**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Кафедра многопроцессорных систем и сетей**

**РАЗРАБОТКА ИНТЕРАКТИВНОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ VR**

Курсовой проект

|  |  |
| --- | --- |
|  | Дорошева Игоря Викторовича  студента 3 курса,  специальность «Информатика»  Научный руководитель:  ассистент кафедры МСС А. С. Гусейнова |

Минск, 2018

**РЕФЕРАТ**

Курсовой проект, 32 стр., 10 рис., 10 ист.

***Ключевые слова:*** виртуальная реальность, VR, опыты, образование, интерактив, Unity, Android, MySQL, Node.js.

***Объект исследования*** – изучение вопроса использования виртуальной реальности в образовании, исследование программы школьных опытов, исследование и изучение средств разработки интерактивных VR приложений.

***Цель работы*** – исследовать и изучить средства разработки интерактивных VR приложений.

***Результат работы*** – изучены средства разработки интерактивных VR приложений и применены для написания демо-приложения.

***Область применения*** – образование, улучшение качества восприятия материала с помощью визуализации объектов исследований в виртуальной реальности.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc532419941)

[ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ, ЦЕЛИ И ПРОБЛЕМЫ ПРОЕКТА 7](#_Toc532419942)

[1.1 Постановка задачи 7](#_Toc532419943)

[1.2 Цели проекта 7](#_Toc532419944)

[1.3 Проблемы проекта 8](#_Toc532419945)

[1.4 Выводы 8](#_Toc532419946)

[ГЛАВА 2. ОПИСАНИЕ ПРИЛОЖЕНИЯ. ОПИСАНИЕ ИНТЕРФЕЙСА. ПОСТАНОВКА ЗАДАЧИ ИГРОКА. 9](#_Toc532419947)

[2.1 Описание приложения 9](#_Toc532419948)

[2.2 Описание интерфейса 9](#_Toc532419949)

[2.3 Постановка задачи пользователя 11](#_Toc532419950)

[ГЛАВА 4. СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ ДЛЯ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ПРИЛОЖЕНИЯ. 12](#_Toc532419951)

[4.1 Средства пользователя 12](#_Toc532419952)

[4.2 Средства разработчика 13](#_Toc532419953)

[ГЛАВА 5. РЕШЕНИЕ ПРИКЛАДНОЙ ЗАДАЧИ И РЕАЛИЗАЦИЯ ПРОЕКТА. 14](#_Toc532419954)

[5.1 Архитектура приложения 14](#_Toc532419955)

[5.2 Реализация базы данных 15](#_Toc532419956)

[5.3 Реализация серверного приложения 17](#_Toc532419957)

[5.4 Реализация клиентского приложения 22](#_Toc532419958)

[5.5 Реализация опытов и управление персонажем 27](#_Toc532419959)

[ГЛАВА 6. РЕЗУЛЬТАТЫ РАЗРАБОТКИ 31](#_Toc532419960)

[ЗАКЛЮЧЕНИЕ 32](#_Toc532419961)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 33](#_Toc532419962)

# **ВВЕДЕНИЕ**

В последнее время всё более широкое распространение получает виртуальная реальность - созданный техническими средствами мир, передаваемый человеку через его ощущения, прежде всего зрение. Виртуальная реальность имитирует как воздействие, так и реакции на воздействие. Для создания убедительного комплекса ощущений реальности компьютерный синтез свойств и реакций виртуальной реальности производится в реальном времени. Объектам виртуальной реальности обычно придают поведение, близкое к поведению аналогичных объектов материальной реальности для большего погружения. Пользователь может воздействовать на эти объекты в согласии с реальными законами физики (гравитация, свойства воды, столкновение с предметами, отражение и т. п.). Однако часто в развлекательных, образовательных или других целях пользователям виртуальных миров позволяется больше, чем возможно в реальной жизни (например, летать, создавать любые предметы и т. п.).

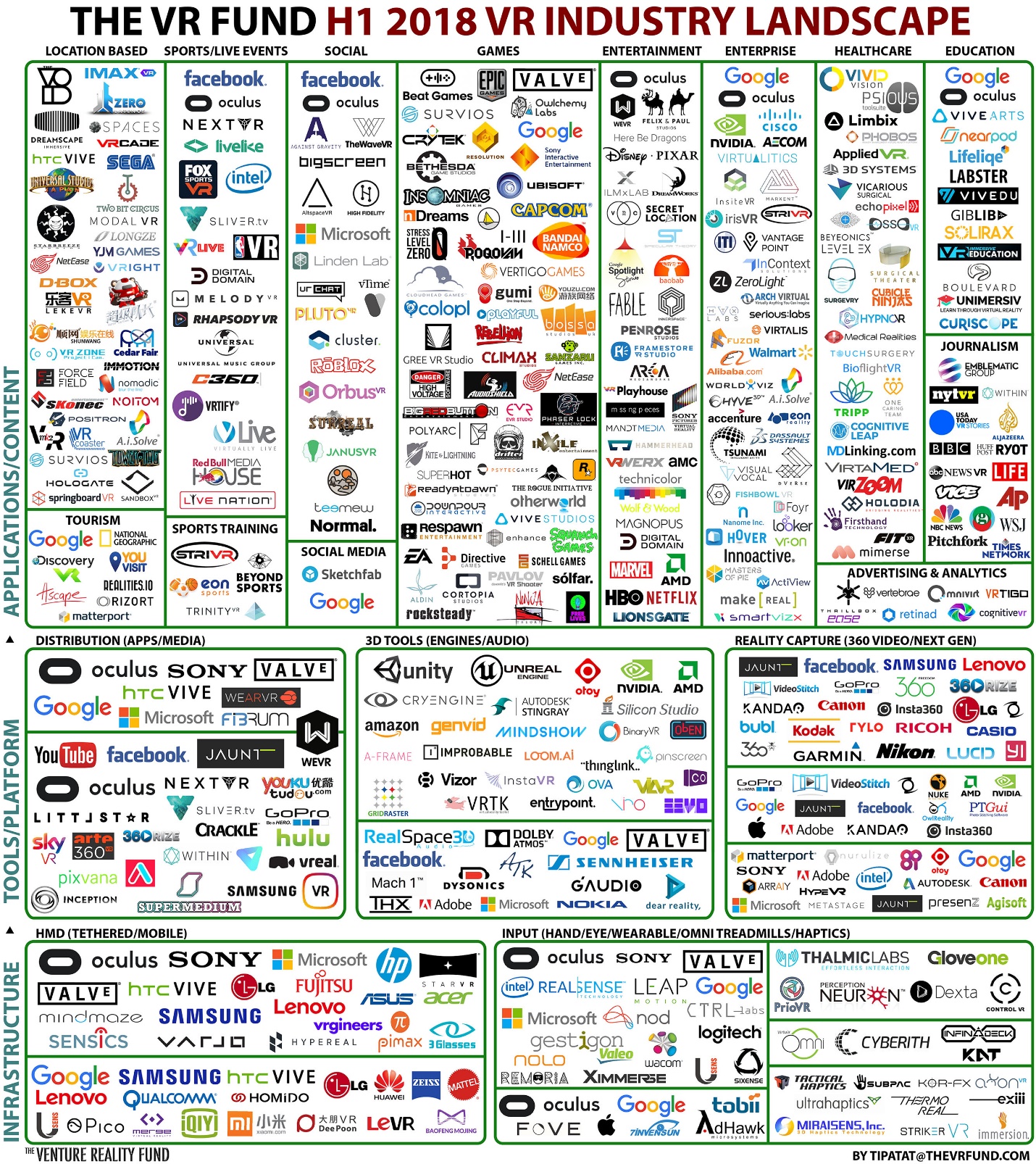
В данный момент технологии виртуальной реальности широко применяются в различных областях человеческой деятельности: проектировании и дизайне, добыче полезных ископаемых, военных технологиях, строительстве, тренажёрах и симуляторах, маркетинге и рекламе, индустрии развлечений и т. д. Объём рынка технологий виртуальной реальности оценивается в 14 млрд долларов в год. В настоящее время множество компаний, такие как Samsung, Sony, Facebook и Google активно вкладываются в развитие VR. По прогнозам компании International Data Corporation, в ближайшие 5 лет темпы роста рынка виртуальной и дополненной реальности составят 58%. Фонд Venture Reality выпустил исследование [1] мирового рынка виртуальной и дополненной реальности (рис. 1), которое распределило десятки работающих в этом направлении компаний по специализации на рынке. Исследование основано на сборе информации, как удалённо, так и в ходе встреч с представителями фигурирующих компаний. Полученная в итоге схема представляет всех, которые специализируются на виртуальной и дополненной реальности, а также определяет наиболее конкурентные сферы.

Рисунок 1 - Компании, инвестирующие в VR (Иллюстрация The Venture Reality Fund)

Проследив динамику по данным исследований The Venture Reality Fund в период с 2016 по 2018 год, можно заметить, что в индустрии виртуальной и дополненной реальности появляются все новые и новые сферы, а также все больше и больше ведущих компаний обращают внимание на данный сегмент.

Задачей курсового проекта является создание мобильного приложения, позволяющего выполнять образовательные опыты в виртуальной реальности. Особенность состоит в том, чтобы сделать среду интерактивной, позволив пользователю не только наблюдать за ходом опытов в виртуальной реальности, но и принимать непосредственное участие в их проведении.

# **ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ, ЦЕЛИ И ПРОБЛЕМЫ ПРОЕКТА**

## **1.1 Постановка задачи**

Курсовой проект ставит задачей разработку обучающего приложения, реализованного в виртуальной реальности. Тема приложения – проведение в виртуальной реальности опытов, связанных с определенным учебным материалом общеобразовательных школ Республики Беларусь. Благодаря широким возможностям виртуальной реальности, существует возможность не ограничиваться определенным возрастом или предметом. А соответствие приложения школьной программе обучения позволяет внедрять и использовать его в различных образовательных учреждениях. Однако, это не ограничивает аудиторию пользователей. Приложение предлагает различные сценарии и уровни сложности, что позволяет его использовать не только как образовательный, но и как развлекательный продукт.

Задача проекта - предоставить доступный продукт с использованием виртуальной реальности, поэтому используются средства разработки, позволяющие получить на выходе приложение, которое возможно запустить на любом смартфоне под управлением одной из наиболее популярных и открытых операционных систем - ОС Android.

## **1.2 Цели проекта**

Проект ставит целью усовершенствовать средства обучения. Приложение не является заменой учебнику, а рассматривается как дополнение к нему. Подобно лабораторным работам по физике, химии, биологии, выполняемым только на бумаге, приложение позволяет на практике закрепить информацию, предоставленную учебником. Опыты, описанные в школьных учебниках, зачастую трудно воспроизвести в школьных условиях, а иногда и вовсе невозможно. В то же время в виртуальной реальности возможно показать любой опыт, не опасаясь за безопасность учеников. Использование приложения не должно вызывать трудностей у современного человека, поэтому необходимо предоставить понятный и простой интерфейс, ясно и четко сформулировать задачу перед пользователем.

## **1.3 Проблемы проекта**

Проблема проекта заключается в том, чтобы использовать технологию виртуальной реальности в образовательных целях, не используя дорогостоящего оборудования, заменив его мобильными устройствами. Это накладывает ограничение на качество и количество объектов, отображаемых в виртуальной реальности.

Также проблемой использования виртуальной реальности является влияние на самочувствие человека. В то время, как пользователь перемещается в приложении, его тело остаётся неподвижным, что может вызвать тошноту и головную боль. Существует несколько вариантов решения данного вопроса, однако ни один из них не является полным [2]. В процессе разработки приложения учтены все возможные способы минимизировать негативное воздействие на самочувствие пользователя. Например, опыты продумываются таким образом, чтобы ученик проводил в очках виртуальной реальности не более 5 минут подряд, что является абсолютно безопасным промежутком времени.

## **1.4 Выводы**

Таким образом, в данной главе поставлены задачи и цели приложения, требования к разработке. Проанализированы проблемы и способы их решения.

# **ГЛАВА 2. ОПИСАНИЕ ПРИЛОЖЕНИЯ. ОПИСАНИЕ ИНТЕРФЕЙСА. ПОСТАНОВКА ЗАДАЧИ ИГРОКА.**

## **2.1 Описание приложения**

В физическом смысле приложение после сборки представляет собой файл с расширением apk для установки основной части приложения на устройство пользователя.

Каждый опыт в приложении является пакетом, устанавливаемым на устройство пользователя с помощью таких же \*.apk файлов, сгенерированных игровым движком Unity 3D, скачать которые можно напрямую из Google Play Market – официального магазина приложений операционной системы Android.

Приложение погружает пользователя в атмосферу, соответствующую его задаче и отображает окружение игрока. В опыты могут быть встроены звуковые эффекты: музыка или подсказки к интерфейсу для упрощения понимания учеником. Пользователь может перемещаться внутри опыта с помощью джойстика, а также взаимодействовать с элементами интерфейса с помощью кнопки на очках виртуальной реальности и таким образом использовать предоставленные возможности по выполнению опыта - интерфейс и перемещение.

Проект является обучающим игровым приложением, нацеленным пояснить теоретический материал, предоставляемый учебником.

## **2.2 Описание интерфейса**

Условно можно разделить части интерфейса на:

* интерфейс меню
* интерфейс внутри сцены

Интерфейс меню - интерфейс, предоставленный пользователю вне игровой сцены. Служит для настройки приложения, выбора необходимого опыта и мониторинга предыдущих результатов. Такой интерфейс включает в себя кнопку входа в меню и пункты меню (рис. 2.1).

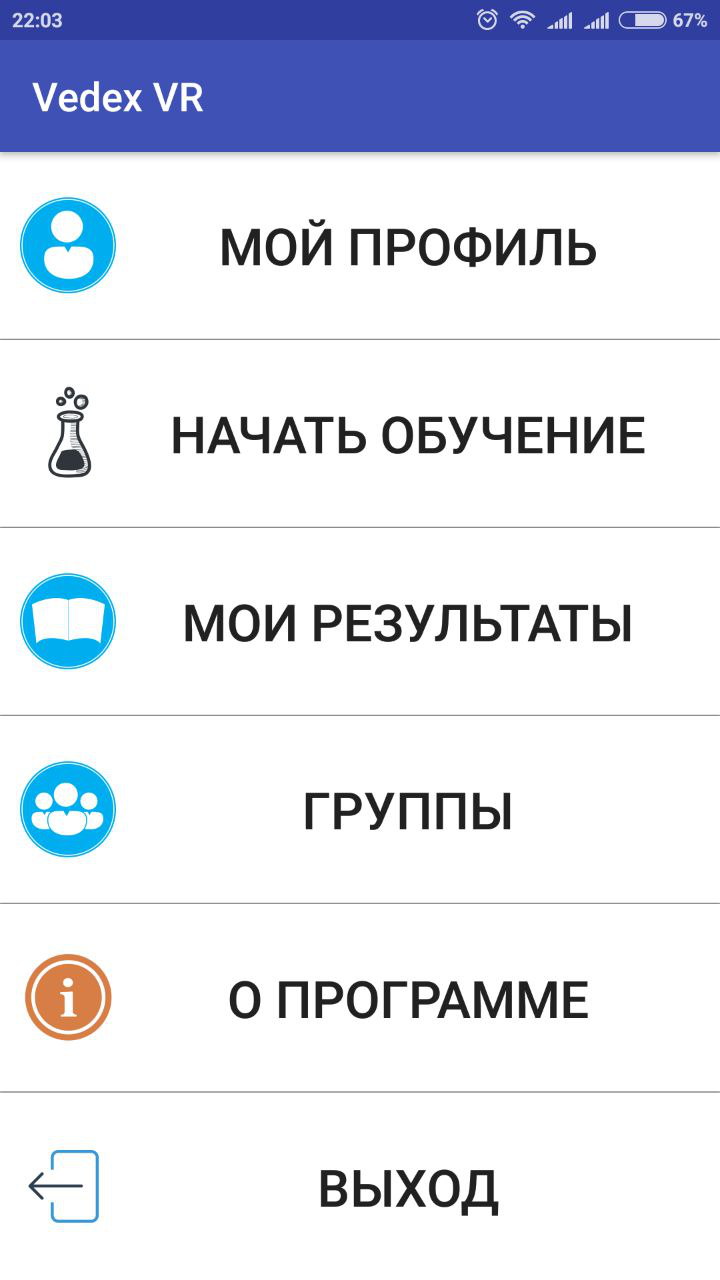


Рисунок 2.1 - Интерфейс меню

Создание меню выполнено с помощью встроенных средств разработки среды Android Studio [3]. Разметка интерфейса и стилизация объектов реализованы с помощью правил, описанных в структуре xml [4].

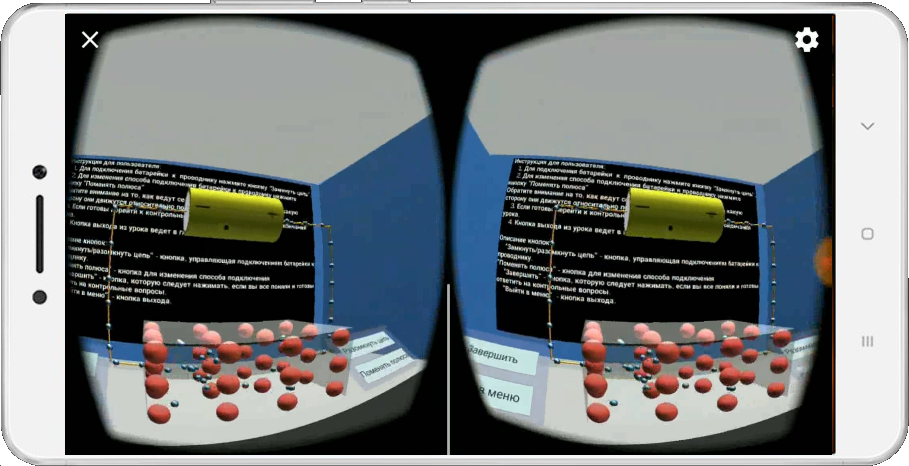
Интерфейс внутри сцены - набор инструментов, позволяющих пользователю управлять игроком в процессе выполнения поставленной задачи на игровом уровне. Такой интерфейс включает элементы управления персонажем (будет описано в инструментах пользователя в следующей главе), рабочую область (рис. 2.2), кнопки и панели инструментов (в зависимости от поставленной перед пользователем в конкретном опыте задачи), постановку задачи, подсказки в управлении, составляющие сцену объекты.

Рисунок 2.2 - Внутриигровой интерфейс

## **2.3 Постановка задачи пользователя**

Каждый уровень ставит перед пользователем определенную задачу. Задача пользователя состоит в выполнении набора заданий по проведению опыта, используя подсказки окружающей среды и средств интерфейса: текста, образов или звука. Например, в одном из опытов предлагается замкнуть электрическую цепь, чтобы пронаблюдать движение электронов под действием электрического тока. При этом, в разомкнутой цепи электроны движутся хаотично. Если поменять полюса источника тока местами, электроны в замкнутой цепи начнут двигаться в другом направлении. Так как приложение в первую очередь ставит целью закрепление знаний по определённому предмету, а приложение является дополнением к учебнику, то задачи, формулируемые на каждом уровне, должны соответствовать программе, описанной в учебнике.

# **ГЛАВА 4. СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ ДЛЯ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ПРИЛОЖЕНИЯ.**

## **4.1 Средства пользователя**

Одной из важных задач проекта было создание *доступного* приложения, поэтому реализация выбрана так, чтобы упростить требования к пользователю относительно необходимого оборудования. Таким образом, прежде всего пользователю потребуется смартфон под управлением операционной системы Android. Также, потребуется наличие очков виртуальной реальности для смартфона. Самой доступными из них являются очки Google Cardboard (рис. 4.1), однако существует множество различных разновидностей очков виртуальной реальности, различающиеся в основном по материалу или наличию крепления на голову. Также рекомендуемо наличие кнопки на очках для возможности нажатия на экран, а, следовательно, позволяющее удобнее взаимодействовать с интерфейсом опытов.



Рисунок 4.1 - Google Cardboard

## **4.2 Средства разработчика**

В качестве платформы для разработки и движка выбрана платформа разработки игр Unity 3D.

Преимуществами данной платформы можно считать кроссплатформенность, популярность, доступность. Платформа поддерживается и постоянно обновляется. Найти необходимую информацию, дополнительное программное обеспечение или расширить возможности стандартной платформы не составит труда.

Однако, по умолчанию Unity не предоставляет средств разработки приложений в виртуальной реальности, поэтому было решено использовать библиотеку Google VR SDK for Unity, распространяемую бесплатно Google Inc. [5].

Данная библиотека упрощает разработку приложений в виртуальной реальности и сборку приложений для устройств под управлением Android. В ней предоставлены стандартные решения для настройки звуковых эффектов под VR-приложения, стандартные исходные коды для поддержки управления с джойстиком, а также стандартная VR-камера. К библиотеке прилагается полная документация, что позволяет быстро находить ответы на возникающие вопросы.

В качестве средства редактирования и проверки исходного кода выбрана среда разработки Microsoft Visual Studio 2017 как новейшая и самая популярная среда разработки для языка C#, используемого для написания кода в Unity.

# **ГЛАВА 5. РЕШЕНИЕ ПРИКЛАДНОЙ ЗАДАЧИ И РЕАЛИЗАЦИЯ ПРОЕКТА.**

## **5.1 Архитектура приложения**

Приложение состоит из трёх частей – базы данных, серверного приложения и клиентского приложения. При этом, серверное приложение выступает прослойкой между клиентским приложением и базой данных (рис. 5.1)



Рисунок 5.1 – Архитектура приложения

На рисунке 5.1 можно наблюдать схему работы приложения. В целях безопасности, пользователь не должен иметь прямого доступа к базе данных. Для достижения этого использована прослойка в виде сервера, который контролирует доступ и запрещает неавторизованным пользователям получать информацию из базы, а также ограничивает доступ авторизованных пользователей таким образом, чтобы они не могли извлечь данные других пользователей.

Коммуникация клиентского и серверного приложений обеспечивается с помощью протокола HTTP, передавая данные согласно концепции REST. Данные сериализуются в формат JSON, после чего отсылается на сервер. Сервер получает информацию, десериализует её и обрабатывает соответствующим образом, в зависимости от запроса. После обработки полученных данных, сервер извлекает необходимую информацию из базы данных, сериализует её в JSON и отправляет клиенту. При этом, после авторизации клиента с помощью логина и пароля, сервер создаёт сессию и поддерживает её на протяжении работы пользователя клиентского приложения. Такой подход позволяет не требовать постоянную авторизацию пользователя для каждого запроса, а лишь авторизоваться однажды, при входе в приложение.

## **5.2 Реализация базы данных**

Для хранения информации о пользователях, их результатах в прохождении опытов и самих опытах необходима база данных – набор таблиц, связанных определёнными отношениями. В качестве базы данных для проекта было решено использовать реляционную базу данных MS SQL.

Разделим информацию, которую необходимо хранить, на группы и выделим для каждой группы элементы, которые будут записаны в базу данных. Рассмотрим, как хранится каждая группа.

Информация о пользователе включает имя и фамилию пользователя – для идентификации в таблице результатов, логин и пароль – для авторизации пользователя, адрес электронной почты – для взаимодействия с пользователем вне приложения, фото пользователя – для отображения в профиле. Сохранение всех фотографий на сервере нецелесообразно по затратам памяти, поэтому фотографии будут автоматически загружаться на хостинг картинок клиентским приложением, а серверу будет передаваться ссылка на фотографию.

Информация о группах пользователей включает имя группы, описание группы и дату её создания. Необходимо гарантировать уникальность поля «имя группы», не позволяя создавать группы с одинаковыми именами.

Для хранения отношений принадлежности пользователя к группе, необходимо создать таблицу, в которой будут храниться пользователи-участники групп, а также группы, в которых они состоят. Каждый пользователь имеет определённую роль в группе. Так, пользователь может быть создателем группы, модератором или обычным её участником.

Для хранения информации о ролях пользователей в группах, создана таблица ROLES, поля которой: ID роли – искусственное поле для уникальной идентификации роли, ACCESS\_LEVEL – уровень доступа пользователя в группе. В зависимости от уровня доступа, пользователь будет получать определённые возможности в группе, такие как добавление новых пользователей, редактирование описания и имени группы. Ещё одно поле в данной таблице – поле NAME, хранящее имя роли.

Таблица, хранящая приглашения пользователей в группы, должна хранить следующие поля: ID приглашающего пользователя – ссылка на таблицу USERS по полю USER\_ID, ID приглашаемого пользователя – ссылка на таблицу USERS по полю USER\_ID, ID группы, в которую приглашается пользователь. Поле, хранящее ID группы, является ссылкой на таблицу GROUPS. Так же необходимо сохранить дату приглашения пользователя в группу и сообщение, которое сопровождает приглашение.

Для хранения сведений об опытах, доступных пользователям, нам понадобятся следующие поля: ID опыта – сгенерированный искусственный номер опыта, название опыта, описание опыта, изображение, которое будет служить превью к опыту, а также информация, позволяющая классифицировать опыты – год обучения в школе, когда опыт рекомендуется проходить и предмет, к которому относится опыт.

Для хранения результатов выполнения опытов пользователями будет создана таблица RESULTS, содержащая следующую информацию: ID результата – сгенерированный автоматически ключ, ID опыта – ссылка на таблицу LESSONS, определяющая опыт, к которому относится запись о результате. Кроме того, необходимо хранить ссылку на профиль пользователя, чей результат будет записываться. В дополнение запишем время – сколько заняло прохождение опыта и дату – когда был пройден опыт.

## **5.3 Реализация серверного приложения**

Для обработки запросов клиентского приложения и обеспечения связи между клиентом и базой данных, а также защитой от несанкционированного доступа к данным, разработано серверное приложение – программа, идентифицирующая клиента и обрабатывающая его запросы.

Сервер реализован на технологии node.js, что делает его платформонезависимым [6].

Клиент отсылает HTTP запросы на сервер, а значит, сервер должен иметь обработчиков для каждого запроса. Данные, передаваемые в таких запросах, сериализованы в виде JSON-объектов. Результаты обработки запросов – ответы клиенту – так же сериализуются в JSON перед отправкой. Все запросы в приложении являются запросами типа POST. Будем называть их POST-запросами.

Рассмотрим основные запросы, обрабатываемые сервером, и структуру данных, которые они передают.

POST-запрос по адресу /doLogin используется для авторизации пользователя в системе и принимает в качестве входных параметров логин или адрес электронной почты пользователя и его пароль. После чего, серверное приложение обращается в базу, чтобы проверить, есть ли в таблице USERS данный пользователь и соответствует ли переданный пароль записанному в базе. В случае, если данные корректны, авторизация проходит успешно и сервер в качестве ответа на запрос возвращает клиенту информацию о его профиле. Кроме того, на данном этапе сервер создаёт сессию с клиентским приложением и при обработке последующих запросов от данного пользователя, ему не придётся проходить авторизацию заново.

POST-запрос по адресу /doRegister используется клиентским приложением для создания новой учётной записи. Входными данными для этого запроса являются логин, пароль, адрес электронной почты, имя и фамилия пользователя. После получения данных о пользователе, сервер проверяет, нет ли такого пользователя в существующей базе данных. В случае, если пользователь уже существует, клиенту возвращается ошибка, сообщающая ему, что пользователь с таким логином уже существует. В противном случае, создаётся новая запись в таблице USERS, а клиенту сообщается об успешной регистрации. При создании новой записи в таблице USERS, сервер в целях безопасности сохраняет только хэш пароля.

POST-запрос по адресу /updatePhoto используется клиентским приложением для изменения фотографии профиля и принимает единственный параметр – URL новой фотографии. В случае, если пользователь авторизован и для него создана сессия, серверное приложение обновляет запись пользователя, устанавливая ему новое значение в поле PHOTO. В качестве ответа на запрос клиенту сообщается об успешной его обработке.

POST-запрос по адресу /updateAccountInformation используется для изменения данных аккаунта пользователя и принимает такие параметры как имя и фамилия, адрес электронной почты и новый пароль пользователя. Если пользователь авторизован, то сервер приступает к обновлению его профиля, обращаясь с запросом на обновление таблицы USERS, заменяя старые данными новыми, извлечёнными из параметров запроса. После чего пользователю сообщается об успешной обработке запроса.

POST-запрос по адресу /getGroupList используется для получения списка групп, в которых состоит на данный момент пользователь. Запрос не имеет входных параметров. Серверное приложение обращается в базу данных, в таблицы GROUPS, GROUP\_MEMBERS, ROLES, извлекая оттуда данные о группах пользователя.

POST-запрос по адресу /getAvailableSubjects возвращает список предметов, опыты по которым доступны для выполнения. Запрос не принимает никаких входных параметров, а ответом на запрос является JSON-массив строк – названий предметов.

POST-запрос по адресу /getLessons принимает один входной параметр – название предмета и используется для получения списка доступных для прохождения опытов по определённому предмету. Ответом пользователю служит массив объектов, описывающих опыты.

POST-запрос по адресу /addResult используется для создания записи о прохождении опыта пользователем. Запрос отсылается после прохождения опыта и содержит такие параметры, как ID урока и время прохождения урока. Сервер оповещает клиента об успешной обработке запроса ответом на этот запрос. Запись создаётся в таблице RESULTS.

POST-запрос по адресу /getMyResults используется для получения списка результатов прохождения опытов пользователем. Входные параметры запроса отсутствуют, однако необходимо существование сессии для его корректной обработки. Результатом обработки запроса является массив, содержащий описание результатов пользователя.

POST-запрос по адресу /createGroup служит для создания групп, в которых пользователи могут объединяться и видеть результаты друг друга. Входные параметры данного запроса – название группы, которую необходимо создать, и её описание. Серверное приложение, обрабатывая запрос, создаёт новую запись в таблице GROUPS, при этом генерируется ID группы, после чего создаётся новая запись в таблице GROUP\_MEMBERS, так как пользователя, отправившего запрос, необходимо назначить создателем группы и наделить его максимальным уровнем доступа к группе. Результатом выполнения запроса является JSON-объект, хранящий ID созданной группы.

POST-запрос по адресу /getInvitations используется в приложении для получения списка приглашений в группу. При обработке запроса, серверное приложение обращается в таблицы INVITATIONS, GROUPS, GROUP\_MEMBERS, откуда извлекает информацию о приглашениях в группы и о самих группы. Запрос возвращает массив JSON-объектов, описывающих приглашения.

POST-запрос по адресу /acceptReject служит для принятия и отклонения приглашений в группу. Входными параметрами являются ID группы, приглашение в которую необходимо обработать, и флаг со значением логического типа, определяющий – принято или отклонено приглашение. В случае принятия приглашения, выполняется вставка соответствующей строки в таблицу GROUP\_MEMBERS, а пользователь, принявший приглашение, наделяется определённой ролью, хранящейся в таблице GROUP\_ROLES. В случае, если приглашение было отклонено, серверному приложению достаточно удалить запись из таблицы INVITATIONS, хранящую приглашение пользователя в определённую группы. Ответом на запрос является сообщение об успешной обработке приглашения.

POST-запрос /getGroupMembers необходим для получения списка пользователей, состоящих в определённой группе. При этом, пользователь, отправивший запрос, должен иметь доступ в группе – состоять в ней и иметь необходимый уровень доступа. Единственным входным параметром является ID группы, информацию о которой хочет получить клиент. Серверное приложение проверяет права доступа пользователя к группе и, если прав достаточно, выгружает необходимую информацию из базы данных – данные каждого пользователя-участника группы.

POST-запрос по адресу /getResults используется для получения информации о результатах выполнения опыта пользователем или участниками группы – в зависимости от входных параметров. Входными параметрами запроса являются ID группы, ID пользователя и название предмета. Выходные данные запроса состоят из JSON-массива, хранящего записи о результатах.

POST-запрос по адресу /editGroup выполняет изменение данных о группе. Пользователь, имеющий достаточный уровень доступа к группе, может отредактировать её данные. Допустимые операции над группой – изменение её имени и изменение описания группы. Серверное приложение проверяет уровень доступа пользователя и выполняет изменение данных о группе, если уровень доступа достаточный. Результатом выполнения служит сообщение об успешной обработке запроса.

POST-запрос по адресу /editMemberRole вызывается для изменения роли и уровня доступа пользователя-участника группы создателем группы. Для успешной обработки запроса, пользователь, который отправляет запрос, должен являться создателем группы. Входные данные запроса содержат ID группы, ID пользователя, название новой роли пользователя. Результатом выполнения запроса является сообщение об успешной его обработке.

POST-запрос по адресу /removeGroup служит для удаления группы и может быть использован только создателем группы. Входным параметром запроса является ID группы. При обработке, удаляются данные из таблиц GROUPS, GROUP\_MEMBERS. Результатом обработки запроса является сообщение об успешном удалении группы.

POST-запрос по адресу /inviteUser используется для приглашения нового пользователя в группу. Пользователь, отсылающий данный запрос, должен обладать необходимым уровнем доступа. Входными параметрами запроса являются ID группы, ID пользователя, а также сообщение, которое будет выслано пользователю вместе с самим приглашением. Если уровень доступа пользователя достаточный, то в таблицу INVITATIONS делается новая запись, описывающая приглашение пользователя в группу. Результатом выполнения запроса является сообщение об успешной его обработке и вставке данных в таблицу INVITATIONS.

POST-запрос по адресу /usersProfile позволяет извлечь данные о пользователях. Для идентификации пользователей, чью информацию необходимо получить, в запрос передаётся в качестве входного параметра массив, содержащий ID пользователей. Серверное приложение обрабатывает запрос, обращаясь в таблицу USERS, и возвращает список профилей пользователей, которые подходят под входные параметры.

POST-запрос по адресу /userList может быть использован для поиска пользователей. Входными параметрами запроса являются имя и фамилия пользователя, а также его логин. Выходным параметром данного запроса является массив, содержащий описания профилей пользователей, которые подошли под описанный фильтр.

POST-запрос по адресу /logOut служит для удаления сессии. Выполнить данный запрос, может только вошедший в систему пользователь. При выполнении запроса, сервер удаляет созданную ранее сессию и «забывает» о пользователе. Входные и выходные параметры в запросе отсутствуют.

Кроме описанных параметров, которые возвращаются каждым из запросом, стоит отметить, что данные всех запросов содержат дополнительное поле, содержащее код ошибки. Если запрос был выполнен корректно, то код ошибки будет равен 0, в противном случае – номер ошибки, которая произошла в ходе выполнения запроса. Коды ошибок описываются в серверном и клиентском приложениях, поэтому пользователь вместо кода ошибки получит сообщение, описывающее суть ошибки.

## **5.4 Реализация клиентского приложения**

Клиентское приложение, разработанное для операционной системы Android, используется пользователем для удобного доступа к опытам, доступных пользователю. Кроме того, пользователь использует клиентское приложение для изменения своего профиля, просмотра результатов, участия в группах.

При запуске приложения, пользователю предлагается войти в систему, указав данные уже существующего аккаунта, или зарегистрировать новый аккаунт (рис. 5.2).

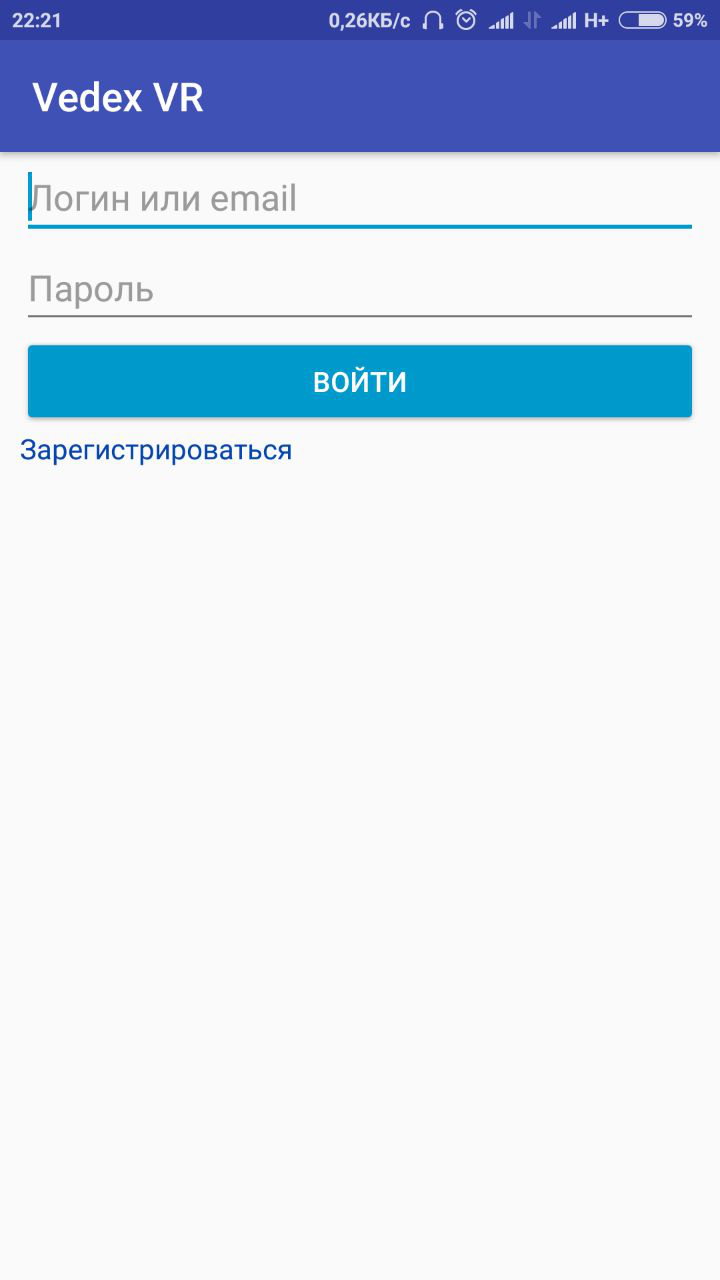


Рисунок 5.2 – Страница входа в аккаунт

На рисунке 5.2 пользователю предлагается ввести данные входа в аккаунт или зарегистрировать новый аккаунт. На странице регистрации пользователя ему предлагается заполнить необходимые поля. Для регистрации нового аккаунта, клиентское приложение отправляет запрос /doRegister на сервер, а после получения подтверждения успешной обработки запроса сервером, выполняет /doLogin, отсылая данные, указанные при регистрации аккаунта. После успешного выполнения входа в аккаунт, открывается главное меню приложения, которое состоит из нескольких разделов (рис. 2.1).

Разделы приложения служат для группировки информации, предоставляемой пользователю, и состоят из:

* Раздел «Мой профиль»
* Раздел «Начать обучение»
* Раздел «Мои результаты»
* Раздел «Группы»
* Раздел «О программе»
* Кнопка «Выход»

Кнопка «Выход» не является разделом, однако она размещена в главном меню для удобства пользователя и позволяет выполнить выход из аккаунта.

Раздел «Мой профиль» содержит информацию о профиле пользователя – его логин, имя и адрес электронной почты. Данные для этого раздела загружаются из памяти приложения.

Главная составляющая приложения – опыты. Для доступа к опытам, доступным пользователю для прохождения, существует раздел «Начать обучение» (рис. 5.3). Перед прохождением опыта пользователь может ознакомиться с краткой информацией о нём – названием и описанием опыта, количеством прохождений другими пользователями.



Рисунок 5.3 – Страница опыта в разделе «Начать обучение»

Раздел «Мои результаты» отображает пользователю результаты прохождения опытов. На данном экране изображён список опытов, пройденных пользователем ранее, и результат прохождения. При нажатии на каждый результат, откроется страница с более подробным описанием опыта. Раздел «Группы» (рис. 5.4) используется для управления группами. Он состоит из списка групп, в которых состоит пользователь. Над списком – две вкладки. Вкладка «Приглашения» отображает количество необработанных приглашений в группы. Число рядом с ней – количество таких приглашений. Также в данном разделе доступен функционал создания новой группы.

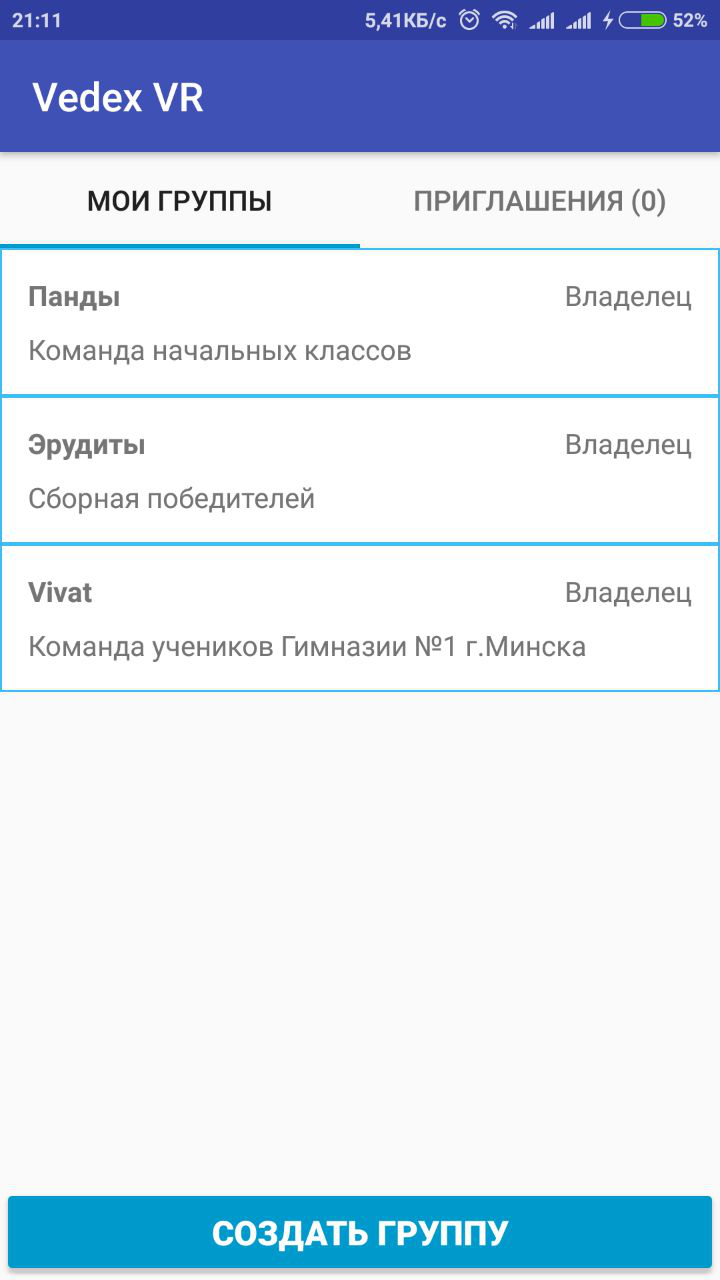


Рисунок 5.4 – Раздел «Группы»

При создании новой группы пользователю необходимо указать название и описание группы. После чего, клиентское приложение выполнит запрос /createGroup, передав введённые данные серверному приложению. В случае успешного выполнения запроса, в списке групп пользователя появится новая группа, а пользователь будет автоматически назначен владельцем такой группы. Каждая группа имеет собственную страницу, где можно увидеть список пользователей, состоящих в ней, а также их результаты.

Функционал групп позволяет пользователем объединяться, взаимодействовать, смотреть результаты друг друга. Кроме того, объединение в группы позволяет классному руководителю и родителям ученика, использующего приложение, контролировать его результаты в режиме онлайн.

Так как опыты занимают достаточно много памяти, они не встроены в приложение изначально. При попытке пройти новый опыт, пользователю будет предложено скачать модуль выбранного опыта из Google Play Market – платформы, на которой размещены приложения. После окончания пользования опытом, его можно удалить с устройства.

## **5.5 Реализация опытов и управление персонажем**

Каждый опыт представляет собой отдельное приложение, загруженное в магазин приложений Google Play Market.

Реализация опытов выполнялась средствами среды разработки Unity 3D. Каждое приложение представлено главной сценой – деревом объектов (рис. 5.5), где каждый объект представляет определённый элемент сцены [7]. Объекты в Unity имеют множество свойств. Основные из них – это координаты объекта, его название и свойства, программные скрипты, управляющие его поведением, а также обработчики событий. Дерево объектов имеет определённую иерархию, в которой объекты-потомки могут перенимать некоторые свойства объекта-предка, в который они вложены. Также перемещения объектов потомков может быть описано относительно двух пространств – глобального и локального. При перемещении по глобальным координатам, смещение объекта указывается в абсолютных величинах – относительно сцены. Если же перемещение выполняется по локальным координатам, смещение объекта указывается относительно его предка.

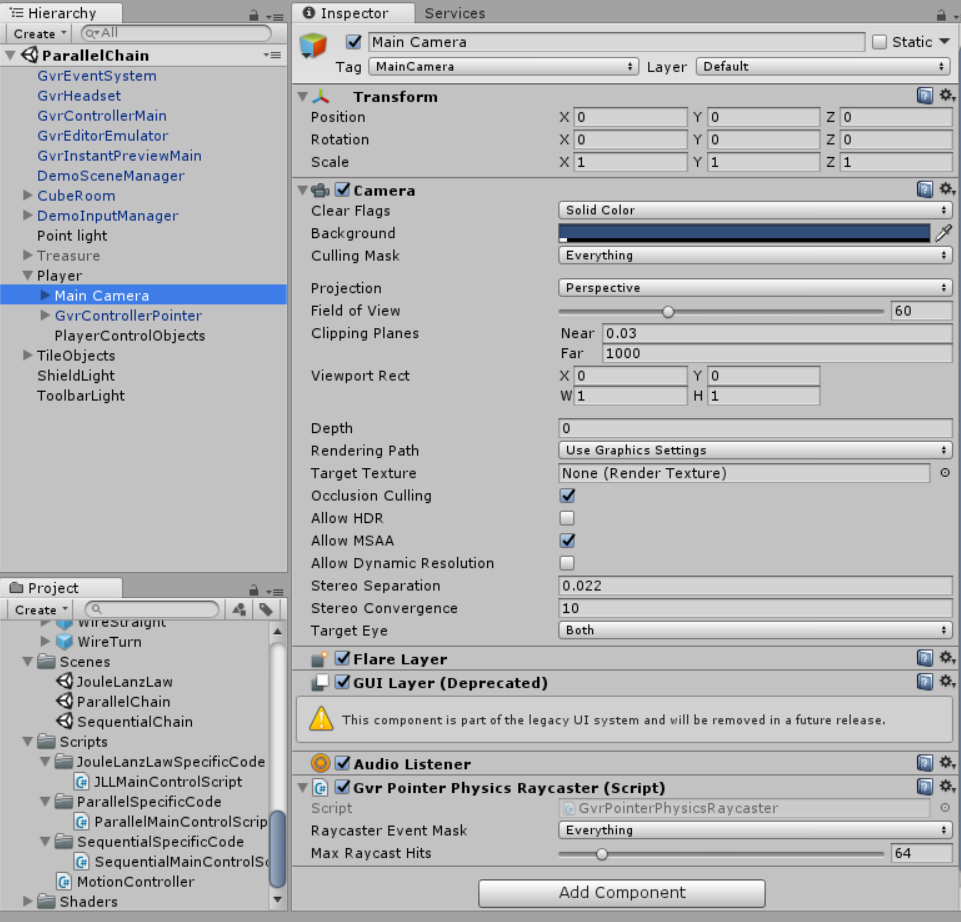


Рисунок 5.5 – Дерево сцены и свойства объектов

На рисунке 5.5 изображена структура сцены – дерево объектов. Справа, в области «Inspector» находятся свойства определённого объекта, выделенного на дереве слева. Обратим внимание, что сама сцена является корнем дерева и предком всех объектов.

За вид сцены для пользователя отвечает объект MainCamera. К нему также прикреплены скрипты, описывающие перемещение пользователя по сцене и взаимодействие с некоторыми объектами.

Перемещение пользователя выполняется с помощью скрипта, перехватывающего сообщения джойстика и соответствующим образом перемещающего игрока по игровой сцене.

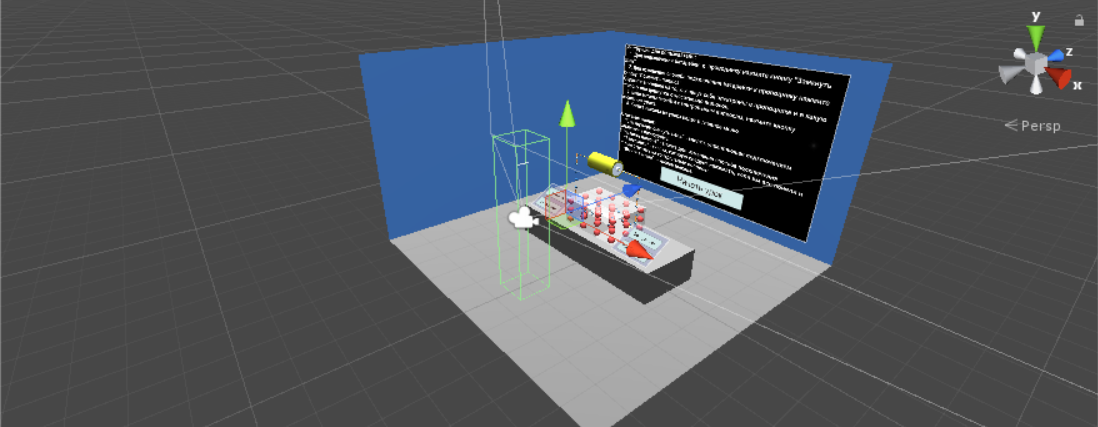


Рисунок 5.6 - Объект персонажа и камера

На рисунке 5.6 можно наблюдать, как выглядит объект персонажа и камера, прикрепленная к нему в опыте с движением электронов в цепи.

В качестве стандартной игровой физики была взята гравитация, материалы объектов, движение объектов.

Так как приложение реализовывает практические опыты по различным предметам, поведение некоторых предметов в демонстрационных целях приходилось изменять, отходя от стандартной физики объектов Unity. Поведение таких специфических объектов описывалось дополнительными скриптами, соответствующе действиям игрока. Так, например, при замыкании электрической цепи, в проводнике начинали двигаться электроны в определённом направлении вместо хаотичного. Более того, само по себе хаотичное перемещение электронов не может быть реализовано стандартными средствами, для чего так же был написан скрипт. Хаотичность перемещения таких объектов обеспечивалась тем, что раз в секунду скорость и направление движения сменялось на новые случайные значения.

Персонаж имеет определённый видимый объект, соответствующий ему. Для игрока персонаж представлен камерой с видом от первого лица. Вращение камерой осуществляется с помощью поворота головы игрока в реальном мире (акселерометр посылает соответствующие сигналы системе). Перемещение игрока осуществляется с помощью джойстика, подключаемого к мобильному устройству с помощью Bluetooth.

Взаимодействие пользователя с объектами сцены осуществляется с помощью компоненты EventTrigger, прикреплённой к объектам, взаимодействие с которыми необходимо обработать. EventTrigger перехватывает события и отсылает их на обработку функциям, указанным пользователем. Основным событием, обработку которого необходимо осуществлять для взаимодействия, является нажатие указателем на объект. Обработка данного события перенаправляется на соответствующие функции, обеспечивая таким образом взаимодействие с объектами. В виртуальной реальности для имитирования нажатия на объект необходимо навести указатель на него и нажать на кнопку очков.

# **ГЛАВА 6. РЕЗУЛЬТАТЫ РАЗРАБОТКИ**

Результатом разработки проекта являются скомпилированные установочные пакеты с расширением apk, составляющие приложение. Один из файлов – клиентское приложение, остальные – скомпилированные сцены опытов. Для удобства доступа к приложению и опытам, они были опубликованы в магазине мобильных приложений для Android – Google Play Market [8]. Публикация в данном магазине подразумевает не только загрузку установочного файла, но и создание страницы приложения. Для этого необходимо использовать аккаунт разработчика Google.

Скачивание и установка приложения осуществляются с мобильного устройства. Пользователь входит на страницу приложения и запускает его загрузку.

При входе с мобильного устройства, функционал приложения позволяет скачать и установить его на устройство пользователя. Пользователю предоставлена возможность оценить интерфейс и управление персонажем.

На сервисе Microsoft Azure [9] запущена база данных MS SQL [10]. Для обеспечения взаимодействия пользователя с базой данных запущен веб-сервер, обеспечивающий обработку пользовательских запросов.

Образовательный продукт был протестирован на курсе «Физика, 8 класс» в СОШ №22 г. Борисова с привлечением к участию учеников 8А класса и директора школы Кушнярова Петра Владимировича. Было проведено две лабораторных работы с использованием приложения по темам «Построение электрической цепи с последовательным соединением проводников» и «Построение электрической цепи с параллельным соединением проводников».

# **ЗАКЛЮЧЕНИЕ**

В процессе выполнения курсового проекта были получены следующие результаты:

* Изучена среда разработки для опытов в VR Unity 3D;
* Изучены база данных MS SQL и серверный фреймворк Node.js;
* Изучена среда разработки Android приложений Android Studio.

Проект был представлен на студенческой конференции, на международном фестивале образовательных проектов «STARTUP EDUTECH FEST», где вошел в число победителей и получил грант в десятом сезоне конкурса социальных проектов Беларуси «Social Weekend», что позволило осуществить закупку очков виртуальной реальности и провести апробацию приложения в реальных условиях в школе. Также проект вышел в суперфинал конкурса «Паскарэнне», стал победителем международного финала олимпиады «ИТ-планета».

В будущем планируется:

- Выполнить качественную озвучку опытов;

- Принять участие в конкурсе инновационных проектов БГУ, а также в конкурсе «100 идей для Беларуси»;

* Провести тестирование приложения в средней школе №208 г. Минска;
* Разработать новые опыты: «Изображение в линзах», «Сечения многогранников», а также опыты по другим предметам, в том числе по химии и астрономии.

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Рынок виртуальной реальности-2018 в одной схеме // The Venture Reality Fund [Электронный ресурс]. — 2018. Режим доступа: <http://www.thevrfund.com>
2. Движение в VR // Unity [Электронный ресурс].— 2017. Режим доступа:

<https://unity3d.com/ru/learn/tutorials/topics/virtual-reality/movement-vr?playlist=22946>

1. Android API // Google [Электронный ресурс]. — 2017. Режим доступа: <https://developer.android.com/reference>
2. Android – Программирование для профессионалов / Б. Харди [и др.]; под общ. ред. Б. Харди — 2-е изд. — СПб.: Питер, 2016. — 640 с.: ил. — (Серия «Для профессионалов»)
3. Google VR в Unity // Google [Электронный ресурс].— 2016. Режим

доступа: https://developers.google.com/vr/unity/get-started

1. Node.js Introduction // w3schools [Электронный ресурс]. — 2017. Режим доступа: <https://www.w3schools.com/nodejs/nodejs_intro.asp>
2. Документация Unity // Unity [Электронный ресурс].— 2016. Режим

доступа: https://docs.unity3d.com/Manual/VROverview.html

1. Инструкция по публикации Android-приложения в Google Play // Habrahabr [Электронный ресурс]. – 2018. Режим доступа: <https://habr.com/company/livetyping/blog/326874/>
2. Getting started with Microsoft Azure // Microsoft [Электронный ресурс]. – 2017. Режим доступа: <https://docs.microsoft.com/en-us/azure/>
3. MS SQL Server Tutorial // tutorialspoint [Электронный ресурс]. — 2017. Режим доступа: <https://www.tutorialspoint.com/ms_sql_server/index.htm>