

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Создание AR-модели системы обучения основам ПДД».

Обучающийся

В.А. Соловьев

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд.пед.наук, доцент, О.М. Гущина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

Е.В. Косс

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2022

Аннотация

Тема выпускной квалификационной работы: Создание AR-модели системы обучения основам ПДД.

Работа была выполнена студентом Тольяттинского Государственного Университета, института математики, физики и информационных технологий, группы ПМИб-1802б, Соловьевым Владимиром Андреевичем.

Выпускная квалификационная работа посвящена разработке программного обеспечения (ПО) для обучения основам правил дорожного движения (ПДД) с использованием AR – модели.

Актуальность работы заключается в решении проблемы упрощения процесса обучения основам ПДД детей, а также увеличения мотивации и интереса к данному процессу.

Выпускная работа разделена на несколько логически связанных частей: анализ актуальности технологии дополненной реальности и постановка задачи, описание математической модели дополненной реальности и разработка программного модуля на основе описанной модели, тестирование и корректировка разработанного программного модуля.

Задачи работы:

- изучить методы реализации AR-моделей,
- изучить и сравнить готовые решения,
- создать AR-модель системы обучения основам ПДД,
- разработать программный модуль на основе описанной модели.

В заключении хотелось бы подчеркнуть, что данная работа актуальна для упрощения процесса обучения детей основам ПДД, а аналогичные технологические и конструктивные решения могут быть использованы при разработке программного обеспечения для изучения других объектов.

Abstract

The topic of the given graduation work is The creation of an AR-model of the training system in the basics of traffic regulations.

The work was performed by Vladimir Solovyov, a student of Togliatti State University, Institute of Mathematics, Physics and Information Technology, group PMIb-1802b.

The graduation work is devoted to the development of software for training in the basics of traffic regulations using the AR model.

The key issue of the thesis is the problem of simplifying the process of training in the basics of traffic regulations to children, as well as increasing motivation and interest in this process.

The graduation work is divided into several logically connected parts which are: the analysis of the augmented reality technology relevance and the problem statement, the description of an augmented reality mathematical model and development of a software module based on the described model, testing and adjustment of the developed software module.

The goals of the work are as follows:

- to study the methods of implementing AR models,
- to study and compare the ready-made solutions,
- to create an AR-model of the system for training in the basics of traffic regulations,
- to develop a software module based on the described model.

In conclusion we'd like to stress this work is relevant to simplifying the process of training in the basics of traffic regulations to children, and the similar technological and constructive solutions can be used for developing software for studying other objects.

Оглавление

| | |
|---|----|
| Введение..... | 5 |
| Глава 1 Анализ актуальности технологии дополненной реальности и постановка задачи | 7 |
| 1.1 Обзор истории развития графических интерфейсов и технологий дополненной и виртуальной реальности..... | 7 |
| 1.2 Анализ актуальности использования технологии дополненной реальности..... | 9 |
| 1.3 Выбор инструментария для реализации программного модуля.. | 12 |
| Глава 2 Описание математической модели дополненной реальности и разработка программного модуля | 16 |
| 2.1 Методика компьютерного зрения в дополненной реальности | 16 |
| 2.2 Описание математической модели дополненной реальности..... | 23 |
| 2.3 Реализация программного модуля | 28 |
| Глава 3 Тестирование и корректировка разработанного программного модуля | 43 |
| 3.1 Проведение вычислительных экспериментов..... | 43 |
| 3.2 Корректировка разработанной информационной технологии для обработки математических данных | 45 |
| Заключение | 50 |
| Список используемой литературы | 51 |

Введение

Дополненная реальность является одной из самых перспективных и быстроразвивающихся технологий в сфере информационных технологий. Область её применения весьма широка: начиная с игр и развлекательных приложений и заканчивая приложениями для подготовки и проведения хирургических операций. Главное достоинство приложения с технологией дополненной реальности - наглядность, которая способствует лучшему усвоению информации. Благодаря этому технология получила широкое распространение в сфере образования.

На протяжении последних двадцати лет школы и другие образовательные учреждения по всему миру спешно применяют информационные технологии для упрощения процесса обучения, а также увеличения степени вовлечённости учеников и студентов в данный процесс.

В данных условиях было решено поставить задачу на создание модели дополненной реальности, а также разработать на её основе мобильное приложение с использованием технологии дополненной реальности для обучения основам правил дорожного движения.

Актуальность AR-моделирования обусловлена тем, что оно открывает новые возможности для изучения теории и совершенствования практических навыков. Практическое объединение виртуального и реального опыта обогащает личностно-ориентированную деятельность обучающихся, позволяет визуализировать, просматривать и изучать совершенно любые учебные пособия, повышая тем самым уровень качества образования.

Цель работы – создание модели дополненной реальности и разработка на её основе программного модуля для обучения детей основам ПДД.

Объектом исследования является изучение и анализ основных алгоритмов и методов для разработки приложения дополненной реальности.

Для достижения поставленной цели в работе решаются следующие задачи:

- анализ существующих реализаций,
- описание модели дополненной реальности и реализация программного модуля,
- тестирование и корректировка программного модуля.

Данная работа включает в себя введение, три главы, заключение и список литературы.

В первой главе рассматривается история возникновения технологии дополненной реальности, актуальность её использования и постановка задачи на исследовательскую работу.

Во второй главе описываются метод компьютерного зрения, математическая модель и разработка программного модуля.

В третьей главе описан процесс тестирования программного модуля, а также корректировки внесённые в код программы.

Глава 1 Анализ актуальности технологии дополненной реальности и постановка задачи

1.1 Обзор истории развития графических интерфейсов и технологий дополненной и виртуальной реальности

В развитии пользовательского интерфейса компьютерных систем выделяют три парадигмы: пакетная обработка, интерфейсы командной строки и графический интерфейс пользователя. Пакетная обработка связана с вводом определенных данных в компьютер и ожиданием результата. При этом взаимодействие не является интерактивным. Так было с первыми электронно-вычислительными машинами, появившимися в середине XX века. Программы и данные набивались на перфокартах или перфолентах и передавались на обработку. Результат нужно было ждать до несколько часов или дней. После присоединения к электронно-вычислительным машинам телетайпов стал возможен интерфейс командной строки. Интерфейс командной строки – это парадигма интерфейса, при которой пользователь взаимодействует с компьютером текстовыми командами.

Начиная с выхода графических пользовательских интерфейсов на массовый рынок примерно 30 лет назад, взаимодействие с компьютером не претерпело существенных изменений. На этом фоне многообещающей выглядит концепция дополненной реальности. Текущие технологии пользовательских интерфейсов имеют фундаментальные ограничения и сфокусированы в основном на взаимодействии человека и компьютера. Технология дополненной реальности предлагает совершенствование интерфейса человека и реального окружающего мира. Она позволяет естественным образом внедрить компьютеры в деятельность человека, не отвлекая от сути решаемых задач.

Дженнифер Уайт и Драган Николич в своей книге «Virtual Reality and the Built Environment» упоминают описывают истоки технологии VR: «Первые

коммерческие пакеты виртуальной реальности стали доступны в конце 1980-х годов с основанием W Industries в Великобритании и VPL Research в США. Исполнительному директору VPL Research Джарону Ланье приписывают изобретение термина «виртуальная реальность» для описания DataGlove его компании, отслеживаемого пользователем дисплея на голове, который погружает пользователей в интерактивное» [26].

Виртуальная реальность — это термин, используемый для 3D-сред, созданных компьютером, которые позволяют пользователю входить в синтетические среды и взаимодействовать с ними. Пользователи могут в разной степени «погрузиться» в искусственный мир компьютеров, который может быть либо имитацией какой-либо формы реальности, либо имитацией сложного явления.

В телеприсутствии (Telepresence, набор технологий, позволяющий пользователю, например с помощью специальных устройств, получить впечатление того, что он находится и/или воздействует на место, отличное от его физического местоположения.), основная цель состоит в том, чтобы расширить сенсомоторные возможности оператора и способности решать проблемы в удаленной среде. В этом смысле телеприсутствие можно определить как систему человек/машина, в которой оператор-человек получает достаточную информацию о телеоператоре и среде выполнения задач, отображаемую достаточно естественным образом, чтобы оператор чувствовал физическое присутствие на удаленном объекте. Очень похоже на виртуальную реальность, в которой мы стремимся создать иллюзию присутствия в компьютерной симуляции, телеприсутствие направлено на создание иллюзии присутствия в удаленном месте.

«AR находится в более общем контексте, называемом смешанной реальностью (MR, Mixed reality), которая относится ко много осевому спектру областей, охватывающих виртуальную реальность (VR, Virtual reality), AR, телеприсутствие и другие связанные технологии. С помощью передовых технологий AR (к примеру, добавление компьютерного зрения и распознавание

объектов) информация об окружающем реальном мире пользователя становится интерактивной и цифровой манипулируемой. Информация об окружающей среде и ее объектах накладывается на реальный мир» [1]. «Одной из основных задач использования дополненной реальности является дополнение реального мира, расширяя воспринимаемую человеком действительность» [3].

Тарек Ахрам в своей книге «Advances in Usability and User Experience» утверждает, что «AR можно рассматривать как технологию между Виртуальной реальностью и телеприсутствием. В то время как в виртуальной реальности среда полностью синтетична, а в телеприсутствии она полностью реальна, в AR пользователь видит реальный мир, дополненный виртуальными объектами. AR обладает уникальным потенциалом воздействия на пользователей, что делает его связующим звеном между реальной деятельностью и цифровыми улучшениями» [27]. Хелен Папагианнис рассказывая о дополненной реальности в книге «Дополненная реальность. Все, что вы хотели узнать о технологии будущего» пишет: «На технологию VR есть свой специфический спрос, но AR позволяет нам глубже погружаться в реальный мир и быть с ним теснее связанными – мир, в котором мы проводим большую часть нашего времени» [15].

1.2 Анализ актуальности использования технологии дополненной реальности

Исследования по тематике дополненной реальности проводятся с 80-90-х гг., но, если в то время требовалось соответствующее оборудование (устройства, обладающие большой вычислительной мощностью для обработки поступающей на вход информации, расчёта трёхмерной графики с последующим её выводом на цветной дисплей), требовала большое количество специального оборудования (камеры, компьютер, дисплеи, элементы питания и др.) и в целом данная технология требовала громоздких решений, и не была

доступна массовому пользователю, то в современном мире ситуация весьма изменилась. Кинг Бретт в своей книге Эпоха дополненной реальности утверждает: «Чем больше высокотехнологичных новинок, тем выше наш изобретательский потенциал – и тем скорее появляются еще более продвинутое технологии. Изобретение печатного станка привело к резкому росту числа образованных людей и к небывалому распространению знаний» [9].

В настоящее время благодаря технологическому прогрессу и развитию потребительской техники использование технологии дополненной реальности стало доступно большинству людей. Сейчас практически каждый человек обладает возможностью работы с дополненной реальностью при помощи своих смартфонов и персональных компьютеров. Современные смартфоны оснащены камерой, мощными, по сравнению с ЭВМ прошлых лет, процессорами, а также цветными дисплеями с большим расширением.

Индустрия виртуальной и дополненной реальности предлагает инновационные технологии, которые могут быть использованы в сфере развлечений, образования, профессиональной подготовки, медицинских и промышленных инноваций [30]. В образовании дополненная реальность пока применяется довольно редко, но всё больше учителей, исследователей и разработчиков начинают двигаться в сторону более интерактивных обучающих методик [18].

«Интерактивность – ключевая причина того, почему VR и AR так привлекательны для образовательных целей» [2]. В основе компьютерного моделирования может находиться математическая модель, лабораторный эксперимент или анимация, которая представляет работу некоторого механизма, протекание процесса, строение сложного предмета и т.д. В моделирующих программах возможно широкое использование интерактивной графики, которое позволяет обучаемому не только наблюдать за происходящим, но и воздействовать на параметры объектов, исследуя при этом эффекты, которые влияют на получаемый результат. Зачастую ученики предпочитают графику и мультимедиа, а не текст и бумажные носители

информации. Вместо чтения главы из учебника они предпочтут посмотреть видеоурок [21]. «Интерактивные обучающие методики позволяют визуализировать смысловую составляющую образовательного процесса, направленную на мотивационную составляющую учащихся. Данное обстоятельство особенно это важно для детей младшего школьного возраста, поскольку дети, которые подходят к обучению осознанно, имеют повышенную мотивацию и стремление к обучению в дальнейшем» [20]. «Внедрение технологии дополненной реальности позволит повысить качество обучения за счет мотивации обучающихся к самообучению, повышения интереса аудитории к излагаемому и изучаемому материалу» [22].

Именно приложения дополненной реальности дают возможность визуализации процессов и предметов, которые необходимо, но невозможно продемонстрировать в учебной аудитории. Поэтому вопрос об использовании технологии дополненной реальности для поддержки образовательного процесса является одним из самых актуальных в сфере использования новых информационных технологий в образовательном процессе.

«Помимо преимуществ использования технологий дополненной реальности в разных социально-экономических сферах существуют и материальные преимущества. Например, не нужно будет использовать любые виды печатной, рекламной продукции. Дополненная реальность экономит средства и ресурсы, работая с виртуальными прототипами» [11].

После анализа проблемы был сделан вывод о том, что применение данной технологии в области обучения актуально. Однако в таких учебных заведениях, как школы, информатизация образования идет весьма медленным путем, а данная технология представляется сотрудникам и учащимся на уровне фантастики, поэтому создаваемое приложение должно быть просто и понятно в использовании.

1.3 Выбор инструментария для реализации программного модуля

Было принято решение создать мобильное приложение, которое можно будет использовать не только в домашних условиях, но и в рамках занятий в учебном заведении.

Таким образом, создаваемое приложение:

- будет использовать технологию дополненной реальности;
- в качестве устройства отображения использует мобильное устройство (смартфон или планшет);
- должно быть использовано как вспомогательный инструмент при решении стереометрических задач из школьного курса геометрии.

У пользователя имеется распечатанный набор задач с маркерами (метками дополненной реальности). В нашем случае маркеры будут размещены на страницах книги. При запуске приложения на мобильном устройстве включается захват видеокамеры. Пользователь наводит камеру так, чтобы маркер был в зоне видимости веб-камеры, на экране отображается привязанный к метке трехмерный объект, представляющий собой визуализацию задачи, например изображает какой-либо объект.

Мобильное приложение, предоставляющее дополненную реальность, может принимать различные формы, которые ограничивают пользователей в использовании его только в определенных операционных системах (так называемые собственные приложения) или, наоборот, позволяют свободно использовать приложение в любой операционной системе (так называемые мультиплатформенные приложения). Сокращение рисков, снижение затрат, усиление терапевтического эффекта, сохранение расходных материалов являются потенциальными результатами использования технологии AR в обучении [8]. «Мобильная дополненная реальность особенно хорошо подходит для таких идей, как “повсеместное обучение”, в рамках которого предполагается, что каждый человек учится постоянно, где бы он ни находился,

когда ему это необходимо. Разработка мобильного AR-проекта основана на принципах моделирования» [14].

Об эффективности применения технологий дополненной реальности в образовании говорится во многих источниках. Приведем несколько фрагментов из статей: «Технология дополненной реальности дает возможность визуализировать объекты, которые трудно себе представить, превращает их в 3D-модели, тем самым облегчая понимание абстрактного и сложного контента. Добавление таких данных как, например, краткая биография человека, интересные факты, исторические данные о событиях, визуальные 3D-модели способствуют улучшению восприятия и облегчают понимание учебного материала. Во многих случаях теоретических знаний недостаточно для получения соответствующих навыков и компетенций в профессиональных областях, поэтому дополненная реальность может служить цели преодоления разрыва между теоретическими знаниями и практикой.» [23]. «До сих пор ситуация описывалась с помощью текста или, в исключительных случаях, аудио с видео или без него. С помощью технологии виртуальной реальности конкретная ситуация может быть запрограммирована несколькими переменными и средами, в которых студент может действовать» [13].

М. Клаудиа том Дик и Тимоти Х. Юнг в своей работе «Augmented Reality and Virtual Reality: New Trends in Immersive Technology.» Сравнивают опыт использования приложения с технологией дополненной реальности и без него, и приходят к следующему выводу: «Существует значительная разница в удовольствии и намерениях между AR ($M = 6,45$) и приложением без AR ($M = 5,1$)» [28]. Несмотря на это у AR приложения так же есть свои недостатки, так например Дюличева Ю.Ю. в статье «О применении технологии дополненной реальности в процессе обучения математике и физике» упоминает: «К недостаткам приложений дополненной реальности можно отнести их узкую направленность на изучение определенного процесса, явления или понятия» [7].

Создание приложения дополненной реальности означает выбор платформы дополненной реальности. Эти платформы являются набором API и инструментов, позволяющих разработчику создавать контент, взаимодействующий с реальным миром [29]. Для реализации технологии дополненной реальности необходимы два основных программных компонента: трекинг и визуализация.

Чтобы определиться с выбором средства разработки дополненной реальности выберем наиболее популярные SDK, и рассмотрим их достоинства и недостатки (таблица 1).

Таблица 1 – Сравнение популярных средств разработки AR

| ПО | Описание | Достоинства | Недостатки |
|---------------------------------|---|--|--|
| DART | Набор инструментов для прототипирования и создания приложений с дополненной реальностью. Выполнен как дополнение к Macromedia Director | Поддержка различных сенсоров для трекинга. | Поддерживает только ARToolkit и ARTag в качестве движков для трекинга. |
| D'Fusion | Коммерческий пакет программ для создания приложений с дополненной реальностью. | Хороший трекинг ключевых точек. | Плохая документация, отсутствие сообщества разработчиков, цена |
| Metaio SDK | Коммерческий пакет программ для разработки дополненной реальности. | Разнообразие функций. | Высокая цена |
| flare | Набор библиотек компьютерного зрения для Adobe Flash/Flex для обнаружения, отслеживания и расчёта позиции двумерных штрих-кодов или изображений в видеопотоке веб-камеры. | Отличный трекинг: устойчивость к искажению маркера и бликам. | Высокая цена. |
| Qualcomm Vuforia (до 2011 QCAR) | Инструментарий для разработки дополненной реальности в виде расширения к Unity | Хороший трекинг. Бесплатен | Закрытый исходный код. |

На настоящий момент исследователями и специалистами была наработана обширная теоретическая и алгоритмическая база для их реализации как в виде различных отдельных компонентов, так и в виде интегрированных программ и наборов средств разработки дополненной реальности. К отдельным компонентам относятся библиотеки и фреймворки компьютерного зрения, движки трёхмерной графики, другие решения.

«С помощью Vuforia можно писать код дополненной реальности на разных языках программирования, ее поддерживает также и Unity, что позволяет создавать мультиплатформенные решения. Vuforia хорошо и стабильно работает на большом количестве устройств» [4]. Исходя из перечисленных преимуществ и недостатков мною был выбран инструментарий Qualcomm Vuforia.

Вывод по главе

В данной главе была рассмотрена история возникновения технологий дополненной и виртуальной реальности, описан принцип работы технологии дополненной реальности, проанализирована актуальность использования технологии в области обучения и поставлена задача на исследовательскую работу.

В главе описано сравнение наиболее популярных инструментариев разработки приложений с использованием технологии дополненной реальности, на основе данного описания был выбран инструментарий Qualcomm Vuforia, при помощи которого в дальнейшем будет вестись разработка программного модуля.

Глава 2 Описание математической модели дополненной реальности и разработка программного модуля

2.1 Методика компьютерного зрения в дополненной реальности

Для внедрения технологии дополненной реальности требуется корректная работа трекинга на распространённых потребительских устройствах. На настоящий момент для таких устройств характерен ограниченный вычислительный ресурс, экономия энергии и наличие чаще всего только одной RGB-камеры. Это ограничивает круг возможных подходов, применимых для реализации трекинга. В этой связи стоит выделить оптический вид трекинга. Он получил большое развитие в исследованиях. Оптический трекинг обладает потенциалом широкого применения на потребительском рынке, так как для его работы достаточно одной камеры без каких-либо специальных дополнительных внешних устройств. Обычно, когда говорится об оптическом трекинге, речь идет о системе дополненной реальности, в которой используется камера, и изображение, поступающее с камеры, используется как в качестве изображения объективной реальности, так и для обработки в целях трекинга. И в данном случае осуществляется отслеживание положения непосредственно камеры как наблюдателя.

«Поскольку основной задачей дополненной реальности является синтез реальных и виртуальных объектов в пространстве, то возникает необходимость в предварительной обработке данных об окружающем пространстве. Чаще всего распознавание маркера реализовывалось через простой алгоритм. Он состоит из шести этапов: перевод цветного изображения в градации серого, бинаризация изображения, определение замкнутых областей, выделение контуров, выделение углов маркера и преобразование координат» [12].

Схема маркеров значительно снижает сложность, с которой сталкиваются традиционные алгоритмы компьютерного зрения. Использование общих алгоритмов компьютерного зрения ограничено контролируемыми средами, и

условиями, в виде освещения или формы объекта. Чтобы смягчить некоторые из этих ограничений и уменьшить сложность алгоритмов обработки изображений, каждый отслеживаемый объект может быть помечен определенным кодом, который его идентифицирует. Таким образом, цель схемы кодирования состоит в том, чтобы однозначно идентифицировать отслеживаемые объекты и оценить их положение без двусмысленности. Коды, которые мы выбрали, представляют собой наборы точек, расположенных по шаблону с равным расстоянием между ними. Необходимым условием является то, что узоры имеют различимые большие и малые оси и что они несимметричны. Мы также предполагаем, что каждая точка в опорном маркере имеет один и тот же цвет, а фон однороден.

В качестве примера мы покажем схему кодирования, основанную на двух осях, как показано на рисунке 1. Количество доступных комбинаций зависит от длины осей. На рисунке 1 мы видим, что с двумя точками на малой оси и тремя точками на большой оси мы можем иметь две не избыточные и несимметричные комбинации (учитывая только повороты вокруг вектора, перпендикулярного странице).

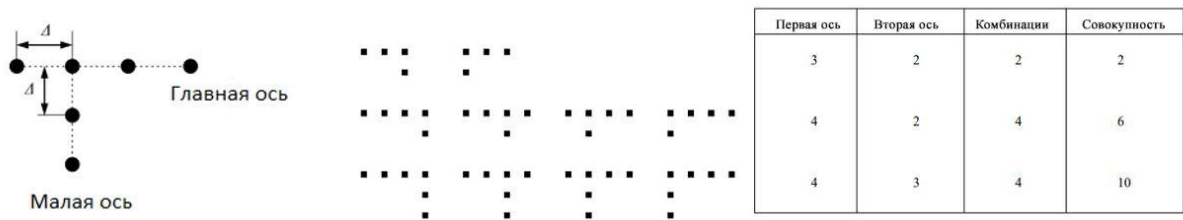


Рисунок 1 - Пример схемы кодирования

С четырьмя точками на главной оси и тремя точками на малой оси комбинации равны четырем, а сумма всех возможных комбинаций равна шести. При большой оси длины равной 10 и малой оси с длиной равной 9 общее совокупное количество объектов, которые могут быть идентифицированы,

составляет 530, что является достаточно большим числом для большинства приложений, требующих отслеживания нескольких объектов. Если используется несколько цветов, узоры могут повторяться.

Изображение сначала свернуто с помощью гауссова ядра, чтобы исключить высокочастотные детали. В качестве следующего шага все области, похожие по интенсивности на опорные точки, будут разделены. Мы используем один из двух методов (в зависимости от того, доступно ли цветное или полутоновое видео): пороговое значение или простая цветовая сегментация. В первом методе мы используем яркие точки на темном фоне (или наоборот) и пороговое значение изображения, чтобы отделить яркие (темные) области от фона. Преимущество простого порогового значения с белыми точками на черном фоне или наоборот заключается в меньшей зависимости от условий освещения. Основным недостатком является то, что больше шума возникает из-за ярких (темных) пятен по всему изображению. Второй метод позволяет использовать несколько цветов точек, а также фон маркеров. Это имеет то преимущество, что снижает уровень шума. Однако цвета точек могут существенно меняться в зависимости от условий освещения, ориентации и т.д., что затрудняет идентификацию действительных точек при различных внешних условиях. Конечным результатом является черно-белое изображение с белыми областями, где обнаруживается точечный цвет.

На следующем этапе уменьшенная версия изображения с пороговым значением (с интервалом выборки 8, 4 или 2 пикселя) сканируется на наличие белых областей. Как только белая область найдена, алгоритм увеличения области определяет и сохраняет центростид каждого отдельного яркого пятна, посещая и отмечая каждый соответствующий пиксель в полном изображении с пороговым значением. Яркие пятна могут быть вызваны отражениями, цветами, нанесенными на объекты, текстурой или частями фона. Чтобы уменьшить шум, а также повысить производительность, участки, которые в процессе увеличения области считаются слишком большими, отбрасываются. После процесса выращивания области участки оцениваются в соответствии с формой.

Вычисляются их соотношение сторон, евклидово расстояние между его центром масс и центром его ограждающей коробки, а также другие соотношения. Накладывая крест, расположенный над центром тяжести пятна, мы считаем пятно действительным, если имеется полное перекрытие во всех четырех сторонах этого креста. Также рассчитываются другие соотношения, связанные с формой, такие как горизонтальные и вертикальные размеры. Исправления, которые выходят за пределы заданных диапазонов для этих коэффициентов, удаляются.

Следующая цель - сгруппировать действительные точки. Размер кластеров должен находиться в пределах количества точек, найденных в наборе маркеров. Алгоритм соединяет центроиды всех выбранных участков, рассматривая их как вершины в графе и генерируя минимальное связующее дерево (МСД) между ними. МСД группирует участки в кластеры в соответствии с локальной близостью. Вычисляются статистические данные, такие как средняя длина и стандартное отклонение всех ветвей МСД, а длина ребер сохраняется в списке смежности или представлении матрицы смежности.

Вторым этапом проблемы зондирования в области сборки является трекинг. Трекинг - это динамическое определение/вычисление пространственных свойств во время работы программы. Чтобы отображать виртуальные объекты в трехмерном пространстве мы должны знать их относительное положение по сравнению с относительным положением местоположений и объектов реального мира. Поскольку AR часто представляет собой процесс в реальном времени, эти измерения должны постоянно обновляться, чтобы они оставались точными. Алгоритм отслеживания состоит из нескольких модулей. К ним относятся: предварительная обработка изображения, обнаружение исправлений, генерация минимального связующего дерева, устранение ветвей выбросов, извлечение осей, идентификация деталей и оценка положения в 3D.

Алгоритм определяет вероятные главные и второстепенные оси в каждом кластере путем измерения ориентации каждой ветви и нахождения двух

вероятных ориентаций осей. Вычисляется длина каждой оси, определяемая как количество точек, и определяется ограничивающая рамка вокруг маркера. Подпрограмма определения оси выводит три неколлинеарные точки, элемент периметра прямоугольника. Правила, используемые для идентификации объектов, могут основываться на:

- общее количество точек,
- длина каждой оси,
- относительное положение осей относительно друг друга.

Эти правила еще больше сократят количество допустимых геометрических кластеров, создаваемых шумом. На рисунке 2 показаны все обнаруженные точки, а также МСД, кластеризация точек и распознавание маркеров.



Рисунок 2 – Процесс обнаружения точек

Точки, соответствующие маркерам, были сгруппированы в подмножества, и единственные кластеры, распознанные как маркеры, получают заключительную рамку. Впоследствии оценивается трехмерное положение маркера. Три неколлинеарные точки, вершины ограничивающей рамки, передаются в процедуру оценки трехмерной позы. Относительное положение

этих точек известно, и оно может быть сохранено в массиве с индексом для каждого маркера.

Правила для идентификации объектов основаны на особенностях кода и могут быть организованы иерархическим образом. Пример правила: если код содержит n точек, то деталь принадлежит классу объектов А. Если на нем $n + 1$ точек, то деталь относится к классу объектов В. Если главная ось имеет m точек, а деталь является элементом множества А, то деталь принадлежит подклассу А1, в противном случае она принадлежит подклассу А2 и т.д. Применение иерархических правил обеспечивает быстрое распознавание и естественную организацию типов маркеров. Правила для распознавания маркеров, которые используются в настоящее время, связаны с размером осей и их относительным положением относительно друг друга.

Приблизительная трехмерная оценка позы с использованием ортоперспективных проекций выполняется быстро. Скорость оценки позы становится проблемой только при очень большом количестве маркеров. Наибольшие угловые погрешности при использовании ортоперспективных методов могут достигать почти 10 градусов. Наибольшие ошибки дальности возникают, когда сторона m_1m_2 очень короткая по отношению к m_1m_2 (рисунок 3). Обычно эти угловые погрешности намного ниже (около 2-3 градусов). Для некоторых применений, в зависимости от требуемой точности, может быть желательно использовать более точные методы.

Отслеживание меток позволяет обойти сложность внешнего вида и видения на основе модели, что позволяет добиться скорости отслеживания в реальном времени. Однако информация об объекте, полученная из маркеров, также гораздо более ограничена. Знание точного положения и ориентации маркера в координатах объекта должно быть сохранено заранее. Кроме того, поскольку маркеры видны только с ограниченного набора точек обзора, возможно, потребуется нанести несколько разных маркеров на разные грани объекта. Предметы неправильной формы или гибкие предметы будут создавать

трудности (например, шланг или пружина). Метки также подходят только для объектов, которые достаточно велики, чтобы их можно было пометить.

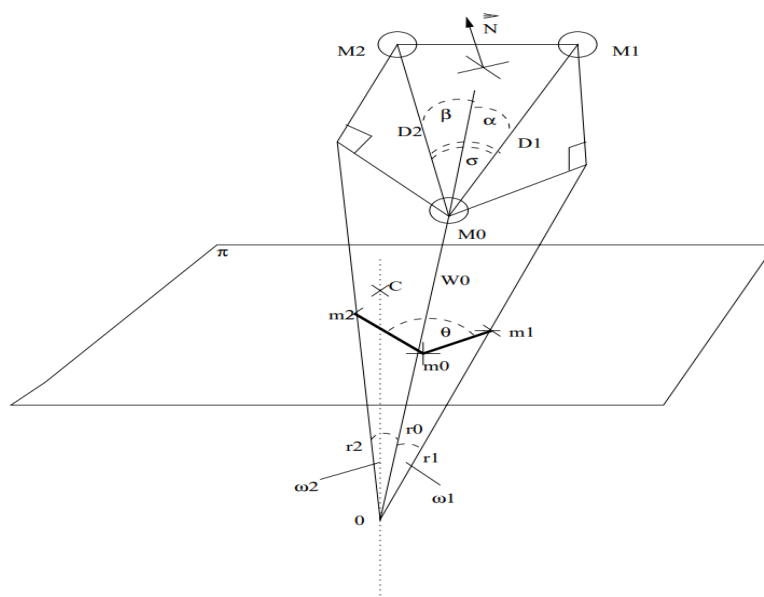


Рисунок 3 – Пример ортогональной проекции

Частичное перекрытие объекта не является проблемой, если маркер виден камере. Оклюзия маркера может быть обработана только частично, не прибегая к использованию функций самого объекта. Отслеживание движения с помощью фильтра Калмана можно использовать для кратковременных периодов окклюзии. Пока часть маркера видна, мы все еще можем отслеживать его в 2-D времени.

Диапазон отслеживания будет зависеть от размера маркера и разрешения видео. Чтобы расширить диапазон, можно использовать маркеры разных размеров. Например, некоторые маркеры могут иметь расстояние между точками d , а другие маркеры могут иметь одно из $n \cdot d$. Стена вдали может быть отмечена большим маркером, а объект поблизости - маленьким маркером. Чтобы использовать одни и те же шаблоны, маркеры разного размера могут иметь разные цвета точек.

2.2 Описание математической модели дополненной реальности

Методы, основанные на моделях, используют модели характеристик отслеживаемых объектов, такие как модели САПР или 2D-шаблоны элемента, основанные на различных признаках. Как только установлено соединение между 2D-изображением и 3D-рамкой мира, можно определить положение камеры, проецируя 3D-координаты объекта в наблюдаемые координаты 2D-изображения и минимизируя расстояние до соответствующих 2D-объектов. Ограничения для оценки положения камеры чаще всего определяются с использованием точечных объектов. Этап восстановления/распознавания использует данные, полученные на первом этапе, для восстановления системы координат реального мира. Предполагая наличие откалиброванной камеры и модели перспективной проекции, если точка имеет координаты $(x, y, z)^T$ в системе координат камеры ее проекция на плоскость изображения равна $(x / z, y / z, 1)^T$.

В точечных ограничениях у нас есть две основные системы координат, как показано на рисунке 4, мировая система координат W и система координат 2D-изображения.

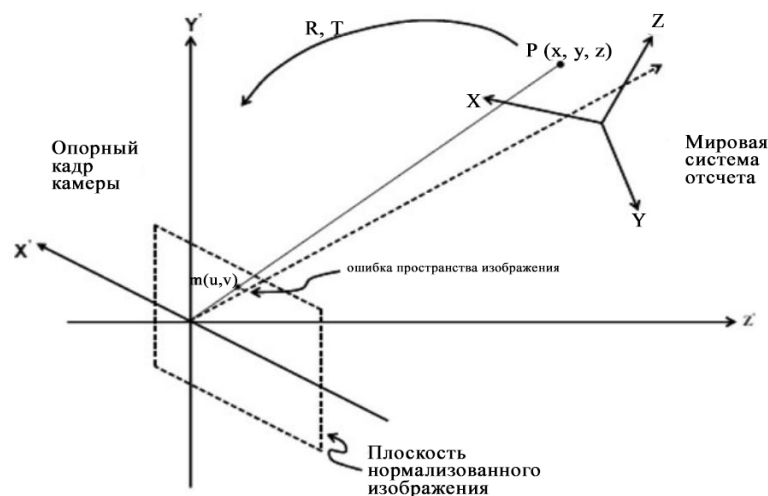


Рисунок 4 - Мировая система координат и система координат изображения

Дан набор 3D неколлинеарных опорных точек в мировой системе координат $P_i(x_i, y_i, z_i)^T$, где $i = 1; \dots; n$, при $n \geq 3$, и $q_i(x'_i, y'_i, z'_i)^T$ - соответствующие координаты камеры в пространстве, p_i и q_i связаны следующим преобразованием (1):

$$q_i = Rp_i + T, \quad (1)$$

где

$$R = \begin{pmatrix} r_1^T \\ r_2^T \\ r_3^T \end{pmatrix} \text{ и } T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \text{ являются матрицей вращения и вектором}$$

перемещения соответственно.

Пусть точка изображения $h_i(u_i, v_i, 1)^T$ является проекцией p_i на нормализованную плоскость изображения. Уравнение коллинеарности, устанавливающее взаимосвязь между h_i и P_i с использованием точечного отверстия камеры, задается следующим образом (2):

$$h_i = \frac{1}{r_3^T p_i + t_z} (Rp_i + T) \quad (2)$$

Ошибка пространства изображения определяет взаимосвязь между 3D-опорными точками, соответствующими точками извлеченного 2D-изображения и параметрами положения камеры и соответствует ограничениям точки. Ошибка пространства изображения задается следующим образом:

$$E_i^p = \sqrt{\left(\hat{u}_i - \frac{r_1^T p_i + t_x}{r_3^T p_i + t_z}\right)^2 + \left(\hat{v}_i - \frac{r_2^T p_i + t_y}{r_3^T p_i + t_z}\right)^2} \quad (3)$$

где $\hat{m}_i(\hat{u}_i, \hat{v}_i, 1)^T$ - наблюдаемые точки изображения.

Некоторые методы предполагают наличие опорных маркеров в среде или объекте с известной 3D-геометрией и используют эти данные. Если вся сцена заранее неизвестна, для отображения относительных положений опорных маркеров или 3D-моделей используется метод одновременной локализации и отображения (SLAM). В случае, когда нельзя сделать никаких предположений о 3D-геометрии сцены, используется метод структуры из движения (SfM). Метод SfM можно разделить на две части: отслеживание характерных точек и оценка параметров камеры. Методы отслеживания в AR в основном зависят от типа среды, в которую будет введено AR-устройство, а также от типа AR-системы. Окружающая среда может быть внутренней, наружной или сочетать то и другое. Точно так же система может быть мобильной или статичной (иметь фиксированное положение). Например, если устройство AR представляет собой устройство с фиксированным положением для реальной внешней среды разработчики могут использовать механическое отслеживание, поскольку все отслеживаемые движения будут механическими, поскольку положение устройства известно. Этот тип среды и системы облегчает отслеживание окружающей среды для улучшения окружающей среды. С другой стороны, если устройство AR является мобильным и предназначено для использования на открытом воздухе, отслеживание становится намного сложнее, а различные методы имеют некоторые преимущества и недостатки.

Виртуальное пространство моделируется с использованием аффинных преобразований, когда вершина для отображения ее на экране преобразуется из трехмерных координат в двухмерные. Входными данными являются координаты, а выходными — изображение. При построении дополненной реальности — наложении виртуальных объектов на изображение, фиксируемое веб-камерой, — необходимо решить обратную задачу, где входными данными будет изображение, а выходными — координаты. Подобные преобразования используются в фотограмметрии для определения размеров объектов. Основная проблема при построении дополненной реальности заключается в определении относительных систем координат сцены (изображения) и видеокамеры, т.е. в

локализации веб-камеры. Эта задача состоит из нескольких подзадач: сопоставить неподвижную систему координат со сценой; выбрать эквивалент виртуальной камеры и установить ее положение, при этом положение камеры определяется ее вращением и смещением относительно центра координат, так чтобы проекции изображений, фиксируемых веб- и виртуальной камерами, совпадали. Как известно, виртуальную камеру можно задать двумя 4×4 -матрицами (4),(5):

$$A = \begin{bmatrix} a & 0 & f & 0 \\ 0 & b & g & 0 \\ 0 & 0 & c & d \\ 0 & 0 & e & 0 \end{bmatrix} \quad (4)$$

$$B = \begin{bmatrix} RY_x & RY_y & RY_z & T_x \\ RP_x & RP_y & RP_z & T_y \\ RR_x & RR_y & RR_z & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Коэффициенты a, b, c, d, e, f, g матрицы A (4) образуют фрустум — пирамидальную видимую область, ограниченную шестью плоскостями, которые строятся исходя из значений угла зрения камеры, отношения высоты виртуального окна к его ширине и значения глубины Z -буфера.

Матрица B (5) состоит из 3×3 -матрицы R вращения вокруг трех осей (RY — yaw, ось направлена вверх; RP — pitch, ось направлена вправо; RR — roll, ось направления) и вектора смещения T вдоль осей x, y, z соответственно. Проекция вершины на ближнюю плоскость экрана строится в соответствии с формулой (6),

$$\begin{bmatrix} S_y \\ S_x \end{bmatrix}^T = \begin{bmatrix} a & 0 & f & 0 \\ 0 & b & g & 0 \\ 0 & 0 & c & d \\ 0 & 0 & e & 0 \end{bmatrix} \times \begin{bmatrix} RY_x & RY_y & RY_z & T_x \\ RP_x & RP_y & RP_z & T_y \\ RR_x & RR_y & RR_z & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} V_x \\ V_y \\ V_z \\ 1 \end{bmatrix} \quad (6)$$

которая преобразует трехмерный вектор $[V_x, V_y, V_z, 1]^T$ виртуального пространства в двухмерный вектор $[S_x, S_y]^T$ экранного пространства.

Для реализации модели дополненной реальности разработаем программный модуль, упрощенная структурная схема которого представлена на рисунке 5.

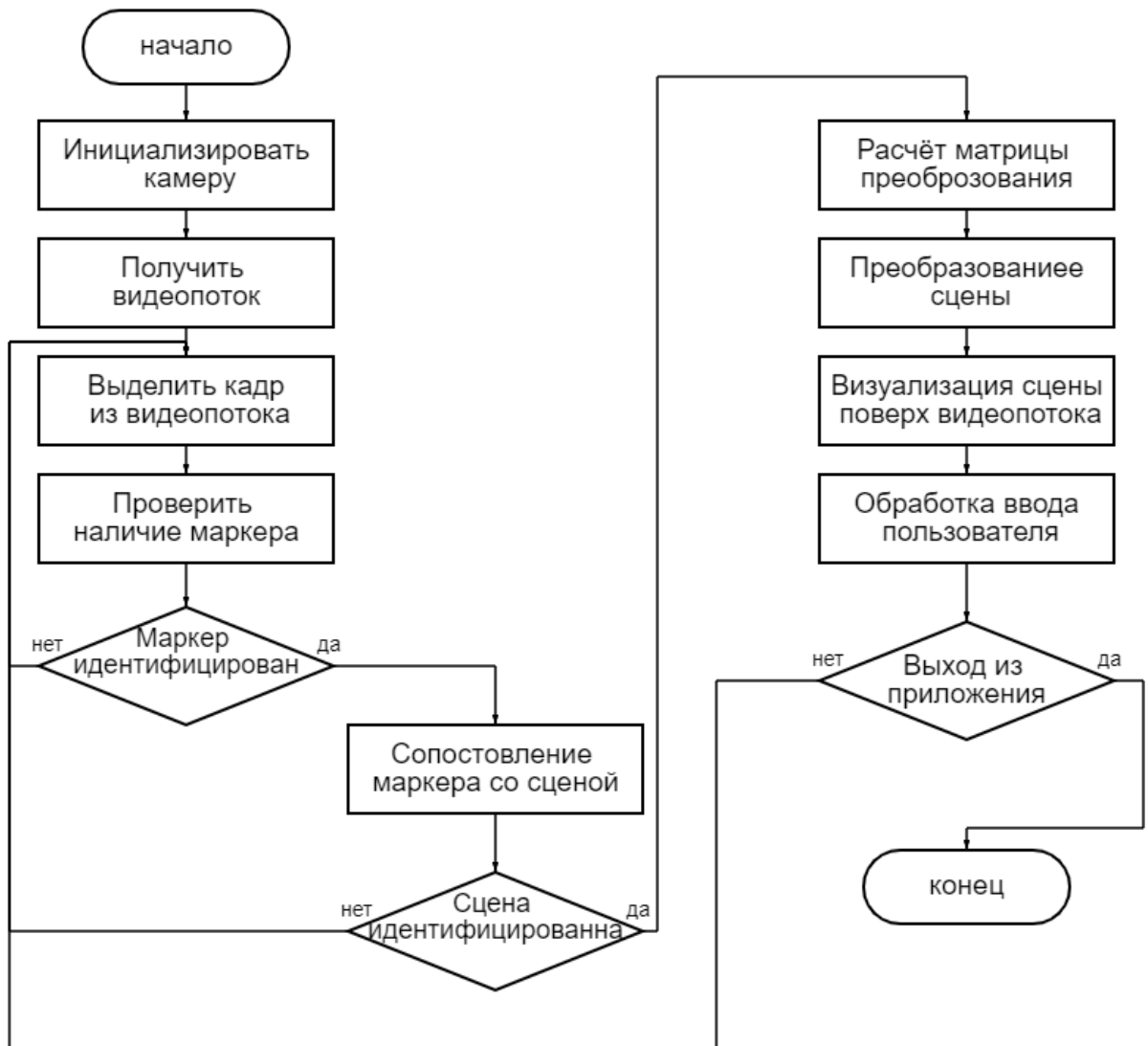


Рисунок 5 - Блок схема программного модуля

При решении задачи локализации камеры 3D- и 2D-векторы в формуле (6) известны, а матрицы A и B, состоящие из внешних и внутренних параметров

веб-камеры, неизвестны, эти параметры и необходимо найти. Матрица В содержит внешние параметры (Extrinsic Parameters) камеры: угол ее поворота и смещение относительно центра сцены. Задача нахождения матрицы В сводится к сопоставлению проекций, фиксируемых реальной и виртуальной камерами. Как только матрица В становится известной, задачу локализации камеры можно считать решенной.

Программа представляет собой виртуальную сцену со свободной камерой; в системе координат сцены, извлеченной из видеоряда с изображением маркера, отображаются главные оси и положение камеры относительно них.

2.3 Реализация программного модуля

Для разработки приложения в среде Unity для начала необходимо определиться с платформой. Так как мы разрабатываем мобильное приложение наилучшим вариантом будет Android. Заходим в настройки проекта (рисунок б), и выставляем желаемую платформу [16].

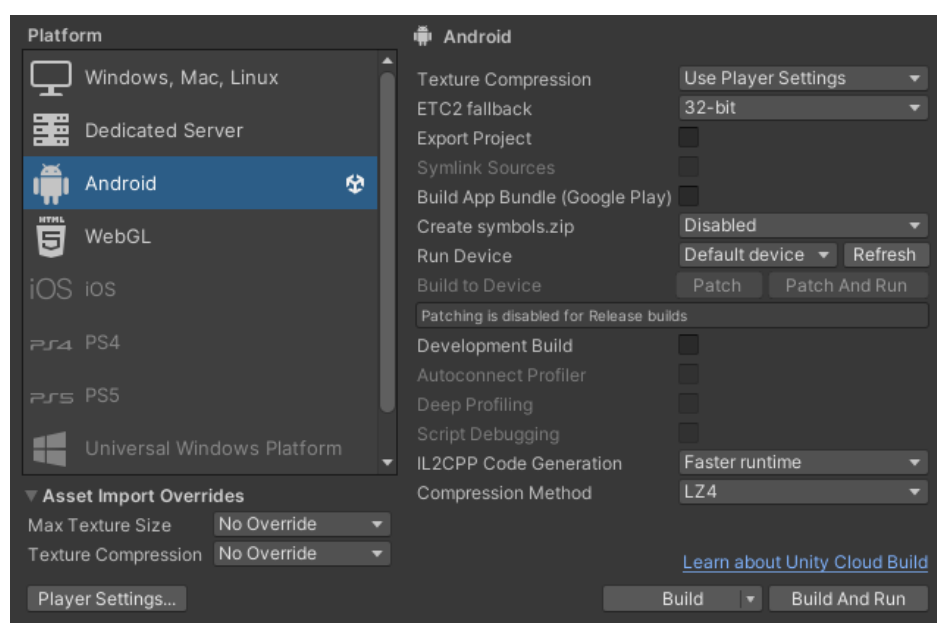


Рисунок б – Выбор платформы

Следующий шаг – создание сцены и добавление в сцену AR-камеры. В последствии для корректной работы приложения и отображения моделей все объекты (маркеры, модели и т.д.) необходимо создавать как дочерние объекты AR-камеры [19]. Теперь нам необходимо создать набор маркеров и импортировать его в наш проект. Было решено создать 5 сцен, соответственно всего будет 5 маркеров (5 страниц в книге, на которых будет происходить “зарисовка” описываемой ситуации). Из заранее подготовленной книги мною были выбраны следующие ситуации:

- ребёнок переходит дорогу на пешеходном переходе,
- ребёнок прогуливается вблизи проезжей части,
- ребёнок ожидает автобус на остановке,
- ребёнок находится внутри транспортного средства,
- ребёнок передвигается на велосипеде.

На рисунке 7 предоставлен листинг иерархии объектов.

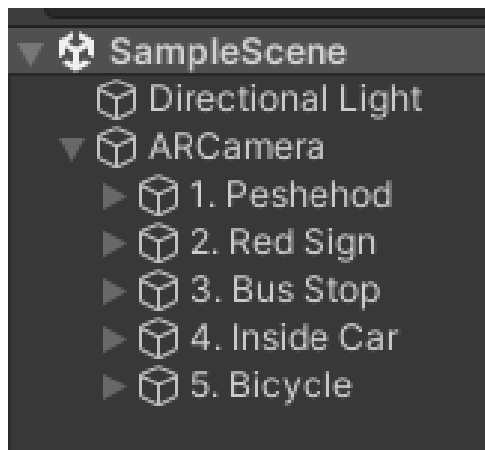


Рисунок 7 – Иерархия объектов

Чтобы начать добавлять в приложение модели и маркеры необходимо создать лицензию и получить лицензионный ключ на сайте Vuforia [17]. Данный ключ (рисунок 8) необходим для дальнейшей разработки в Unity.

For TSU Diploma

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

License Key

Usage

Please copy the license key below into your app

```
AXhFoJ3/////AAABmT/IRF/1iEj0jLVpu6Qv5xkx/F8+K26jie/1dn0UrDFp1E1VAeBKwoYifALwwhQ7kYIq1+X01PtsCaNKj968Bhr  
WVeSwfbDXctXB4Z2Wgp1/6dH9chxJYWAzyYnICYLf/vzTPmkFwizGnmgiE2DoK2rDxYYYi7aD5xgXZdc/phJI73t6cuAIYbw0K02E5t  
SqnLKYaata07MSraECFq1KbjuhVEecVfD9dB4iC1Q1r+p+2kjttqC6bfedETC2oOh7x56Y9LN8cWUVjFAzWfn07XNdBZGKf4H1yU2+T  
RR5z7D1hJ1cW7ke/xG02RwoeGoffPeQUq2MoF4mKa0ZL8nwXGkOoV8tatXurDVCF1Jkxv9Z
```

Plan Type: Basic**Status:** Active**Created:** Apr 19, 2022 20:24**License UUID:** 1db33712607547ff95a70dc41879b6aa**Permissions:**

- Advanced Camera
- External Camera
- Model & Area Targets
- Watermark

Рисунок 8 – Ключ лицензии

После создания лицензии и лицензионного ключа необходимо создать маркер, на котором будет проходить отображение модели. Возьмём данные правила за основу: одинаковое расстояние между точками, максимальная длина осей и максимальное покрытие изображения. Чтобы положение маркера определялось максимально точно так же добавим точки в углы изображения. На рисунке 9 продемонстрирован процесс создания первого маркера.

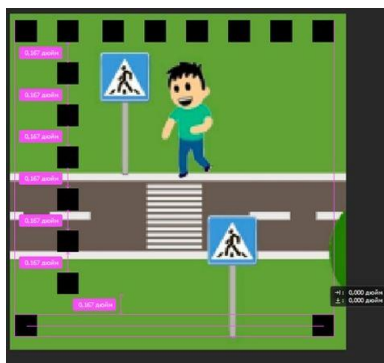
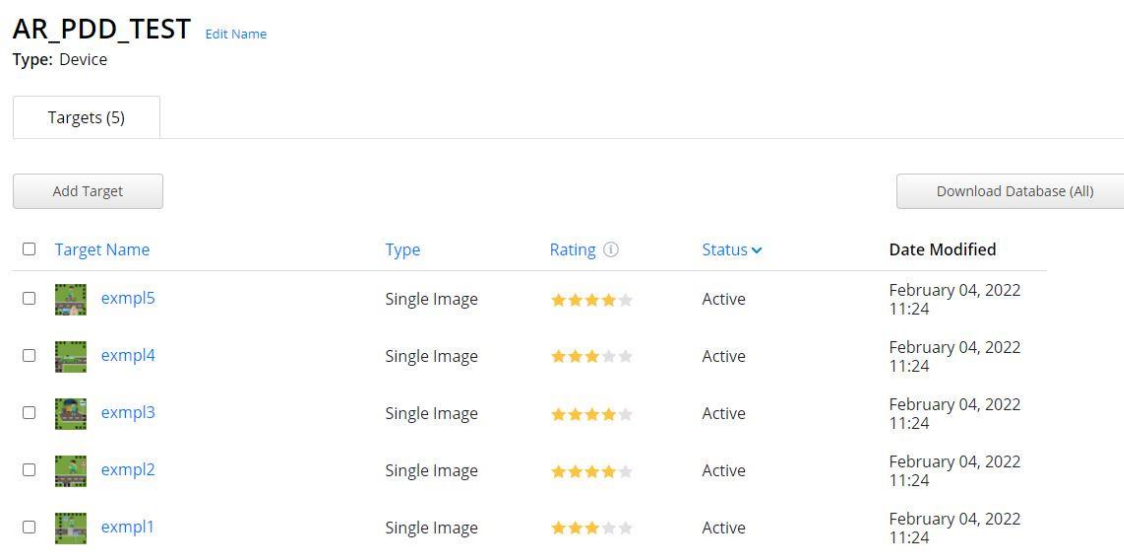


Рисунок 9 – Создание первого маркера

По аналогии расставляем точки для оставшихся маркеров, не забывая при этом каждый раз использовать оригинальную комбинацию расположения точек. «Маркеры, загруженные в портал, заранее обрабатываются алгоритмом и скачиваются специальным пакетом для Unity. После распаковки в ImageTarget выбирается пакет маркеров» [6]. Система оценивает по пятибалльной шкале распознаваемость маркера, а также позволит нам импортировать данные маркеры в разрабатываемое приложение (рисунок 10).



The screenshot shows a web interface for an AR application named "AR_PDD_TEST". It displays a list of 5 targets. Each target has a checkbox, a small image icon, a name (exmpl1 to exmpl5), a type (Single Image), a rating (4 stars), a status (Active), and a date modified (February 04, 2022 11:24). There are buttons for "Add Target" and "Download Database (All)".






| <input type="checkbox"/> | Target Name | Type | Rating ^① | Status [▼] | Date Modified |
|--------------------------|--|--------------|---------------------|---------------------|-------------------------|
| <input type="checkbox"/> |  exmpl5 | Single Image | ★★★★☆ | Active | February 04, 2022 11:24 |
| <input type="checkbox"/> |  exmpl4 | Single Image | ★★★★☆ | Active | February 04, 2022 11:24 |
| <input type="checkbox"/> |  exmpl3 | Single Image | ★★★★☆ | Active | February 04, 2022 11:24 |
| <input type="checkbox"/> |  exmpl2 | Single Image | ★★★★☆ | Active | February 04, 2022 11:24 |
| <input type="checkbox"/> |  exmpl1 | Single Image | ★★★★☆ | Active | February 04, 2022 11:24 |

Рисунок 10 – База данных маркеров

Как мы можем наблюдать наши отметки были оценены системой в 3-4 звезды. Это означает что маркеры могут хорошо распознаваться и отслеживаться приложением, но также могут возникнуть проблемы при трекинге. Далее необходимо скачать нашу ново созданную базу данных маркеров, и импортировать в приложение (рисунок 11).

Добавляем к нашей камере дочерний объект типа «Target» и в свойствах данного объекта выбираем базу данных маркеров, а затем и конкретный маркер (рисунок 12).

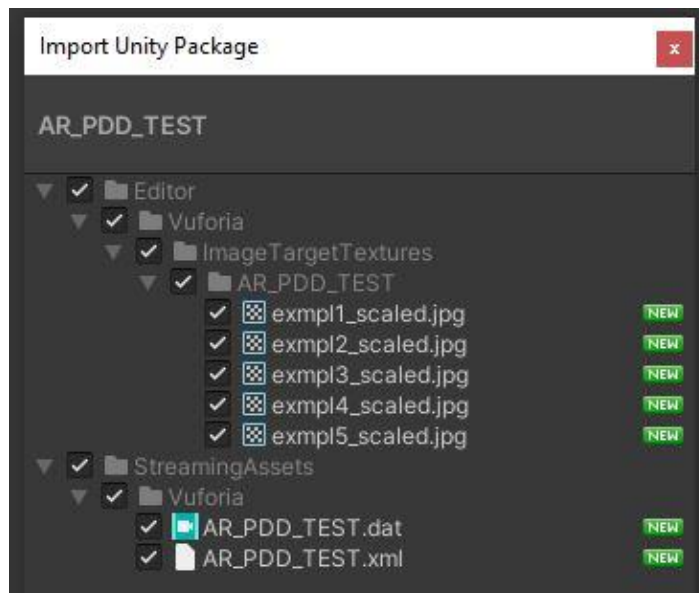


Рисунок 11 – Импорт базы данных маркеров в проект

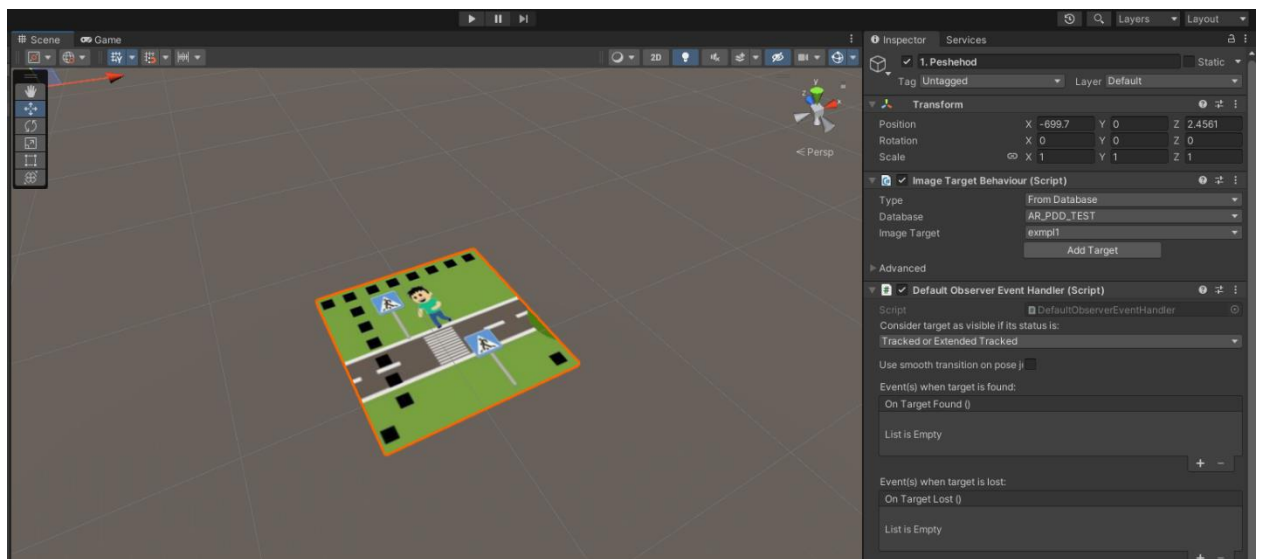


Рисунок 12 – Маркер помещён в виртуальное пространство

Следующий шаг – создание самой модели. Для этого нам необходимо обратиться к ресурсам с бесплатными игровыми ассетами (англ. Asset – модель или текстура). Так как по задумке действие сцены будет происходить на странице книги необходимо разместить основание сцены на маркере пропорционально размерам страницы книги. На рисунке 13 продемонстрирована основа модели сцены на маркере.

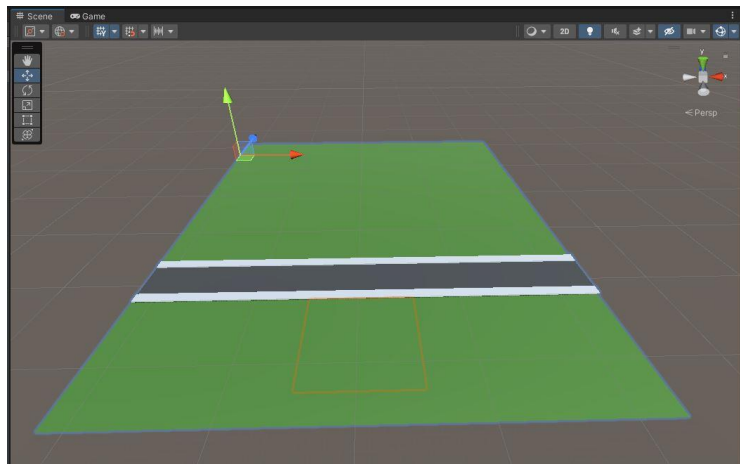


Рисунок 13 – Основание модели на маркере

Как видно на рисунке 14 красными границами обозначено местоположение нашего маркера. Затем необходимо создать макет города используя заготовленные 3Д модели и текстуры. Для начала разместим дома и тропинки.

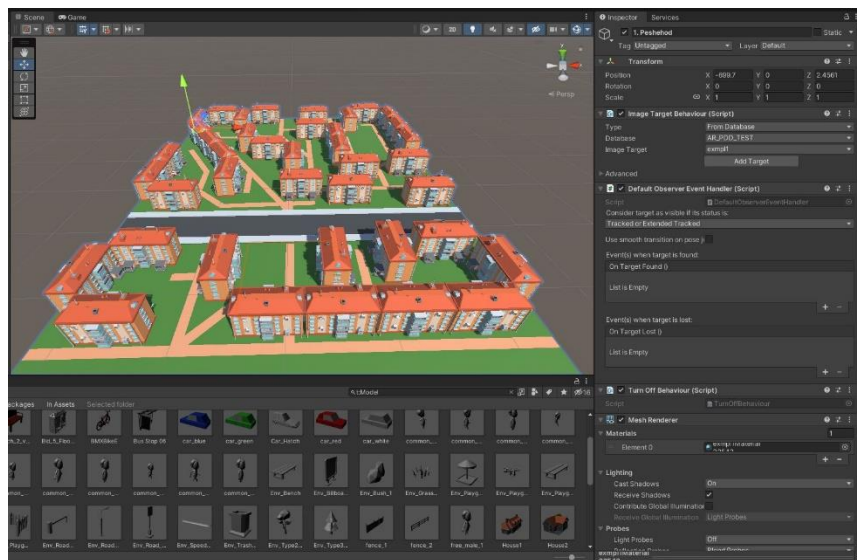


Рисунок 14 – Дома и тропинки

Далее заполним дворы и аллеи скамейками, растительностью и элементами детского уличного игрового оборудования (детские площадки). Результат можно увидеть на рисунке 15.

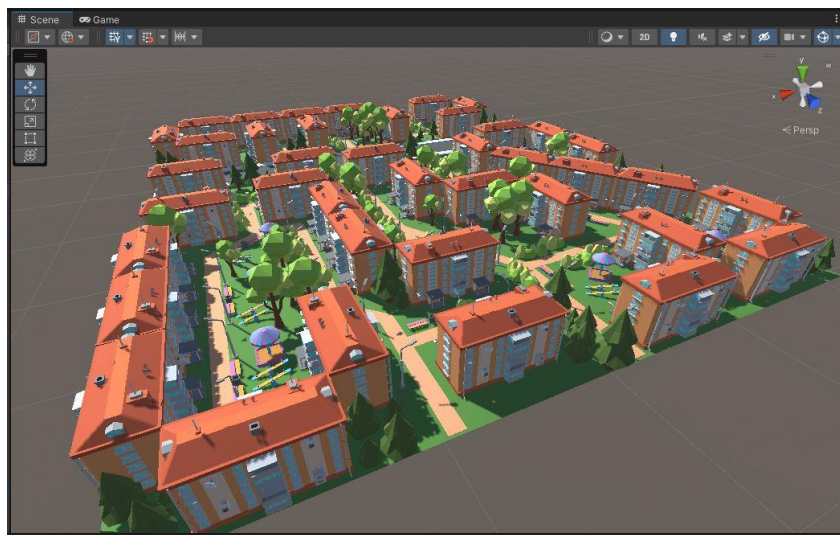


Рисунок 15 – Готовая модель города в виртуальном пространстве

Остаётся лишь добавить объекты, которые в последствии будут анимированы (к ним относятся машины, движущиеся по проезжей части, а также наш переходящий дорогу персонаж), а также специфичные для этой сцены элементы. В данном случае (рисунок 16) это будут лежащие полицейские, знаки пешеходного перехода и непосредственно сам пешеходный переход.

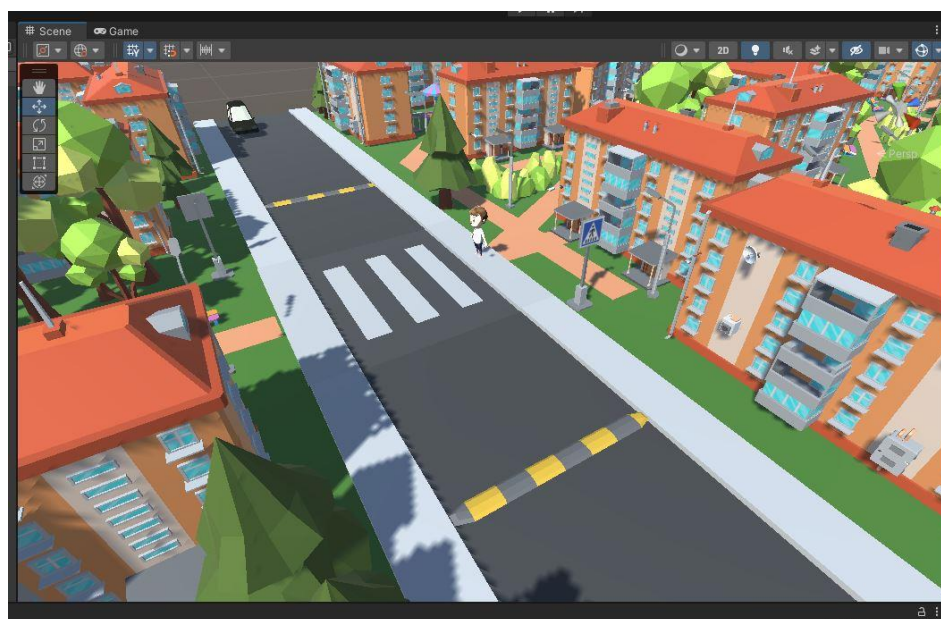
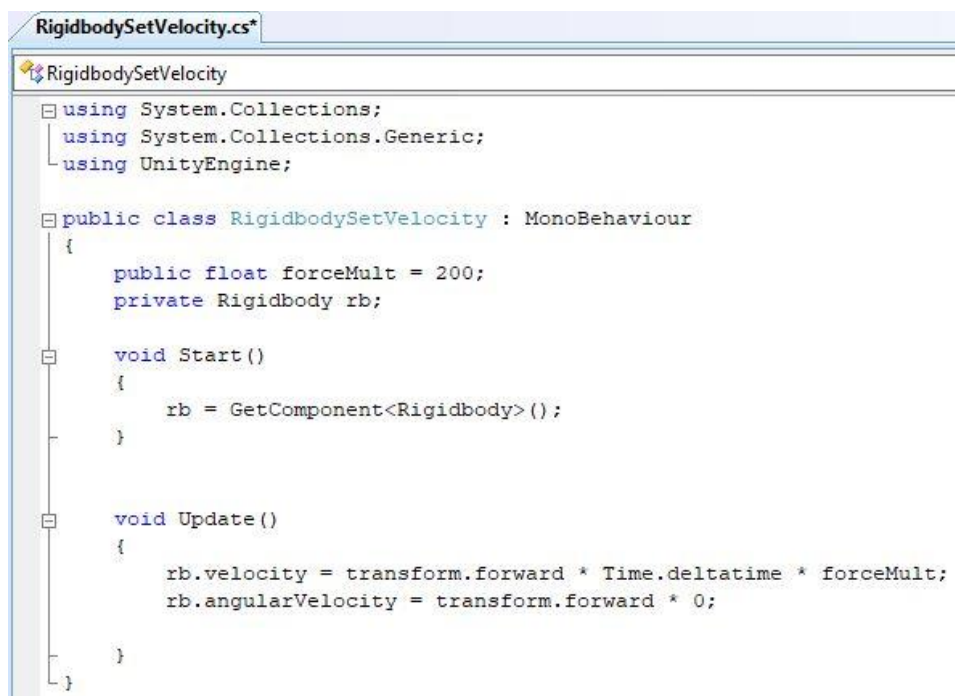


Рисунок 16 – Персонаж и элементы дороги

Теперь, когда отображаемый на метке макет города готов необходимо задать движение персонажу, а также машинам. Для этого создадим класс `RigidbodySetVelocity`, листинг которого предоставлен на рисунке 17.



```
RigidbodySetVelocity.cs*
RigidbodySetVelocity
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RigidbodySetVelocity : MonoBehaviour
{
    public float forceMult = 200;
    private Rigidbody rb;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

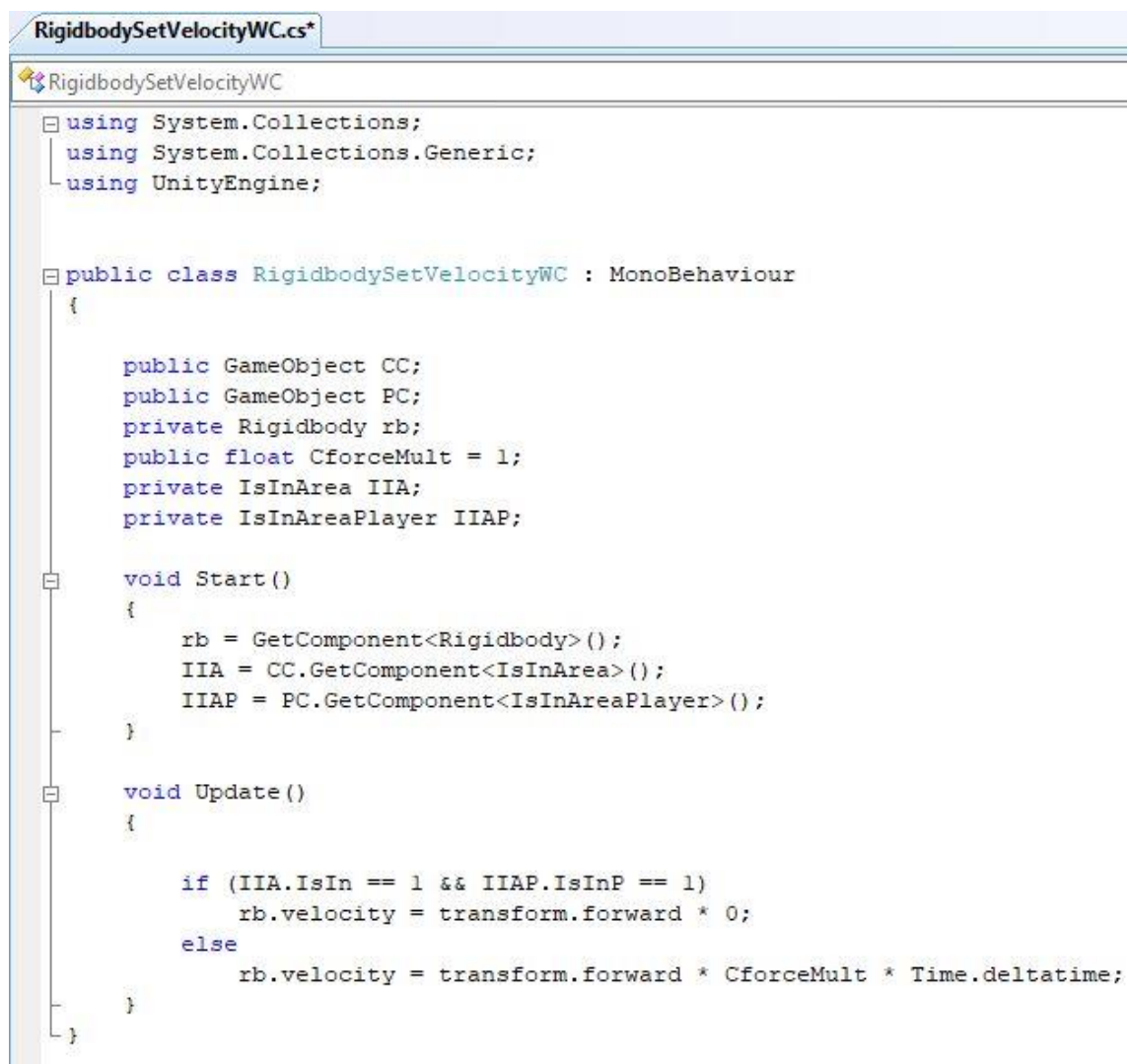
    void Update()
    {
        rb.velocity = transform.forward * Time.deltaTime * forceMult;
        rb.angularVelocity = transform.forward * 0;
    }
}
```

Рисунок 17 – Листинг класса `RigidbodySetVelocity`

В данном классе используется функция `Update`. «Приложение это что-то вроде анимации, в которой кадры генерируются на ходу. Ключевой концепт в программировании игр заключается в изменении позиции, состояния и поведения объектов в игре прямо перед отрисовкой кадра. Такой код в Unity обычно размещают в функции `Update`. `Update` вызывается перед отрисовкой кадра и перед расчётом анимации» [5]. В нашем случае перед отрисовкой каждого кадра мы присваиваем скорости объекта значение равное произведению направления, дельты времени и силы.

Привязав этот класс к моделям машины и персонажа мы увидим и запустив приложение мы увидим, как наши объекты начнут двигаться вперёд. Но простого движения вперёд нам недостаточно, ведь рано или поздно объекты просто столкнутся друг с другом, другими объектами, или вовсе покинут

область модели города. Нам необходимо задать поведение объектов, при котором наш персонаж не будет идти по дороге, когда в расстоянии между лежащими полицейскими находятся машины. Для этого создадим новый класс `RigidbodySetVelocityWC`, листинг которого представлен на рисунке 18.



```
RigidbodySetVelocityWC.cs*
RigidbodySetVelocityWC
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RigidbodySetVelocityWC : MonoBehaviour
{
    public GameObject CC;
    public GameObject PC;
    private Rigidbody rb;
    public float CforceMult = 1;
    private IsInArea IIA;
    private IsInAreaPlayer IIAP;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        IIA = CC.GetComponent<IsInArea>();
        IIAP = PC.GetComponent<IsInAreaPlayer>();
    }

    void Update()
    {
        if (IIA.IsIn == 1 && IIAP.IsInP == 1)
            rb.velocity = transform.forward * 0;
        else
            rb.velocity = transform.forward * CforceMult * Time.deltaTime;
    }
}
```

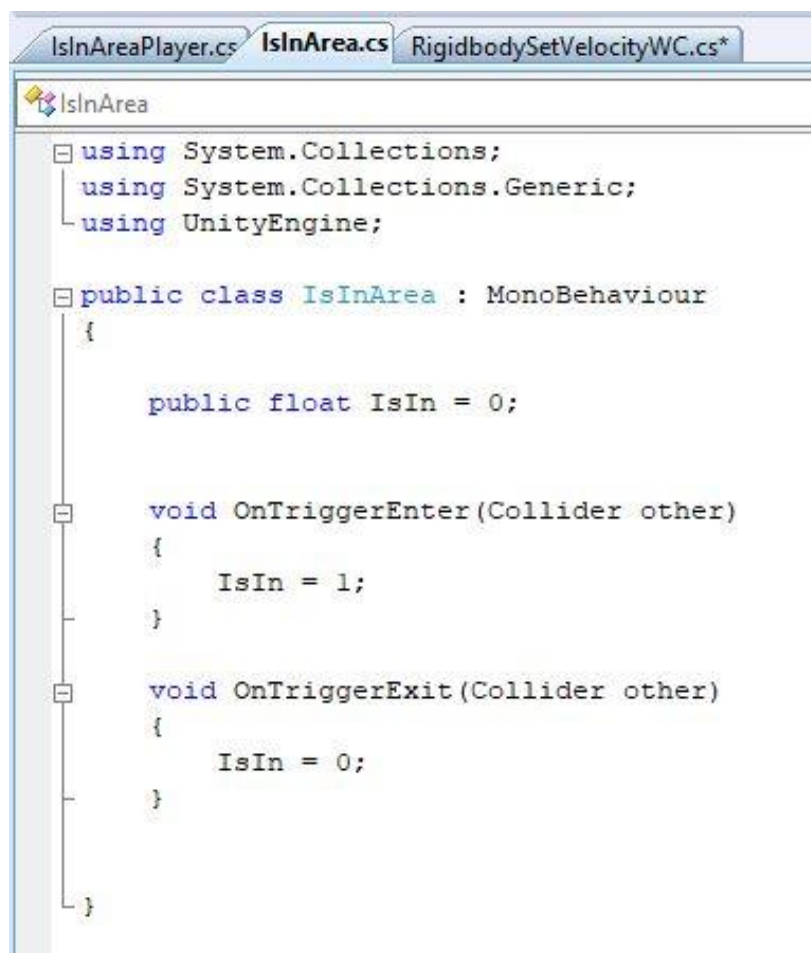
Рисунок 18 – Листинг класса `RigidbodySrtVelocityWC`

За основу возьмём прошлый класс, но при этом добавим несколько условий. Во-первых, объявим переменную `IsInArea` (IIA) которая будет индикатором нахождения машины в области пешеходного перехода, а также переменную `IsInAreaPlayer` (IIAP) которая будет индикатором нахождения персонажа перед пешеходным переходом. «Когда член класса обозначается

спецификатором `public`, он становится доступным из любого другого кода в программе, включая и методы, определенные в других классах» [25].

В функции `Update` поставим следующие условия: если персонаж перед пешеходным переходом и машина в области пешеходного перехода, то скорость персонажа равна нулю, иначе скорость персонажа равна произведению направления, дельты времени и силы. Таким образом наш персонаж, подходя к пешеходному переходу, будет пропускать машины, прежде чем сам начнёт переходить дорогу.

Чтобы отслеживать нахождение в обозначенных областях персонажа и машин создадим классы `IsInArea` и `IsInAreaPlayer`. Листинг данных классов отображён на рисунках 19 и 20.



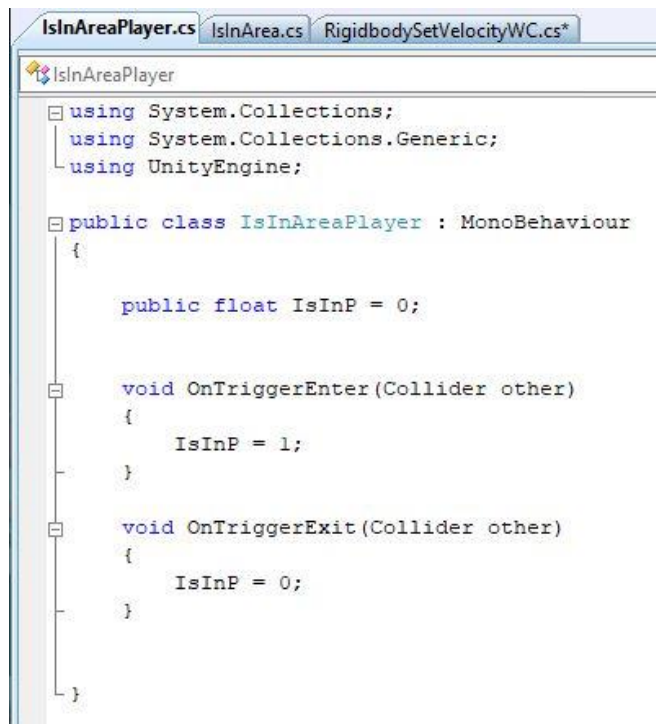
```
IsInAreaPlayer.cs | IsInArea.cs | RigidbodySetVelocityWC.cs*
IsInArea
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class IsInArea : MonoBehaviour
{
    public float IsIn = 0;

    void OnTriggerEnter(Collider other)
    {
        IsIn = 1;
    }

    void OnTriggerExit(Collider other)
    {
        IsIn = 0;
    }
}
```

Рисунок 19 - Листинг класса `IsInArea`



```
IsInAreaPlayer.cs | IsInArea.cs | RigidbodySetVelocityWC.cs*
IsInAreaPlayer
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class IsInAreaPlayer : MonoBehaviour
{
    public float IsInP = 0;

    void OnTriggerEnter(Collider other)
    {
        IsInP = 1;
    }

    void OnTriggerExit(Collider other)
    {
        IsInP = 0;
    }
}
```

Рисунок 20 - Листинг класса IsInAreaPlayer

Данные классы используют функцию OnTriggerEnter, которая будет исполняться при контакте персонажа или машины с обозначенной областью, а также OnTriggerExit, которая будет исполняться, как только персонаж или машина перестаёт контактировать с объектом. Таким образом переменные ПА и ПАР будут равны единице, когда объекты находятся в области, и нулю – когда нет.

Далее необходимо создать объекты, которые будут играть роль границ области, и применить к ним написанные классы. На рисунке 21 показана созданная область.

«Наблюдая за тем, как учатся дети, вы наверняка замечаете, что они действуют по шаблону. Они подвергают его испытанию – ведь чтобы научиться чему-то, недостаточно одних только наставлений. Учиться нужно и на своих ошибках. Ребёнок испытывает норму на прочность, чтобы проверить, насколько сильно она прогнётся. Он готов прокручивать одну и ту же ситуацию снова и снова, снова и снова» [10]. В нашем же приложении все объекты

движутся без остановки вперёд, и запуская приложение мы увидим анимацию лишь один раз.



Рисунок 21 – Область пешеходного перехода

Чтобы решить данную проблему создадим новый класс Teleporting, с помощью которого мы сможем зациклить анимацию объектов, благодаря чему пользователь приложения сможет без перезапуска раз за разом пересматривать анимацию с различных сторон. Листинг класса предоставлен на рисунке 22.

```
Teleporting.cs | IsInAreaPlayer.cs | IsInArea.cs | RigidbodySetVelocityWC.cs*
Teleporting
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Teleporting : MonoBehaviour
{
    public Transform teleportTarget;
    public GameObject thePlayer;

    void OnTriggerEnter(Collider other)
    {
        thePlayer.transform.position = teleportTarget.transform.position;
    }
}
```

Рисунок 22 – Листинг класса Teleporting

Данный класс так же использует функцию `OnTriggerEnter`, благодаря которой в момент соприкосновения персонажа с первым объектом [24] (назовём его входом) персонажу будут присваиваться координаты второго объекта (выхода) (рисунок 23). Остаётся лишь создать вход и выход для каждого объекта, и привязать к ним класс `Teleporting`. Чтобы получить эффект реалистичности точки входа и выхода для персонажа установим в подъездах домов, таким образом персонаж будет не просто телепортироваться из одной точки в другую, а как будто выходить и входить в подъезд.

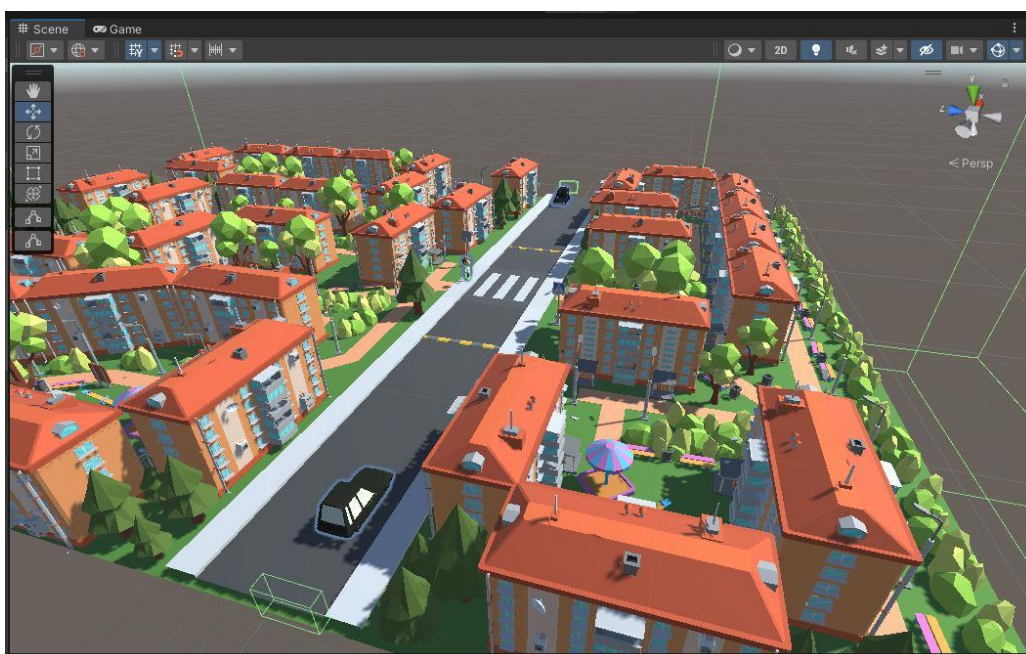


Рисунок 23 – Точки входа и выхода для движущихся объектов

Теперь, запустив приложения и поднеся к камере маркер, мы увидим (рисунок 24) созданный нами макет города, а также анимацию – движущиеся по дороге машины, и переходящий дорогу персонаж.

Используя загруженные модели, созданные классы и макет города по аналогии реализуем оставшиеся 4 сцены, адаптируя объекты и параметры класса под происходящую ситуацию. Для второй, третьей и пятой сцены будет необходимо немного поменять ландшафт города, поменять местоположение персонажа и точек входа и выхода. Во второй сцене так же необходимо

добавить модель остановки. Четвёртую сцену придётся создавать с нуля, так как её действие происходит в лесу.



Рисунок 24 – Персонаж переходит дорогу

Воссоздать правдоподобный лес довольно быстро можно путём копирования групп деревьев различных размеров, расставленных в хаотичном порядке. На рисунке 25 продемонстрированы все готовые сцены.

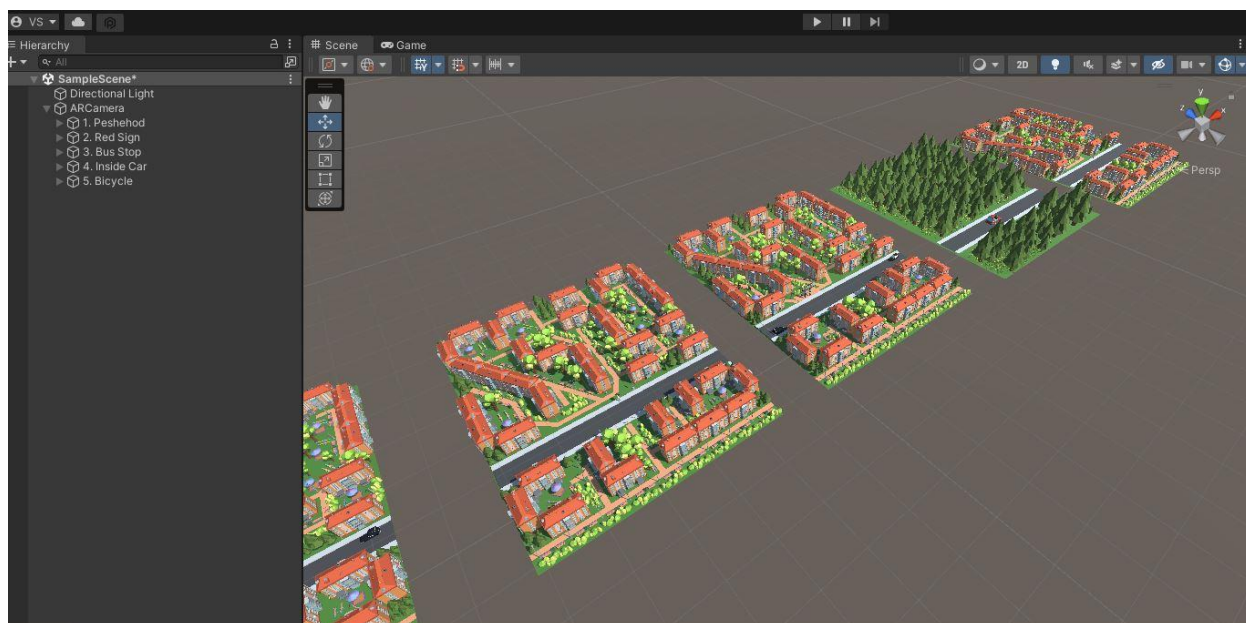


Рисунок 25 – Все макеты

Последний шаг – собрать проект. Для этого в свойствах проекта нажимаем Build Project, и по завершению процесса получаем файл в расширении ark, который в дальнейшем необходимо отправить на мобильный телефон и установить.

Вывод по главе

Во второй главе рассмотрена методика компьютерного зрения, описана математическая модель дополненной реальности.

Описан процесс создания маркеров на основе рассмотренной методики компьютерного зрения, а также процесс разработки приложения с использованием данных маркеров и описанной модели.

Глава 3 Тестирование и корректировка разработанного программного модуля

3.1 Проведение вычислительных экспериментов

Теперь, когда приложение установлено на нашем устройстве запустим его и проверим работоспособность, а также качество распознавания и отслеживания меток. Наведя камеру на первую метку, мы увидим созданную нами модель (рисунок 26). Засечём время, которое приложение потратит на распознавание метки и размещение модели.



Рисунок 26 - Модель на маркере в реальном мире

Замерив время распознавания отметки, получаем следующий результат:

- сцена с пешеходным переходом: 6.3 секунд;

- сцена у проезжей части: 7.6 секунд;
- сцена с автобусной остановкой: 4.8 секунд;
- сцена за городом: 3.2 секунды;
- сцена с велосипедом: 7.8 секунд.

Рассчитаем среднее время распознавания отметок – 5.94 секунды. Результат не самый лучший, так как в идеале для предоставления пользователю лучшего опыта приложение должно практически моментально распознавать маркеры. Далее необходимо проверить как хорошо приложение отслеживает метку и размещает поверх неё модель. На рисунке 27 отображена работа приложения при наведении камеры на метку под углом.



Рисунок 27 – Некорректное отображение модели

Как мы можем заметить модель не совсем корректно размещается поверх маркера. Причиной данного поведения скорее всего является то, что

приложение не может адекватно отследить положение точек маркера, находящихся под углом. Так же если мы захотим рассмотреть модель города в верхней части сцены, то модель просто перестанет отображаться, так как маркер, находящийся в нижней части страницы полностью пропадает из зоны видимости.

Ещё одна проблема приложения – скорость воспроизведения анимации напрямую зависит от устройства, на котором приложение запущено. На устройствах разных ценовых категорий приложение будет показывать разное количество кадров в секунду, а так функции классов привязаны к количеству кадров в секунду на устройствах с меньшей производительностью анимация будет проигрываться медленнее, чем на устройствах с хорошей производительностью.

3.2 Корректировка разработанной информационной технологии для обработки математических данных

Чтобы модель корректно отображалась на странице книги под любым углом можно расширить область маркера до размеров всей страницы. Таким образом процесс отслеживания для программы будет упрощён – даже если часть страницы и точек маркера будут вне поля зрения камеры. К тому же в таком случае пользователь сможет рассмотреть любую часть модели не боясь того, что она пропадёт с экрана. На рисунке 28 показана новая версия маркера.

Вместо монотонно чёрных используем чёрно-белые точки. Таким образом мы повысим количество отслеживаемых деталей с 4 до 12 у каждой точки, что должно значительно улучшить качество трекинга [17]. Повторяем данный процесс для каждой из страниц, и загружаем новые метки на сайт Vuforia. На этот раз система оценила каждый маркер в пять звёзд (рисунок 29), что означает, что данные маркеры будут отлично отслеживаться приложением.



Рисунок 28 – Новая версия маркера

ARTR_BOOK_updated [Edit Name](#)
Type: Device

Targets (5)

Add Target Download Database (All)






| <input type="checkbox"/> | Target Name | Type | Rating ^① | Status [▼] | Date Modified |
|--------------------------|--|--------------|---------------------|---------------------|----------------------|
| <input type="checkbox"/> |  NewPageMark5 | Single Image | ★★★★★ | Active | March 08, 2022 13:57 |
| <input type="checkbox"/> |  NewPageMark4 | Single Image | ★★★★★ | Active | March 08, 2022 13:57 |
| <input type="checkbox"/> |  NewPageMark3 | Single Image | ★★★★★ | Active | March 08, 2022 13:56 |
| <input type="checkbox"/> |  NewPageMark2 | Single Image | ★★★★★ | Active | March 08, 2022 13:56 |
| <input type="checkbox"/> |  NewPageMark1 | Single Image | ★★★★★ | Active | March 08, 2022 13:56 |

Рисунок 29 - Модель на обновлённом маркере (маркер отмечен красной рамкой)

По аналогии с прошлым разом импортируем в проект новый набор меток и размещаем модель на метке относительно размера страницы (рисунок 30).

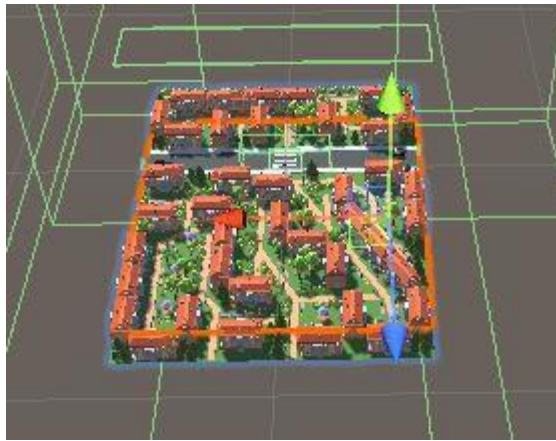


Рисунок 30 – Модель поверх нового маркера

Мы можем использовать `FixedUpdate` вместо `Update`, что позволит нам вне зависимости от частоты кадров вызывать функцию 50 раз в секунду [25], благодаря чему пропадёт зависимость скорости объекта от частоты кадров. Листинг обновлённых классов предоставлен на рисунках 31, 32.

```
RigidbodySetVelocity.cs
RigidbodySetVelocity
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RigidbodySetVelocity : MonoBehaviour
{
    public float forceMult = 2;
    private Rigidbody rb;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate()
    {
        rb.velocity = transform.forward * forceMult;
        rb.angularVelocity = transform.forward * 0;
    }
}
```

Рисунок 31 – Обновлённый листинг класса `RigidbodySetVelocity`

```
RigidbodySetVelocityWC.cs  RigidbodySetVelocity.cs
RigidbodySetVelocityWC
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RigidbodySetVelocityWC : MonoBehaviour
{
    public GameObject CC;
    public GameObject PC;
    private Rigidbody rb;
    public float CforceMult = 1;
    private IsInArea IIA;
    private IsInAreaPlayer IIAP;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        IIA = CC.GetComponent<IsInArea>();
        IIAP = PC.GetComponent<IsInAreaPlayer>();
    }

    void FixedUpdate()
    {
        if (IIA.IsIn == 1 && IIAP.IsInP == 1)
            rb.velocity = transform.forward * 0;
        else
            rb.velocity = transform.forward * CforceMult;
    }
}
```

Рисунок 32 - Обновлённый листинг класса RigidbodySetVelocity

Сохранив проект и собрав приложение, отправим его на мобильное устройство. На рисунке 33 продемонстрировано отображение модели на улучшенной метке.

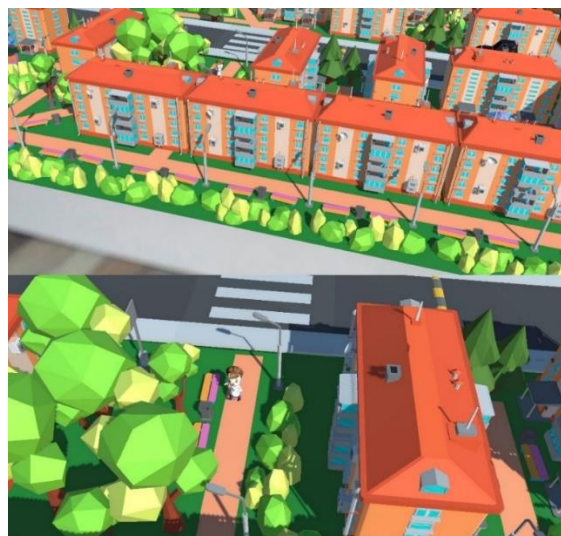


Рисунок 33 – Демонстрация отображения модели на метке

Также необходимо проверить трекинг и наложение модели на маркер при наведении камеры под углом (рисунок 34).



Рисунок 34 – Модель на метке под углом

Как мы можем наблюдать на рисунке 34, благодаря расширению области маркера до размера целой страницы приложению гораздо легче отследить положения маркера. Теперь сцена отображается практически моментально при наведении камеры мобильного устройства на страницу – метку, что является отличным результатом. Также процесс трекинга стал более плавным – модель не смещается с метки и хорошо отображается под углом и даже при частичной видимости маркера.

Вывод по главе

В третьей главе описан процесс тестирования функционала и производительности разработанного приложения.

Были внесены корректировки в код приложения и произведено повторное тестирование с целью сравнения стабильности работы и производительности приложения. В результате корректировок работа приложения значительно улучшилась.

Заключение

В выпускной квалификационной работе были рассмотрены методы компьютерного зрения и трекинга, описана математическая модель дополненной реальности, разработан, а также протестирован, программный модуль для реализации данной модели.

В результате работы была получена математическая модель дополненной реальности, которая может быть использована для разработки приложений с технологией дополненной реальности. На основе описанной методики компьютерного зрения созданы маркеры для их дальнейшего трекинга приложением.

Также было реализовано приложение с использованием технологии дополненной реальности для обучения детей основам правил дорожного движения. Данное приложение может быть использовано в обучении как наглядный пример основных правил дорожного движения.

Для создания реализации приложения была выбрана платформа Unity, позволяющая разработать мультиплатформенное приложение, не занимающее большого пространства в памяти устройства, а также AR – библиотека Vuforia.

Разработанное приложение было протестировано на предмет правильности работы и производительности. В ходе тестирования выявлены недостатки в виде долгого распознавания маркера приложением, а также временами некорректного отображения модели на распознанном маркере.

Найденные недочёты были исправлены, на основе описанной методики компьютерного зрения созданные маркеры были изменены для лучшего распознавания и процесса трекинга. В ходе тестирования новой версии приложения проблем с работоспособностью выявлено не было, отмечено значительное улучшение производительности.

Список используемой литературы

1. Акулич М. Дополненная, виртуальная, смешанная реальность и маркетинг. ISBN 978-5-4490-2111-3. 2021 г.
2. Антониади К.С., Грубич Т.Ю. Применение VR и AR технологий в образовании // Новые импульсы развития: вопросы научных исследований. 2020. №2. С. 26 – 30. URL: <https://cyberleninka.ru/article/n/primeneniye-vr-i-ar-tehnologiy-v-obrazovanii> (дата обращения: 15.02.2022).
3. Бажина П.С., Куприенко А.А. Опыт применения технологии дополненной реальности в образовании // МНКО. 2018. №3. С. 244 – 247. URL: <https://cyberleninka.ru/article/n/opyt-primeneniya-tehnologii-dopolnennoy-realnosti-v-obrazovanii> (дата обращения: 15.02.2022).
4. Биткин В.В. Дополненная реальность, её виды и инструменты создания // Скиф. 2021. №5. С. 106 – 109. URL: <https://cyberleninka.ru/article/n/dopolnennaya-realnost-eyo-vidy-i-instrumenty-sozdaniya> (дата обращения: 15.02.2022).
5. Бонд Д.Г. Unity и C#. Геймдев от идеи до реализации. Россия: Издательство: Питер, 2019.
6. Демедюк А.С. Первые шаги разработки дополненной реальности // E-Scio. 2020. №3 . С 129 – 135. URL: <https://cyberleninka.ru/article/n/pervyye-shagi-razrabotki-dopolnennoy-realnosti> (дата обращения: 15.02.2022).
7. Дюличева Ю.Ю. О применении технологии дополненной реальности в процессе обучения математике и физике. // Открытое образование. 2020;24(3): С. 44 – 55. URL: <https://doi.org/10.21686/1818-4243-2020-3-44-55> (дата обращения: 15.02.2022).
8. Иванова А.В. Технологии виртуальной и дополненной реальности: возможности и препятствия применения // CPPM. 2018. №3. С. 88 – 107. URL: <https://cyberleninka.ru/article/n/tehnologii-virtualnoy-i-dopolnennoy-realnosti-vozmozhnosti-i-prepyatstviya-primeneniya> (дата обращения: 12.04.2022).

9. Кинг Б. Эпоха дополненной реальности. Россия: Олимп-Бизнес. 2018 г.
10. Костер Р. Разработка игр и теория развлечений. Россия: ДМК. 2018 г.
11. Маслова Ю.А., Белов Ю.С. Технологии дополненной реальности // E-Scio. 2022. №2 (65). 2022 г. С. 312 – 322. URL: <https://cyberleninka.ru/article/n/tehnologii-dopolnennoy-realnosti> (дата обращения: 15.02.2022).
12. Михайлов А.С. Обнаружение маркеров в технологии дополненной реальности // Актуальные проблемы авиации и космонавтики. 2019 г. №. С. 149 – 151. URL: <https://cyberleninka.ru/article/n/obnaruzhenie-markerov-v-tehnologii-dopolnennoy-realnosti> (дата обращения: 15.02.2022).
13. Мухамадиева К.Б. Дополненная и виртуальная реальность в образовании // Образование и проблемы развития общества. 2021 г. №1 (14). С. 68 – 75. URL: <https://cyberleninka.ru/article/n/dopolnennaya-i-virtualnaya-realnost-v-obrazovanii> (дата обращения: 13.04.2022).
14. Набокова Л.С. Загидуллина Ф.Р., Перспективы внедрения технологий дополненной и виртуальной реальности в сферу образовательного процесса высшей школы // Профессиональное образование в современном мире. 9(2):2710/2719. 2019 г.
15. Папагианнис Х. Дополненная реальность. Все, что вы хотели узнать о технологии будущего. Россия: Бомбора (Эксмо). 2019 г.
16. Руководство Unity [Электронный ресурс] URL: <https://docs.unity3d.com/ru/current/Manual/index.html>. (дата обращения: 13.04.2022).
17. Руководство Vuforia Engine Unity [Электронный ресурс] URL: <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html> (дата обращения: 13.04.2022).
18. Сайын И.С., Морозов Б.Б. Анализ использования дополненной реальности в образовании // Вестник науки. 2020 г. №12 (33). URL:

<https://cyberleninka.ru/article/n/analiz-ispolzovaniya-dopolnennoy-realnosti-v-obrazovanii> (дата обращения: 15.02.2022).

19. Смолин А.А., Жданов Д.Д., Потемин И.С., Меженин А.В., Богатырев В.А. системы виртуальной, дополненной и смешанной реальности. Санкт-Петербург: Университет ИТМО. 2018 г.

20. Сотников А.М., Тычков А.Ю., Золотарев Р.В., Николаева М.А., Петкилева А.А. дополненная и виртуальная реальность в образовании как инструмент осознанного обучения // Вестник ПензГУ. 2021 г. №4 (36). С. 117 – 122. URL: <https://cyberleninka.ru/article/n/dopolnennaya-i-virtualnaya-realnost-v-obrazovanii-kak-instrument-osoznannogo-obucheniya> (дата обращения: 15.02.2022).

21. Суворова Е.Ю. образовательный потенциал дополненной реальности // Известия ВГПУ. 2021 г. №4 (157). С. 30 – 35. URL: <https://cyberleninka.ru/article/n/obrazovatelnyy-potentsial-dopolnennoy-realnosti> (дата обращения: 15.02.2022).

22. Таран В.Н. Применение дополненной реальности в обучении // Проблемы современного педагогического образования. 2018 г. №60-2. С. 333 – 337. URL: <https://cyberleninka.ru/article/n/primenenie-dopolnennoy-realnosti-v-obuchanii> (дата обращения: 15.02.2022).

23. Филимоненкова Т.Н. Дополненная реальность как инновационная технология образовательного процесса // Проблемы современного педагогического образования. 2018 г. №58-1. С. 246 – 251. URL: <https://cyberleninka.ru/article/n/dopolnennaya-realnost-kak-innovatsionnaya-tehnologiya-obrazovatel'nogo-protsesssa> (дата обращения: 12.04.2022).

24. Хокинг Д. Unity в действии. Мультиплатформенная разработка на C#. Россия: Издательство: Питер. 2019 г.

25. Шилдт Г. C# 4.0. Полное руководство. М.: Издательский дом «Вильямс». 2019 г.

26. Whyte J., Dragana N. Virtual Reality and the Built Environment. CRC Press. 2018 г.

27. Ahram T., Christianne F. *Advances in Usability and User Experience*. Springer; 1st ed. 2019 г.
28. Claudia M.t.D., Timothy H.J., Sandra M.C.L. *Augmented Reality and Virtual Reality: New Trends in Immersive Technology*. Springer; 1st ed. 2021 г.
29. Pangilinan E., Steve L., Vasanth M. *Creating Augmented and Virtual Realities: Theory and Practice for Next-Generation Spatial Computing*. O'Reilly Media; 1st edition. 2019 г.
30. Tyagi A. *Multimedia and Sensory Input for Augmented, Mixed, and Virtual Reality*. IGI Global; 1st edition. 2021 г.