

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики

Кафедра информационных и управляющих систем

Направление подготовки 09.03.01 – Информатика и вычислительная техника

Направленность (профиль) образовательной программы Автоматизированные системы обработки информации и управления

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

 А.В. Бушманов

« 29 » июль 2020 г.

**БАКАЛАВРСКАЯ РАБОТА**

на тему: Разработка приложения дополненной реальности для изучения школьного курса физики

Исполнитель  
студент группы 653об

 19.06.2020

(подпись, дата)

Ю.Е. Шикунова

Руководитель  
доцент к.ф.-м.н.

 19.06.2020

(подпись, дата)

В.В. Ерёмина

Консультант:  
по безопасности жизне-  
деятельности доцент

 19.06.2020

(подпись, дата)

А.Б. Булгаков

Нормоконтроль

 19.06.2020

(подпись, дата)

В.В. Ерёмина

Благовещенск 2020

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
(ФГБОУ ВО «АмГУ»)

Факультет Математики и информатики

Кафедра Информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

 А.В. Бушманов

« 29 » июль 2020 г.

**ЗАДАНИЕ**

К выпускной квалификационной работе студента Шикунова Ю.Е.

1. Тема выпускной квалификационной работы: Разработка мобильного приложения дополненной реальности для изучения школьного курса физики  
(утверждена приказом от 30.04.2020 №810 –уч)

2. Срок сдачи студентом законченной работы 15.06.2020

3. Содержание выпускной квалификационной работы: анализ предметной области; анализ и выбор программных средств; разработка и эксплуатация программного продукта; безопасность и экологичность.

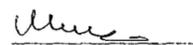
4. Перечень материалов приложения: UML-диаграмма классов, UML-диаграмма использования, экранные формы, листинг программы.

5. Дата выдачи задания 01.09.2019

Руководитель бакалаврской работы В. В. Ерёмина, доцент к.ф.-м.н.

(фамилия, имя, отчество, должность, уч.степень, уч.звание)

Задание принял к исполнению (01.09.2019)



(подпись студента)

## РЕФЕРАТ

Выпускная квалификационная работа 80с., 41 рисунок, 13 таблиц, 20 источников, 5 приложений.

AR – AUGMENTED REALITY, SDK – SOFTWARE DEVELOPMENT KIT, UML – UNIFIED MODELING LANGUAGE, UI – USER INTERFACE, ПО – ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

В работе была изучена технология дополненной реальности, ее применение в сфере школьного образования, выполнена разработка мобильного приложения с дополненной реальностью.

Цель работы – разработка мобильного приложения дополненной реальности для изучения школьного курса физики.

Объект исследования – технология дополненной реальности.

Результатом работы является мобильное приложение с дополненной реальностью для изучения школьного курса физики.

## СОДЕРЖАНИЕ

|  |    |
|--|----|
| ВВЕДЕНИЕ   | 6  |
| 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ  | 8  |
| 1.1 Определение дополненной реальности   | 8  |
| 1.2 Принцип работы технологии AR   | 8  |
| 1.3 Устройства, реализующие AR   | 10 |
| 1.4 Применение технологии дополненной реальности в сфере школьного образования | 12 |
| 2 АНАЛИЗ И ВЫБОР ПРОГРАММНЫХ СРЕДСТВ   | 17 |
| 2.1 Кроссплатформенный движок Unity  | 17 |
| 2.2 Анализ SDK для дополненной реальности                                      | 18 |
| 2.2.1 Платформа Apple ARKit  | 18 |
| 2.2.2 Платформа GoogleARCore   | 19 |
| 2.2.3 Библиотека AR Foundation   | 19 |
| 2.2.4 Библиотека Vuforia   | 20 |
| 2.2.5 Платформа ARToolKit  | 21 |
| 2.2.6 Платформа 8th Wall   | 21 |
| 2.3 Графические редакторы для 3д-моделирования                                 | 22 |
| 2.3.1 Редактор 3Ds Max   | 22 |
| 2.3.2 Редактор Blender   | 22 |
| 2.3.3 Редактор Modo  | 23 |
| 2.4 Языки программирования в разработке AR приложений                          | 24 |
| 2.4.1 Язык программирования C#   | 24 |
| 2.4.2 Язык программирования C++  | 24 |
| 2.4.3 Язык программирования JavaScript   | 25 |
| 2.5 Текстовые редакторы для написания кода                                     | 26 |
| 2.5.1 Редактор Visual Studio Code  | 26 |
| 2.5.1 Редактор sublime text  | 26 |
| 2.6 Метод SLAM   | 27 |
| 2.7 Обоснование выбора программных средств                                     | 28 |
| 3 РАЗРАБОТКА И ЭКСПЛУАТАЦИЯ ПРОГРАММНОГО ПРОДУКТА                              | 29 |

|       |  |    |
|-------|--|----|
| 3.1   | Постановка задачи и определение требований к программному продукту   | 29 |
| 3.2   | Создание проекта и внедрение AR                                      | 32 |
| 3.3   | Виртуальный контент, разработка 3д-моделей                           | 34 |
| 3.3.1 | Модель солнечной системы   | 35 |
| 3.3.2 | Модель автомобиля  | 38 |
| 3.3.3 | Модель реализации правила правой руки                                | 40 |
| 3.4   | Реализация компонентов, написание программного кода                  | 41 |
| 3.4.1 | Отслеживание целей, программный код добавления виртуального контента | 44 |
| 3.4.2 | Программный код управления объектами                                 | 48 |
| 3.4.3 | Реализация экранов главного меню и теоретических материалов          | 56 |
| 3.5   | Сборка и тестирование проекта  | 56 |
| 4     | БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ   | 59 |
| 4.1   | Безопасность   | 59 |
| 4.1.1 | Пользовательский интерфейс   | 59 |
| 4.1.2 | Организация рабочего места   | 61 |
| 4.2   | Экологичность  | 66 |
| 4.3   | Чрезвычайные ситуации  | 68 |
|       | ЗАКЛЮЧЕНИЕ   | 72 |
|       | БИБЛИОГРАФИЧЕСКИЙ СПИСОК   | 73 |

## ВВЕДЕНИЕ

Дополненная реальность является одним из наиболее перспективных и востребованных направлений среди информационных технологий. AR-разработка имеет широкое применение среди различных сфер деятельности, таких как: медицина, игровая индустрия, реклама, торговля, образование, повседневная жизнь и другое.

Не следует путать VR и AR. Основная идея AR — это дополнить наш реальный физический мир, каким мы его видим, компьютерной графикой (3d- и 2d-модели, звук и другой контент), что позволит расширить визуальные возможности представления информации. Идея же VR-технологии — это полная замена реального мира искусственным, созданным разработчиком с помощью специального ПО.

В сфере образования использование дополненной реальности позволяет сфокусировать внимание учеников и представить процесс обучения в виде игры, представить объект изучения в виде визуальной модели, наглядно показать какой-либо процесс или явление, вследствие чего ученик может наглядно представить и выделить ключевую информацию. Технология дополненной реальности появилась относительно недавно и на данный момент используется больше в развлекательных и торговых сферах. В образовательной сфере AR используется реже. Говорить о массовом переходе школьников от простых бумажных учебников к мобильным или иным приложениям с AR пока рано, но в некоторых учебных заведениях все же подобные технологии используются. Нет сомнений что, скоро AR войдет в повседневную школьную дисциплину, поэтому сейчас разработка в этом направлении является актуальной и востребованной.

Предмет исследования — разработка мобильного приложения для изучения школьного курса физики с AR, на платформу android с помощью AR Foundation SDK на базе кроссплатформенного движка Unity. Данное приложение должно содержать теоретическую и практическую (сцены с AR) части.

Разрабатываемый интерфейс приложения должен быть интуитивно понятен и доступен для пользователя. Объект исследования в данной работе — платформа дополненной реальности AR Foundation.

В первой главе содержится описание предметной области, определение дополненной реальности, необходимые технические устройства, применение AR-технологий в сфере образования. Вторая глава содержит обзор и выбор программных средств, используемых при разработке AR-приложений. В третьей главе описан процесс разработки и описание готового продукта. В четвертой главе содержатся основы безопасности жизнедеятельности при разработке данного ПО.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Определение дополненной реальности

Дополненная реальность — это среда, в реальном времени дополняющая физический мир, каким мы его видим, цифровыми данными с помощью каких-либо устройств. Дополненную реальность (augmented reality, AR) надо отличать от виртуальной (virtual reality, VR) и смешанной (mixed reality, MR). В дополненной реальности виртуальные объекты проецируются на реальное окружение.

Виртуальная реальность — это созданный техническими средствами мир, передаваемый человеку через органы чувств.

Смешанная или гибридная реальность объединяет оба подхода.

То есть, виртуальная реальность создает свой мир, куда может погрузиться человек, а дополненная добавляет виртуальные элементы в мир реальный. Выходит, что VR взаимодействует лишь с пользователями, а AR — со всем внешним миром.

## 1.2 Принцип работы технологии AR

Специфика дополненной реальности заключается в программном совмещении двух независимых пространства: реальный мир вокруг нас и виртуальный мир, воссозданный на компьютере. Новая виртуальная среда образуется путем наложения цифровых виртуальных объектов поверх видеосигнала с камеры, и становится интерактивной путем использования специальных маркеров. Основа технологии дополненной реальности – система оптического трекинга. Камера распознает маркеры в реальном мире, «переносит» их в виртуальную среду, накладывает один слой реальности на другой и таким образом создает мир дополненной реальности.

Базовый упрощенный алгоритм работы дополненной реальности можно описать следующим образом:

1. Камера устройства считывает изображение реального объекта.

2. Программное обеспечение анализирует и обрабатывает полученные с камеры данные.

3. Осуществляется поиск в базе данных виртуального объекта, который соответствует обработанным с камеры данным. Если виртуальный объект найден, то он подготавливается для вывода на устройство визуализации.

4. Реальное изображение дополняется виртуальным объектом, и результат (итоговое изображение) выводится на устройство.

Существуют три основных направления в развитии этой технологии:

- «Безмаркерная» технология — на окружающий ландшафт, снятый камерой, накладывается виртуальная «сетка». На этой сетке программные алгоритмы находят некие опорные точки, по которым определяют точное место, к которому будет «привязана» виртуальная модель. Преимущество такой технологии в том, что объекты реального мира служат маркерами сами по себе и для них не нужно создавать специальных визуальных идентификаторов.
- Технология на базе специальных маркеров, или меток, удобна тем, что они проще распознаются камерой и дают ей более жесткую привязку к месту для виртуальной модели. Такая технология гораздо надежнее «безмаркерной» и работает практически без сбоев.
- Технология дополненной реальности, основанная на пространственном расположении объекта. В ней используются данные GPS/ГЛОНАСС, гироскопа и компаса, встроенного в мобильный телефон. Место виртуального объекта определяется координатами в пространстве. Активация программы дополненной реальности происходит при совпадении координаты, заложенной в программе, с координатами пользователя.

Для работы с технологией дополненной реальности необходимы следующие компоненты:

- Устройство съемки. Это может быть смартфон с камерой, 3D-очки и так далее;

- Дисплей. Экран телефона, телевизор, монитор, проекционный экран и так далее;
- Камера. Благодаря камере мы получаем «слепок» реального мира, на который специальное программное обеспечение накладывает виртуальные объекты;
- Метки, или маркеры при необходимости;
- Программное обеспечение. Математические алгоритмы, базы данных, которые позволяют камере увидеть и распознать метку (маркер) в окружающем пространстве, а затем определить, какая именно модель программно «привязана» к метке и «положить» эту модель на метку.

### 1.3 Устройства, реализующие AR

Как уже говорилось выше, для работы дополненной реальности необходимы определенные технические и программные средства. На сегодняшний день существующие технические устройства для этой задачи можно условно разделить на три группы: мобильные, стационарные и специальные устройства.

К мобильным устройствам относят:

- Планшеты и смартфоны на которых должно быть установлено специальное ПО, также к этой категории существует ряд требований: версия устройства, четкость камеры, оперативная память и другое;
- Очки дополненной реальности — отдельное полноценное устройство, для работы с AR.



Рисунок 1 — Очки дополненной реальности HoloLens 2 компании Microsoft

Они могут проецировать голограммы и информацию в реальное пространство, без привязывания к физическим объектам. Фактически это просто экран перед глазами. На рисунке 1 показаны очки дополненной реальности HoloLens 2 компании Microsoft;

- Линзы для дополненной реальности пока еще остаются технологией будущего. Разработчики стремятся превратить линзы в прозрачный экран, содержащий систему управления, миниатюрную камеру, антенну, светодиоды и другие оптоэлектронные компоненты;
- Перчатки для дополненной реальности с помощью встроенных датчиков способны отслеживать движения руки пальцев и передавать данные другим мобильным или стационарным устройствам. На рисунке 2 показаны перчатки дополненной реальности Gest от компании Apotact Labs;



Рисунок 2 — Перчатки дополненной реальности Gest от компании Apotact Labs

К стационарным устройствам относят: телевизор, экран компьютера, игровой компьютер типа Kinect.

Специальные средства. К ним относятся, например, специализированные шлемы. На стекло шлема выводится необходимая пользователю важная информация, и он может воспринимать ее, не переводя взгляд на приборную панель, экономя тем самым драгоценные секунды. Многие подобные системы позволяют осуществлять целеуказание путём поворота головы или движением глазных яблок пилота. На рисунке 3 показан Шлем дополненной реальности CrossHelmet X1.



Рисунок 3 — Шлем дополненной реальности CrossHelmet X1

#### **1.4 Применение технологии дополненной реальности в сфере школьного образования**

Эдгар Дейл, американский педагог, разработал «Конус опыта» в 1960-х годах. Согласно «Конусу опыта» или «Пирамиде обучения», учащиеся сохраняют больше знаний и информации благодаря тому, что они «делают», а не тому, что они «читают», «слышат» или «наблюдают». Это исследование также поощряло идею «учиться на собственном опыте», которая позже стала известна как «Экспериментальное обучение» или «Обучение на практике»



Рисунок 4 — Конус обучения Эдгара Дейла

Обучение на основе опыта намного эффективнее, чем другие методы обучения. AR - это технология, которая способствует кинестетическому обучению, предоставляя школам интересный и всеобъемлющий опыт погружения, который можно контролировать в классе. Эти технологии способны изменить образовательный процесс, делая обучение интерактивным и увлекательным.

Еще год назад VR/AR-проекты для детского образования в России были исключением, а уже сегодня тендерные площадки полны заказов на разработку технологичных проектов для школ, Вузов, центров качества образования. В этом же году в российских школах появилось несколько десятков классов с VR/AR-оборудованием. Стоит отметить, что для для реализации технологии AR необходимо менее затратное и простое оборудование чем для VR.

Неким тормозом в распространении AR-технологий, как и VR, пока что является недостаток качественного и полезного контента, подходящего для школьных программ.

На сегодняшний день проекты в дополненной реальности могут быть созданы под индивидуальный заказ (например, для школы с уклоном на определенный предмет) или являться универсальными решениями, которые

могут подойти абсолютно любому учителю, вне зависимости от специфики преподавания. При этом в обязательном порядке технологичные уроки разрабатываются AR-компаниями совместно с педагогами, методистами, преподавателями вузов или учеными. Использование новых технологий на уроках не требует от учителя специальных знаний. Педагогу достаточно уметь скачивать приложение на смартфоны, а также правильно наводить технику на поверхность для воспроизведения контента. Контент, который воспроизводится на интерактивных панелях, при помощи установленного приложения, ученики класса могут видеть на собственных телефонах или планшетах, например, при наведении на учебник. Технология при этом повышает интерес к теме, стимулирует детей к самостоятельному изучению предмета и упрощает понимание сложного материала. При этом есть и проблемы внедрения технологии. По словам преподавателей, не все школы и родители сегодня готовы разрешить детям использовать личную технику на уроках, даже если речь идет о повышении эффективности обучения. Кстати, по данным Business Wire, почти 97% учащихся хотели бы, чтобы часть уроков в их учебных заведениях проводилась с использованием технологии AR. Преподавателей, которые впервые увидели и использовали AR-проекты, попросили рассказать о проблемах внедрения AR посредством опросника. Интересно, что из 77 опрошенных учителей 89% утвердительно ответили на вопрос «могут ли быть полезны AR в образовании?». Из них 92% ответили, что готовы внедрять такие технологии в свои уроки уже сейчас. При этом больше всего опасений вызвал вопрос о влиянии технологий на здоровье — 63% опрошенных, а среди проблем внедрения AR в школы учителя отметили недостаточное оснащение школ.

Примером реализации AR в школьном образовании является приложение Мастерская науки. Продукт: Лаборатория виртуальной и дополненной реальности. Профильные эксперты методически и содержательно разработали, а затем реализовали в программном виде контент в дополненной (для планшетов) и виртуальной (для шлемов) реальностях для следующих учебных курсов:

- «Клеточная биология»,

- «Симметрия молекул и кристаллов»,
- «Стереометрия — сечения многогранников»,
- «Мозг человека».

Все занятия согласованы с основной школьной образовательной программой, планируется расширение программ и углубление контента, в том числе программы для подготовки к ЕГЭ. На рисунке 5 показана реализация данного приложения.



Рисунок 5 — Приложение «Мастерская науки»

Мобильное приложение AR «Занимательная химия» реализует проведение интересных экспериментов из школьного курса химии без специального оборудования и реактивов. В версии «Занимательная химия AR» 1.0 доступны опыты:

- Реакция алюминия с йодом при добавлении воды.
- Создание и расщепление молекулы воды.



Рисунок 6 — Приложение «Занимательная химия»

## 2 АНАЛИЗ И ВЫБОР ПРОГРАММНЫХ СРЕДСТВ

### 2.1 Кроссплатформенный движок Unity

Unity — межплатформенная среда разработки компьютерных игр и приложений, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Выпуск Unity состоялся в 2005 году и с того времени идет постоянное развитие.

Как правило, игровой движок предоставляет множество функциональных возможностей, позволяющих их задействовать в различных играх, в которые входят моделирование физических сред, карты нормалей, динамические тени и многое другое. В отличие от многих игровых движков, у Unity имеется два основных преимущества: наличие визуальной среды разработки и межплатформенная поддержка. Первый фактор включает не только инструментарий визуального моделирования, но и интегрированную среду, цепочку сборки, что направлено на повышение производительности разработчиков, в частности, этапов создания прототипов и тестирования. Под межплатформенной поддержкой предоставляется не только места развертывания (установка на персональном компьютере, на мобильном устройстве, консоли и т. д.), но и наличие инструментария разработки (интегрированная среда может использоваться под Windows и Mac OS).

Третьим преимуществом называется модульная система компонентов Unity, с помощью которой происходит конструирование игровых объектов, когда последние представляют собой комбинированные пакеты функциональных элементов. В отличие от механизмов наследования, объекты в Unity создаются посредством объединения функциональных блоков, а не помещения в узлы дерева наследования. Такой подход облегчает создание прототипов, что актуально при разработке игр.

В качестве недостатков приводятся ограничение визуального редактора при работе с многокомпонентными системами, когда в сложных сценах визуальная работа затрудняется. Вторым недостатком называется отсутствие поддержки Unity ссылок на внешние библиотеки, работу с которыми программистам приходится настраивать самостоятельно, и это также затрудняет командную работу. Еще один недостаток связан с использованием шаблонов экземпляров. С одной стороны, эта концепция Unity предлагает гибкий подход визуального редактирования объектов, но с другой стороны, редактирование таких шаблонов является сложным. Также, WebGL-версия движка, в силу специфики своей архитектуры (трансляция кода из C# в C++ и далее в JavaScript), имеет ряд нерешенных проблем с производительностью, потреблением памяти и работоспособностью на мобильных устройствах.

## **2.2 Анализ SDK для дополненной реальности**

AR SDK — это программный движок, который обеспечивает работу технологий дополненной реальности и AR-приложений. Роль AR SDK заключается в выполнении самой трудной задачи — объединении цифрового контента и реального мира. Именно SDK, в конечном итоге, лежит в основе функций и возможностей AR-приложения, поэтому очень важно выбрать правильную платформу в соответствии с требованиями разрабатываемого проекта.

### **2.2.1 Платформа Apple ARKit**

ARKit можно использовать при создании приложений как для iPhone, так и для iPad, у которых под капотом процессор A9, или выше. Он позволяет разработчикам проектировать и разрабатывать приложения, которые расширяют возможности каждого пользователя с помощью множества функций, таких как:

- Обнаружение и отслеживание 2D-изображений
- Распознавание и размещение 3D-объектов
- Обнаружение горизонтальной и вертикальной плоскостей
- Трекинг лица
- Стабильное и быстрое отслеживание движения

- Расчет масштаба

### 2.2.2 Платформа GoogleARCore

Google ARCore — это один из самых популярных SDK для создания приложений для смартфонов и планшетов. Удивительно, что его AR-приложения поддерживаются не только на Android, но и на устройствах на базе iOS, позволяя разрабатывать кроссплатформенные AR-приложения. ARCore основан на двух элементах: отслеживании позиции и распознавании объектов. Некоторые из его выдающихся особенностей включают в себя:

- Оценка освещенности в реальном времени
- Точное размещение виртуальных объектов
- Легкое отслеживание для создания реалистичных объектов
- Определение размера и местоположения вертикальных, горизонтальных и наклонных поверхностей
- Отслеживание движения в соответствии с положением телефона

### 2.2.3 Библиотека AR Foundation

AR Foundation позволяет вам работать с платформами дополненной реальности на многоплатформенной основе в Unity. Этот пакет предоставляет интерфейс для использования разработчиками Unity, но сам по себе не реализует никаких функций AR. Чтобы использовать AR Foundation на целевом устройстве, вам также потребуется отдельный пакет для этой платформы (например, ARKit XR Plugin на iOS или ARCore XR Plugin на Android).

AR Foundation — это набор MonoBehaviours и API для работы с устройствами, которые поддерживают следующие концепции:

- Отслеживание мира: отслеживание положения и ориентации устройства в физическом пространстве.
- Обнаружение плоскости: обнаружение горизонтальных и вертикальных поверхностей.
- Облака точек, также известные как характерные точки.

- Якорь: произвольная позиция и ориентация, которые отслеживает устройство.
- Оценка освещенности: оценки средней цветовой температуры и яркости в физическом пространстве.
- Проверка среды: средство для создания карты куба для представления определенной области физической среды.
- Отслеживание лица: обнаружение и отслеживание человеческих лиц.
- Отслеживание изображений: обнаружение и отслеживание 2D-изображений.

#### **2.2.4 Библиотека Vuforia**

Vuforia является одним из самых популярных SDK для разработки приложений дополненной реальности, которые позволяют компаниям предоставлять своим клиентам захватывающий опыт AR. Благодаря доступности API через Unity, Vuforia можно использовать для разработки собственных приложений под iOS и Android. Она также считается полным SDK с обширным набором функций для приложений AR.

- Идентификация и отслеживание целевых изображений, текстов на английском языке и 3D-объектов в режиме реального времени
- Размещение виртуальных объектов, таких как 3D-модели, в реальной среде
- Многоцелевые 3D-конфигурации
- Vuforia Engine Area Targets вместе с Area Target Generator
- Отсканированные Model Targets
- Расширенные Model Targets — Обнаружение нескольких моделей
- Продолжение работы при приостановке приложений
- Режим симуляции
- Vuforia Engine Tracking Scale

### 2.2.5 Платформа ARToolKit

ARToolKit — это SDK с открытым исходным кодом, который предлагает множество сложных функций для эффективного и быстрого создания AR-приложений и ПО для смарт-очков. Он предлагает скомпилированные SDK для платформ iOS, Android, Windows, Mac OS и Linux. Это один из первых AR SDK с простым в использовании интерфейсом и быстрой работой. Его многочисленные функции включают в себя:

- Интеграция GPS и компаса
- Поддержка Unity3D и OpenSceneGraph
- Распознавание 2D-объектов
- Одновременное отслеживание
- Поддержка как одной, так и двух камер сразу
- Сопоставление дополнительных элементов через OpenGL
- Интеграция со смарт-очками

### 2.2.6 Платформа 8th Wall

Как и все веб-приложения, дополненная реальность из платформы 8th Wall не требует загрузки приложений, устраняя большой источник трения между разработчиками и потребителями. Доступ к веб-интерфейсу на совместимом мобильном устройстве так же прост, как сканирование QR-кода или переход по адресу в браузере.

- Поддержка SLAM
- Поддержка маркеров ( Image Target )
- Поддержка React, Vue.js, A-Frame, Babylon.js, Three.js
- Распределенная система контроля версий для работы удаленных специалистов
- Разные стадии развертывания, включая промежуточную среду, защищенную паролем

## **2.3 Графические редакторы для 3д-моделирования**

### **2.3.1 Редактор 3Ds Max**

3Ds Max – «пионер» среди 3D редакторов, очень популярный инструмент, №1 в выборе многих начинающих и продвинутых специалистов. Занимает ведущие позиции в сфере дизайна и архитектурной визуализации. Часто используется в игровой индустрии. интеграция композитинга. Возможности:

- моделирование на основе полигонов, сплайнов и NURBS,
- мощная система частиц,
- модуль волосы/шерсть,
- расширенные шейдеры Shader FX,
- поддержка новых и усовершенствованных механизмов Iray и mental ray.
- анимация толпы,
- импорт из Revit и SketchUp,
- интеграция композитинга.

И многое другое.

Плюсы: огромный функционал, множество плагинов и обучающей информации.

Минусы: не так прост в освоении, «старожилу» требуются серьезные обновления.

### **2.3.2 Редактор Blender**

Единственный в списке бесплатный 3D пакет, который практически не уступает по функционалу платным приложениям. Blender включает в себя средства для 3D моделирования, анимации, а также набор опций для создания игр, визуальных эффектов и скульптинга. Отличная альтернатива «монстрам» 3D анимации. Благодаря поддержке Blender Foundation, программа очень быстро и стабильно развивается. Возможности:

- полигональное моделирование, сплайны, NURBS-кривые и поверхности;
- режим лепки;
- система частиц;

- динамика твердых и мягких тел: жидкость, шерсть/волосы и т.д.;
- скелетная анимация;
- встроенные механизмы рендеринга и интеграция со сторонними визуализаторами;
- редактор видео;
- функции создания игр и приложений (Game Blender).

Плюсы: доступность, открытый код, кроссплатформенность, небольшой размер (около 50 мегабайт), широкий функционал, возможность создания игр.

Минусы: отсутствие документации в базовой поставке.

### **2.3.3 Редактор Modo**

Modo – полноценный продукт для моделирования, рисования, анимации и визуализации. Включает также инструменты скульптинга и текстурного окрашивания. Благодаря удобству пользования и высокой производительности, у Modo репутация одного из самых быстрых инструментов моделирования. Modo популярен в сфере рекламы, разработки игр, спецэффектов и архитектурной визуализации.

Возможности:

- полигональное и моделирование SDS;
- современные инструменты анимации;
- динамика твердых и мягких тел;
- система рисования;
- материал Fur (мех) для создания волос, травы и меха;
- инструменты лепки;
- быстрая и качественная визуализация.

Плюсы: мощный и понятный инструментарий, высокая производительность.

Минусы: мало информации.

## **2.4 Языки программирования в разработке AR приложений**

### **2.4.1 Язык программирования C#**

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

### **2.4.2 Язык программирования C++**

C++ — компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком C, — наибольшее внимание

уделено поддержке объектно-ориентированного и обобщенного программирования.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder и другие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на Java и C#.

Синтаксис C++ унаследован от языка C. Одним из принципов разработки было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C.

### **2.4.3 Язык программирования JavaScript**

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией стандарта ECMAScript (стандарт ECMA-262).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком корпорации Oracle в США.

## **2.5 Текстовые редакторы для написания кода**

### **2.5.1 Редактор Visual Studio Code**

Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

Visual Studio Code основан на Electron и реализуется через веб-редактор Monaco, разработанный для Visual Studio Online.

### **2.5.1 Редактор sublime text**

Sublime Text — проприетарный текстовый редактор. Поддерживает плагины на языке программирования Python.

Разработчик позволяет бесплатно и без ограничений ознакомиться с продуктом, однако программа уведомляет о необходимости приобретения лицензии. Некоторые возможности:

- Быстрая навигация (Goto Anything)
- Командная палитра (Command Palette)
- API плагинов на Python
- Одновременное редактирование (Split Editing)
- Высокая степень настраиваемости (Customize Anything)

Sublime Text поддерживает большое количество языков программирования и имеет возможность подсветки синтаксиса для C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL и XML.

В дополнение к тем языкам программирования, которые включены по умолчанию, пользователи имеют возможность загружать плагины для поддержки других языков.

Менеджер пакетов Sublime Text может быть оснащен менеджером пакетов, который позволяет пользователю находить, устанавливать, обновлять и удалять пакеты без перезагрузки программы. Менеджер поддерживает установленные пакеты в актуальном состоянии, загружая новые версии из репозитория. Кроме того, он предоставляет команды для активации и деактивации установленных пакетов.

## 2.6 Метод SLAM

В основе работы библиотек дополненной реальности для распознавания целей лежит технология SLAM — simultaneous localization and mapping. SLAM карта является графиком 3D точек, которые представляют собой разреженное облако, где каждая отметка соответствует координатам оптического объекта (например, угол таблицы).

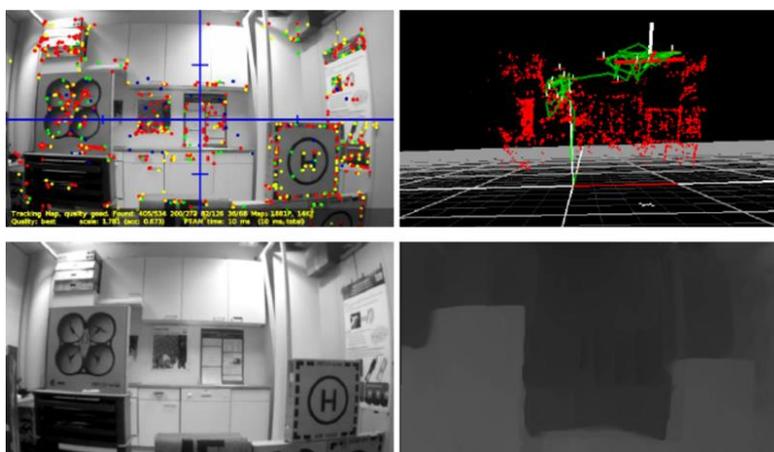


Рисунок 7 — SLAM карта

Также как и с измерениями на основе ускорения, SLAM опирается на карту точек, которые могут быть более или менее надежными.

Основная задача SLAM — построение и обновление карты неизвестной среды при одновременном отслеживании местоположения агента внутри нее.

## 2.7 Обоснование выбора программных средств

Проведя анализ программных средств и технологий для разрабатываемого приложения, было принято решение использовать следующие инструменты:

- Кроссплатформенный движок Unity дает возможность разрабатывать приложения под различные технические устройства, имеет огромное множество встроенных модулей, необходимых для реализации сложных визуальных сцен;
- Библиотека дополненной реальности AR Foundation реализовывает функции ARCore и ARKit, имеет интеграцию в проект Unity;
- Язык программирования C# традиционно применяется в разработке проектов в Unity, а также реализует работу библиотеки AR Foundation;
- Редактор 3д-моделирования Blender предоставляет все необходимые функции для разработки 3д-моделей в данной работе;
- Редактор кода Visual Studio Code — является удобной средой для написания программного кода на языке C#.

## 3 РАЗРАБОТКА И ЭКСПЛУАТАЦИЯ ПРОГРАММНОГО ПРОДУКТА

### 3.1 Постановка задачи и определение требований к программному продукту

Необходимо разработать программный продукт — мобильное приложение, работающее на платформе android 7+, с технологией дополненной реальности для изучения школьного курса физики. Данное приложение должно содержать следующие AR-сцены:

- модель солнечной системы — содержит анимированные модели планет и солнца, а также представление осей по которым осуществляется движение планет. Пользователь должен иметь возможность остановить и возобновить анимацию с помощью кнопок “пауза” и “начать”. Каждая планета и солнце имеет информационный блок с описанием объекта (радиус, скорость, масса и так далее). Для активации информационного блока пользователь должен нажать на 3д-модель объекта, при этом анимация сцены должна быть остановлена для удобного просмотра и восприятия информации. При остановки анимации также необходимо скрывать линии осей, для того чтобы не было наложения между ними и текстом. На рисунке 8 представлено изображение солнечной системы, оно использовано в качестве образца моделируемой системы ;

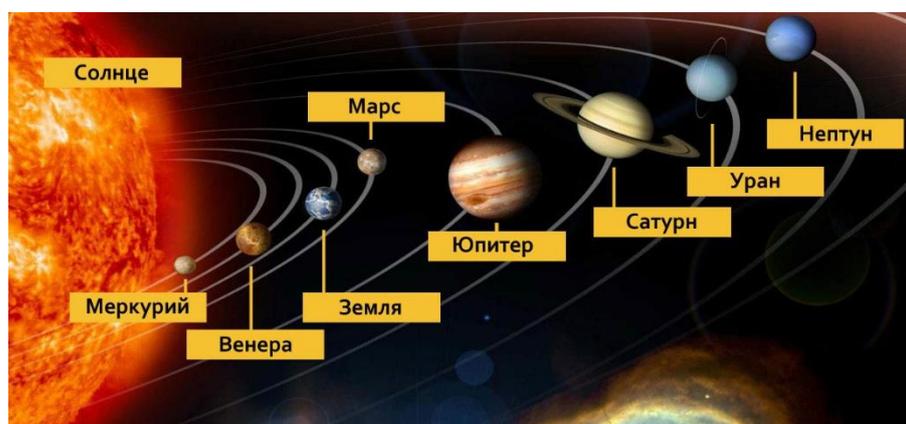


Рисунок 8 — Изображение солнечной системы

- модель машины с изображением векторов действующих на автомобиль когда он в покое и движении. У пользователя должна быть возможность управлять 3д-моделью машины с помощью элемента пользовательского интерфейса “джойстика”. Когда управления объектом нет, на машину действуют силы тяжести и реакции опоры, при движении объекта, в зависимости от выходных данных “джойстика” указывается вектор движения модели автомобиля, а также показывается вектор силы трения относительно колес машины. На рисунке 9 показано, как должны быть направлены векторы действующий сил относительно колес во время движения автомобиля;

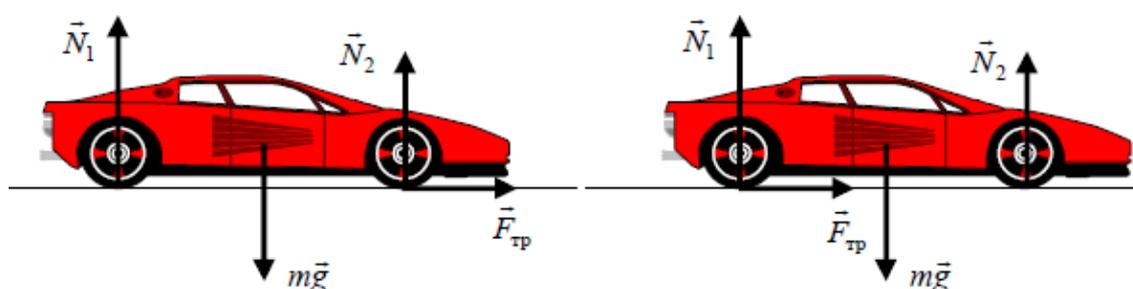


Рисунок 9 — Векторы действующих сил на автомобиль во время движения

- модель правила левой руки — с помощью специального изображения-маркера, который крепится к ладони пользователя, проецируются векторы сил, согласно правилу левой руки: если расположить ладонь левой руки так, чтобы линии индукции магнитного поля входили во внутреннюю сторону ладони, перпендикулярно к ней, а четыре пальца направлены по току, то отставленный на 90 градусов большой палец укажет направление силы, действующей со стороны магнитного поля на проводник с током. На рисунке 10 показано, как располагаются данные векторы.

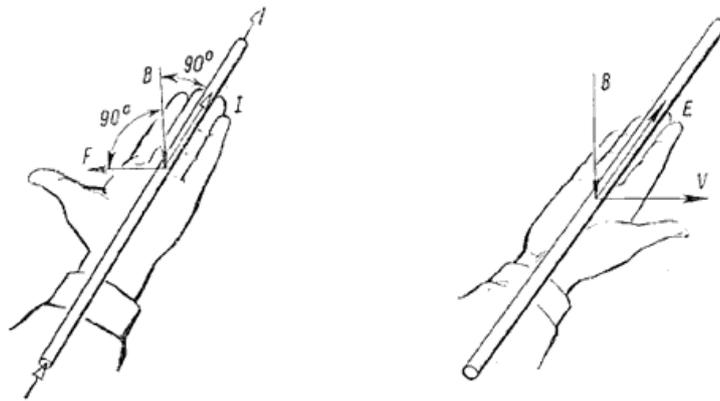


Рисунок 10 — Модель правила левой руки

Кроме сцен с дополненной реальность, приложение содержать экраны с главным меню выбора раздела, а также экраны с теоретическим материалом к выбранному разделу.

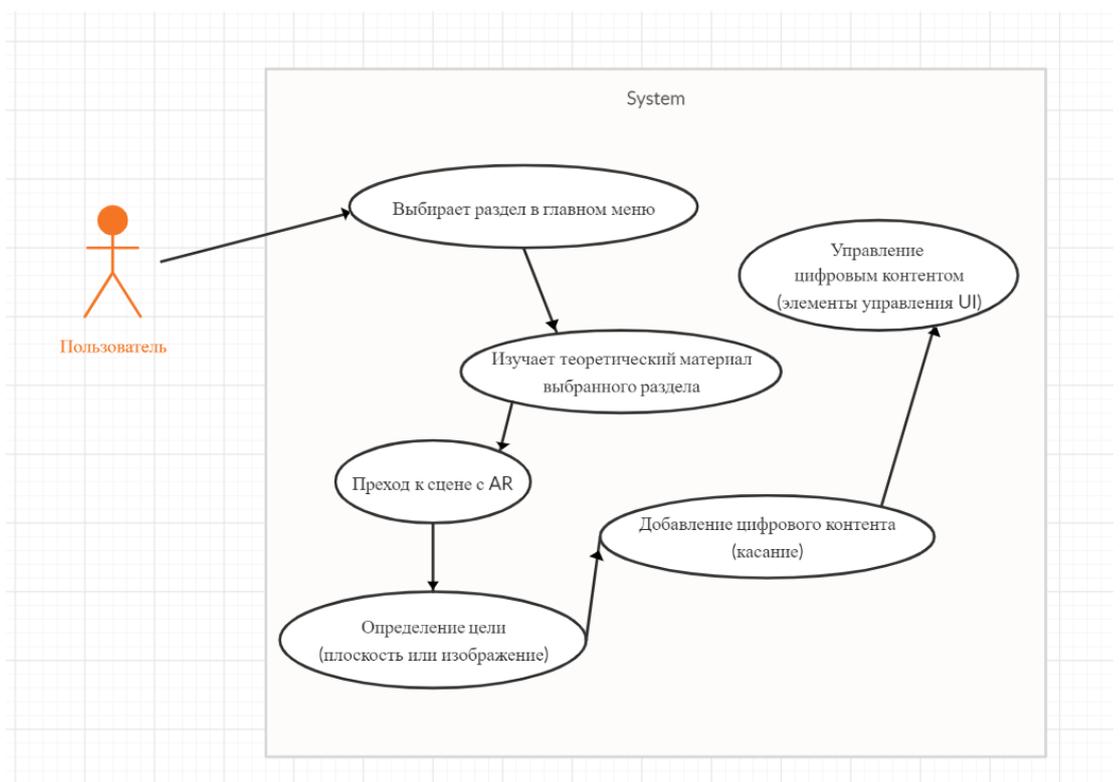


Рисунок 11 — Диаграмма использования приложения

Интерфейс приложения должен быть интуитивно понятен и доступен для пользователя, давать возможность перехода между сценами с дополненной реальностью и информационными разделами. Детальное взаимодействие

пользователя с приложением описано в диаграмме вариантов использования, рисунок 11.

### 3.2 Создание проекта и внедрение AR

Для создания проекта в Unity используется инструмент Unity Hub, он позволяет осуществлять управление и отслеживание версий Unity, а также добавлять программные модули, отвечающие за сборку проекта под различные платформы на устройствах. Данное приложение разрабатывается под мобильную платформу Android, поэтому для его разработки и сборки необходимо добавить в Unity Hub платформу Android Build Support, в которую будут включены модули Android SDK & NDK Tools и Open JDK, рисунок 12.

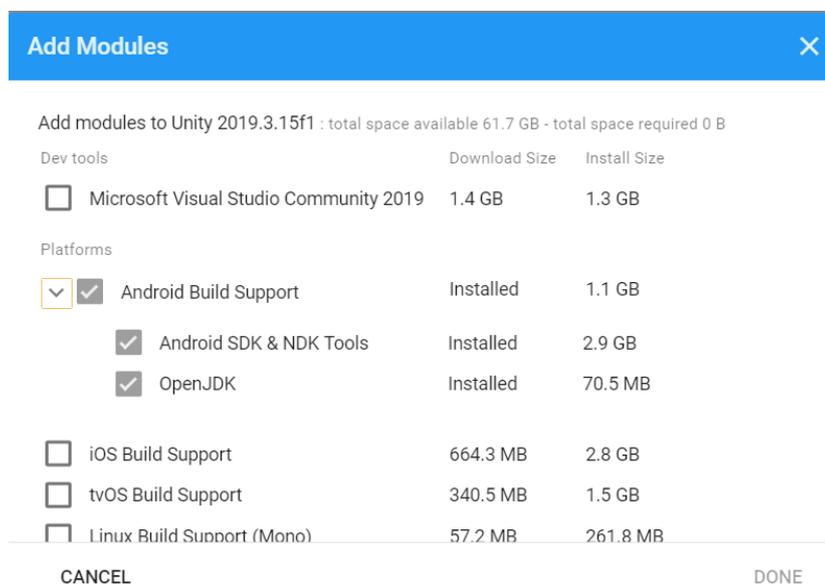


Рисунок 12 — Добавление Android Build Support

На момент разработки данного приложения использовалась версия Unity 2019.3.15f1.

Для создания проекта в окне запуска необходимо нажать на кнопку “NEW” (создать новый проект), рисунок 13.

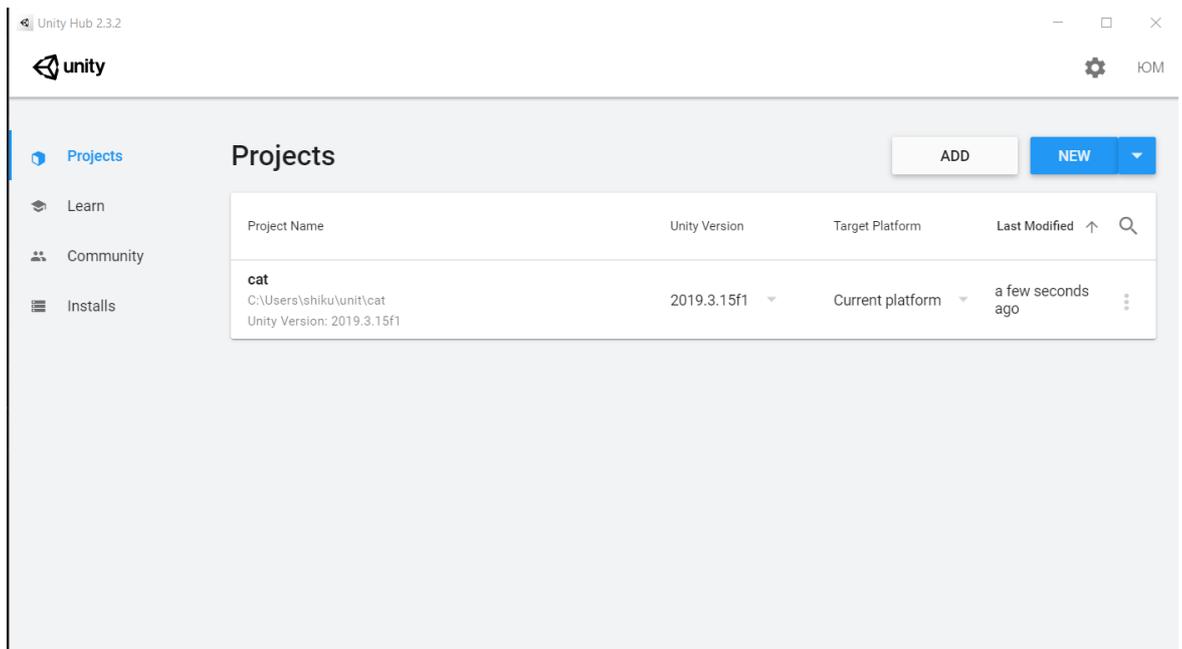


Рисунок 13 — Окно создания проекта в Unity

В окне создания проекта необходимо выбрать шаблон сцен, в данном случае нужен шаблон “3д”. Далее указывается имя проекта и его путь размещения, рисунок 14.

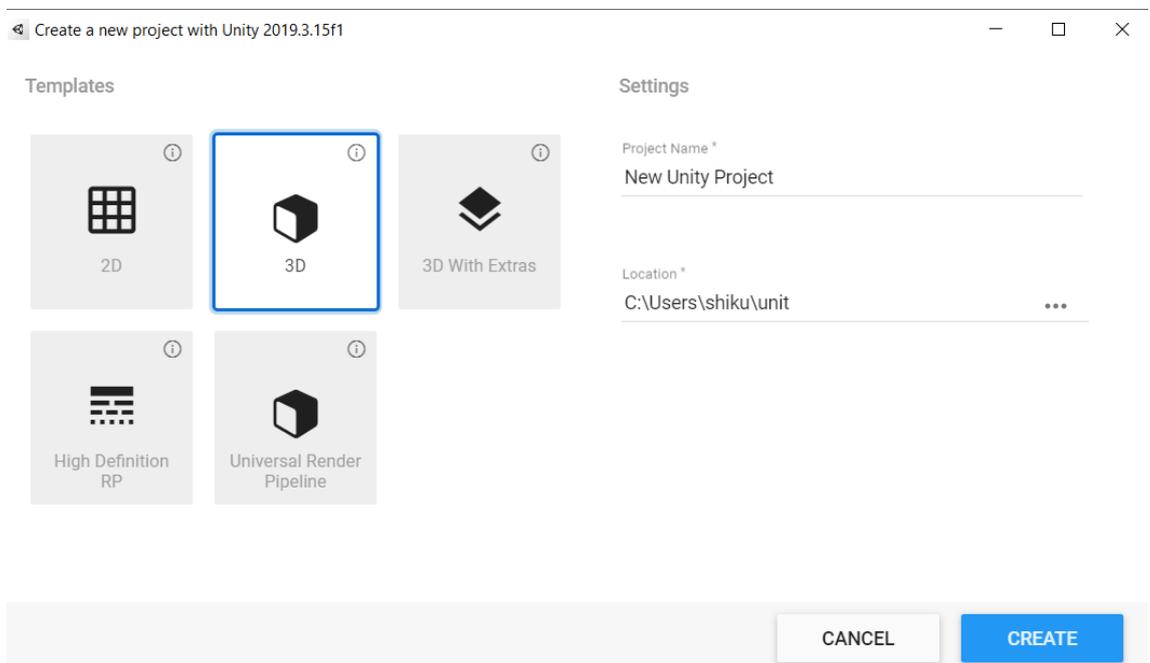


Рисунок 14 — Окно создания проекта в Unity, настройки

После создания проекта для работы с дополненной реальностью, с помощью инструмента Package Manager, необходимо импортировать библиотеку ARFoundation и необходимые для ее работы компоненты ARKit XR Plugin и ARCore XR Plugin, рисунок 15. На момент написания данной работы, в проекте использовалась библиотека ARFoundation и компоненты ARKit XR Plugin и ARCore XR Plugin последних версий 3.1.3. Данные версии библиотеки дополненной реальности и движка Unity имеют полноценный набор функций для реализации поставленных системе задач. Использование более низких версий библиотек может вызвать нежелательные конфликты при эксплуатации программного продукта.

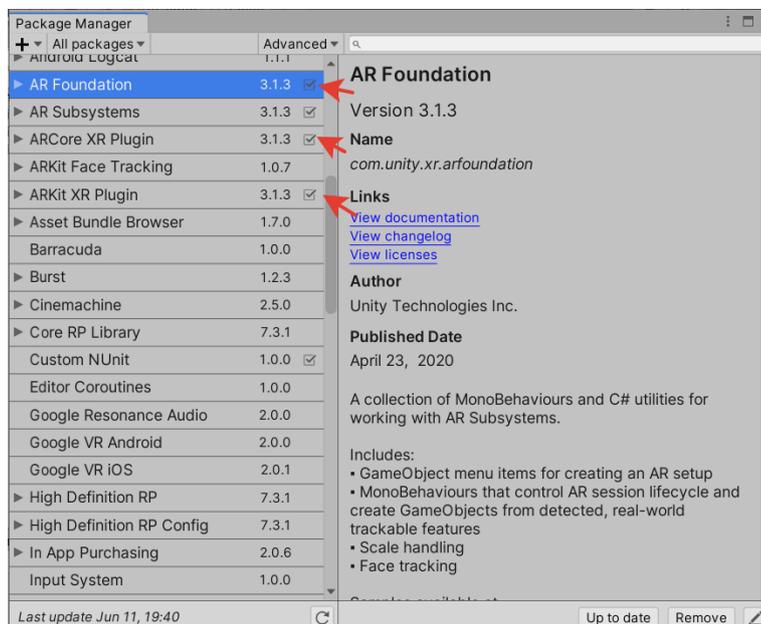


Рисунок 15 — Импорт инструментов для дополненной реальности

### 3.3 Виртуальный контент, разработка 3д-моделей

Основная идея приложения с дополненной реальностью — дополнить “реальный” мир цифровым контентом. В данном приложении в качестве добавляемого цифрового контента выступают 3д-модели. Поэтому от их разработки напрямую зависит дальнейшее написание всего программного продукта. Для моделирования объектов использовалось программное обеспечение Blender.

### 3.3.1 Модель солнечной системы

Для последующей разработки анимации и взаимодействия системы с пользователем, было принято решение о разработке моделей планет и солнца отдельно, затем соединить модели в единую систему на сцене в Unity. В качестве каркаса для моделей планет и солнца служит примитивный объект сфера. Кольца Сатурна были сделаны путем наложения и обрезания boolean groups двух плоскостей окружности. Для масштабирования моделей, а также задания значений скорости анимации вращения и расположения объектов в системе было использовано логарифмическое отношение оригинальных и моделируемых значений. Благодаря чему модели выглядят пропорционально и наглядно относительно друг друга, несмотря на очень большую разницу своих значений характеристик. Затем на каркасные примитивы были наложены текстуры соответствующих объектов. Получившиеся модели планет представлены на рисунке 16.

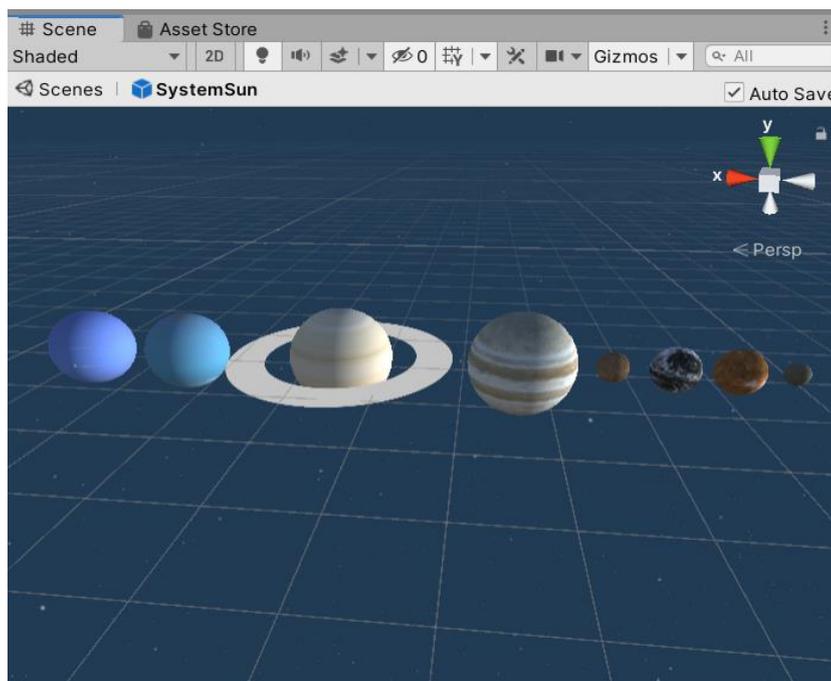


Рисунок 16 — 3д-модели планет

Отдельным процессом была разработка модели солнца в редакторе Unity. Для имитации короны (последней внешней оболочки Солнца) и эффекта “горения” энергетических извержений использовался компонент реализации системы частиц Particle System, рисунок 17. С его помощью была разработана

непрерывная анимация огненных извержений путем добавления к объекту солнца плоских искаженных 2д “спрайтов” частиц.

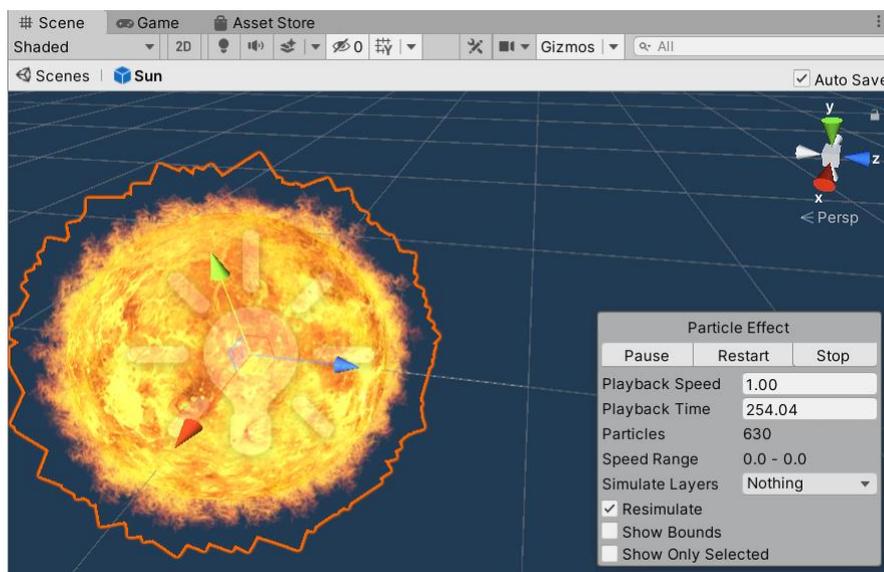


Рисунок 17 — 3д-модель солнца

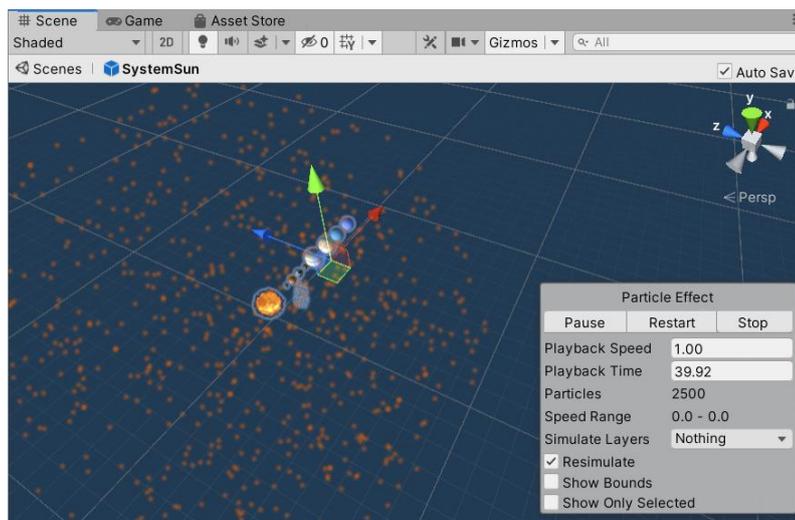


Рисунок 18 — Система движущихся частиц в пространстве солнечной системы

Данный компонент был также использован для имитации движущихся частиц в пространстве системы, рисунок 18.

Для моделирования линий осей планет был использован компонент “Line Renderer”. Его характеристики приведены на рисунке 19, в следующем разделе

приведен листинг реализации данного компонента в процессе инициализации сцены.

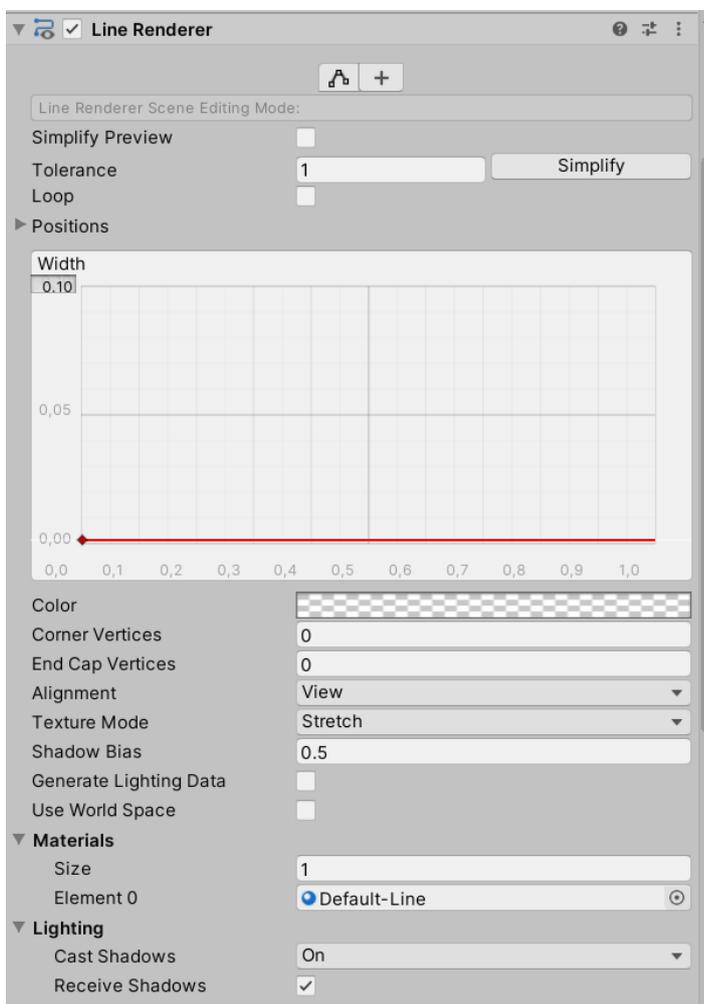


Рисунок 19 — Настройки компонента “Line Renderer” для моделирования линий осей движения моделей планет

Для моделирования информационных текстовых “подсказок” к объектам солнечной системы был использован компонент “TextMesh Pro”. Данный компонент позволяет заменить UI.Text на объект текста в 3д-пространстве, благодаря чему возможно отобразить текст относительно заданной 3д-модели, рисунок 20.

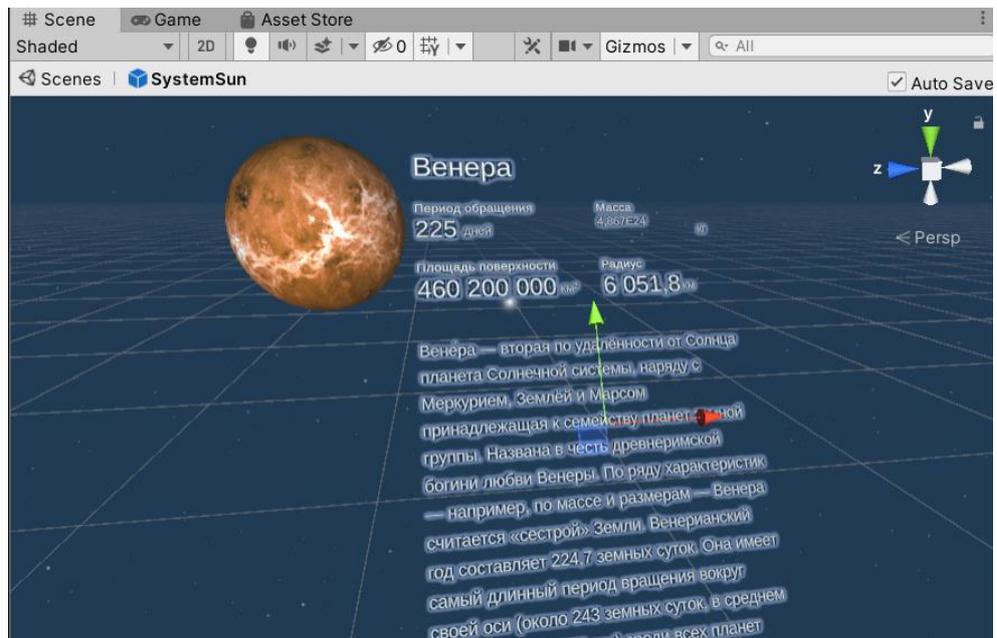


Рисунок 20 — Моделирование информационных блоков

### 3.3.2 Модель автомобиля

В качестве объекта сцены была выбрана 3д-модель автомобиля BMW M6, рисунок 22. Чертеж данной модели в различных ракурсах представлен на рисунке 21. Для составных деталей кузова 3д-модели, металлических объектов, стекол и других поверхностей были созданы и наложены соответствующие текстуры, рисунок 23.

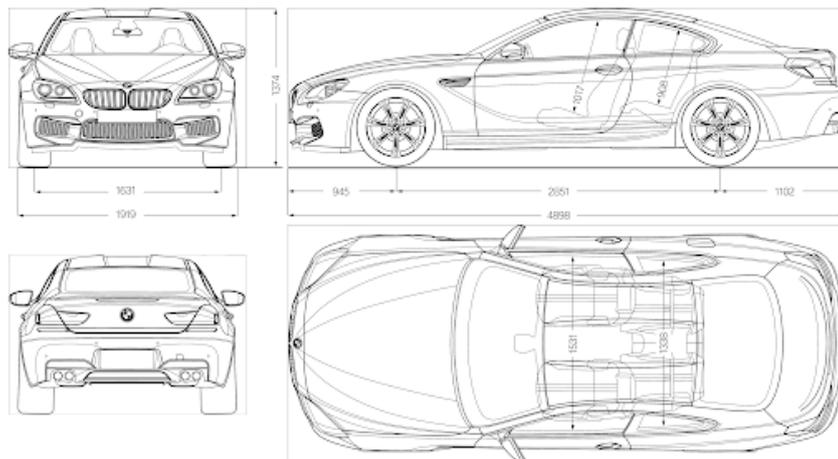


Рисунок 21 — Чертеж модели автомобиля BMW M6

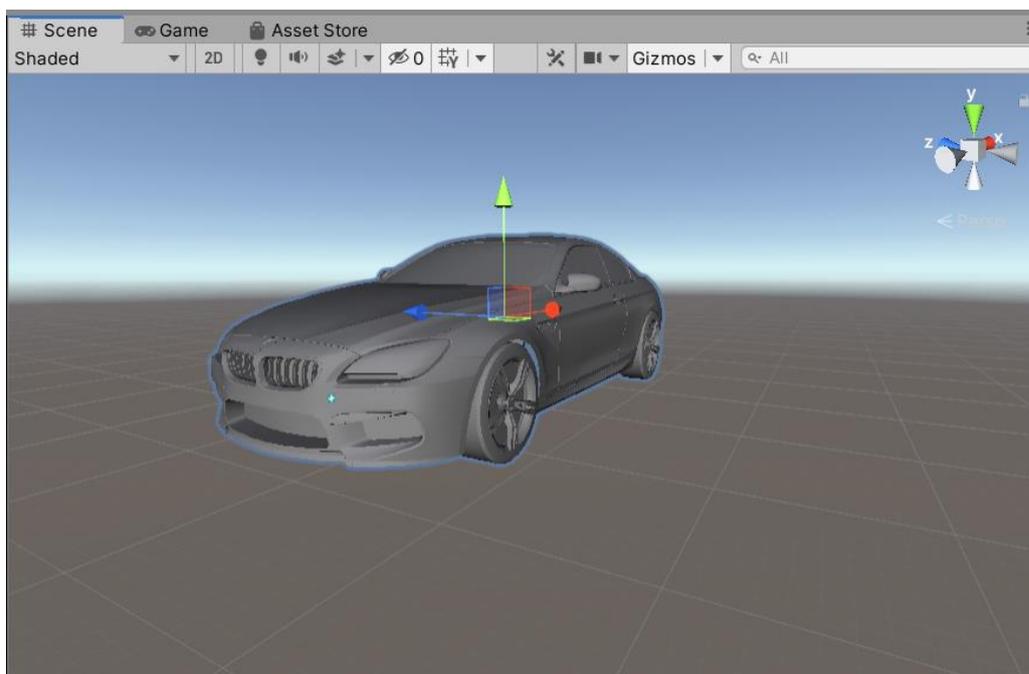


Рисунок 22 — 3д-модель автомобиля BMW M6

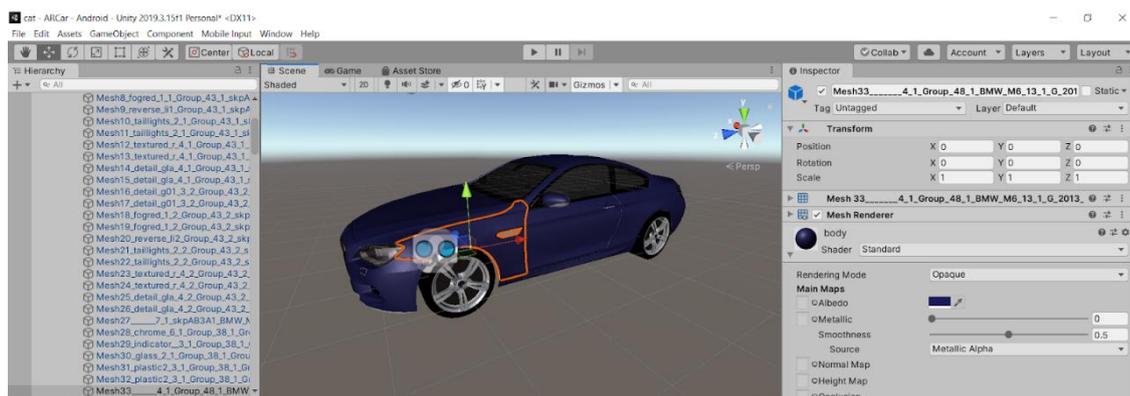


Рисунок 23 — Наложение текстур на 3д-модель автомобиля

Для реализации векторов движения и силы трения были использованы разработанные 2д-спрайты. Стрелки векторов силы реакции опоры и силы тяжести представлены в виде 3д-фигуры состоящей из объектов цилиндра и конуса. Для подписи наименования векторов также использовался компонент “TextMesh Pro”. Полученная модель изображена на рисунке 24.

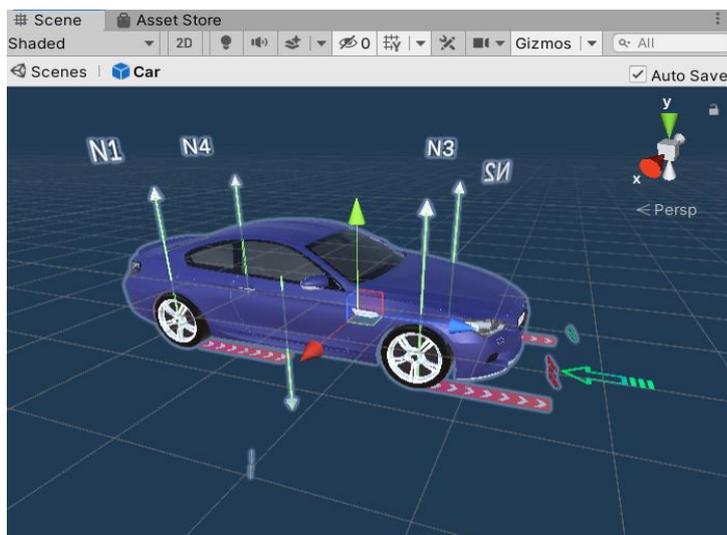


Рисунок 24 — Модель автомобиля и действующих на него физических сил

### 3.3.3 Модель реализации правила правой руки

Для моделирования векторов направления тока и силы, действующей со стороны магнитного поля на проводник использованы 2д-спрайты, а линия индукции представлена 3д-объектом, состоящим из моделей цилиндра и конуса. Название векторов реализовано с помощью компонента “TextMesh Pro”. Полученная модель показана на рисунке 25.

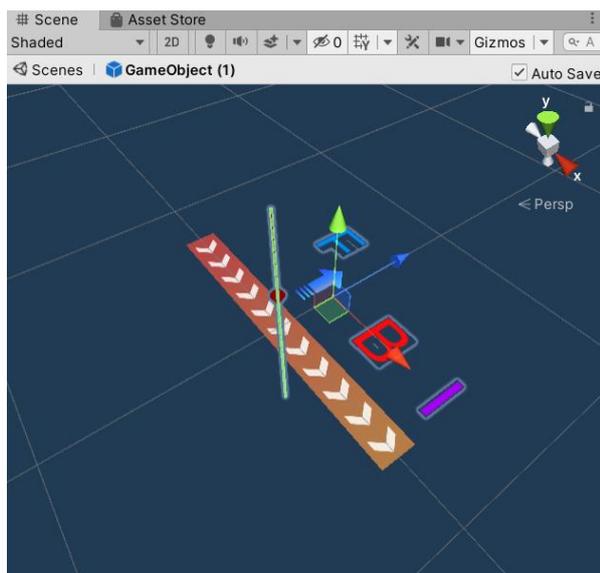


Рисунок 25 — Модель правила левой руки

### 3.4 Реализация компонентов, написание программного кода

Для реализации логики программы и поведения объектов в Unity используется компонентный подход. Данный подход реализуется путем написания неких скриптов, в которых описываются все свойства и методы объекта, его поведение и роль в системе. Далее такой скрипт добавляется к объекту в виде компонента. Каждый скрипт реализуется путем наследования встроенного класса “MonoBehaviour”, который связывает компонент с внутренними механизмами в Unity.

Каждый такой компонент имеет жизненный цикл, описанный в классе “MonoBehaviour”.

Основные этапы жизненного цикла компонентов:

1. Первая загрузка сцены
  - Эти функции вызываются, когда сцена стартует.
    - Awake: Эта функция всегда вызывается до начала любых функций, а также сразу после инициализации префаба.
    - OnEnable: (вызывается, если объект является активным): Эта функция вызывается только после того, как объект будет включен.
2. До первого обновления кадров
  - Start: вызывается перед прорисовкой первого фрейма, только если сценарий определен.
3. В промежутке между кадрами
  - OnApplicationPause: Это событие вызывается в конце кадра, когда обнаружена пауза, фактически между обычными обновлениями кадров. После OnApplicationPause прорисовывается один дополнительный кадр для того, чтобы показать окно, которое отображается во время паузы.
4. Порядок обновления

Для отслеживания логики программы, взаимодействия и анимации объектов, положения камеры и т.д., есть несколько различных событий. Общий

механизм для выполнения большинства задач находится в функции Update(), но есть и другие функции.

- FixedUpdate: FixedUpdate() не зависит от Update(), и может вызываться как чаще него так и реже.
- Update: Update() вызывается один раз за кадр. Это основное событие для прорисовки кадра.
- LateUpdate: LateUpdate() вызывается один раз в кадре, после завершения Update().

#### 5. Разрушение объектов

- OnDestroy: Эта функция вызывается для последнего кадра существования объекта).

#### 6. При выходе

Эти функции вызываются для всех активных объектов в сцене:

- OnApplicationQuit: Эта функция вызывается для всех игровых объектов перед закрытием приложения. В редакторе это происходит, когда пользователь прекращает PlayMode.
- OnDisable: Эта функция вызывается, когда объект отключается или становится неактивным.

UML диаграмма разработанных классов приведена на рисунке 26. Детальное описание приведенных классов рассмотрено в следующих разделах.

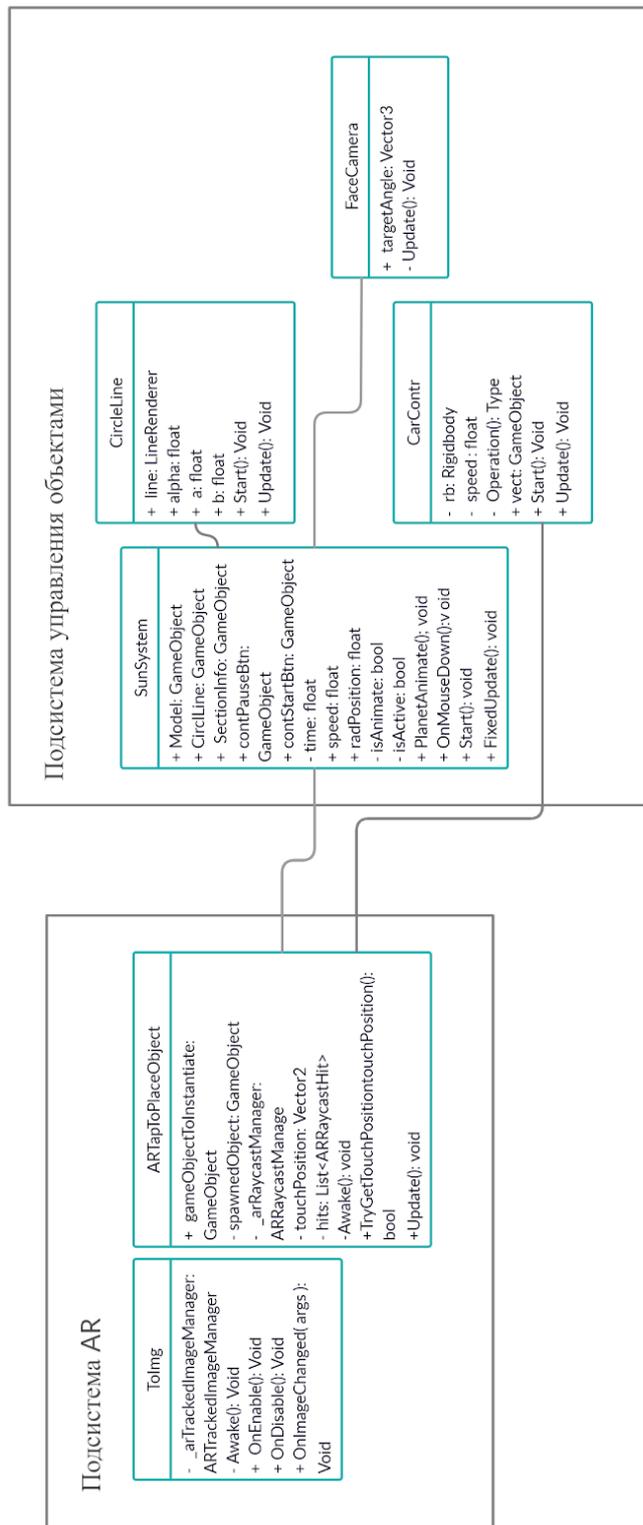


Рисунок 26 — UML диаграмма классов

### 3.4.1 Отслеживание целей, программный код добавления виртуального контента

Связанные с AR подсистемы AR Foundation определены в пакете ARSubsystems и используют пространство имен UnityEngine.XR.ARSubsystems.

Базовая иерархия AR-сцен выглядит следующим образом, рисунок 27:

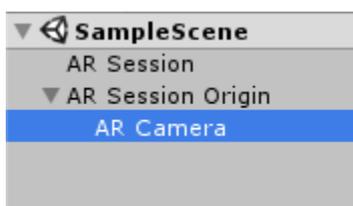


Рисунок 27 — Базовая иерархия AR-сцен

Каждая AR сцена должна включать объект ARSession. Сеанс AR управляет жизненным циклом взаимодействия с AR, включая или отключая AR на целевой платформе. ARSession Может быть на любом GameObject.

Цель ARSessionOrigin состоит в том, чтобы преобразовать отслеживаемые объекты (такие как плоские поверхности и характерные точки) в их конечное положение, ориентацию и масштаб в сцене Unity. Поскольку устройства AR предоставляют свои данные в «пространстве сеанса», немасштабированном пространстве относительно начала сеанса AR, ARSessionOrigin выполняет соответствующее преобразование в пространство Unity. ARSessionOrigin также позволяет масштабировать виртуальный контент и применить смещение к камере.

В AR Foundation под «отслеживаемым» понимается все, что можно обнаружить и отследить в реальном мире. Плоскости, облака точек, контрольные точки, зонды окружающей среды, лица, изображения и трехмерные объекты — все это примеры отслеживаемых объектов. В данной работе будет реализовано два типа отслеживания: горизонтальной плоскости и 2д-изображения. За реализацию данных функций отвечают отслеживаемые менеджеры ARPlaneManager и ARTrackedImageManager соответственно.

Для добавления виртуального контента к отслеживаемой цели необходимо написать компонент, который будет добавлен к объекту ARSessionOrigin.

Таблица 1 — Поля класса “ARTapToPlaceObject”

| Название                | Тип                | Модификатор доступа | Описание   |
|-------------------------|--------------------|---------------------|--|
| gameObjectToInstantiate | GameObject         | public              | Модель объекта, которая будет добавлена на сцену |
| spawnedObject           | GameObject         | private             | Добавленный объект на сцену                      |
| _arRaycastManager       | ARRaycastManager   | private             | Компонент библиотеки ARRaycastManager            |
| touchPosition           | Vector2            | private             | Вектор для хранения координат добавления объекта |
| hits                    | List<ARRaycastHit> | static              | Массив точек в пространстве                      |

Таблица 2 — Методы класса “ARTapToPlaceObject”

| Название            | Возвращаемый тип | Аргументы     | Модификатор доступа | Описание                  |
|---------------------|------------------|---------------|---------------------|---------------------------|
| Awake               | void             | -             | private             | Инициализация сцены       |
| TryGetTouchPosition | bool             | touchPosition | public              | Проверка на touchPosition |
| Update              | void             | -             | public              | Выполняется каждый фрейм  |

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;
[RequireComponent(typeof(ARRaycastManager))]

public class ARTapToPlaceObject : MonoBehaviour
{
    public GameObject gameObjectToInstantiate;
    private GameObject spawnedObject;
    private ARRaycastManager _arRaycastManager;
    private Vector2 touchPosition;
    static List<ARRaycastHit> hits = new List<ARRaycastHit>();

    private void Awake()
    {
        _arRaycastManager = GetComponent<ARRaycastManager>();
    }

    bool TryGetTouchPosition(out Vector2 touchPosition)
    {
        if(Input.touchCount > 0)
        {
            touchPosition = Input.GetTouch(0).position;
            return true;
        }

        touchPosition = default;
        return false;
    }

    void Update()
    {
        if(!TryGetTouchPosition(out Vector2 touchPosition))
            return;

        if(_arRaycastManager.Raycast(touchPosition, hits, TrackableType.PlaneWithinPolygon))
        {
            var hitPose = hits[0].pose;

            if(spawnedObject == null)
            {
                spawnedObject = Instantiate(gameObjectToInstantiate, hitPose.position, hitPose.rotation);
            }
        }
    }
}

```

Рисунок 28 — Листинг класса ARTapToPlaceObject

Листинг написанного класса ARTapToPlaceObject представлен на рисунке 28, его поля и методы обозначены в таблицах 1 и 2 соответственно. Данный класс наследует свойства базового класса MonoBehaviour. После инициализации ARRaycastManager, с помощью метода TryGetTouchPosition, каждый фрейм

производится проверка на событие касания пользователем экрана. Как только это произойдет, в вектор `touchPosition` будут записаны координаты касания, которые потом будут преобразованы в набор точек `hitPose`, содержащий значения позиции и ротации виртуального объекта.

Таблица 3 — Поля класса “ToImg”

| Название                            | Тип                                | Модификатор доступа  | Описание                                     |
|-------------------------------------|------------------------------------|----------------------|--|
| <code>_arTrackedImageManager</code> | <code>ARTrackedImageManager</code> | <code>private</code> | Компонент <code>ARTrackedImageManager</code> |

Таблица 4 — Методы класса “ToImg”

| Название                    | Возвращаемый тип  | Аргументы         | Модификатор доступа  | Описание                |
|-----------------------------|-------------------|-------------------|----------------------|-------------------------|
| <code>Awake</code>          | <code>void</code> | -                 | <code>private</code> | Инициализация сцены     |
| <code>OnEnable</code>       | <code>void</code> | -                 | <code>public</code>  | Обнаружение изображения |
| <code>OnDisable</code>      | <code>void</code> | -                 | <code>public</code>  | Скрытие изображения     |
| <code>OnImageChanged</code> | <code>void</code> | <code>args</code> | <code>public</code>  | Добавление контента     |

Листинг класса `ToImg` представлен на рисунке 29, его поля и методы обозначены в таблицах 3 и 4 соответственно. Данный класс также наследует свойства `MonoBehaviour` и привязывает виртуальный объект к 2д-изображению, если оно находится в области видимости камеры, с помощью отслеживания опорных точек данной цели.

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

```

using UnityEngine.XR;
using UnityEngine.XR.ARFoundation;

public class ToImg : MonoBehaviour
{
    private ARTrackedImageManager _arTrackedImageManager;

    private void Awake()
    {
        _arTrackedImageManager = FindObjectOfType<ARTrackedImageManager>();
    }

    public void OnEnable()
    {
        _arTrackedImageManager.trackedImagesChanged += OnImageChanged;
    }

    public void OnDisable()
    {
        _arTrackedImageManager.trackedImagesChanged -= OnImageChanged;
    }

    public void OnImageChanged(ARTrackedImagesChangedEventArgs args)
    {
        foreach (var trackedImage in args.added)
        {
            Debug.Log(trackedImage.name);
        }
    }
}

```

Рисунок 29 — Листинг класса ToImg

### 3.4.2 Программный код управления объектами

Исходя из поставленных требований к разрабатываемому приложению модель солнечной системы должна наглядно показывать не только как выглядят объекты в реальности, но и их поведение, например движение планет. В сцене из раздела “механика, сила трения” необходимо наглядно изобразить, как и какие силы действуют на автомобиль когда он находится в покое или совершает движение. Для решения поставленных задач, в системе необходимо реализовать управление объектами. Так в классе SunSystem реализовано поведение объектов в модели солнечной системы, а также взаимодействие данной сцены с пользователем, его листинг изображен на рисунке 30, поля и методы описаны в таблицах 5 и 6 соответственно.

Таблица 5 — Поля класса “SunSystem ”

| Название     | Тип        | Модификатор доступа | Описание                   |
|--------------|------------|---------------------|----------------------------|
| Model        | GameObject | public              | Модель объекта             |
| CirclLine    | GameObject | public              | Линии осей                 |
| SectionInfo  | GameObject | public              | Объект текста              |
| contPauseBtn | GameObject | public              | Кнопка “Пауза”             |
| contStartBtn | GameObject | public              | Кнопка “Старт”             |
| time         | float      | private             | Время                      |
| speed        | float      | public              | Скорость                   |
| radPosition  | float      | public              | Положение                  |
| isAnimate    | bool       | private             | Анимация есть или нет      |
| isActive     | bool       | private             | Когда сцена стоит на паузе |

Таблица 6 — Поля класса “SunSystem”

| Название      | Возвращаемый тип | Аргументы | Модификатор доступа | Описание                 |
|---------------|------------------|-----------|---------------------|--------------------------|
| PlanetAnimate | void             | -         | public              | Анимация                 |
| OnMouseDown   | void             | -         | public              | Нажатие на объект        |
| Start         | void             | -         | public              | Начало выполнения        |
| FixedUpdate   | void             | -         | public              | Выполняется каждый фрейм |

using System.Collections;

using System.Collections.Generic;

```

using UnityEngine;
using UnityEngine.UI;

public class SunSystem: MonoBehaviour
{
    public GameObject Model;
    public GameObject CirclLine;
    public GameObject SectionInfo;
    public GameObject contPauseBtn;
    public GameObject contStartBtn;
    private float time = 0f;
    public float speed;
    public float radPosition;
    private bool isAnimate;
    private bool isActive;

    void PlanetAnimate()
    {
        time += 0.5f*speed*Time.deltaTime;
        var positionX = radPosition*Mathf.Cos(time);
        var positionZ = radPosition*Mathf.Sin(time);
        var y = time;
        PlanetModel.transform.position = new
Vector3(positionX, 1, positionZ);
    }

    void OnMouseDown()
    {
        if(!isAnimate)
        {
            if(isActive)
            {
                if(SectionInfo) SectionInfo.SetActive(true);
                isActive = true;
            }
            else
            {
                if(SectionInfo) SectionInfo.SetActive(false);
                isActive = false;
            }
        }
    }

    void Start()
    {
        isAnimate = true;
        isActive = false;

        pauseBtn.onClick.AddListener(delegate {
            isAnimate = false;
            contPauseBtn.SetActive(false);
            contStartBtn.SetActive(true);
            CirclLine.SetActive(false);
        });

        startBtn.onClick.AddListener(delegate {
            isAnimate = true;
            contStartBtn.SetActive(false);
            contPauseBtn.SetActive(true);
            CirclLine.SetActive(true);
        });
    }

    void FixedUpdate()
    {
        if (isAnimate)
        {
            PlanetAnimate();
        }
    }
}

```

Рисунок 30 — Листинг класса SunSystem

Для большей четкости в модели линии осей движения объектов прорисовываются с помощью компонента “LineRenderer”, для моделирования нужной формы и траектории необходимо написать компонент, в котором будет описан алгоритм рисования осей. Этот компонент будет принимать данные настроек компонента “LineRenderere”. Листинг разработанного класса “CircleLine” приведен в рисунке 31, его поля и методы описаны в таблицах 7 и 8 соответственно.

Таблица 7 — Поля класса “CircleLine”

| Название | Тип | Модификатор доступа | Описание |
|----------|-----|---------------------|----------|
|----------|-----|---------------------|----------|

|       |              |        |                        |
|-------|--------------|--------|------------------------|
| line  | LineRenderer | public | Компонент LineRenderer |
| alpha | float        | public | Угол альфа             |
| a     | float        | public | Ширина                 |
| b     | float        | public | Высота                 |

Таблица 8 — Методы класса “CircleLine”

| Название | Возвращаемый тип | Аргументы | Модификатор доступа | Описание                 |
|----------|------------------|-----------|---------------------|--------------------------|
| Start    | void             | -         | public              | Начало работы            |
| Update   | void             | -         | public              | Выполняется каждый фрейм |

```

using UnityEngine;

public class CircleLine : MonoBehaviour
{
    LineRenderer line;
    [SerializeField] float alpha;
    [SerializeField] float a;
    [SerializeField] float b;

    void Start()
    {
        line = GetComponent<LineRenderer>();
    }

    void Update()
    {
        for(int count = 0; count < line.positionCount; count++)
        {
            alpha = alpha + 360 / (line.positionCount - 1) * Mathf.Deg2Rad;
            line.SetPosition(count, new Vector3(a * Mathf.Sin(alpha), 0, b * Mathf.Cos(alpha)));
        }

        alpha = 0;
    }
}

```

Рисунок 31 — Листинг класса CircleLine

Исходя из ранее описанных требований к пользовательскому интерфейсу, управление 3д-моделью автомобиля должно осуществляться с помощью элемента “джойстика”. Для решения данной задачи необходимо импортировать в проект пакет Standard Assets, который содержит необходимые инструменты для реализации элементов управления объектами сцены, на платформе

мобильного устройства, смартфона или планшета. После импорта на сцену необходимо добавить “префаб” `MobileSingleStickControl`. Для того чтобы задействовать добавленный элемент, необходимо написать программный код, в котором будет реализован алгоритм управления 3д-моделью. Листинг разработанного компонента представлен на рисунке 32, его поля и методы описаны в таблицах 9 и 10 соответственно. Также для реализации данного алгоритма необходимо добавить к 3д-модели автомобиля компонент `Rigidbody`, который отвечает за физику объекта, в нем будут отключены функции использования к объекту гравитации и кинематики, иначе модель будем проваливаться сквозь твердой плоскости. Разрабатываемый класс будет наследником класса `MonoBehaviour` и использовать пространство имен `UnityStandardAssets.CrossPlatformInput`. Каждый фрейм происходит отслеживание использования `MobileSingleStickControl` и запись в вектор `movement` выходных данных координат осей X и Y, затем `Rigidbody`, используя полученный вектор и заданное время реализует движение объекта. Важно не только осуществить передвижение модели, но и менять ее угол поворота, за что отвечает вектор `transform.eulerAngles`. Во время движения модели также показываются модели вектора силы трения и передвижения.

Таблица 9 — Поля класса “CarContr”

| Название | Тип        | Модификатор доступа | Описание                               |
|----------|------------|---------------------|--|
| rb       | Rigidbody  | private             | Компонент Rigidbody                    |
| speed    | float      | private             | Скорость                               |
| vect     | GameObject | public              | Модель векторов силы трения и движения |

Таблица 10 — Методы класса “CarContr”

| Название | Возвращаемый тип | Аргументы | Модификатор доступа | Описание      |
|----------|------------------|-----------|---------------------|---------------|
| Start    | void             | -         | public              | Начало работы |

|        |      |   |        |                             |
|--------|------|---|--------|-----------------------------|
| Update | void | - | public | Выполняется<br>каждый фрейм |
|--------|------|---|--------|-----------------------------|

```

using UnityEngine;
using UnityEngine.CrossPlatformInput;

public class CarCntr : MonoBehaviour
{
    private Rigidbody rb;
    private float speed = 1f;
    public GameObject vect;
    void Start()
    {
        rb = GetComponent <Rigidbody> ();
    }

    void Update()
    {
        float x = CrossPlatformInputManager.GetAxis ("Horizontal");
        float y = CrossPlatformInputManager.GetAxis ("Vertical");

        Vector3 movement = new Vector3(x, 0, y);

        rb.velocity = movement * speed;

        if (x != 0 && y != 0)
        {
            vect.SetActive(true);
            transform.eulerAngles = new Vector3 (transform.eulerAngles.x, Mathf.Atan2 (x, y) * Mathf.Rad2Deg,
transform.eulerAngles.z);
        }
        else {
            vect.SetActive(false);
        }
    }
}

```

Рисунок 32 — Листинг класса CarCntr

Важным требованием к пользовательскому интерфейсу в реализации сцен с дополненной реальностью является наличие информационных блоков, относящихся к какому-либо 3д-объекту. Например, в сцене реализующей правило левой руки важно не только показать направление векторов действующий значений, но и указать название этих самых значений. Для этого в сцену был добавлен объект с текстом, имеющий компонент TextMesh Pro, который преобразует наш 2д текст в 3д. Положение данного объекта как и других объектов сцены “привязано” к отслеженной цели — к плоскости или изображению. Но если пользователь с телефоном, например, встанет с другой стороны к добавленному виртуальному объекту, то с нового ракурса объект с

текстом не будет отображаться корректно. Все потому что данный объект после инициализации никак не зависит от положения камеры в пространстве. Для решения данной проблемы к объекту необходимо написать компонент, который будет управлять ротацией объекта в зависимости от положения камеры. Разработанный программный код приведен на рисунке 33, поля и методы описаны в таблицах 11 и 12.

Таблица 11 — Поля класса “FaceCamera”

| Название    | Тип     | Модификатор доступа | Описание                  |
|-------------|---------|---------------------|---------------------------|
| targetAngle | Vector3 | public              | Вектор для ротации текста |

Таблица 12 — Методы класса “FaceCamera”

| Название | Возвращаемый тип | Аргументы | Модификатор доступа | Описание                 |
|----------|------------------|-----------|---------------------|--------------------------|
| Update   | void             | -         | public              | Выполняется каждый фрейм |

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[ExecuteInEditMode]
public class FaceCamera : MonoBehaviour
{
    Vector3 targetAngle = Vector3.zero;

    void Update()
    {
        transform.LookAt(Camera.main.transform);
        targetAngle = transform.localEulerAngles;
        targetAngle.y = targetAngle.y - 180;
        transform.localEulerAngles = targetAngle;
    }
}

```

Рисунок 33 — Листинг класса FaceCamera

### 3.4.3 Реализация экранов главного меню и теоретических материалов

Перед тем как приступить к практическому изучению, реализованному в сценах с дополненной реальность, пользователю необходимо ознакомиться с теоретическим материалом темы. А для этого, в самом начале работы с приложением, ему необходимо выбрать ту или иную тему. Для всего этого в

приложении реализованы сцены, содержащие экраны главного меню и теоретического материала.

Необходимые инструменты для работы с пользовательским инструментом содержатся в пакет Unity UI. В разработке данного приложения были использованы следующие объекты:

- Canvas — является родителем для всех остальных UI объектов;
- Button — объект кнопки, его компоненты;
- Panel — объект панели;
- Text — объект текста;
- Scroll View — объект скролла;

### **3.5 Сборка и тестирование проекта**

Заключительный этап проектирования — сборка и тестирование приложения. Для сборки проекта в Unity необходимо указать настройки Build Settings, рисунок 34, выбрать, на какую платформу будет собрано приложение и перейти в настройки плеера, рисунок 35. В настройках проекта указываются основные характеристики, версия платформы устройства, название проекта, поддержка сторонних библиотек и платформ, и так далее. Перед заданием данных настроек необходимо ознакомиться с официальной документацией используемых сторонних инструментов в проекте, а также согласование их используемых версий. При отсутствии каких-либо возникших конфликтов или системных ошибок, будет сгенерирован .apk файл, который необходимо скачать и установить на мобильное устройство.

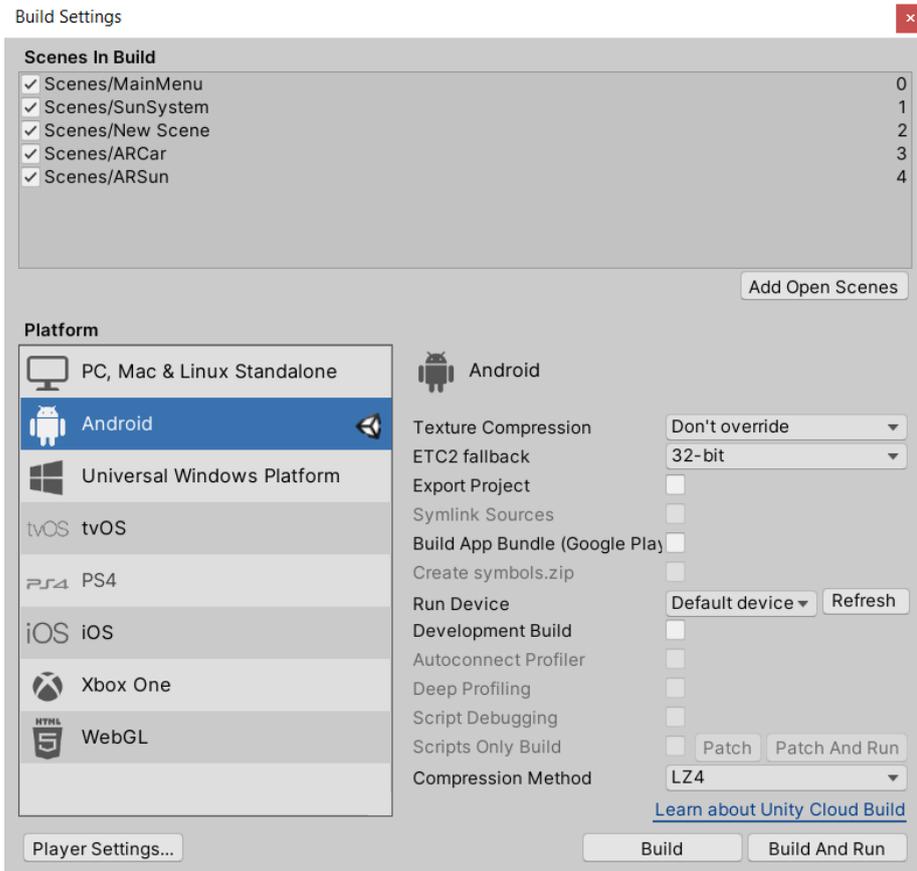


Рисунок 34 — Настройки Build Settings

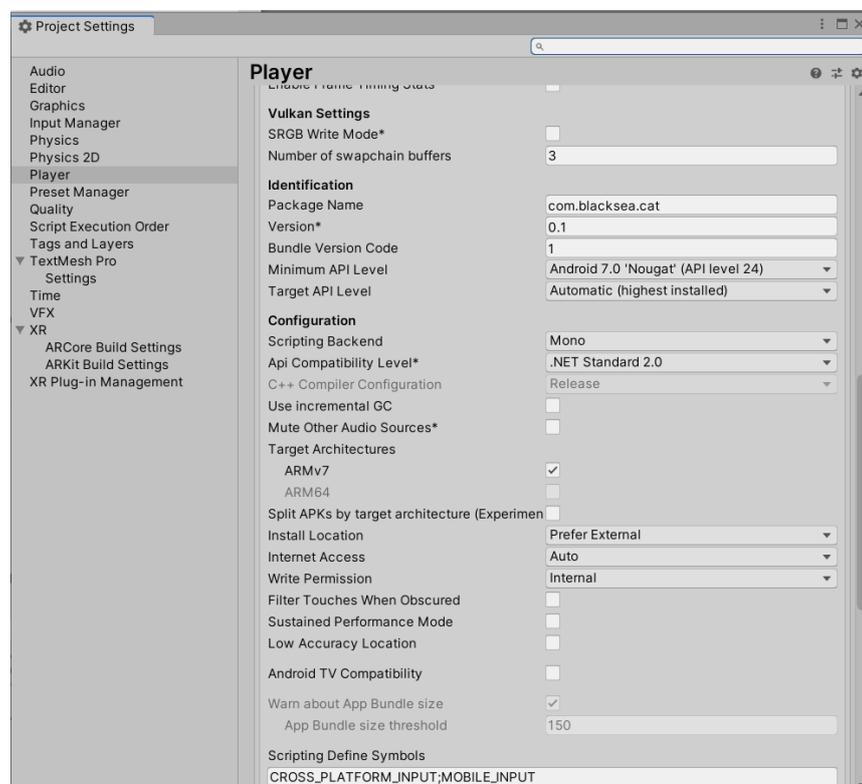


Рисунок 35 — Настройки плеера

Технические характеристики полученного приложения приведены в таблице 13. При установке приложения необходимо разрешение на использование камеры в настройках смартфона.

Таблица 13 — Технические характеристики приложения

|                          |                   |
|--------------------------|-------------------|
| Название                 | AR Физика         |
| Поддерживаемая платформа | Android 7 и более |
| Версия                   | 0.1               |
| Использование камеры     | да                |
| Использование микрофона  | нет               |
| Память                   | 58,01 МБ          |

Рисунки скриншотов выполнения программы приведены в приложении.

## 4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

### 4.1 Безопасность

В данном разделе рассмотрены основные аспекты безопасности работника, осуществляющего трудовую деятельность во время разработки программного обеспечения, а также основные положения о безопасности пользователя при взаимодействии с программным продуктом.

#### 4.1.1 Пользовательский интерфейс

Одним из основных направлений исследований в области обеспечения безопасности пользовательских интерфейсов, и, в частности, визуальных интерфейсов пользователя, является разработка моделей информационной безопасности при условии комплексного учета информационных, функциональных, психофизиологических и экологических аспектов безопасности. Это связано, прежде всего, с включением информационного фактора в состав факторов среды систем человек-компьютер и информационным характером почти всех происходящих в области распространения ИП процессов. Наименее разработанным областям проблематики защиты информации в системе человек-компьютер (СЧК) соответствуют такие угрозы, как:

- искажение воспринимаемой пользователем информации за счет её зашумления источниками среды на рабочем месте пользователя;
- потеря или искажение воспринимаемой пользователем информации из-за физической, семантической или синтаксической несогласованности её представления пользователю;
- искажение представлений пользователя о реальном состоянии объекта управления за счет скрытых информационных воздействий и неадекватное принятие им решений в процессе решения задач в рамках СЧК.

#### Визуальные и звуковые сигналы

Для правильного восприятия пользователем визуальный сигнал должен попадать в его поле зрения в течение всего периода выполнения необходимой операции.

Визуальный сигнал должен иметь только одно значение, но при разных условиях оно может быть разным. В случае сигнала, относящегося к обеспечению безопасности, цветовые характеристики органа управления и его окружения, если таковые применяются, должны соответствовать IЕС 60073.

Звуковой сигнал может подаваться в ответ на активизацию органа управления, в особенности применительно к стандартным операциям. Однако ограничиваться звуковым сигналом как единственным средством идентификации органа управления не рекомендуется.

При использовании звукового сигнала его характер и громкость должны быть адекватны прогнозируемому уровню акустического шума окружающей среды и расстоянию до предписанного рабочего места оператора.

Для надежного распознавания звукового сигнала последний должен иметь достаточную длительность или повторяться до тех пор, пока оператор не предпримет необходимых действий в ответ на сообщенную информацию, касающуюся безопасности. Когда звуковой сигнал используется для индикации информации, связанной с обеспечением безопасности, не должны предоставляться никакие средства понижения его уровня ниже порога слышимости.

Смысловое значение звукового сигнала должно быть предельно ясно и однозначно для пользователя.

### **Сенсорные области (органы управления) монитора (VDU)**

В тех случаях, когда орган управления представляется на экране дисплея сенсорной областью (например, активируемой с помощью курсора, светового пера или прикосновения пальцем), должны выполняться основные требования, излагаемые ниже. Размеры каждой из сенсорных областей - в особенности сенсорных полей - должны быть достаточно большими, чтобы они могли быстро и однозначно идентифицироваться пользователем и быть удобными для

применения надлежащих средств активизации. Область, относящаяся к функции обеспечения безопасности, должна иметь больший размер по сравнению с обычным рабочим полем, а между этой областью и другими органами управления должны существовать свободные позиции (по горизонтали и вертикали). Должны быть полностью предотвращены воздействия, которые могут привести к неопределенному или опасному состоянию в результате случайного прикосновения к сенсорной области.

#### 4.1.2 Организация рабочего места

Для обеспечения нормальных условий деятельности человека параметры микроклимата нормируются. Нормы производственного микроклимата установлены двумя нормативными документами:

- ГОСТ 12.1.005-88. ССБТ «Общие санитарно-гигиенические требования к воздуху рабочей зоны».
- Санитарные правила и нормы. СанПиН 2.2.4.2.2.1340-03

Уровень температуры, влажности и скорости движения воздуха регламентируется с учетом тяжести физического труда: «легкая», «средняя» и «тяжелая» работа. Помимо этого, учитывается сезон года: холодный период года (среднесуточная температура наружного воздуха ниже +10°C) и теплый период (температура +10°C и выше).

Работа офисного сотрудника характеризуется небольшим напряжением, сидя, на постоянном рабочем месте. Исходя из этого, ее можно охарактеризовать как работу I категории – легкой.

В таблице 14 приведены оптимальные параметры микроклимата для данной категории.

| Период года | Категория работ | Темп.воздуха, гр. С не более | Относит.влажность воздуха, % | Скорость движения воздуха, м/с |
|-------------|-----------------|------------------------------|------------------------------|--------------------------------|
|             |                 |                              |                              |                                |

|          |           |       |       |     |
|----------|-----------|-------|-------|-----|
| Холодный | легкая –1 | 22-24 | 40-60 | 0,1 |
| Теплый   | легкая -1 | 23-25 | 40-60 | 0,1 |

Таблица 14 – Оптимальные параметры микроклимата

При обеспечении оптимальных показателей микроклимата температура внутренних поверхностей, ограждающих рабочую зону конструкций (стен, пола, потолка) или устройств, а также температура наружных поверхностей технологического оборудования или его ограждающих устройств не должны выходить более чем на 2°С за пределы оптимальных величин температуры воздуха.

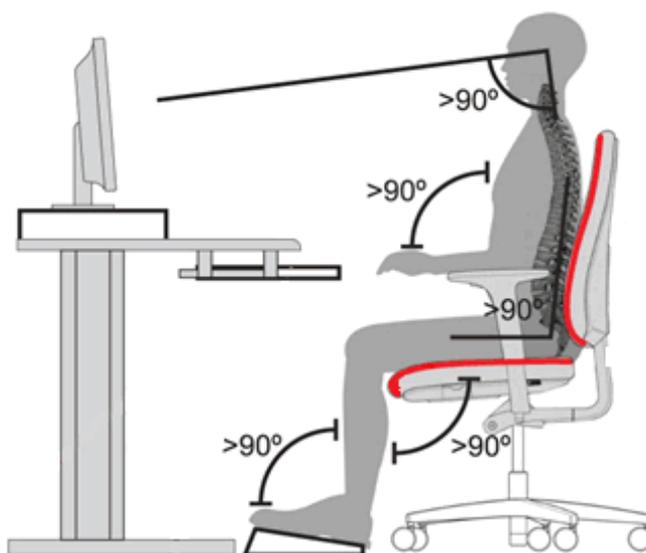


Рисунок 36 – Организация рабочего места

Рабочие места сотрудников соответствуют требованиям стандарта ГОСТ 12.2.032-78 «Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования». Верхний край монитора находится примерно на одном уровне с глазами, а угол между этим уровнем и центром экрана составляет 15 градусов. Расстояние от глаз до экрана составляет 45-70 см. Стул обеспечивает жесткую и удобную поддержку. Высота стула установлена таким образом, чтобы бедра находились горизонтально, а ноги доставали до пола. Конструкция рабочего стола поддерживает рациональную рабочую позу при работе на ПЭВМ. Плечо и предплечье

образовывают приблизительно прямой угол, запястье и рука находятся примерно на одной прямой. На рабочем месте соблюдается оптимальная высота стола или выдвижной полки для клавиатуры, которая составляет 68 – 73 см. над полом. Высота стула и стола подобрана так, чтобы минимально напрягать мышцы плеч, рук и запястий. Помимо выполнения требований по устройству рабочих мест, устанавливаются обязательные короткие, но частые перерывы в работе с компьютером.

### **Электробезопасность**

Для обеспечения электробезопасности используется система организационных мероприятий, электротехнических способов и средств. Для обеспечения безопасности персонала при непосредственном выполнении работ в электроустановках применяются следующие технические мероприятия:

- изоляция токоведущих частей ЭВМ;
- защитное заземление для обеспечения защиты от поражения электрическим током при прикосновении к металлическим нетокоевущим частям, которые могут оказаться под напряжением в результате повреждения изоляции.

Кабинет сотрудников по опасности поражения электрическим током можно отнести к 1 классу, т. е. это помещение без повышенной опасности (сухое, беспыльное, с нормальной температурой воздуха и не токопроводящими полами). Электробезопасность обеспечивается в соответствии с ГОСТ 12.1.030.-81. Система стандартов безопасности труда. Электробезопасность. Защитное заземление, зануление.

### **Электромагнитные поля**

Электростатическое поле и рентгеновское излучение отсутствуют у жидкокристаллических экранов. Помимо этого генерировать поля могут преобразователь напряжения питания, схемы управления и формирования информации на дискретных жидкокристаллических экранах и другие элементы аппаратуры. Портативные компьютеры располагаются гораздо ближе к

жизненно важным органам пользователя, соответственно облучение увеличивается.

В кабинете работника уровни напряженности электростатических полей составляют не более 20 кВ/м. Поверхностный электростатический потенциал не превышает 500 В. Для предотвращения действия излучения установлены обязательные пятнадцатиминутные перерывы в течении полутора часов работы и применяются защитные экраны. Защитный экран, изготовляемый из мелкой сетки или стекла, собирает на себе электростатический заряд. Для снятия заряда экран монитора заземлен.

Допустимые уровни электромагнитного излучения, т.е. плотность потока электромагнитной энергии, отражаются в нормативах, которые устанавливают государственные компетентные органы.

Работа разработчика ПО по виду трудовой деятельности относится к группе В – творческая работа в режиме диалога с ЭВМ, а по напряженности работы ко II категории тяжести (СанПиН 2.2.2/2.4.1340-03). В соответствии с этим для защиты от электромагнитного излучения для оператора устанавливаются перерывы в работе суммарной продолжительностью не менее 1 часа. Помимо этого применяются защитные экраны, способные защитить организм человека от электромагнитных полей.

### **Освещение**

Рациональное освещение рабочего места является одним из важнейших факторов, влияющих на эффективность трудовой деятельности человека, предупреждающих травматизм и профессиональные заболевания. Правильно организованное освещение создает благоприятные условия труда, повышает работоспособность и производительность труда.

Существует три вида освещения – естественное, искусственное и совмещенное.

Естественное освещение – освещение помещений дневным светом, проникающим через световые проемы в наружных ограждающих конструкциях помещений.

Искусственное освещение применяется при работе в темное время суток и днем, когда не удастся обеспечить нормированные значения коэффициента естественного освещения. Освещение, при котором недостаточное по нормам естественное освещение дополняется искусственным, называется совмещенным освещением.

Кабинет работника соответствует требованиям к освещенности в помещениях, где установлены компьютеры.

Поскольку зрительная работа операциониста является работой средней точности, обеспечивается общая освещенность 200 лк, а комбинированная – 300 лк.

Кроме того все поле зрения освещено достаточно равномерно – это основное гигиеническое требование. Поддерживается примерно одинаковый уровень освещения помещения и яркость экрана компьютера, т.к. яркий свет в районе периферийного зрения значительно увеличивает напряженность глаз и, как следствие, приводит к их быстрой утомляемости.

Работа разработчика характеризуется высокой степенью точности зрительной работы. Значение коэффициента естественной освещенности для производственных помещений с данным типом зрительной работы равно 1,2%.

Естественное освещение обеспечивается наличием окон большой площади. Поддержание же уровня освещенности достигается благодаря ежеквартальной чистке стекол.

Недостаток естественного света восполняется искусственным освещением. В качестве источников света при искусственном освещении применяются преимущественно люминесцентные лампы. При этом источники света по отношению к рабочему месту расположены таким образом, чтобы исключить попадания в глаза прямого света

Уровень искусственной освещенности на рабочих местах (на клавиатуре и поверхности стола) находится в пределах нормы и составляет 350 лк. Освещенность на экране дисплеев не превышает 250 лк.

Расчет общего освещения производится по формуле 1.

$$\Phi = \frac{E_{\text{норм}} \cdot K_z \cdot S \cdot z}{q \cdot f}, \text{ где (1)}$$

$E_{\text{н}}$  – нормируемая минимальная освещенность, 200 Лк;

$K_z$  – коэффициент запаса, 1,5;

$S$  – освещаемая площадь, 42 м<sup>2</sup>;

$z$  – коэффициент неравномерности освещенности, 1,1;

$q$  – коэффициент использования потока;

$f$  – коэффициент затемнения, принимается равным 0,9. Индекс помещения определяется по формуле 2:

$$i = \frac{s}{h \cdot (A+B)} = \frac{42,5}{3 \cdot (6,75+6,3)} = 1,08 \approx 1, \text{ где (2)}$$

$A$  – длина помещения, 7 м;

$B$  – ширина помещения, 6 м;

$s$  – площадь помещения, 42 м<sup>2</sup>;

$h$  – высота подвеса светильников над рабочей поверхностью, 3,5 м.

$$i = 42 / (3,5 \cdot (7+6)) = 0,92$$

Коэффициенты отражения стен и потолка примем равными  $R_{\text{ст}}=30$ ,  $R_{\text{п}}=50$ .

Для индекса  $i=0,9$ , коэффициентов  $R_{\text{ст}}=30$ ,  $R_{\text{п}}=50$ , коэффициент использования  $q=0,27$ .

Следовательно, получаем:  $\Phi = 57037$  лм.

Число  $N$  ламп, необходимых для организации общего освещения определяется по формуле 3.

$$N = \Phi / \Phi_{\text{л}}, \text{ где (3)}$$

$\Phi_{\text{л}}$  – световой поток лампы,  $\Phi_{\text{л}} = 4800$  лм.

$$N = 57037 / 4800 = 12$$

Таким образом, для организации освещения требуется 12 люминесцентных ламп ЛБ – 65.

## 4.2 Экологичность

Важным аспектом в вопросе об экологичности окружающей среды для предприятий офисного типа является процесс утилизации неработоспособной оргтехники. Списание и утилизация оргтехники – особая процедура, предусмотренная законодательством. Важность грамотной утилизации обусловлена следующими причинами:

- Исключение вредного воздействия на окружающую среду. Оргтехника и офисное оборудование относятся к категории опасных отходов – при их производстве используются токсичные вещества (свинец, ртуть, мышьяк и т. д.), способные отравлять почву, воду и воздух, разрушать озоновый слой.
- Повторное использование составных элементов и материалов. В конструкции оргтехники присутствуют элементы, содержащие цветные и драгоценные металлы (золото, серебро, платину, палладий и т. д.). Такие детали оборудования можно использовать повторно, кроме того драгметаллы нуждаются в особой процедуре списания с активов производства.

Предприятие организует процедуру утилизации оргтехники в соответствии с законодательством РФ. Регламентируют утилизацию неисправной и отработавшей оргтехники несколько правовых актов, основными являются:

- Кодекс РФ об административных правонарушениях (статья 8.2.);
- Постановление Правительства РФ от 23 апреля 2016 г. № 340;
- Постановление Правительства РФ от 26 августа 2006 г. № 524;
- Федеральный закон от 24 июня 1998 г. № 89;
- Федеральный закон от 26 марта 1998 г. № 41;
- Приказ Минприроды РФ от 27 сентября 2016 г. № 499.

Нарушение норм утилизации оргтехники грозит дисциплинарными, административными, уголовными или гражданско-правовыми санкциями – как к частным лицам, так и к организациям применимы выговоры, увольнения,

штрафы, требования о возмещении причиненного вреда окружающей среде, приостановка или прекращение деятельности.

### **4.3 Чрезвычайные ситуации**

Чрезвычайная ситуация – это обстановка на определенной территории, сложившаяся в результате аварии, опасного природного явления, катастрофы, стихийного или иного бедствия, которые могут повлечь или повлекли за собой человеческие жертвы, ущерб здоровью людей или окружающей природной среде, значительные материальные потери и нарушение условий жизнедеятельности людей.

Наиболее вероятной чрезвычайной ситуацией на исследуемом объекте является пожар.

Причинами возгорания в кабинете оператора могут быть неисправности в электросети, нарушение правил использования электроэнергии, использование открытого огня или оставленные без присмотра электронагревательные приборы, включенные в электросеть.

Пожарная безопасность обеспечивается системой предотвращения пожара и системой пожарной защиты.

Для тушения пожара предусмотрены два вида средств пожаротушения: стационарные установки и огнетушители.

Стационарные установки предназначены для тушения пожаров в начальной стадии их возникновения без участия людей. В помещении установлен датчик системы охранно-пожарной сигнализации, который предназначен для круглосуточного контроля охраняемого объекта, а также для раннего оповещения об обнаружении признаков пожара или задымления.

Огнетушители по виду огнетушащих средств подразделяют на жидкостные, углекислотные, химпенные, воздушно-пенные, хладоновые, порошковые и комбинированные. К углекислотным относятся ОУ-2, ОУ-3 ОУ-5, ОУ-8 огнетушители, которые применяются для тушения пожаров электроустановок, находящихся под напряжением. В помещении имеется один углекислый огнетушитель ОУ-2.

Здание, в котором располагается кабинет оператора, по пожарной опасности строительных конструкций в соответствии с ГОСТ 30403-96 «Конструкции строительные. Метод определения пожарной опасности» и ФЗ № 123 «Технический регламент о требованиях пожарной безопасности» относится к категории К1 (малопожароопасное). По конструктивным характеристикам здание можно отнести к зданиям с несущими и ограждающими конструкциями из естественных или искусственных каменных материалов, бетона или железобетона, где для перекрытий допускается использование деревянных конструкций, защищенных штукатуркой или трудно горючими листовыми, а также плитными материалами. Следовательно, степень огнестойкости здания можно определить как третью (III).

Класс конструктивной пожарной опасности зданий, сооружений, строений и пожарных отсеков должен устанавливаться в зависимости от их этажности, класса функциональной пожарной опасности, площади пожарного отсека и пожарной опасности происходящих в них технологических процессов. Степень конструктивной пожарной опасности здания можно определить как С0.

Основными документами, регламентирующими нормы пожарной безопасности в офисе, являются Федеральный закон № 123-ФЗ от 22 июля 2008 г. «Технический регламент о требованиях пожарной безопасности» и Правила противопожарного режима Российской Федерации, утвержденные постановлением Правительства РФ от 25 июля 2012 г. № 390 «О противопожарном режиме».

Инструкция по пожарной безопасности в офисе содержит следующие положения:

- общие требования пожарной безопасности;
- порядок содержания здания, сооружений (в том числе и эвакуационных путей) и прилегающей территории;
- требования безопасности перед началом работы, во время ее и после ее окончания;
- действия работников в случае возникновения пожара в офисе;

- перечень средств пожаротушения и порядок их применения.

Инструкцией устанавливаются лица, ответственные за обеспечение пожарной безопасности, в том числе за своевременное сообщение в пожарную охрану о возникновении очага возгорания, за организацию эвакуации сотрудников, за отключение электроэнергии (при необходимости) и прекращение всех работ в офисе.

Новые работники допускаются к работе в офисе только после прохождения первичного инструктажа. Кроме первичного, в офисе могут также проводиться повторный, внеплановый и целевой инструктажи, сведения о которых в обязательном порядке заносятся в журнал.

На предприятии установлен распорядительный документ, в соответствии с которым организуется офисное пространство и устанавливается противопожарный режим. Его основные положения включают:

- Наличие табличек с номером телефона вызова пожарной охраны.
- Наличие планов эвакуации при пожаре.
- Наличие знаков пожарной безопасности в офисе. К ним относятся информационные стенды и плакаты по пожарной безопасности и информационные таблички
- Наличие журнала инструктажа.

На основании установленных законодательством правил и норм пожарной безопасности в офисе введены следующие средства пожаротушения:

- Углекислые огнетушители ОУ-3 для тушения возгораний различных горючих веществ, горение которых не может происходить без доступа воздуха. Расстояние от очага пожара до места расположения огнетушителя не превышает 20 метров, размещение огнетушителей производится на видных местах на высоте не более 1,5 метров так, чтобы не препятствовать эвакуации людей.
- Противопожарная сигнализация состоит из:
  1. Приемно-контрольного прибора, который отвечает за прием сигнала, обнаружение места возгорания и выдачу сигналов для управления

устройствами дымоудаления, СОУЭ, вентиляции и кондиционирования, тушения пожара. Прибор снабжается аккумуляторной батареей – это обеспечивает его бесперебойную работу даже при отключении электричества.

2. Извещателей. Их функция – обнаружение очагов возгорания путем регистрации увеличения температуры, улавливания тепла пламени или появления дыма в помещении.
3. Оповещателей о пожаре – звуковых, световых и речевых устройств, которые предназначены для донесения информации о возникновении пожара до персонала офиса.

При возникновении пожара в одном из помещений офиса необходимо:

1. срочно сообщить в пожарную охрану о возгорании по телефону 01 (для сотовых 101, 010, 112);
2. если самостоятельно ликвидировать очаг возгорания невозможно, то закрыть окна и форточки, чтобы перекрыть доступ кислороду, и немедленно покинуть помещение;
3. оповестить о пожаре коллег в соседних помещениях;
4. отключить электроэнергию и вентиляцию (по возможности);
5. покинуть опасную зону и далее действовать по указанию ответственных за пожарную безопасность;
6. если по какой-то причине покидать помещение опасно (сильное задымление в коридоре), уплотнить дверь, заткнув щели подручными материалами и приоткрыть окно для проветривания, известить администрацию о месте своего нахождения, а после прибытия пожарных попросить помощи.

## ЗАКЛЮЧЕНИЕ

В ходе проделанной работы была изучена технология дополненной реальности, рассмотрены принципы ее работы, а также устройства, реализующие данную технологию. Кроме этого был проведен анализ применения AR в сфере школьного образования. По итогу которого можно сказать, что хоть данное направление и имеет огромное множество плюсов, а это в первую очередь улучшение процесса обучаемости школьника, но на практике такая технология не используется массово, а только как дополнение в образовательном процессе. При этом наблюдается активный интерес к AR как у школьников, так и у преподавателей. Основными проблемами в внедрении и использовании этой технологии в школьном образовательном процессе являются: недостаток качественных программных продуктов и недостаточное оснащение школ соответствующим оборудованием. Но в целом спрос на AR в данной сфере все же есть.

Предметом исследования в данной работе была — разработка мобильного приложения для изучения школьного курса физики с AR, на платформе android.

Реализовано приложение с помощью библиотеки AR Foundation SDK на базе кроссплатформенного движка Unity. Полученное приложение удовлетворяет поставленным требованиям и реализовывает возможности использования дополненной реальности в процессе обучения.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Документация ARCore. [Электронный ресурс]. URL: <https://clck.ru/NN4iN> (дата обращения 05.01.2020).
2. Документация ARFoundation. [Электронный ресурс]. URL: <https://clck.ru/NN3cM> (дата обращения 05.01.2020).
3. Документация Vuforia. [Электронный ресурс]. URL: <https://clck.ru/NnLwY> (дата обращения 07.01.2020).
4. Хокинг Д., Unity в действии. Мультиплатформенная разработка на C#. / Д. Хокинг. - М.: СПб, 2018. – 352с. (дата обращения: 10.01.2020).
5. Документация Unity. [Электронный ресурс]. URL: <https://docs.unity3d.com/ru/2019.3/Manual/UnityManual.html> (дата обращения 11.01.2020).
6. Троелсен Э. Язык программирования C# 7 и платформы .NET и .NET Core/ Э. Троелсен, Ф. Джепикс. - 8-е изд., изд-во. – Вильямс, 2018. – 1328 с. (дата обращения: 01.02.2020).
7. Мартин Р. Чистый код. Создание, анализ и рефакторинг. / Р. Мартин. – М.: Альфа-Книга, 2013. – 465 с. (дата обращения: 05.02.2020)
8. Мартин Р. Чистая архитектура. / Р. Мартин. – М.: Альфа-Книга, 2015. – 436 с. (дата обращения: 07.02.2020)
9. Курс по основам Blender. [Электронный ресурс]. URL: <https://www.youtube.com/watch?v=KO0wNQdiivs&list=PLuuJ7EJSjEfMETY8txzRpXHPH08Eg7kA6&index=2&t=0s> (дата обращения: 03.01.2020)
10. Фаулер М. Рефакторинг. Улучшение проекта существующего кода. / М. Фаулер. – М.: Альфа-Книга, 2017. – 448 с. (дата обращения: 04.12.2019)
11. Томас К. Алгоритмы: построение и анализ. / Томас Кормен, Чарльз Лейзерсон: Альфа-Книга, 2018. – 302 с. (дата обращения: 01.12.2019)
12. Уоррен Г., Алгоритмические трюки для программистов. / Г. Уоррен. - М.: Вильямс, 2014. – 512. (дата обращения: 10.11.2019).

13. Учебник по физике. Школьный курс. [Электронный ресурс]. URL: <https://mathus.ru/phys/book.pdf> (дата обращения: 03.12.2019)
14. Майкл Ф., Эффективная работа с унаследованным кодом. / Майкл Ф.: , 2018. – 412. (дата обращения: 10.11.2019).
15. Билл В., Эффективное программирование на C#. 50 способов улучшения кода. / Билл В.: , 2017. – 782. (дата обращения: 20.10.2019).
16. Бхаргава А., Грокаем алгоритмы. / Бхаргава А.: , 2018. – 282. (дата обращения: 10.03.2020).
17. Фленов М., Библия C#. / Фленов М.: , 2017. – 762. (дата обращения: 14.01.2020).
18. Давыдова Н., Программирование. / Давыдова Н.: , 2017. – 354. (дата обращения: 10.11.2019).
19. Бонд Г. Unity и C#. Геймдев от идеи до реализации . / Г. Бонд. – М.: Питер, 2019. – 928с. (дата обращения: 09.12.2019)
20. Гейг М. Unity и C#. Разработка игр на Unity 2018 за 24 часа . / М. Гейг. М.: Бомбора, 2020. – 464с. (дата обращения: 06.01.2020)

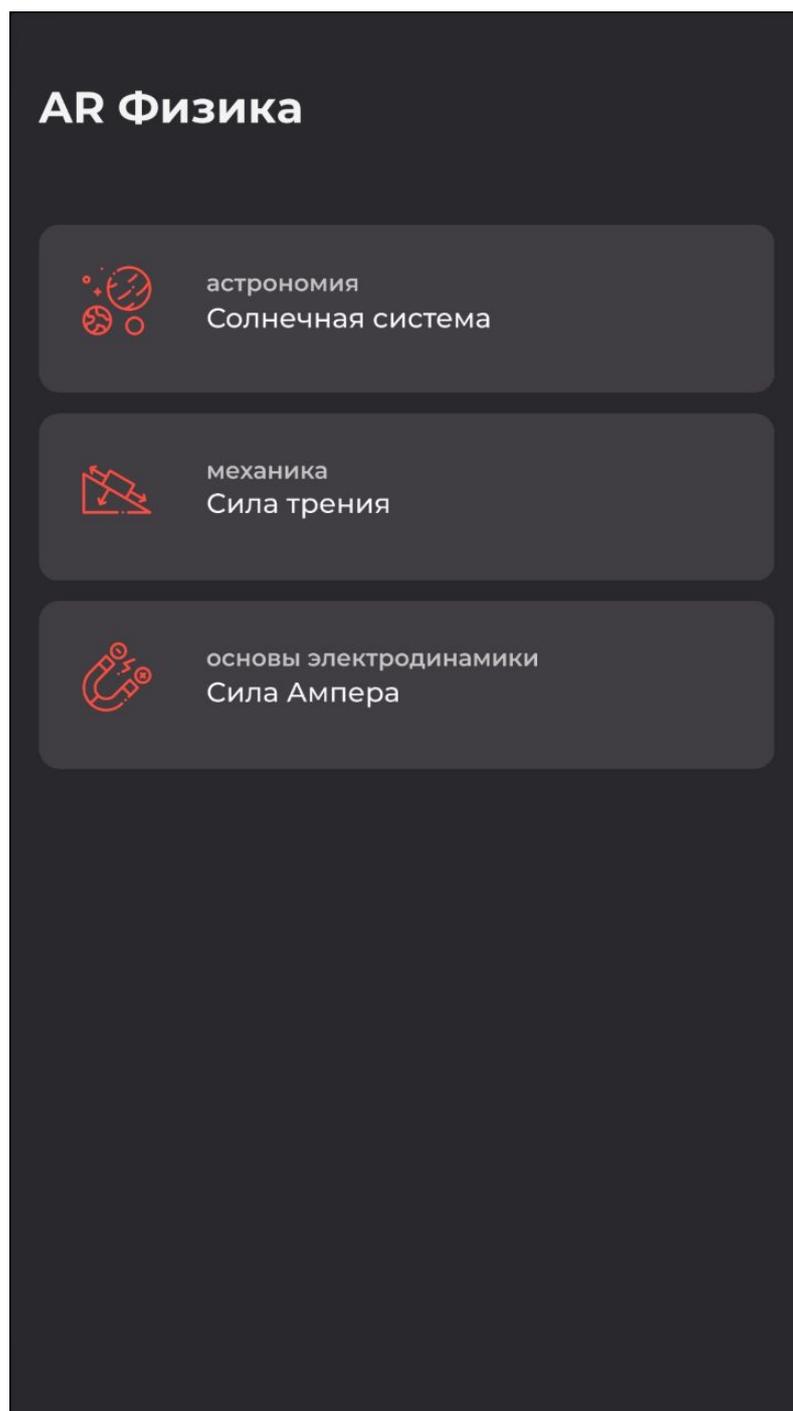


Рисунок 37 — Экран “Главное меню”



## Солнечная система

Солнечная система — планетная система, включает в себя центральную звезду — Солнце — и все естественные космические объекты, вращающиеся вокруг Солнца. Она сформировалась путём гравитационного сжатия газопылевого облака примерно 4,57 млрд лет назад[2].

Общая масса Солнечной системы составляет около 1,0014  $M_{\odot}$ . Большая часть её приходится на Солнце; оставшаяся часть практически полностью содержится в восьми отдалённых друг от друга планетах, имеющих близкие к круговым орбиты, лежащие почти в одной плоскости — плоскости эклиптики. Из-за этого наблюдается противоречащее ожидаемому распределение момента импульса между Солнцем и планетами (т. н. «проблема моментов»): всего 2 % общего момента системы приходится на долю Солнца, масса которого в ~740 раз больше общей массы планет, а остальные 98 % — на ~0,001 общей массы Солнечной системы[18].

Четыре ближайшие к Солнцу планеты, называемые планетами земной группы, — Меркурий, Венера, Земля[19] и Марс. Четыре остальные планеты — Юпитер, Сатурн, Уран и Нептун (также называемые

Смотреть в 3D



Рисунок 38 — Экран “Теоретический материал”



Рисунок 39 — Экран “Модель солнечной системы AR”

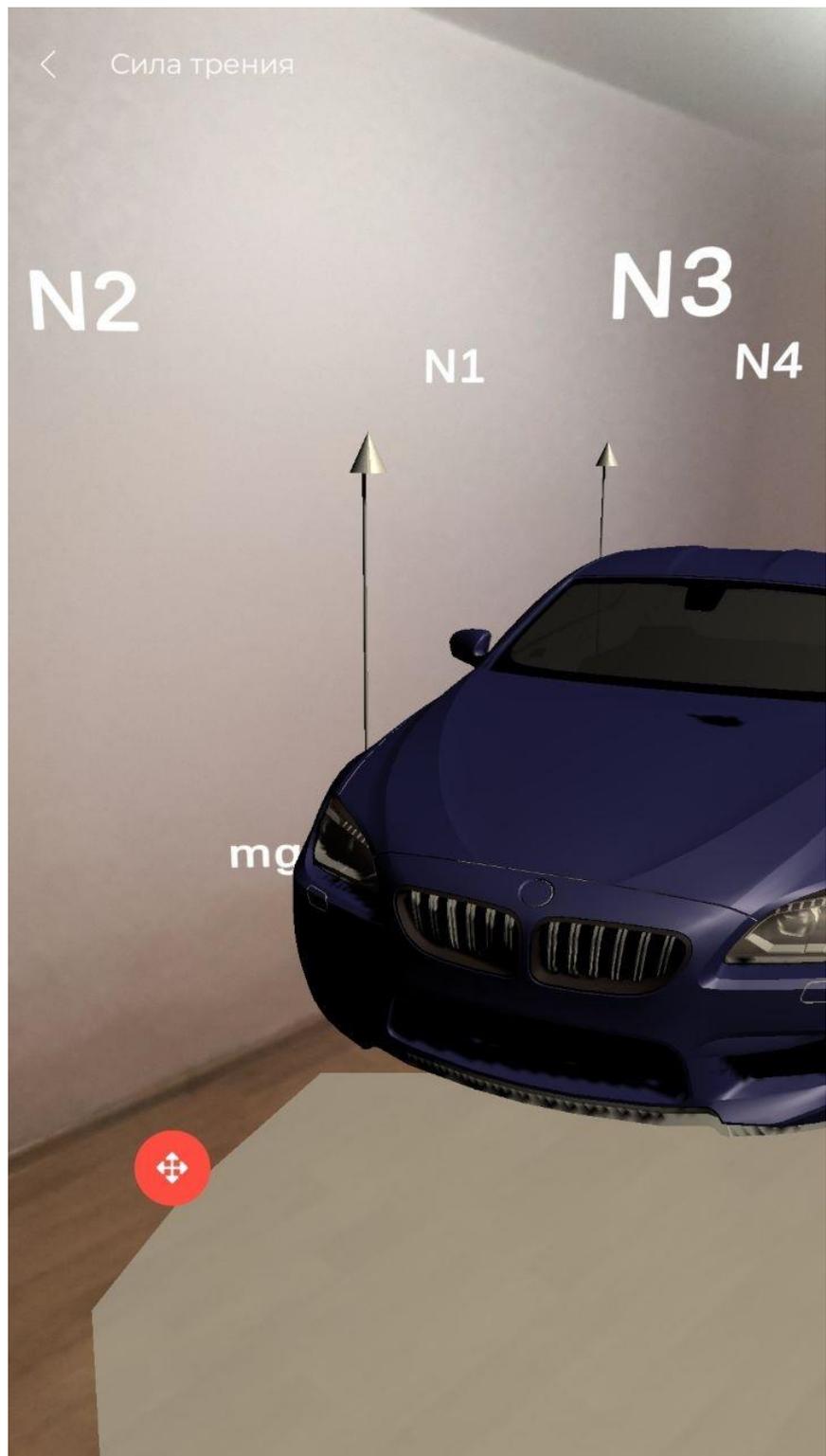


Рисунок 40 — Экран “Модель автомобиля AR”

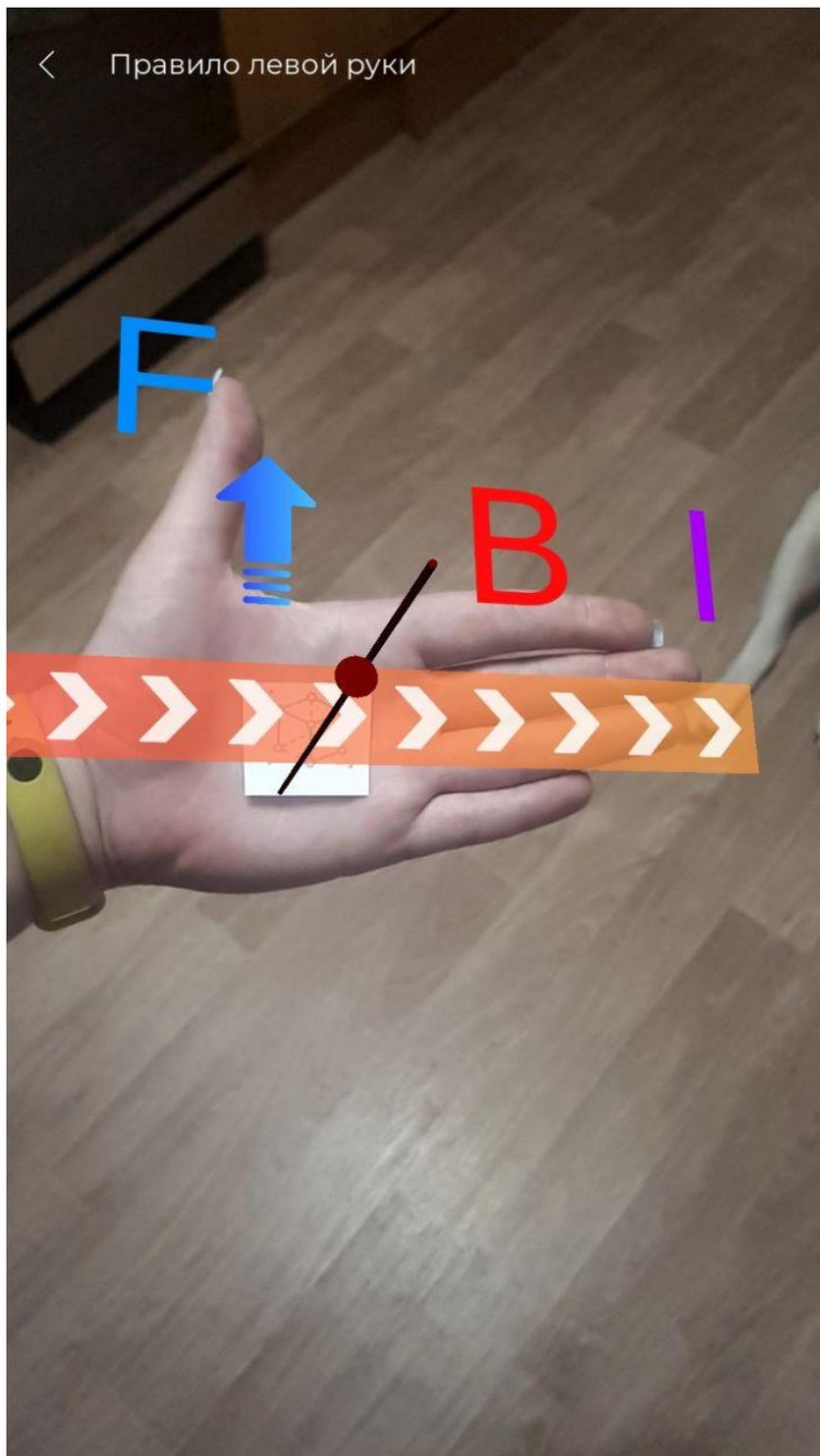


Рисунок 41 — Экран “Модель правила левой руки AR”