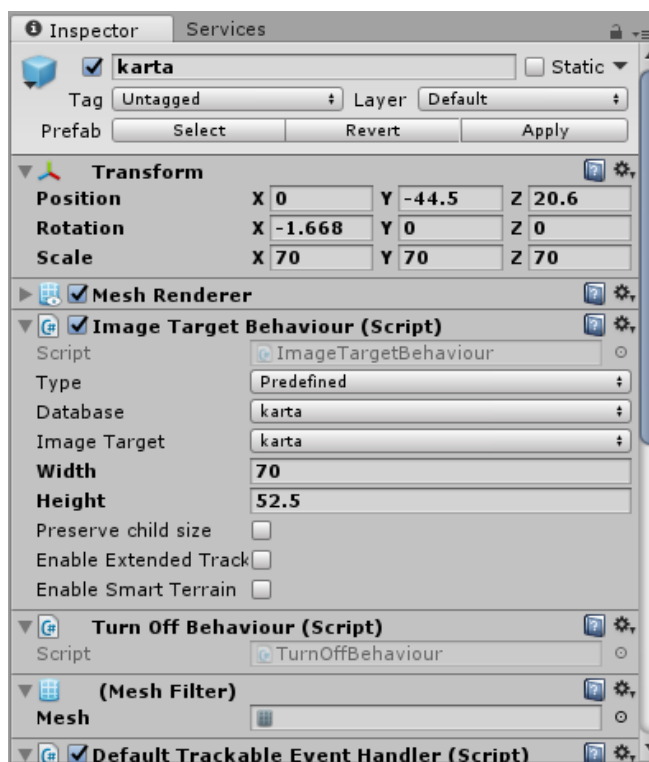


Рисунок 3.16 – Выбор базы для ImageTarget.

Следующим этапом разработки является наложение 3D моделей на уже созданные мишени. Ранее уже был произведен импорт всех материалов: 3D моделей, текстур, картинок, баз данных с Vuforia и т.д. в папке Assets создаем папку для хранения всех 3D моделей – Models (рис.3.17).

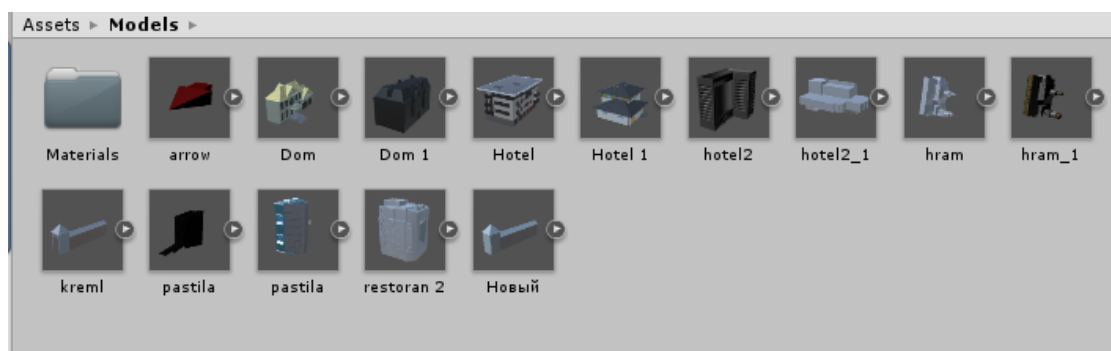


Рисунок 3.17 – Расположение 3D моделей в проекте.

Масштабировать, перемещать и поворачивать модели можно при помощи панели инструментов, находящейся в верхней части рабочей области.

Панель состоит из семи основных элементов управления. Каждый относится к различным частям редакторов.



– инструменты преобразования;



– transform контейнер, влияют на дисплей сцены;



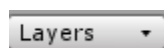
– воспроизведение / пауза / шаг;



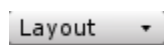
– кнопка «Облако» открывает окно Unity Services;



– используется для доступа к учетной записи;



– элементы управления, отображающиеся в сцене;



– схема расположения элементов управления [19].

Для дальнейшей работы с моделями требуется создать для них теги. Менеджер тегов и слоев (Tags and Layers Manager) позволяет вам настраивать Layers, Tags и Sorting Layers (слои, теги и слои сортировки). Чтобы просмотреть менеджер, выберите меню Edit > Project Settings > Tags and Layers. Теги (Tags) - маркеры, которые могут использоваться для идентификации объектов в вашем проекте. Добавить новые теги можно путём ввода нового тега в пустую форму внизу списка тегов или увеличив значение Size. Уменьшение Size удалит теги из конца списка.

Слои сортировки (Sorting Layers) используются в совокупности со спрайтовой графикой в 2D системе. Под “сортировкой” подразумевается порядок наложения разных спрайтов. Вы можете добавлять или убирать сортировочные слои используя кнопки +/- внизу слева списка и изменить их порядок перетягивая за левый край элемента слоя.

Слои (Layers) используются в Unity как способ группирования объектов, которые разделяют определённые характеристики. В основном слои используются для запрещения различных операций, вроде рейкастинга или рендеринга, таким образом, чтобы это касалось только определённой группы объектов. В менеджере, первые 8 слоёв по умолчанию используются Unity и их

нельзя изменить. Однако, слоям с 8 по 31 можно задать имя, просто введя его в соответствующую текстовую форму [19].

Добавляем каждой модели тег в соответствии с ее названием (рис. 3.18).

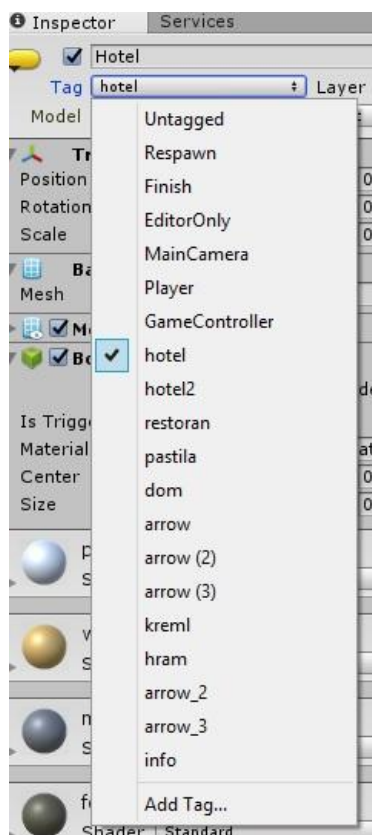


Рисунок 3.18 – Добавление тега (идентификатора) к 3D модели.

3.2.4. Разработка пользовательского интерфейса

Для взаимодействия пользователя и приложения необходим интерфейс.

Canvas это область, внутри которой находятся все элементы пользовательского интерфейса (рис. 3.19). Так же все элементы интерфейса должны являться «детьми» Canvas.

Для создание нового элемента пользовательского интерфейса, таких как изображения, кнопки и так далее, используем меню `GameObject > UI > Image`. Автоматически создается Canvas, если его еще не было в сцене. Элемент пользовательского интерфейса создается как ребенок к этому Canvas.

Область Canvas показана в виде прямоугольника в сцене. Это позволяет легко позиционировать элементы пользовательского интерфейса без

необходимости иметь Game View видимым постоянно. Canvas использует объект EventSystem, чтобы помочь системе в обмене сообщениями.



Рисунок 3.19 – Сцена информационного окна, созданная через Canvas.

На главной сцене приложение присутствуют 2 панели с кнопками. Правое меню позволяет манипулировать объектами, а именно, выбрать одну из трех категорий (рис. 3.20). Первая кнопка меню позволяет отображать все категории на карте. Вторая – оставляет в поле видимости только объекты категории «Достопримечательности». Третья – категории «Места отдыха» и четвертая – «События». Последняя кнопка «Выход» осуществляет выход из приложения. Левое контекстное меню служит в вспомогательных целях. Левая кнопка выдает окно с инструкцией эксплуатации данного приложения, а правая – дает общие сведения о городе Коломна. В информационных вспомогательных сценах (пример приведен на рис. 3.19) присутствует кнопка «Назад», позволяющая пользователю вернуться на главный экран. Информационное окно для все моделей сделано по единому шаблону. Для создания дизайна кнопок был использован Adobe Illustrator CC 2017.

Иллюстратор уже давно является стандартом, когда речь идет о векторной иллюстрации и дизайна. С момента своего первого выпуска в далеком 1987 году, это было идеально для тех, кто хочет создать бесконечно масштабируемые графики. Во время своего путешествия к своей текущей CC 2017 инкарнации - всего через 30 лет после первоначального выпуска -

приложение превратилось в нечто большее, чем инструмент «иллюстрация». Преимуществом данного контекстного меню состоит в том, что была произведена полная привязка каждого элемента к Canvas. Кнопку Anchor Preset можно найти в верхнем левом углу компонента Rect Transform. При нажатии кнопки появляется выпадающий список Anchor Preset. Здесь вы можете быстро выбрать некоторые из наиболее распространенных вариантов привязок. Вы можете закрепить элемент пользовательского интерфейса для сторон или середины родителя, или растягиваться вместе с родительским размером [2].

Когда якоря разделены, поля могут изменяться частично или полностью влево, вправо, вверх и вниз. Эти поля определяют отступы внутри прямоугольника, определенного якорями. Поля влево и вправо используются, если якоря разделены горизонтально, а сверху и снизу поля используются, если они разделены по вертикали.

Следует отметить, что изменение значений в анкерных или шарнирных полях, как правило, встречным настроить значение позиционирования для того, чтобы разместить прямоугольник на нужное место. В случаях, когда это не требуется, режим Raw можно включить с помощью небольшой кнопки в инспекторе. Это приводит к тому, что якоря и значение поворота, чтобы иметь возможность быть изменены без каких - либо других значений, изменяющихся в результате.



Рисунок 3.20 – Меню выбора категорий.

Без программного кода кнопки работать не будут. Для этого следует написать Script. Scripting является одним из основных этапов, при создании приложения. Даже самое простое приложение нуждается в сценарии, чтобы ответить на какие-либо действия пользователя. Кроме того, сценарии могут быть использованы для создания графических эффектов, контролировать физическое поведение объектов или даже создать пользовательскую систему искусственного интеллекта.

Unity позволяет создавать свои собственные компоненты с помощью сценариев. Это позволит вам вызвать события приложения, изменять свойства компонентов с течением времени и реагировать на ввод данных пользователем.

Unity поддерживает два языка программирования:

- C#, отраслевой стандарт языка похож на Java или C++;
- UnityScript, язык, разработанный специально для использования с Unity и смоделирован JavaScript.

В отличие от большинства других активов, сценарии обычно создаются непосредственно в Unity. Вы можете создать новый сценарий из меню «Создать» в левом верхнем углу панели проекта или выбрав Assets > Create > C# Script (или JavaScript) из главного меню. Новый сценарий будет создан в зависимости от того, какая папка выбрана в панели Project. При двойном щелчке актива скрипт в Unity, он будет открыт в текстовом редакторе. По умолчанию, Unity будет использовать MonoDevelop. Начальное содержимое файла будет выглядеть следующим образом (рис. 3.21) [19]:


```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class NewBehaviourScript : MonoBehaviour {
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12    // Update is called once per frame
13    void Update () {
14
15    }
16 }
17

```

Рисунок 3.21 – Начальное содержимое файла Script.

Основные вещи, которые стоит отметить, эти две функции, определенные внутри класса. Update функция является местом, чтобы поместить код, который будет обрабатывать обновление кадров для GameObject. Это может включать в себя движение, вызывая действия и реагировать на ввод пользователя, в основном все, что должно быть обработано в течение долгого времени во время игры. Start функция будет вызываться Unity перед началом приложения (то есть до того, как функция Update вызывается в первый раз) и является идеальным местом, чтобы сделать инициализацию [18].

Программный код для подключения библиотек и обозначения переменных приведен в листинге 3.1.

Листинг 3.1 «Подключение библиотек. Обозначения переменных»

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
public class Script1 : MonoBehaviour {

    public GameObject hotel2;
    public GameObject Hotel;
    public GameObject kreml;
    public GameObject Restoran;
    public GameObject arrow;
    public GameObject arrow_2;
    public GameObject arrow_3;

```



```
public GameObject pastila;  
public GameObject Dom;  
public GameObject hram;
```

В приложении при нажатии на любую из моделек, появляющихся на экране, можно перейти на страничку с дополнительной информацией о данном объекте. Луч из камеры чаще всего используют для совершения рейкаста в сцену. Рейкаст отправляет воображаемый “лазерный луч” вдоль программного луча из точки испускания до тех пор, пока он не встретит на пути коллайдер в сцене. В результате столкновения с коллайдером, возвращается RaycastHit с координатами и объектом столкновения. Это очень полезный способ нахождения объекта по его изображению на экране. Для использования данного метода в программном коде описывается нахождение моделей через созданные оригинальные теги. При столкновении луча с объектов, он определяет его тег и перенаправляет пользователя на нужную страницу с информацией (листинг 3.2).

Листинг 3.2 «Создание луча. Переход на информационную страницу»

```
void Update()  
{  
    hotel2 = GameObject.FindGameObjectWithTag("hotel2");  
    Hotel = GameObject.FindGameObjectWithTag("hotel");  
    kreml = GameObject.FindGameObjectWithTag("kreml");  
    Restoran = GameObject.FindGameObjectWithTag("restoran");  
    arrow = GameObject.FindGameObjectWithTag("arrow");  
    arrow_2 = GameObject.FindGameObjectWithTag("arrow_2");  
    arrow_3 = GameObject.FindGameObjectWithTag("arrow_3");  
    pastila = GameObject.FindGameObjectWithTag("pastila");  
    Dom = GameObject.FindGameObjectWithTag("dom");  
    hram = GameObject.FindGameObjectWithTag("hram");  
    if(Input.GetMouseButtonDown(0))  
    {  
        Ray ray = Camera.main.ScreenPointToRay(new Vector3(Input.mousePosition  
.x, Input.mousePosition.y, Input.mousePosition.z));  
        RaycastHit hit;
```

```
        if (Physics.Raycast (ray, out hit, Mathf.Infinity))
    {
        if (hit.transform.tag == "hotel")
        {
            SceneManager.LoadScene ("info_hotel");
        }
        if (hit.transform.tag == "hotel2")
        {
            SceneManager.LoadScene ("info_hotel2");
        }
        if (hit.transform.tag == "arrow")
        {
            SceneManager.LoadScene ("info_arrow");
        }
        if (hit.transform.tag == "arrow_2")
        {
            SceneManager.LoadScene ("info_arrow_2");
        }
        if (hit.transform.tag == "arrow_3")
        {
            SceneManager.LoadScene ("info_arrow_3");
        }
        if (hit.transform.tag == "kreml")
        {
            SceneManager.LoadScene ("info_kreml");
        }
        if (hit.transform.tag == "pastila")
        {
            SceneManager.LoadScene ("info_pastila");
        }
        if (hit.transform.tag == "dom")
        {
            SceneManager.LoadScene ("info_dom");
        }
        if (hit.transform.tag == "hram")
        {
            SceneManager.LoadScene ("info_hram");
        }
        if (hit.transform.tag == "restoran")
        {
            SceneManager.LoadScene ("info_restoran");
        }
    }
```

Действие при нажатии на кнопку «Все категории» вывод на экран все скрытые модели. Программный код представлен в Листинге 3.3. Для этого создаем новую функцию, которой указываем, что если та или иная модели в данный момент скрыты, то ее нужно отобразить. Программный код для остальных кнопок контекстного меню в правой и левой частях экрана представлены в Приложении Б.

Листинг 3.3 «Действие при нажатии на кнопку «Все категории»

```
public void fun1()
{
    if (hotel2.GetComponent<Renderer>().enabled == false)
    {
        hotel2.GetComponent<Renderer>().enabled = true;
    }
    if (Hotel.GetComponent<Renderer>().enabled == false)
    {
        Hotel.GetComponent<Renderer>().enabled = true;
    }
    if (kreml.GetComponent<Renderer>().enabled == false)
    {
        kreml.GetComponent<Renderer>().enabled = true;
    }
    if (Restoran.GetComponent<Renderer>().enabled == false)
    {
        Restoran.GetComponent<Renderer>().enabled = true;
    }
    if (arrow.GetComponent<Renderer>().enabled == false)
    {
        arrow.GetComponent<Renderer>().enabled = true;
    }
    if (arrow_2.GetComponent<Renderer>().enabled == false)
    {
        arrow_2.GetComponent<Renderer>().enabled = true;
    }
    if (arrow_3.GetComponent<Renderer>().enabled == false)
    {
        arrow_3.GetComponent<Renderer>().enabled = true;
    }
    if (pastila.GetComponent<Renderer>().enabled == false)
```

```
{
    pastila.GetComponent<Renderer>().enabled = true;
}
if (Dom.GetComponent<Renderer>().enabled == false)
{
    Dom.GetComponent<Renderer>().enabled = true;
}
if (hram.GetComponent<Renderer>().enabled == false)
{
    hram.GetComponent<Renderer>().enabled = true;
}
}
```

3.2.5. Уменьшение размера файла сборки. Build Settings

Сведение размера файла встроенного приложения к минимуму является важным аспектом, особенно для мобильных устройств или для магазинов приложений, которые налагают ограничение на размер. Первым шагом в уменьшении размера является определение, какие активы вносят наибольший вклад, потому что эти активы являются наиболее вероятными кандидатами на оптимизацию. Эта информация доступна в редакторе только после того, как вы выполнили сборку. Перейдите в окно консоли (меню: Window < Console), нажмите на маленькую раскрывающуюся панель в правом верхнем углу, и выберите Open Editor Log.

Редактор журнала предоставляет сводку активов в разбивке по типу, а затем перечисляются все отдельные активы в порядке размера вклада. Как правило, такие вещи, как текстуры, звуки и анимация занимают большую часть памяти, в то время как скрипты, уровни и шейдеры обычно имеют наименьшее влияние. Заголовки файлов, указанные в списке, не являются активами - они на самом деле дополнительные данные, которые добавляют в «сырые» файлы активов для хранения ссылок и настроек.

Редактор журнала поможет вам определить активы, которые вы можете удалить или оптимизировать, но вы должны учитывать следующее:

- В Unity импортировали активы в свои собственные внутренние форматы, поэтому выбор типа источника активов не является актуальным. Например, если у вас есть многослойные Photoshop текстуры в проекте, то они сплющиваются и сжимаются перед сборкой. Экспорт текстуры как .png файл не имеет никакого значения для построения размера, так что вы должны придерживаться формата, который наиболее удобен для вас в процессе разработки.
- Unity наиболее неиспользованных активов в течение сборки, так что вручную удалить активы из проекта не получится. Единственные активы, которые не удаляются скрипты, которых, как правило, очень мало в любом случае, и активы в папке Resources (потому что Unity не может определить, какие из них необходимы, а какие нет). Имея это в виду, вы должны убедиться в том, что только активы, находящиеся в папке Resources, являются теми, что нужны для приложения [19].

Текстуры обычно занимают большую часть пространства в сборке. Первое решение этой проблемы состоит в использовании сжатых форматов текстур. Если это не уменьшает размер файла достаточно, нуэно попытаться уменьшить физический размер (в пикселях) текстуры изображений. Чтобы сделать это без изменения фактического содержания источника, выберите текстуру в окне проекта и в окне инспектора уменьшить максимальный размер. Изменение максимального размера текстуры не влияет на текстуры актива, только его разрешение в игре.

Окно Build Settings позволяет выбрать целевую платформу, настроить параметры для сборки и запуска процесса сборки. Чтобы получить доступ к окну Build Settings, выберите File > Build Settings.... После того, как вы определили параметры сборки, вы можете нажать кнопку Build для создания сборки, или нажмите Build And Run, чтобы создать и запустить сборку на платформе, которую вы указали (рис. 3.22).

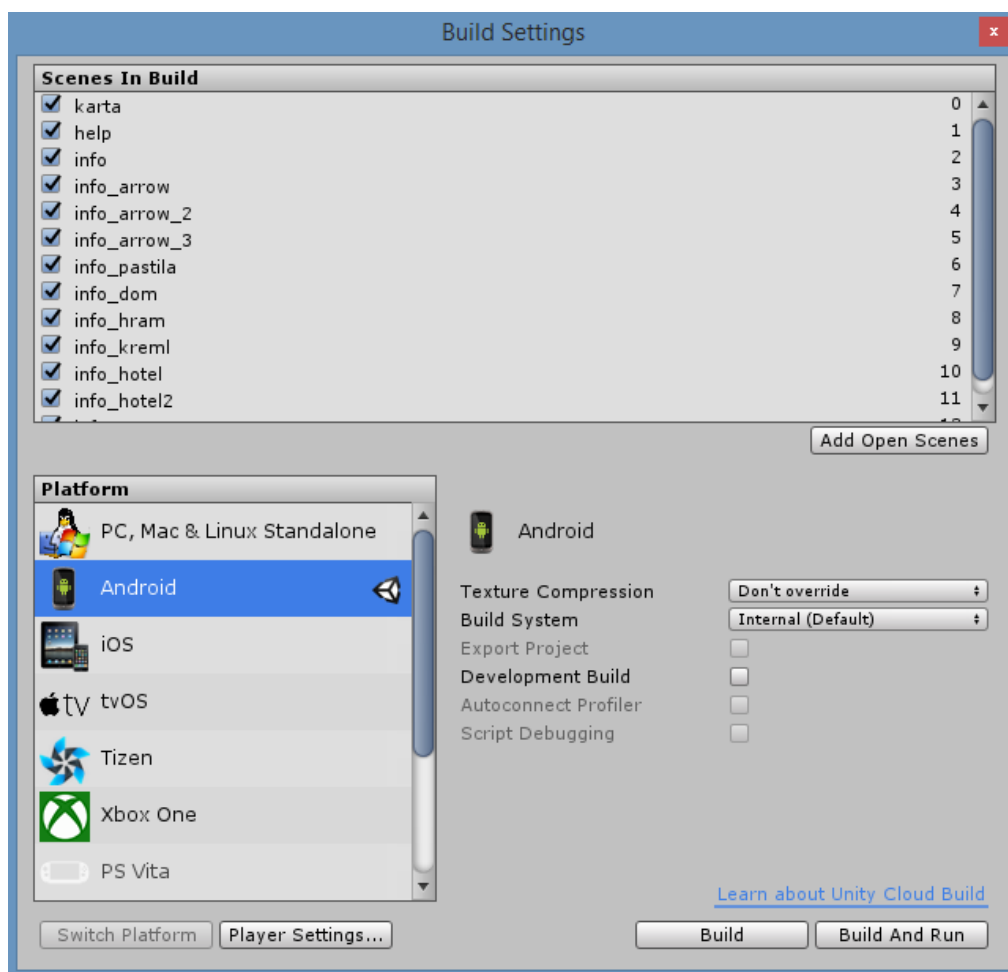


Рисунок 3.22 – Окно Build Settings.

Scenes in Build. Эта часть окна показывает сцены из вашего проекта, которые будут включены в сборку. Если нужная сцена не отображается, то вы можете использовать кнопку «Add Current» (Добавить текущую), чтобы добавить текущую сцену в сборку, или вы можете перетаскивать ресурсы сцены в это окно из окна проекта. Вы также можете убрать галочку сцены в этом списке, чтобы исключить ее из сборки, не удаляя ее из списка. Список сцен будет использоваться для управления порядком сцены загрузки. Вы можете изменить порядок сцен, перемещая их вверх или вниз.

Площадь платформы под сцену в списке области сборки всех платформ, которые доступны для вашей версии Unity. Некоторые платформы могут быть неактивны, чтобы указать, что они не являются частью вашей версии или пригласить вас для загрузки платформы конкретных параметров сборки. Выбор

одной из платформ будет контролировать, какая платформа будет построена. Чтобы изменить целевую платформу, вам необходимо нажать на кнопку «Switch Platform», чтобы применить изменения. Это может занять некоторое время, делая переход, потому что ваши активы, возможно, потребуется повторно импортировать в форматах, которые соответствуют вашей целевой платформе. Выбранная в данный момент платформа обозначается значком Unity справа от названия платформы. Для выбранной платформы будет показан список опций, которые могут быть скорректированы для сборки. Каждая платформа может иметь различные варианты. Эти параметры перечислены ниже [19].

3.3. Вывод к главе 3

В данной части пояснительной записки были рассмотрены все этапы разработки приложения, инструменты по созданию 3D моделей, маркеров (мишеней), импорта данных в Unity, а также сжатие и «постройка» приложения. Подробно рассмотрены способы создания пользовательского интерфейса, создание Script для работы интерфейса. Завершением данного этапа разработки стало создание руководства пользователя.

ЗАКЛЮЧЕНИЕ

В ходе дипломного проектирования была поставлена следующая цель: создание мобильного приложения для городского ориентирования по конкретному туристическому кластеру, которое поможет путешественникам выбрать наиболее интересные для изучения объекты и эргономично рассчитать время для их посещения.

Реализация данной цели потребовала содержательного и методического решения следующих задач:

- были рассмотрены понятия и сферы применения технологии «дополненной реальности»;
- был произведен анализ и выбор программного обеспечения для создания AR-приложения;
- был создан опытный образец мобильного приложения для системы Android.

В первой главе была подробно изучена предметная область, сформулированы требования к конечному продукту, были подробно описаны технологии для его разработки. Результатом аналитической работы стало создание технического задания.

Во второй главе были рассмотрены технологии и программное обеспечение, которые использовались в процессе разработки приложения для городского ориентирования с элементами дополненной реальности. В результате анализа было выбрано следующее ПО для разработки приложения:

- Фреймворк Vuforia SDK;
- Кроссплатформенный движок Unity3D;
- Графический 3D редактор Blender.

Так же была рассмотрена область исследования проекта с представлением примеров используемого материала.

В третьей главе были рассмотрены все этапы разработки приложения, инструменты по созданию 3D моделей, маркеров (мишеней), импорта данных в

Unity, а также сжатие и «постройка» приложения. Подробно рассмотрены способы создания пользовательского интерфейса, создание Script для работы интерфейса. Завершением данного этапа разработки стало создание руководства пользователя.

СПИСОК ЛИТЕРАТУРЫ

1. 3ds max 6 / И.Н. Чумаченко. - Москва: ДМК Пресс, 2004. - 416 с.
2. Adobe Illustrator CC 2017 // Уроки и руководства
URL: <http://www.design-lessons.info/illustrator/adobe-illustrator>.
3. Advances in Modelling, Animation and Rendering / John Vince, Rae A. Earnshaw. - Москва: Мир, 2004. - 440 с.
4. Blender – горячие клавиши // Роман Филенко
[Электронный ресурс] URL:
<http://roman.filenko.ru/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D0%B8/3d/5-blender-hotkeys.html>Lightwave.
5. 3D 8 Cartoon Character Creation, Volume 1: Modeling & Texturing (LightWave 3D 8 Cartoon Character Creation) / Jonny Gorden. - Москва: Мир, 2004. - 496 с.
6. Mental Ray / Iray. Мастерство визуализации в Blender3D / А. Плаксин. - Москва: ДМК Пресс, 2014. - 105 с.
7. Update to Blender3D Training CD / Laurent M. Abecassis. - Москва: РГГУ, 2006. - 836 с.
8. Vuforia: немного магии в нашей реальности / Сайт компания EligoVision (ООО «ЭлигоВижн»), раздел R&D и ноу-хау – Vuforia. [Электронный ресурс]. URL:
<https://habrahabr.ru/post/198862/>.
9. Web Animation Expert: All That You Need to Create Your Own Fantastic Web Animations / Graham Davis. - Санкт –Петербург: 2010. - 562 с.
10. Storytelling through Animation (Graphics) / Mike Wellins. - Москва: Высшая школа, 2005. - 435 с.
11. 7 плюсов и минусов Unity // Авторские статьи об OpenSource
[Электронный ресурс] URL:
http://vasilisc.com/unity_criticism.
12. Дизайн архитектуры и интерьеров в 3ds Max Design 2010 / О. Миловская. - Москва: БХВ-Петербург, 2010. - 502 с.

13. Дополненная реальность: пространство между реальностью и виртуальностью. / Подберезкина А. [Электронный ресурс]. URL: <http://zillion.net/ru/blog/236/dopolniennaia-rieal-nost-prostranstvo-miezhdu-rieal-nost-iu-i-virtual-nost-iu/>.
14. Достопримечательности Коломны // Тонкости Туризма [Электронный ресурс] URL: https://tonkosti.ru/Достопримечательности_Коломны.
15. Интерактивные технологии / Сайт компания EligoVision (ООО «ЭлигоВижн»), раздел [EV]³ R&D и ноу-хау — AR студия. [Электронный ресурс]. URL: <http://www.eligovision.ru/inventions/ar-studio/>.
16. История, особенности и перспективы технологии дополненной реальности // Яковлев Б.С., Пустов С.И. // Известия ТулГУ. Технические науки. 2013. №3 . [Электронный ресурс] // URL: <http://cyberleninka.ru/article/n/istoriya-osobennosti-i-perspektivy-tehnologii-dopolnennoy-realnosti/>.
17. На пороге дополненной реальности: к чему готовиться разработчикам. Бабичев А. [Электронный ресурс] URL: <http://lib.custis.ru/213-Waiting-Augmented-Reality-add-2010/>.
18. Практика разработки систем измерения и управления на C# / Ю.С. Магда. - Москва: ДМК Пресс, 2016. - 190 с.
19. Руководство Unity // Unity DOCUMENTATION [Электронный ресурс] URL: <https://docs.unity3d.com/ru/current/Manual/index.html>.
20. Самоучитель Blender3D (+ DVD-ROM) / Ольга Миловская. - Москва: БХВ-Петербург, 2008. - 336 с.
21. Что такое «дополненная реальность»? // Журнал Chip №8 (август 2010) [Электронный ресурс] URL: http://www.ichip.ru/stati/praktika/2010/11/chto-takoe-abdopolnennaya-realnostbb?b_start:int=1.

ПРИЛОЖЕНИЕ А
Техническое Задание

1. Наименование и область применения

Дипломный проект представляет собой мобильное приложение для городского ориентирования с элементами дополненной реальности. Основная цель, которая стоит перед разработчиком – сделать приложение максимально информативным, красочным и запоминающимся, чтобы пользователь выбрал представленный в приложении объект. В качестве объекта для разработки приложения по городскому ориентированию предлагается один из туристических центров Подмосковья – древний город Коломна. В городе находятся большое количество памятников архитектуры и истории, разноплановых музеев и мест отдыха. Приложение нацелено помочь туристам выбрать для посещения интересные для него объекты и наиболее эргономично рассчитать время их посещения.

2. Основание для разработки

Основанием для разработки является задание для дипломного проекта, утвержденное кафедрой «Информатика и информационные технологии» Высшей школы печати и медиаиндустрии Московского политехнического университета.

3. Исполнитель

Исполнителем является студент кафедры «Информатика и информационные технологии» Высшей школы печати и медиаиндустрии Московского политехнического университета группы ДЦасБ 4-1 Власова Алина Олеговна.

4. Назначение разработки

4.1 Назначение системы

Помощь путешественникам в выборе наиболее интересных для изучения объектов и эргономичного расчета времени для их посещения.

4.2 Цели создания системы

1. Исследование требований, методов и алгоритмов решения поставленной задачи.
2. Разработка структуры программного обеспечения.

3. Организация удобного интерфейса для пользователей.
4. Создание опытного образца программного обеспечения.

5. Технические требования к системе

5.1 Задачи, подлежащие решению

В процессе разработки мобильного приложения для городского ориентирования с элементами дополненной реальности необходимо решить следующие задачи:

- Описать предметную область.
- Определить функции, выполняемые системой.
- Разработать и отладить интерфейс системы.
- Создать опытный образец программного обеспечения.

5.2 Требования к функциональным характеристикам

Мобильного приложения для городского ориентирования с элементами дополненной реальности необходимо решить следующие функции:

- отображение трехмерных моделей;
- помощь путешественникам в выборе наиболее интересных для изучения объектов;
- эргономичный расчета времени для посещения объектов.

5.3 Требования к входным и выходным данным

Входными данными для приложения являются снимки, созданные в роли мишеней:

- таргет основной карты;
- таргеты снимков-помощников.

Выходными данными для приложения является информация, выдаваемая в виде вспомогательных информационных страниц:

5.4 Требования к составу программных компонентов

Мобильное приложение для городского ориентирования с элементами дополненной реальности должно состоять из следующих основных модулей:

- Форма – Сцена отображения 3D моделей;
- Форма – Пользовательский интерфейс;

- Форма – Сцены с дополнительной информацией.

5.5 Требования к компонентам программного продукта

Приложение должно реализовывать следующие функции:

- 3D модели должны иметь структуру, обеспечивающую полную функциональность системы. Разрабатывается на основе 3D графического редактора Blender.
- Маркеры должны быть полностью распознаваемыми. Разрабатывается на основе Vuforia SDK.
- Интерфейс загрузки и отображения контента должен обеспечивать диалог пользователя с системой: принятие информации, ввод и отображение информационных страничек на основании запросов пользователей. Данный модуль должен разрабатываться в среде Unity3D;

5.6 Требования к составу и параметрам технических средств

5.6.1 Требования к составу программного обеспечения

- Операционная система Windows XP/Vista/7/8/10;
- Blender 3D (любая версия)
- Unity3D 5.5;

5.6.2 Требования к аппаратному обеспечению

- процессор с тактовой частотой не менее 166 МГц;
- 64Мб памяти и выше;
- Монитор с разрешением 1280x800;

5.7 Условия эксплуатации

Мобильного приложения для городского ориентирования с элементами дополненной реальности должно быть установлено на устройстве на системе Android 4.1 и выше. Пользователь должен знать основы работы с оконным интерфейсом операционной системы Windows. Условия окружающей среды должны соответствовать требованиям по эксплуатации аппаратного обеспечения.

6. Стадии и этапы разработки

Укрупнённый план-график выполнения этапов работы при разработке мобильного приложения для городского ориентирования с элементами дополненной реальности приведён в следующей таблице:

Название этапа	Дата реализации этапа
Поиск и исследование литературы по теме дипломной работы	15.12.16-17.12.16
Исследование предметной области	19.12.16-22.12.16
Выполнение аналитической части	25.12.16-24.01.17
Настройка ПО	27.01.17-05.02.17
Создание 3D моделей	10.02.17-20.03.17
Разработка приложение на платформе Unity3D	25.03.17-20.05.17

7. Техническая документация, представляемая по окончании работы

По окончании разработки приложения для городского ориентирования с элементами дополненной реальности предъявляются следующие документы:

1. Программа;
2. Пояснительная записка.

ПРИЛОЖЕНИЕ Б
Программный код

Листинг 1 «Действие при нажатии на кнопку «Достопримечательности»

```
public void fun2()
{
    if (hotel2.GetComponent<Renderer>().enabled == true)
    {
        hotel2.GetComponent<Renderer>().enabled = false;
    }
    if (Hotel.GetComponent<Renderer>().enabled == true)
    {
        Hotel.GetComponent<Renderer>().enabled = false;
    }
    if (kreml.GetComponent<Renderer>().enabled == false)
    {
        kreml.GetComponent<Renderer>().enabled = true;
    }
    if (Restoran.GetComponent<Renderer>().enabled == true)
    {
        Restoran.GetComponent<Renderer>().enabled = false;
    }
    if (arrow.GetComponent<Renderer>().enabled == true)
    {
        arrow.GetComponent<Renderer>().enabled = false;
    }
    if (arrow_2.GetComponent<Renderer>().enabled == true)
    {
        arrow_2.GetComponent<Renderer>().enabled = false;
    }
    if (arrow_3.GetComponent<Renderer>().enabled == true)
    {
        arrow_3.GetComponent<Renderer>().enabled = false;
    }
    if (pastila.GetComponent<Renderer>().enabled == false)
    {
        pastila.GetComponent<Renderer>().enabled = true;
    }
    if (Dom.GetComponent<Renderer>().enabled == false)
    {
        Dom.GetComponent<Renderer>().enabled = true;
    }
    if (hram.GetComponent<Renderer>().enabled == false)
    {
        hram.GetComponent<Renderer>().enabled = true;
    }
}
```

Листинг 2 «Действие при нажатии на кнопку «Места отдыха»

```
public void fun3()
{
    if (hotel2.GetComponent<Renderer>().enabled == false)
    {
        hotel2.GetComponent<Renderer>().enabled = true;
    }
    if (Hotel.GetComponent<Renderer>().enabled == false)
    {
        Hotel.GetComponent<Renderer>().enabled = true;
    }
    if (kreml.GetComponent<Renderer>().enabled == true)
    {
        kreml.GetComponent<Renderer>().enabled = false;
    }
    if (Restoran.GetComponent<Renderer>().enabled == false)
    {
        Restoran.GetComponent<Renderer>().enabled = true;
    }
    if (arrow.GetComponent<Renderer>().enabled == true)
    {
        arrow.GetComponent<Renderer>().enabled = false;
    }
    if (arrow_2.GetComponent<Renderer>().enabled == true)
    {
        arrow_2.GetComponent<Renderer>().enabled = false;
    }
    if (arrow_3.GetComponent<Renderer>().enabled == true)
    {
        arrow_3.GetComponent<Renderer>().enabled = false;
    }
    if (pastila.GetComponent<Renderer>().enabled == true)
    {
        pastila.GetComponent<Renderer>().enabled = false;
    }
    if (Dom.GetComponent<Renderer>().enabled == true)
    {
        Dom.GetComponent<Renderer>().enabled = false;
    }
    if (hram.GetComponent<Renderer>().enabled == true)
    {
        hram.GetComponent<Renderer>().enabled = false;
    }
}
```

Листинг 3 «Действие при нажатии на кнопку «События»

```
public void fun4()
{
    if (hotel2.GetComponent<Renderer>().enabled == true)
    {
        hotel2.GetComponent<Renderer>().enabled = false;
    }
    if (Hotel.GetComponent<Renderer>().enabled == true)
    {
        Hotel.GetComponent<Renderer>().enabled = false;
    }
    if (kreml.GetComponent<Renderer>().enabled == true)
    {
        kreml.GetComponent<Renderer>().enabled = false;
    }
    if (Restoran.GetComponent<Renderer>().enabled == true)
    {
        Restoran.GetComponent<Renderer>().enabled = false;
    }
    if (arrow.GetComponent<Renderer>().enabled == false)
    {
        arrow.GetComponent<Renderer>().enabled = true;
    }
    if (arrow_2.GetComponent<Renderer>().enabled == false)
    {
        arrow_2.GetComponent<Renderer>().enabled = true;
    }
    if (arrow_3.GetComponent<Renderer>().enabled == false)
    {
        arrow_3.GetComponent<Renderer>().enabled = true;
    }
    if (pastila.GetComponent<Renderer>().enabled == true)
    {
        pastila.GetComponent<Renderer>().enabled = false;
    }
    if (Dom.GetComponent<Renderer>().enabled == true)
    {
        Dom.GetComponent<Renderer>().enabled = false;
    }
    if (hram.GetComponent<Renderer>().enabled == true)
    {
        hram.GetComponent<Renderer>().enabled = false;
    }
}
```

Листинг 4 «Действие при нажатии на кнопку «Выход»

```
public void fun8()
{
    Application.Quit();
}
```

Листинг 5 «Действие при нажатии на левое контекстное меню»

```
public void fun5()
{
    SceneManager.LoadScene ("karta");
}
public void fun6()
{
    SceneManager.LoadScene ("help");
}
public void fun7()
{
    SceneManager.LoadScene ("info");
}
```

ПРИЛОЖЕНИЕ В
Примеры target



Рисунок 1.1 – Рисунок с изображением метки таргета №1.



Рисунок 1.2 – Рисунок с изображением метки таргета №2.

ПРИЛОЖЕНИЕ Г
Презентация

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ



ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ
институт Принтмедиа и информационных технологий
кафедра Информатики и информационных технологий
направление подготовки 09.03.01 «Информатика и вычислительная техника»

Тема: «Разработка приложения для городского ориентирования с элементами дополненной реальности»

Выполнил: Власова Алина Олеговна

Руководитель: Попов Дмитрий Иванович д.т.н., профессор, зав.каф. ИиИТ Попов Д. И.

Выполнил:	Власова А. О.
Руководитель:	д.т.н., проф. Попов Д. И.

Цель

Создание мобильного приложения для городского ориентирования по конкретному туристическому кластеру, которое поможет путешественникам выбрать наиболее интересные для изучения объекты и эргономично использовать время для их посещения.

Задачи

- Анализ области исследования
- Исследование требований, методов и алгоритмов решения поставленной цели
- Разработка структуры программного обеспечения
- Создание опытного образца программного обеспечения

Выполнил:	Власова А. О.	Слайд №
Руководитель:	д.т.н., проф. Попов Д. И.	2

Используемое программное обеспечение



Blender 3D 2.78c



Qualcomm Vuforia



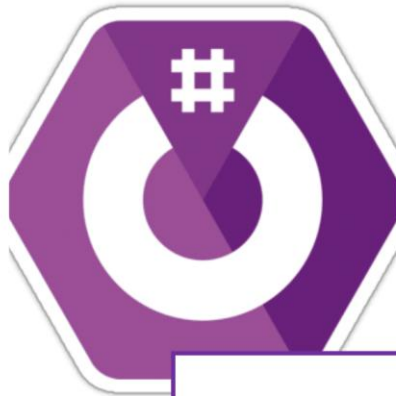
Unity 3D 5.5.1f1

Выполнил:	Власова А. О.	Слайд №
Руководитель:	д.т.н., проф. Попов Д. И.	3

Используемая технология и язык программирования



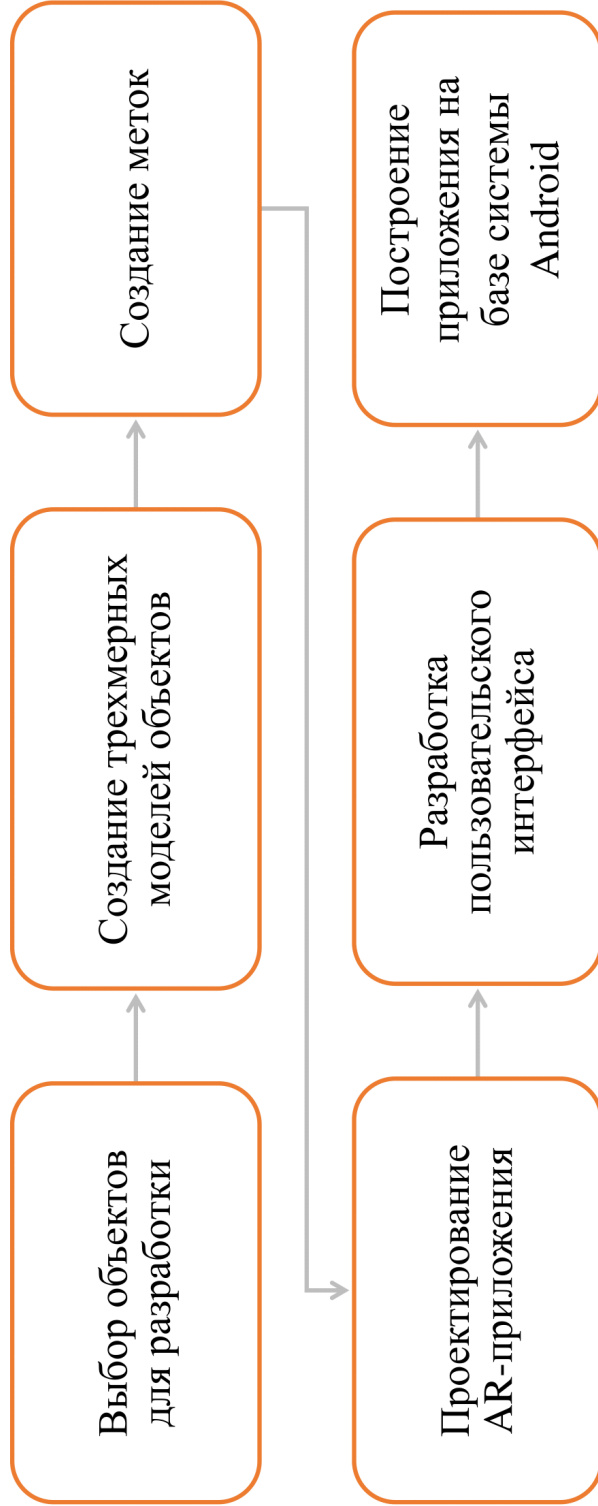
AR
ТЕХНОЛОГИЯ



C#

Выполнил:	Власова А. О.	Слайд №
Руководитель:	д.т.н., проф. Попов Д. И.	4

Этапы разработки приложения



Выполнил:

Власова А. О.

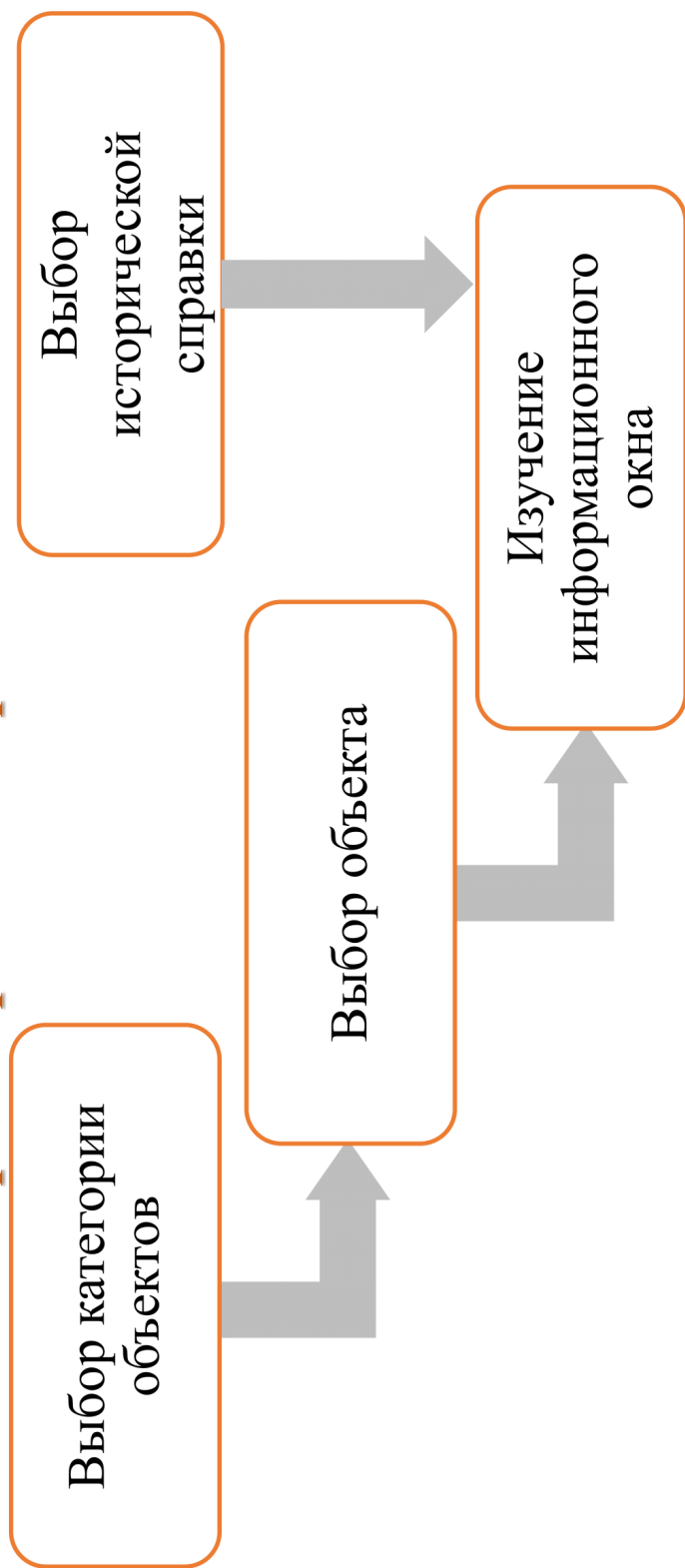
Слайд №

Руководитель:

д.т.н., проф. Попов Д. И.

5

Порядок работы с приложением



Выполнил:	Власова А. О.	Слайд №
Руководитель:	д.т.н., проф. Попов Д. И.	6

В процессе создания приложения были решены следующие задачи:

- Разработаны 3D модели выбранных объектов
- Спроектировано приложение с использованием технологий дополненной реальности
- Разработан удобный и функциональный интерфейс
- Реализовано приложение на базе системы Android
- Проанализированы пути дальнейшего развития приложения

Выполнил:	Власова А. О.	Слайд №
Руководитель:	д.т.н., проф. Попов Д. И.	7

ПРИЛОЖЕНИЕ Д
Антиплагиат


О документе

Оригинальность: 78.16%
 Зачислования: 21.84%
 Цитирование: 0%
 Дата: 09.06.2017
 Источников: 16

В кабинет
В кабинет Пояснительная записка.doc

История отчетов Выгрузить .arfx Выгрузить .pdf Краткая информация Версия для печати Руководство

№	%	Источник	Ссылка	Дата	Найдено в
[1]	4.41%	Vufolia: немного магии в нашей реальности / Хабрахабр	http://habrahabr.ru	28.10.2014	Модуль поиска Интернет
[2]	3.73%	И.В. Бойченко, А.В. Лежанкин. Дополненная реальность: состояние, проблемы и пути решения	http://masters.donntu.edu...	раньше 2011 года	Модуль поиска Интернет
[3]	3.55%	ТЕМАТИКИ НАЦИОНАЛЬНОГО ИССЛЕДОВАТЕЛЬСКОГО УНИВЕРСИТЕТА ВЫСШАЯ ШКОЛА ЭКОНОМИКИ . (2/2)	http://fullref.ru	15.05.2016	Модуль поиска Интернет

Еще найдено источников – 13, заимствования – 11.13%

Получить полный отчет