**Реферат**

СОДЕРЖАНИЕ

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 6](#_Toc135050515)

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 8](#_Toc135050516)

[ВВЕДЕНИЕ 9](#_Toc135050517)

[1. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ 11](#_Toc135050518)

[1.1. Описание предметной области 11](#_Toc135050519)

[1.2. Обзор существующих решений 11](#_Toc135050520)

[1.3. Возможные пути решения задач их достоинства недостатки обоснование выбора 17](#_Toc135050521)

[1.4. Анализ применяемых технологий и инструментов 18](#_Toc135050522)

[1.5. Требования к прототипу решения 24](#_Toc135050523)

[1.5.1. Требования к функциональным характеристикам 24](#_Toc135050524)

[1.5.2. Требования к составу и параметрам технических средств 25](#_Toc135050525)

[1.5.3. Условия эксплуатации 25](#_Toc135050526)

[1.5.4. Требования к информационной и программной совместимости 25](#_Toc135050527)

[2. РАЗРАБОТКА ПРЕДПОЛАГАЕМОГО РЕШЕНИЯ 27](#_Toc135050528)

[2.1. Разработка архитектуры прототипа решения 27](#_Toc135050529)

[2.1.1. Архитектура системы 27](#_Toc135050530)

[2.1.2. Диаграмма последовательности 27](#_Toc135050531)

[2.1.3. Диаграмма компонентов 27](#_Toc135050532)

[2.1.4. Структура сцены 28](#_Toc135050533)

[2.1.5. Входные выходные данные 32](#_Toc135050534)

[3. РЕАЛИЗАЦИЯ ПРЕДПОЛАГАЕМОГО РЕШЕНИЯ 33](#_Toc135050535)

[3.1. Моделирование 33](#_Toc135050536)

[3.2. Разработка проекта 34](#_Toc135050537)

[3.2.1. Создание комнат 34](#_Toc135050538)

[3.2.2. Настройка клиент-сервера 36](#_Toc135050539)

[3.2.3. Создание логики приложения 37](#_Toc135050540)

[3.2.4. Сборка проекта 42](#_Toc135050541)

[3.3. Проблемы и пути их решения 43](#_Toc135050542)

[3.4. Экспериментальна проверка 44](#_Toc135050543)

[ЗАКЛЮЧЕНИЕ 45](#_Toc135050544)

[СПИСКОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 46](#_Toc135050545)

[ПРИЛОЖЕНИЕ А 51](#_Toc135050546)

[ПРИЛОЖЕНИЕ Б 52](#_Toc135050547)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Asset – Цифровой объект, который включает в себя модель, анимацию или программный код

FBX – технология и формат файлов разработанный Kaydara

Lobby (лобби) - внутриигровая комната ожидания, внутри которой производятся настройки будущего сеанса подключения

MeshRenderer – компонент, отвечающий за отрисовку вершин и полигонов для объекта

Peer-to-peer – Одноранговые, децентрализованные сети — это компьютерные сети, основанные на равноправии участников. В таких сетях отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и сервером

Prefab (префаб) — это шаблон для объекта в игровом движке Unity

Spawn (Спавн) – создание какого-либо объекта в сцене во время работы приложения

UV-развёртка — Это соответствие между координатами на поверхности трёхмерного объекта (X, Y, Z) и координатами на текстуре (U, V)

Аватар - текстовые, двухмерные или трехмерные графические модели, при помощи которых пользователи взаимодействуют друг с другом внутри виртуальной реальности

Виртуальная реальность – это разновидность человеко-компьютерного интерфейса, обеспечивающего присутствие человека в моделируемой среде посредством поддержки его погружения и взаимодействия с объектами этой среды в реальном времени

Клиент-сервер - вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами

Метавселенная - представляет собой крупномасштабную и интероперабельную сеть трехмерных виртуальных миров, визуализируемых в реальном времени, которые могут синхронно и постоянно восприниматься фактически неограниченным числом пользователей с индивидуальным ощущением присутствия и с непрерывностью данных, таких как личность, история, права, объекты, коммуникации и платежи

Мультиплеер – режим игры, при котором в одну игру играет несколько человек

Мультиюзер - способность компьютера поддерживать одновременную работу многих пользователей

Погружение – свойство моделируемой среды, вовлекающее человека через его сенсомоторные средства в поведение среды и снабжающее его непрерывным потоком стимулов, сопоставимым с естественным

Присутствие – человек, испытывающий данное чувство, переживает виртуальную среду, происходящие с ним там события и свое нахождение в ней, как нечто реальное, а не созданное с помощью технических средств

Якорь – пустой объект в сцене, к которому во время работы приложения будет прикреплен другой объект

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

VR - Virtual Reality

ВР – Виртуальная реальность

P2P – Peer-To-Peer

# ВВЕДЕНИЕ

В современном мире потребность людей в различном цифровом интерактивном контенте приобретает все большее значение. В связи с этим растет развитие разнообразия производства вычислительной техники, что привело к появлению различных устройств виртуальной реальности. На сегодняшний день технологии виртуальной реальности применяются в различных сферах жизнедеятельности людей в обществе.

Согласно Указу Президента Российской Федерации от 09.05.2017 г. № 203[1], а так же распоряжению правительства от 21.12.2021 № 3759-р[2], обеспечение национальных интересов при развитии информационного общества осуществляется путем развития информационной и коммуникационной инфраструктуры РФ, создание и применение российских информационных и коммуникационных технологий и обеспечение их конкурентоспособности на международном уровне. Одним из приоритетов цифровой трансформации является увеличение доли отечественного оборудования и информационных систем, используемых в образовательных организациях высшего образования и научных организациях. В целях формирования информационного пространства знаний студентов необходимо усовершенствовать систему обмена знаний, использовать и развивать различные дистанционные образовательные технологии с целью обеспечить насыщение рынка доступными, качественными и легальными медиапродуктами и сервисами российского производства.

В качестве примера использования VR технологий в образовании можно привести работу с химикатами в виртуальной реальности на уроках химии в школах. Плюсами такого подхода является безопасность обучаемого и отсутствие необходимости закупки новых химикатов.

Пандемия 2020 года так же внесла свои коррективы в образовательный процесс. Часть образовательных организаций была вынуждена перейти на удаленный формат обучения. Однако не по всем дисциплинам можно проводить занятия в подобном формате. Например, лабораторные работы по предмету «Цифровая схемотехника» требуют наличия необходимого профессионального оборудования, которого у обучающихся дома, как правило, нет.

При использовании ресурсов для удаленного обучения, преподавателю трудно контролировать вовлеченность студентов на дистанционных занятиях. Следовательно, на сегодняшний день общество нуждается в таком программном обеспечении, которое позволит не только обеспечить вербальное общение на дистанционном занятии, но и создаст эффект реального присутствия в аудитории. Преподаватель и обучающиеся должны иметь возможность видеть аватары друг друга, а также взаимодействовать с интерактивными объектами внутри виртуальной сцены.

Целью работы является разработка многопользовательского приложения виртуальной реальности с голосовым чатом и аватарами для проведение дистанционных занятий.

Для достижения цели должны быть решены следующие задачи:

1) изучение способов, инструментов, технологий моделирования в программной среде Blender;

2) изучение способов и инструментов для создания многопользовательского приложения;

3) изучение инструментов для создания голосового чата;

4) изучение видов аватаров в различных многопользовательских приложениях;

5) составление сравнительного анализа игровых движков, подходящих для создания многопользовательского VR-приложения;

6) выработка технических требований к разрабатываемой программе;

6) разработка архитектуры системы;

7) создание VR-проекта с мультюзером;

8) реализация голосового общения в мультиюзере;

9) проведение тестирования разработанного приложения;

# 1. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

## 1.1. Описание предметной области

Виртуальная реальность – это разновидность человеко-компьютерного интерфейса, обеспечивающего присутствие человека в моделируемой среде посредством поддержки его погружения и взаимодействия с объектами этой среды в реальном времени.[3]

Одной из характеристик VR оборудования являются степени свободы. Шесть степеней свободы — это движение шлема по осям X, Y, Z, XZ, XY, ZY. Поэтому если в руках есть контроллеры или трекеры, можно перемещаться в любую точку пространства виртуального и при этом видеть самого себя в виде аватара. Шесть степеней свободы — это стандарт для современного VR.[4]

## 1.2. Обзор существующих решений

All-In-One Sports VR

Многопользовательская VR игра от корейской компании «Appnori Inc». В данной игре несколько игроков могут соревноваться друг с другом в одном из 10 видов спорта (см. рисунок 1).[5]



1. – Многопользовательский волейбол

Не смотря на возможность многопользовательского использования данное приложения является развлекательным и не предназначено для образовательных целей.

Реализуемое приложение будет иметь возможность проведения лекционных, лабораторных занятий в специально отведенных для этого виртуальных комнатах.

VRChat

Разработанный VRChat Inc., VR-аналог Skype/Discord и подобных приложений для связи. Связь осуществляется при помощи очков виртуальной реальности. Пользователи взаимодействую друг с другом при помощи аватаров, внутри виртуальных комнат.

VR-chat предоставляет огромное разнообразие виртуальных комнат для общения. Комнаты обставлены различными декорациями, начиная от лоснящегося от официоза зала и заканчивая космической станцией где-то на задворках вселенной. Пользователи так же могут добавлять собственные комнаты.[6]

Аватары используют анимированные конечности, отслеживание движения глаз и даже синхронизацию губ. Аватары подвержены пользовательской кастомизации, так же пользователи могут создавать и добавлять собственные аватары, что позволяет пользователям создавать уникальные собственные аватары (см. рисунок 2-3).



1. – Комната vr-chat



1. – Аватары пользователей

Однако VRChat имеет ряд недостатков, не позволяющих использовать его в качестве основной платформы для дистанционного обучения: VRChat создавался как развлекательная платформа позволяющая людям самовыражаться, в следствии чего данное приложение предоставляет обширные возможности для модификации, кастомизации каждым пользователем. Подобные кастомизации и модификации со стороны студентов отрицательно скажутся на восприятии ими учебного материала. Так же данные модификации могут и вовсе испортить сцены учебного процесса.

Реализуемое приложение будет иметь простые комнаты и аватары, не подвергаемые пользовательским модификациям.

Meta Horizon Worlds

Часть meta-вселенной компании «Facebook». Данное приложение является VR-социальной сетью, поддерживающей связь со всеми продуктами компании.

Пользователи взаимодействуют при помощи настраиваемых аватаров. Причем данные аватары могут использоваться во всех VR приложениях компании.

Meta Horizon Worlds позволяет создавать и настраивать собственные комнаты для общения, а так же присоединяться к уже существующим (см. рисунок 4).



1. – Общение при помощи аватаров в meta horizon worlds

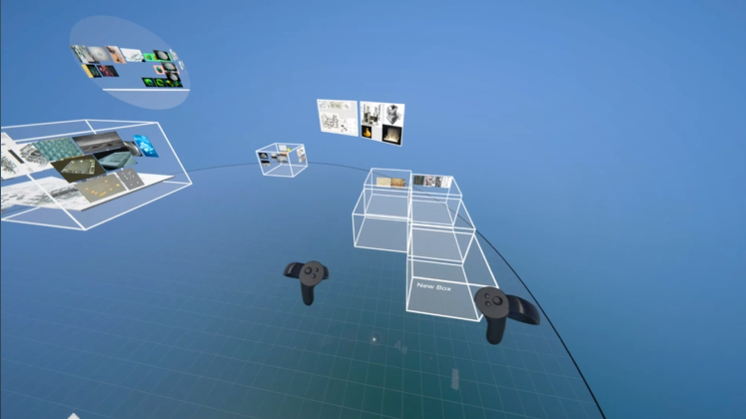
Компания «Facebook» ведет мониторинг происходящего в Meta Horizon Worlds, что означает фильтрацию нежелательного контента. Пользователи нарушающие правила использования сервиса (ведущие не допустимое поведение, создающие не допустимые/оскорбительные материалы), будут подвергаться санкциям со стороны компании. А нежелательный контент будет удален. Таким образом Horizon Worlds создает безопасную для общения среду.[7]

Основным недостатком Meta Horizon Worlds является его главное достоинство: компания «Facebook». Именно ее сервисы и политика используются для взаимодействия с данным приложением. И даже если нас устраивает правила использования сервиса, компания «Facebook» признана экстремисткой на территории Российской Федерации, а также сама компания отказывается сотрудничать с гражданами Российской Федерации.

Реализуемое приложение будет доступно на территории Российской федерации. Сторонние компании не будут иметь доступ к работе реализуемого приложения.

Softspace

Softspace это приложение от российских разработчиков от «Softspace inc». Softspace представляет из себя комнаты в дополненной реальности для дизайнеров и исследователей (см. рисунок 5-6).[8]



1. – Создание окружения



1. – Аватары пользователей

Softspace все еще находиться в разработке. Приложение создается в основном для дизайна, что может помешать интегрированию в него более комплексных механик.

Реализуемое приложение будет иметь фиксированные комнаты, не подверженные модификациям пользователей во время работы в этих комнатах.

Часть подобных приложений в данный момент недоступна на территории Российской Федерации. Значит, при выборе стороннего приложения как платформы для создания образовательного процесса в VR, существует риск лишиться всех наработок из-за отказов стороннего приложения. Так же при использовании стороннего приложения, особенно не предназначенного для предоставления образовательных услуг, может не оказаться возможным интегрировать собранную образовательную платформу с другими платформами образовательного учреждения. Если мы не готовы рисковать внезапной потерей приложения и хотим в дальнейшем интегрировать его в информационную среду образовательного учреждения, то необходимо разработать собственный независимые аналог.[9]

## 1.3. Возможные пути решения задач их достоинства недостатки обоснование выбора

Для создания сеанса связи возможно использовать технологии P2P и клиент-сервер. [10]

При клиент-серверной архитектуре, имеется один (или несколько) сервер(-ов), на котором(-ых) хранятся все данные приложения и происходят все вычислительные процессы. Клиент – сто стороннее устройство, предоставляющее пользователю доступ к удаленному серверу.

При P2P архитектуре нет центрального сервера, управляющего сетью. Напротив, все компьютеры в сети подключены друг к другу и делятся ресурсами, такими, как файлы, приложения и программы. В одноранговой сети каждый компьютер может быть клиентом или сервером, так как он может запрашивать или предлагать услуги.

1. Преимущества и недостатки технологий для создания сеанса связи

|  |  |  |
| --- | --- | --- |
| Технология | P2P | Клиент-сервер |
| Преимущества | Отсутствие расходов на центральный сервер;  Легкость масштабирования – новые клиенты просто подключаются в уже существующую сеть;  Если одно устройство сети выйдет из строя – остальные продолжат работу; | Централизованное управление облегчает синхронизацию данных;  Данные, физически хранимые в одном месте, легче защитить;  Обновление серверной части приложения не требует остановки работ клиентов. |
| Недостатки | Сложность защиты данных. На каждом устройстве необходимо обеспечить защиту данных этого устройства;  Производительность и услуги в рамках всей системы могут быть медленнее, поскольку каждый компьютер выполняет более одной задачи и может быть доступен другим компьютерам. | Расходы на закупку, содержание серверов или аренду серверов;  При отказе центрального сервера, все клиенты данного сервера прекращают свою работу;  Сложность масштабирования системы. Чем больше клиентов, тем больше нагрузка на сервер, а значит необходимо искать решения для распределения этой нагрузки. |

Для реализации приложения была выбрана клиент-серверная архитектура. Так как она облегчает синхронизацию, контроль и защиту общих данных.

## 1.4. Анализ применяемых технологий и инструментов

Для разработки приложения потребуются: игровой движок (см. таблица 0-3), библиотека для сетевой синхронизации, приложение для создания 3d моделей, шлем виртуальной реальности.

Выбор игрового движка

1. Сравнение игровых движков часть 1.

|  |  |  |
| --- | --- | --- |
| Движок | Unity | Unreal Engine |
| Язык программирования | С#, UnityScript | C++, Blueprint |
| Цена | Условно бесплатный. При использовании движка в коммерческих целях, он будет бесплатным, пока оборот компании не превышает 100000$/год. После этого необходимо оформить подписку 125$/месяц или купить бессрочную версию за 3000$ | Условно бесплатный. При использовании движка в коммерческих целях, он будет бесплатным, пока созданная вами игра не принесёт доход суммой более одного миллиона долларов. После этого выплаты составят 5% от прибыли. |
| Платформы | Windows, Mac, Linux, Android, IOS, tvOS, Xbox, PS4, WebG | Windows, Linux, Mac, Xbox, PS, Android, IOS, Nintendo Switch |
| Минимальные системные требования | Процессоры Intel Core® i5-2500K @ 3.3 GHz или AMD FX-8350 @ 4.0 GHz;  RAM 6Гб;  Не менее 50 ГБ свободного места;  NVIDIA GeForce GTX 680 или AMD Radeon HD 7970 | Процессоры Quad-core Intel или AMD, 2.5 GHz;  RAM 8Гб;  Не менее 256 ГБ свободного места;  NVIDIA GeForce GTX 970 |
| Ассеты | Магазин Unity Assets Store | Магазин Unreal Engine Marketplace |
| Поддерживаемые форматы 3d моделей | fbx, .dae (Collada), .3ds, .dxf , .obj | .obj и .fbx |
| Поддержка VR технологий | OpenXR | OpenXR |

1. Сравнение игровых движков часть 2.

|  |  |  |
| --- | --- | --- |
| Движок | Unigine | Godot |
| Язык программирования | C++, C#, UnigineSckipt | GDScript |
| Цена | Бесплатная community версия с ограничением на разработку оборонной промышленности,  азартных игр,  электроэнергетики,  добычу нефти, газа и полезных ископаемых. | Полностью бесплатен |
| Платформы | [Windows](https://ru.wikipedia.org/wiki/Windows), Linux, Mac | Windows, macOS, X11 (Linux, \*BSD), Web, Android (экспериментальный) |
| Минимальные системные требования | Процессоры x64 с поддержкой SSE4.2, начиная с четырехъядерных процессоров Intel или AMD с частотой 2,5+ ГГц;  RAM 4Гб;  Не менее 50 ГБ свободного места;  NVIDIA GeForce GTX 1080 | Процессоры Intel или AMD (Dual Core);  RAM 4Гб;  Nvidia GeForce 8 серии или AMD Radeon HD 7000  500 Мб свободного места |
| Ассеты | Магазин Unigine Asset Store | Магазин Godot Asset Library |
| Поддерживаемые форматы 3d моделей | fbx | .obj и .dae |
| Поддержка VR технологий | OpenVR | OpenVR; OpenXR |

Для данного проекта был выбран игровой движок Unity, из-за его низких системных требований и простоты освоения.

Выбор библиотеки для сетевой реализации

Между Photon, Mirror и Unity multiplayer Services, был выбран Photon из-за его преимуществ над остальными вариантами.

В сравнении с Unity Multiplayer Services, Photon легче подключается к Unity, проще настраивается сетевое соединение, Photon имеет бесплатную версию с ограничением в 500 сообщений в секунду, Photon из коробки предлагает функцию миграции хоста.

В сравнении с Mirror, Photon легче настраивается, Mirror не имеет функции миграции хоста. Несмотря на то, что Mirror бесплатна и не имеет ограничений на использование, сложность его освоения и малое количество обучающих материалов заставляют сделать выбор в пользу Photon.

Так же Photon имеет встроенный модуль Photon Voice предоставляющий возможность реализации голосового чата.

Выбор шлема виртуальной реальности

1. Сравнение шлемов виртуальной реальности.[11][12][13]

|  |  |  |  |
| --- | --- | --- | --- |
| Шлем | Oculus Quest 2 | HTC Vive Pro | Pico 4 |
| Цена в рублях | 40000 | 120000 | 35000 |
| Разрешение экрана | 3664х1920 | 2880х1600 | 4320х2160 |
| Дисплей | LCD | AMOLED 3,5 дюйма | LCD |
| Угол обзора в градусах | 90 | 110 | 105 |
| Оперативная память, Гб | 6 | 4 | 8 |
| Вес, г | 470 | 520 | 295 |
| Время работы аккумулятора в часах | 2 | 2,5 | 3 |
| Отслеживание рук | + | - | + |
| Дата выпуска | 13.08.2020 | 05.04.2018 | 18.08.2022 |

Для выполнения данного проекта был выбран шлем Oculus Quest 2. Т.к. он дешевле HTC Vive Pro. Он существует на рынке дольше чем Pico 4 из-за чего более стабилен в работе и более изучен для разработки приложений под данное устройство.

Выбор типа аватара

Аватары бывают минималистичные, с половиной тела, в полный рост.[14]

Минималистичные – такой аватар содержит только голову (или объект, имитирующий ее) и руки/контроллеры, парящие в воздухе. Просты в реализации, но мало ассоциируются с гуманоидами, что влияет на погружение (см. рисунок 7).



1. – Минималистичные аватары

Аватары с половиной тела. Такие аватары не имеют нижней части тела, но голова и руки соединены туловищем. Такие аватары сложнее в реализации, чем минималистичные, но вызывают большие ассоциации с гуманоидами (см. рисунок 8).



1. – Аватары с половиной тела

Аватары в полный рост. Такие аватары учитывают так же положение ног пользователя (см. рисунок 9).



1. – Аватар в полный рост

Для разрабатываемого приложения выбраны аватары с половиной тела, т.к. они обеспечивают большее погружение чем минималистичные и не требуют полного отслеживания тела пользователя, что избыточно для разрабатываемого приложения.

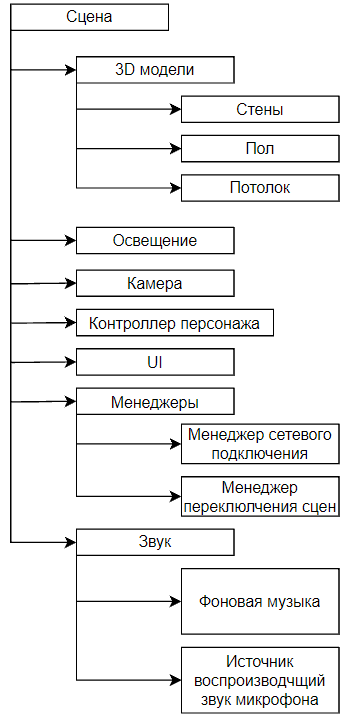
Готовые аватары можно создать при помощи системы readyplayer.me.[15]

В качестве приложение для создания 3d моделей был выбран Blender, так как это самое популярное бесплатное приложение для создания 3d моделей.

## 1.5. Требования к прототипу решения

#### 1.5.1. Требования к функциональным характеристикам

##### 1.5.1.1. Оснащение сцены



1. – Оснащение сцены

Сцена должна содержать пол, потолок и стены, для ориентации пользователя в пространстве сцены. Должен присутствовать источник естественного света.

В сцене должна быть камера, имеющая 6 степеней свободы.

В сцене должен находиться контроллер, с помощью которого пользователь будет контролировать свое перемещение внутри сцены.

Сцена может содержать UI, а значит в ней должны быть элементы позволяющие использовать VR UI.

В сцене должны находиться компоненты, отвечающие за сетевое взаимодействие и переключения между различными сценами.

В сцене могут присутствовать источники звука, как источники фоновой музыки, так и источник голоса игрока.

##### 1.5.1.2. Требования к аватарам, жестам и микрофону

Аватары должны иметь голову, контроллеры и руки.

У пользователя должна быть возможность использовать микрофон.

Система должна иметь возможность распознавать жесты. Некоторые интерактивные предметы должны иметь возможность взаимодействия с ними при помощи определенных жестов.

#### 1.5.2. Требования к составу и параметрам технических средств

Шлем виртуальной реальности Oculus Quest 2, с включенным отслеживанием рук и подключением к интернету;

Контроллеры Oculus Touch;

Библиотека для сетевой синхронизации Photon

#### 1.5.3. Условия эксплуатации

В помещении не должно быть объектов, препятствующих перемещению пользователя.

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Для работы в приложении устройство VR должно иметь подключение к интернету.

#### 1.5.4. Требования к информационной и программной совместимости

Используемые программы должны быть представлены лицензионной версией ОС не ниже 29 версии Android.

Версия Oculus Quest 2 должна быть не ниже 43-й.

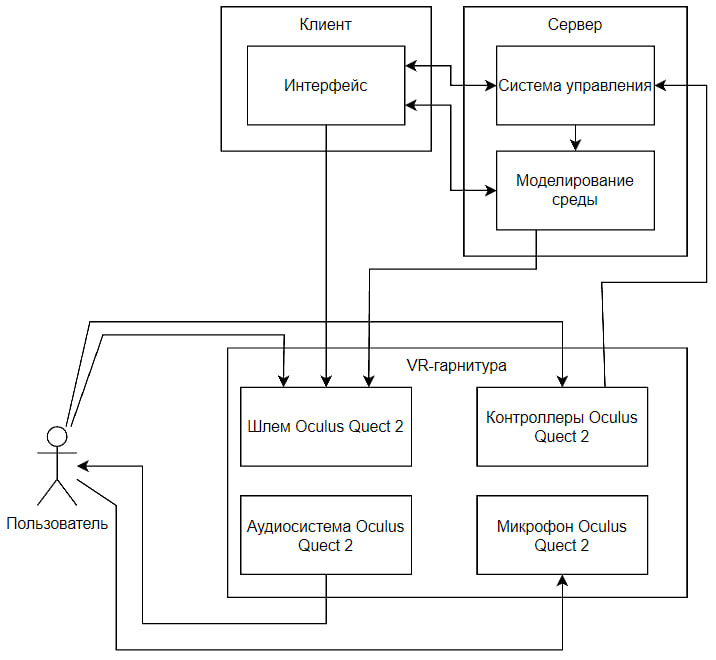
Все технические решения, использованные при создании системы должны соответствовать действующим требованиям по защите от несанкционированного доступа и защите персональных данных.

# 2. РАЗРАБОТКА ПРЕДПОЛАГАЕМОГО РЕШЕНИЯ

## 2.1. Разработка архитектуры прототипа решения

#### 2.1.1. Архитектура системы

Диаграмма архитектуры системы рассматривается на примере одного пользователя. Диаграмма системы представлена на рисунке 11.



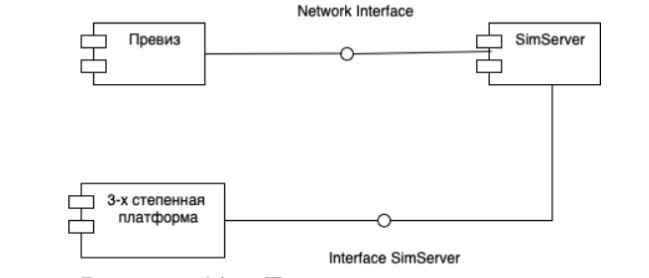
1. – Архитектура системы

#### 2.1.2. Диаграмма последовательности

Пользователь подключение к серверу, подключение к комнате, передача данных, покинуть комнату,

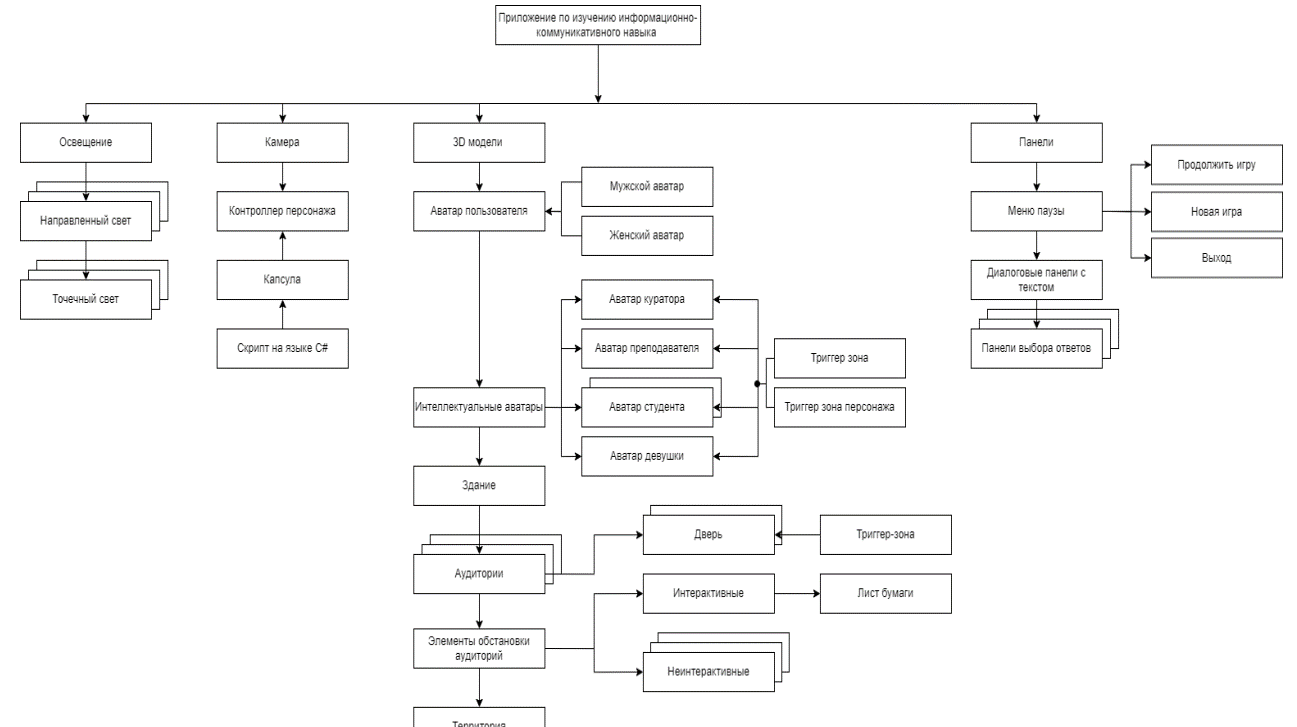
#### 2.1.3. Диаграмма компонентов

Пример:



#### 2.1.4. Структура сцены

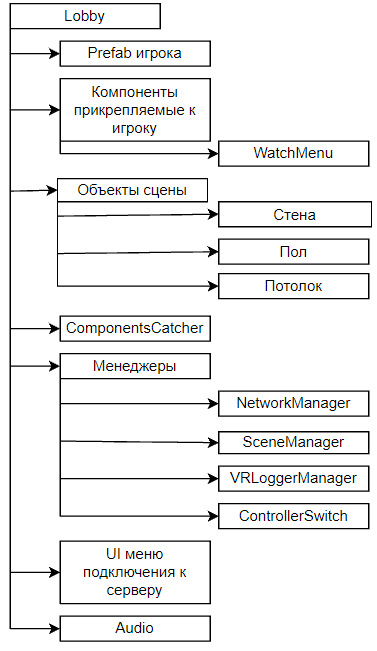
*Пример:*



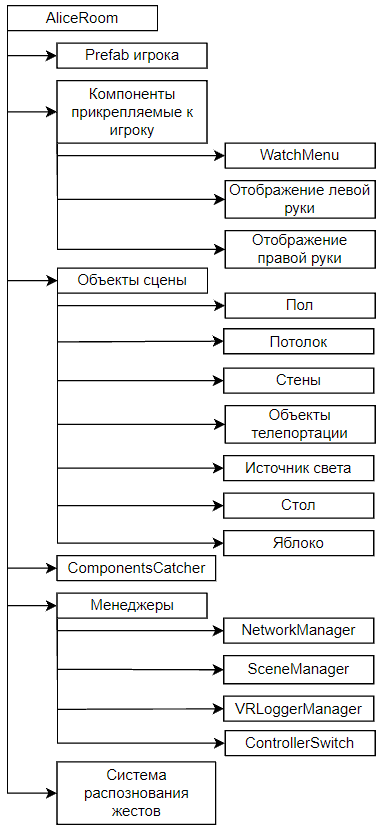
*Мои наработки*



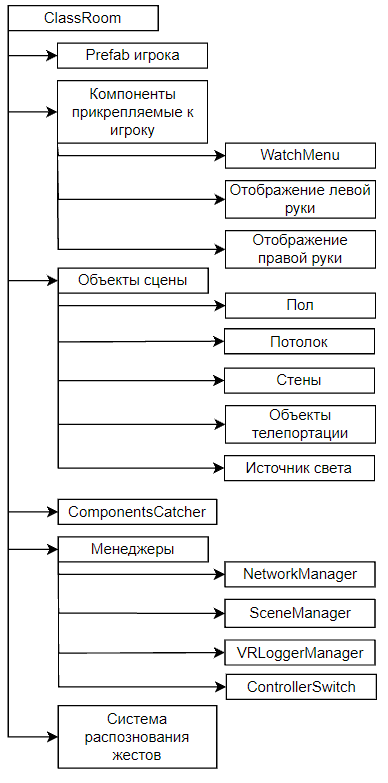
1. – Общая структура сцен



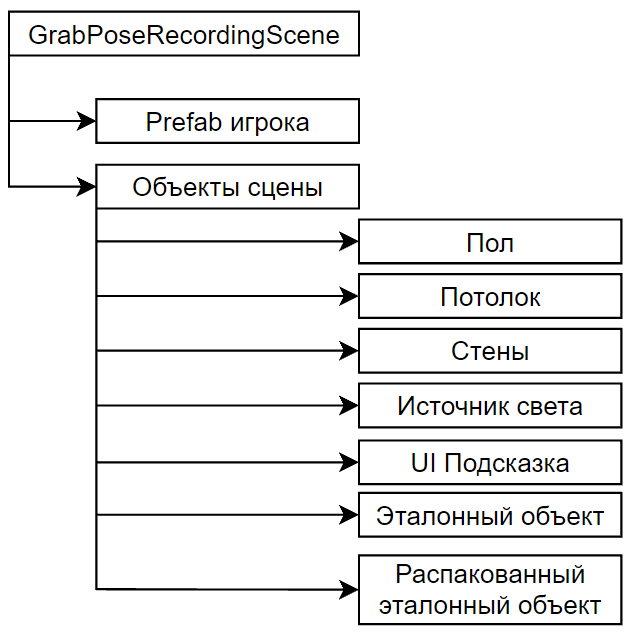
1. - Структура Lobby



1. - Структура AliceRoom



1. - Структура ClassRoom



1. - Структура GrabPoseRecordingScene

#### 2.1.5. Входные выходные данные

Входные данные:

Положение контроллеров, рук, головы;

Нажатие кнопок на контроллерах;

Складывание рук в жесты;

Голос.

Выходные данные:

Изображение;

Звук.

# 3. РЕАЛИЗАЦИЯ ПРЕДПОЛАГАЕМОГО РЕШЕНИЯ

## 3.1. Моделирование

откуда я что взял и как подключил в сцену

Модель стола взята из Unity Asset Store: Table with chairs x3 Free



1. - Модель стола

Модель стола взята из Unity Asset Store: Food and Kitchen Props Pack



1. - Модель яблока

## 3.2. Разработка проекта

#### 3.2.1. Создание комнат

Первая сцена представляет из себя лобби. В данной сцене игрок подключается к серверу, выбирает тип комнаты и комнату, к которой хочет присоединиться или создает новую комнату.

В данной сцене помимо игрока находятся:

UI с которым игрок взаимодействует для подключения к серверу, выбора типа комнаты, подключения к комнате, создания новой комнаты;

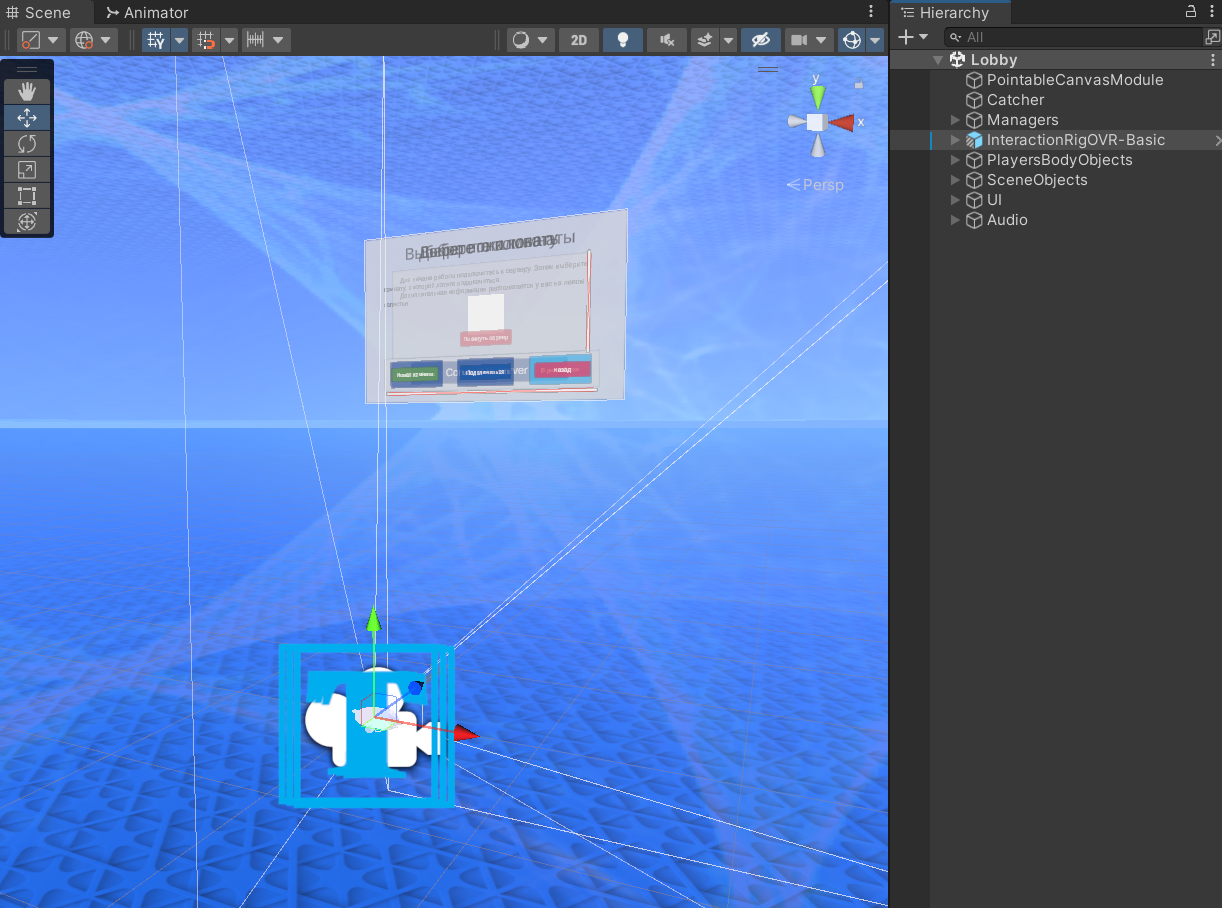
Предметы окружения – пол, потолок, кольцеобразная стена;

Предметы, крепящиеся к якорям, расположенным на игроке;

Источник звука, проигрывающий фоновую музыку;

Компоненты, отвечающие за игровую логики и взаимодействие с сервером;

Источник света.



1. – Сцена «Lobby»

Сцена первой комнаты. В данную сцену игроки попадают после подключения к комнате с типом «тип». Здесь игроки могут взаимодействовать друг с другом и с окружением. В данной сцене помимо игрока находятся:

Предметы окружения – пол, потолок, стены, \_;

Интерактивные компоненты - \_;

Предметы, крепящиеся к якорям, расположенным на игроке;

Компоненты, отвечающие за игровую логики и взаимодействие с сервером.

Источник света.

1. – Сцена «название»

Сцена второй комнаты. В данную сцену игроки попадают после подключения к комнате с типом «тип». Здесь игроки могут взаимодействовать друг с другом и с окружением. В данной сцене помимо игрока находятся:

Предметы окружения – пол, потолок, стены, \_;

Предметы, крепящиеся к якорям, расположенным на игроке;

Интерактивные компоненты - \_;

Компоненты, отвечающие за игровую логики и взаимодействие с сервером.

Источник света.

1. - Сцена «название»

Сцена для создания жестов к интерактивным объектам. В данную сцену нельзя попасть из работающего приложения. Она предназначена для разработчиков. В данной сцене настраивается взаимодействие с предметами через систему жестов Hand Grab Pose Recorder. Она минималистична и помимо игрока в ней присутствуют:

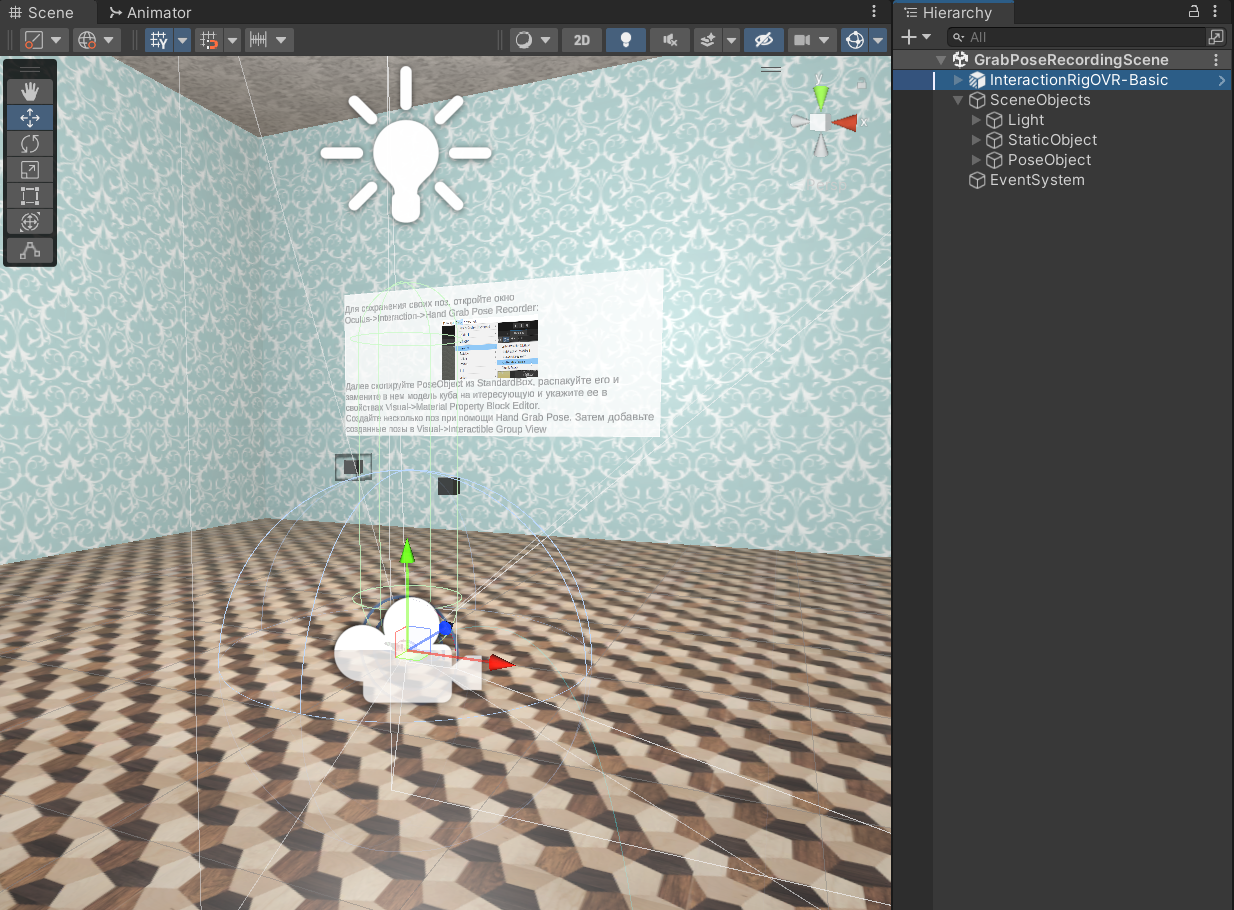
Предметы окружения – пол, потолок, стены;

UI инструкция по настройке жеста;

Prefab эталонного предмета, которому можно настраивать жесты;

Распакованный эталонный предмет;

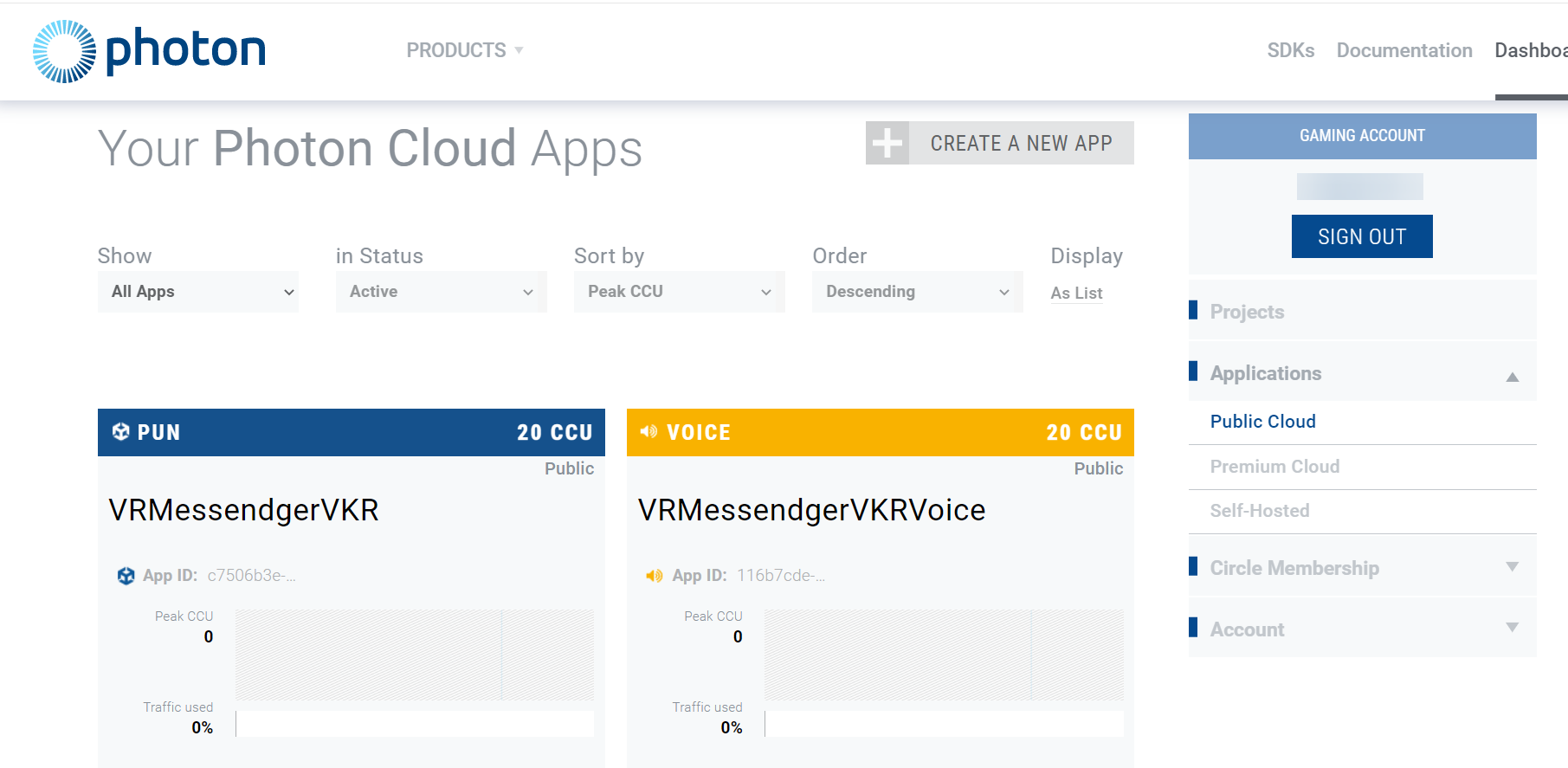
Источник света.



1. – Сцена «GrabPoseRecordingScene»

#### 3.2.2. Настройка клиент-сервера

Был создан аккаунт на сайте Photon Engine[6]. Были созданы 2 приложения photon. Первое для передачи положения объектов, второе для передачи голоса (см. рисунок 23).



1. – Приложения photon engine

В проект были импортированы следующие ассеты:

PUN 2 – FREE;

Photon Voice 2;

Данные этих приложений были добавлены в настройки ассетов PUN 2 – FREE и Photon Voice 2 соответственно.

В дальнейшем, при создании логики приложения использовались методы библиотек Photon.

#### 3.2.3. Создание логики приложения

Основным скриптом является NetworkManager. В нем осуществляется подключение к серверу, отключение от сервера, создание и подключение к лобби, создание и подключение к комнате, отключение от комнаты, отключение от лобби, отключение от сервера, спавн игрока на сервере. В полях данного скрипта указываются типы доступных комнат, а также имя prefab-а игрока спавнемого на сервере. Данный prefab должен храниться в папке Resources в корне проекта.

Скрипту NetworkManager необходимо загружать различные сцены. Для этой цели ему передается скрипт SceneChanger, отвечающий за смену сцен.

В процессе работы приложения регулярно необходимо будет синхронизировать состояния объектов с сервером. Для отправки текущего состояния на сервер, а также обработки пришедших на сервер состояний используется скрипт NetworkVariables.

Для отладки приложения необходимо логгировать отлаживаемые компоненты. При разработке не всегда удобно, а иногда и невозможно смотреть в консоль движка Unity. С этой целью был написан скрипт VRLogger, выводящий приходящие в него логи на текстовое поле интерфейса внутри приложения. Таких скриптов в сцене может быть много. Мы можем дать каждому игроку по экрану лога. Что бы при логгировании не искать каждый раз все VRLogger в сцене, был написан скрипт VRLoggersManager. VRLoggersManager выводит лог во все VRLogger на сцене.

Игрок может переключаться между типами контроллеров. Он может осуществлять взаимодействие с виртуальной реальностью при помощи контроллеров Oculus Touch, а может отложить их в сторону и начать взаимодействовать при помощи своих собственных рук. Подобное переключение необходимо отслеживать. Это нужно как для корректного отображения типа контроллеров на сервере, так и для использования корректных якорей на теле игрока. С целью отслеживания данного переключения создан скрипт ControllerEvents;

Скрипты NetworkManager, SceneChanger, NetworkVariables, VRLoggersManager, ControllerEvents должны всегда оставаться в сцене, и более того, при смене сцены они должны сохранять свое состояние. Для этого к ним применен паттерн Singleton, реализованный в скрипте EasySingleton.

NetworkManager при подключении к комнате спавнит на сервере prefab игрока, который он берет из папки Resources в корне проекта. Рассмотрим данный prefab:

\_\_\_

1. - Prefab спавнемого игрока

Данный prefab имеет голову, левую и правую руки.

Голова состоит из модели головы и компонента Speaker, отвечающего за передачу голоса на сервер.

В руках храниться компонент ControllerTypeController, отвечающий за смету типа отображаемых на сервере контроллеров. Внутри этого компонента хранятся отображаемые контроллеры: Controller и Hand. Controller управляется скриптом ControllerAnimationUpdater. Этот скрипт осуществляет синхронизацию анимации контроллера, отображаемого локально и контроллера, отображаемого на сервере. Hand, в свою очередь, управляется скриптом GostHandTransformUpdater. Данный скрипт синхронизирует положение и поворот всех суставов отображаемой на сервере руки и HandSynthetic локального игрока.

Весь prefab игрока контролируется скриптом NetworkPlayer, который обеспечивает синхронизацию положения контроллеров/рук и головы игрока локального и отображения этих частей на сервере. Так же данный скрипт следит за синхронизацией состояния типа контроллеров.

Если NetworkPlayer работает на стороне сервера, то на стороне клиента присутствует компонент HandView, к которому NetworkPlayer обращается за текущим положением и состоянием рук.

HandView должен быть прикреплен к рукам локального игрока, чтобы корректно передавать их положение на сервер. Причем при смене типа контроллера он должен перепрыгивать с рук на контроллеры и наоборот. Вручную размещать данные компоненты на игроке неудобно. А потому был написан скрипт AttachToPlayersBody. Данный скрипт прикрепляет компонент к якорю, указанному в его свойствах. Таким образом, мы создаем якорь на руках и контроллерах, а скрипт AttachToPlayersBody во время запуска игры крепит к данным якорям указанный компонент. Это позволяет хранить все прикрепляемые к игроку объекты в отдельном месте, а prefab игрока открывать только для настроек игрока и добавления якорей.

В лобби присутствуют UI кнопки, которые должны обращаться к NetworkManager, но NetworkManager нельзя добавлять в OnClick события, т.к. после смены сцен, в лобби вернется текущий NetworkManager, а указанный в изначальной сцене будет уничтожен. С целью исправить это, а также предоставить кнопкам удобный интерфейс взаимодействия с NetworkManager через OnClick события, был создан компонент NetworkManagerProvider.

Развивая идеи NetworkManagerProvider, появился компонент ComponentCatcher. В сцене могут появляться объекты изначально там не находящиеся. Это singleton объекты пришедшие из другой сцены, спавнемые аватары или просто объекты. И всем им могут быть нужны различные объекты из текущей сцены. ComponentCatcher это компонент, настраиваемый instance которого находиться в каждой сцене. Данный компонент при старте сцены отлавливает все часто используемые объекты и объекты, до которых не добраться во время редактирования сцены: VRLoggersManager, NetworkManager, ControllerEvents и подобные. В дальнейшем, если какому-либо скрипту понадобиться NetworkManager, он обратиться за ним к ComponentCatcher.

При взаимодействии игрока с предметами возникал ряд ошибок, следующие скрипты были написаны для исправления данных ошибок:

- IgnoreGrabbableObjects – исправление ошибки, из-за которой коллайдер взятого объекта сталкивался с коллайдером игрока, и игрок двигал сам себя.

- OwnershipTransfer - В Photon у интерактивных объектов есть владелец. И Photon синхронизирует с сервером только данные владельца объекта. Владельцем становится первый схвативший объект. Но нам необходимо, что бы все пользователи могли взаимодействовать с объектами. Для этого необходимо передавать владение объектом тому пользователю, который взял объект. OwnershipTransfer совершает передачу прав владения объектом тому, кто взял этот объект.

- RigidBodySycn – Photon синхронизирует положение, скорость, ускорение объекта. Но не синхронизирует isKinematic. Из-за чего происходит ошибка: когда игрок берет предмет, локально, данный предмет становиться Kinematic, а на сервере получается, что объект поднят в воздух и он начинает падать, пока Photon снова не синхронизирует его положение. Из-за данной ошибки взятый в руку предмет начинает прыгать в руку и вниз. RigidBodySycn синхронизирует isKinematic объекта с сервером и исправляет данную ошибку.

Взаимодействие с микрофоном осуществляется скриптами MicrophoneController и MicrophoneNetworkSettings. MicrophoneController контролирует локальные действия с микрофоном, а MicrophoneNetworkSettings отвечает за синхронизацию локальных действий на сервере.

В некоторых сценах играет фоновая музыка. Она может отвлекать или просто не нравиться пользователю. Для контроля нескольких источников звука, в частности фоновой музыки, используется скрипт AudioController.

В лобби состояние главного меню должно изменяться в зависимости от текущего состояния подключения к серверу. Например, пока происходит подключение меню должно отображать preloader, а когда подключение завершилось скрыть его и показать пункты меню. За динамическую смену слоев интерфейса главного меню в зависимости от сигналов приходящих от NetworkManager, отвечает компонент ChangeUILayer.

При выборе комнаты игроку должен предоставляться список комнат активных в данный момент. Этот список реализуется скриптами RoomListItem – элемент списка, и UIRoomListController – непосредственно список отрисовывающий RoomListItem в UI.

У игрока на запястье присутствует несколько меню. Эти меню не всегда включены и когда одно меню становится видимым, предыдущее исчезает. Для скрытия/появления меню используется скрипт InterfaceHider. А для смены одного меню на другое используется скрипт MenuSwapper.

Для распознавания жестов используется абстрактный класс GestureDetector. Его наследники могут отслеживать жесты и осуществлять действия при их обнаружении.

Компонент GestureProperties хранит в себе информацию о пальцах, загнутых в указанном жесте.

Вспомогательные маленькие скрипты:

- ButtonTextChanger – при наведении на UI кнопку записывает в указанное текстовое поле указанный текст;

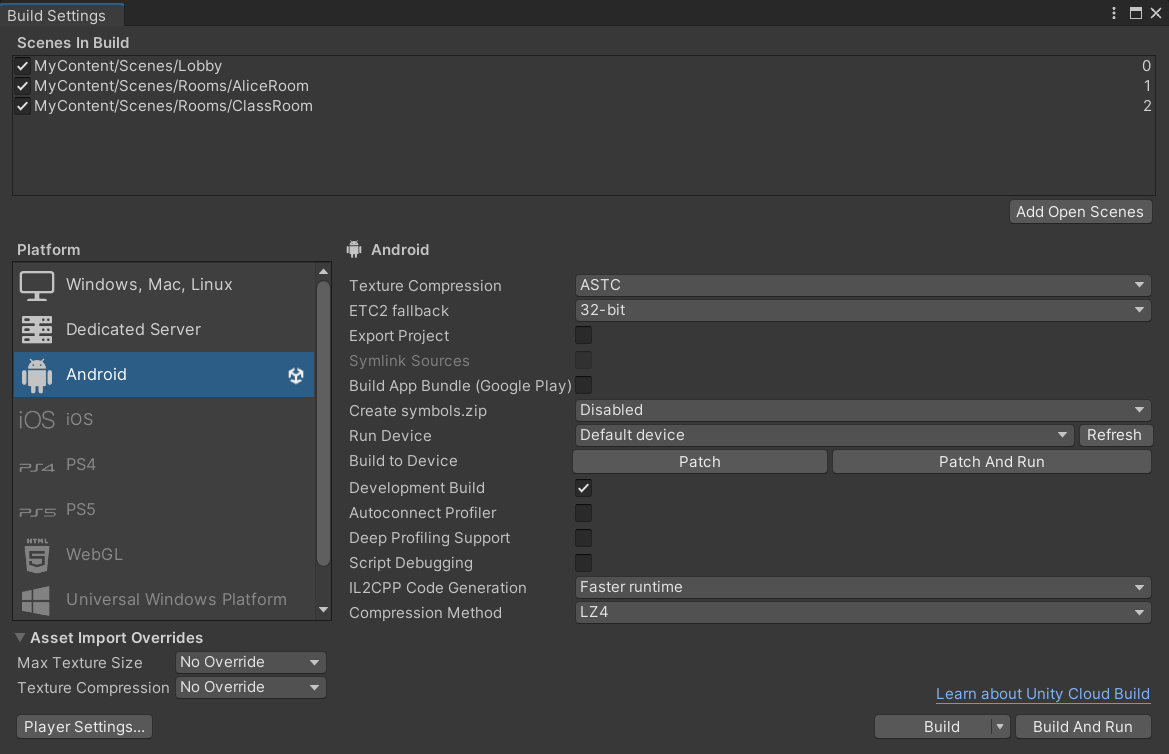
- GifAnimation – создает анимацию из входных кадров;

- SetupRoomName – записывает имя текущей комнаты в указанное текстовое поле.

Дописать скрипты, которые появятся в дальнейшем.

#### 3.2.4. Сборка проекта

Для сборки проекта указываем платформу Android. Добавляем в сборку необходимые сцены. Настройки сборки проекта представлены на рисунке 25.



1. - Build settings проекта

## 3.3. Проблемы и пути их решения

При взаимодействии игрока с предметами возникал ряд ошибок, следующие скрипты были написаны для исправления данных ошибок:

Коллайдер взятого объекта сталкивался с коллайдером игрока, и игрок двигал сам себя. Решение данной ошибки – написание скрипта IgnoreGrabbableObjects, благодаря которому колайдер игрока игнорирует колайдеры grabbable объектов.

Только один игрок в сцене может перемещать объект на сервере. В Photon у интерактивных объектов есть владелец. И Photon синхронизирует с сервером только данные владельца объекта. Владельцем становится первый схвативший объект. Но нам необходимо, что бы все пользователи могли взаимодействовать с объектами. Для этого необходимо передавать владение объектом тому пользователю, который взял объект. OwnershipTransfer совершает передачу прав владения объектом тому, кто взял этот объект.

Взятый в руку предмет, отображаясь на сервере, начинает прыгать в руку и вниз. Photon синхронизирует положение, скорость, ускорение объекта. Но не синхронизирует isKinematic. Из-за чего и происходит данная ошибка. Решением стало написание скрипта RigidBodySycn, синхронизирующего isKinematic объекта с сервером.

Проблема переключения типов контроллеров – с контроллера на руки и наоборот. Локально это просто, а вот как синхронизировать это переключение с сервером. Решением стало написание скрипта ControllerEvents, следящего за текущим состоянием локальных контроллеров и при их смене отправляющим сигнал синхронизации на сервер.

При подключении к комнате у всех аватаров выбраны контроллеры, как текущий тип контроллеров. Решение – при подключении к комнате отправлять на сервер запрос о синхронизации текущего клиента.

При попытках записать жесты взаимодействия с предметами, GhostHands не отображались. Обновление OculusIntegration до версии 51.0 исправило ошибку.

Невозможность передать на сервер HandsOVR. Первым вариантом решения проблемы было создать собственную модель рук. Создать анимацию сгибания каждого пальца. И при помощи жестов активировать данную анимацию на сервере. Однако этот вариант делал крайне сложным передачу сложных положений пальцев. Вторым вариантом было передача на сервер моделей используемых для захвата предметов жестами и синхронизация положения и поворота всех сгибающихся частей этих моделей.

На сервере синхронизируется только одна рука. Проблема в перегрузке компонента PhotonView. Один компонент следил за всеми частями тела игрока и просто не справлялся. Решением стало добавление каждой руке собственного PhotonView, что сняло нагрузку с PhotonView игрока и руки стали нормально синхронизироваться.

Не удается получить доступ к изменению громкости микрофона. Решение – написать скрипт, контролирующий громкость источника звука микрофона на сервере.

## 3.4. Экспериментальна проверка

Скриншоты работы приложения

# ЗАКЛЮЧЕНИЕ

*кратко пройтись по требованиям и результатам и сделать вывод о том, что цель диплома выполнена. Из заключения всем должно быть понятно, насколько это хороший диплом*

# СПИСКОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

|  |  |
| --- | --- |
| 1 | Официальный интернет-портал правовой информации «Указ Президента Российской Федерации от 09.05.2017 г. № 203» [Электронный ресурс] Режим доступа: http://www.kremlin.ru/acts/bank/41919, свободный (дата обращения 09.05.2023) |
| 2 | Официальный интернет-портал правовой информации «Распоряжение Правительства Российской Федерации от 21.12.2021 № 3759-р» [Электронный ресурс] Режим доступа: http://publication.pravo.gov.ru/Document/View/0001202112250002, свободный (дата обращения 09.05.2023) |
| 3 | Виртуальная и дополненная реальность: учеб. пособие / Д. А. Булгаков, Е. Е. Майн, А. В. Никитин, Н. Н. Решетникова, И. А. Ситников. – СПб.: ГУАП, 2022. – 210 с. |
| 4 | Виртуальная реальность в Unity. / Пер. с англ. Рагимов Р. Н. – М.: ДМК Пресс, 2016. – 316 с.: ил. |
| 5 | Страница игры «All-In-One Sports VR» в Steam [Электронный ресурс] Режим доступа: <https://store.steampowered.com/app/1514840/AllInOne_Sports_VR___C_VR/> , свободный (дата обращения 10.05.2023) |
| 6 | Статья «Что такое VRChat и почему все про него говорят?» [Электронный ресурс] Режим доступа: https://vgtimes.ru/articles/44931-chto-takoe-vrchat-i-pochemu-vse-pro-nego-govoryat-mozhno-li-ispolzovat-vrchat-bez-vr-shlema-kakie-igry-i-razvlecheniya-est-v-vrchat.html, свободный (дата обращения 08.05.2023) |
| 7 | Статья «Meta Horizon Worlds Current and Future State» [Электронный ресурс] Режим доступа: https://insidetelecom.com/meta-horizon-worlds-current-and-future-state/ (дата обращения 08.05.2023) |
| 8 | Сайт проекта «Softspace» [Электронный ресурс] Режим доступа: https://soft.space/, свободный (дата обращения 10.05.2023) |
| 9 | Статья «Top 10 VrChat Alternatives & Competitors» [Электронный ресурс] Режим доступа: https://www.g2.com/products/vrchat/competitors/alternatives (дата обращения 08.05.2023) |
| 10 | Статья «В чём отличие между одноранговой и клиент-серверной сетью» [Электронный ресурс] Режим доступа: https://community.fs.com/ru/blog/client-server-vs-peer-to-peer-networks.html, свободный (дата обращения 10.05.2023) |
| 11 | Статья «Чем HTC Vive Pro лучше Oculus Quest 2» [Электронный ресурс] Режим доступа: https://portal-shop.com/blog/obzory/chem-htc-vive-pro-luchshe-oculus-quest-2/, свободный (дата обращения 10.05.2023) |
| 12 | Статья «Pico 4 vs Oculus Quest 2 | Что купить?» [Электронный ресурс] Режим доступа: https://portal-shop.com/blog/obzory/pico-4-vs-oculus-quest-2-chto-kupit/, свободный (дата обращения 10.05.2023) |
| 13 | Статья «VR шлем Pico 4 — ожидание и реальность» [Электронный ресурс] Режим доступа: https://vr419.ru/blog/vr-shlem-pico-4-ozhidanie-i-realnost/, свободный (дата обращения 10.05.2023) |
| 14 | Статья «Что такое аватары и зачем они нужны в метавселенной» [Электронный ресурс] Режим доступа: https://mitsoftware.com/ru/что-такое-аватары-и-зачем-они-нужны-в-метавселенной/, свободный (дата обращения 10.05.2023) |
| 15 | Официальный сайт Ready Player Me [Электронный ресурс] Режим доступа: https://readyplayer.me/ru, свободный (дата обращения 15.05.2023) |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
|  |  |
|  | Официальная документация «Unity» [Электронный ресурс] Режим доступа: https://unity.com/, свободный (дата обращения 08.05.2023) |
|  | Статья «Основы виртуальной реальности» [Электронный ресурс] Режим доступа: https://www.teacherjournal.ru/categories/24/articles/1316, свободный (дата обращения 10.05.2023) |
|  | Официальный сайт Photon Engine [Электронный ресурс] Режим доступа: https://dashboard.photonengine.com/, свободный (дата обращения 08.05.2023) |
|  |  |
|  | https://pikabu.ru/story/zamechatelnyie\_lyudi\_pridumali\_modeli\_i\_miryi\_dlya\_vrchat\_5615728 |
|  | https://www.youtube.com/watch?v=BO0HCQpdQzs |
|  | https://store.steampowered.com/app/407060/AltspaceVR/ |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# ПРИЛОЖЕНИЕ А

Все скрипты

# ПРИЛОЖЕНИЕ Б

Картинки