

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Кафедра

вычислительной техники

«Вычислительная техника»

Специальность: 09.05.01 «Применение и эксплуатация
автоматизированных систем специального
назначения»
Специализация: «Эксплуатация вычислительных машин, комплексов,
систем и сетей специального назначения»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
СПЕЦИАЛИСТА

на тему

ПРОГРАММА ЗАПОЛНЕНИЯ ТАКТИЧЕСКИХ
КАРТ В ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Студент Халак Данил Сергеевич (подпись, дата) 16.06.22 (ФИО полностью)
Руководитель Митрохин М.А. (подпись, дата) 16.06.22 (фамилия, инициалы)
Нормоконтролёр Бычков А.С. (подпись, дата) 16.06.22 (фамилия, инициалы)
Рецензент Афонин А.Ю. (подпись, дата) 22.06.2022 (фамилия, инициалы)
(должность, место работы) каф. вычисл. тех.

Работа допущена к защите (протокол зас. кафедры от 02.06.22 г. № 13)

Заведующий кафедрой М.А. Митрохин (подпись) (фамилия, инициалы)

Работа защищена с отметкой Отлично (протокол заседания ГЭК от
24 июня 2022 г. № 14)

Секретарь ГЭК Е.И. Гурин

Пенза 2022

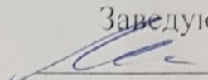
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»

Заведующий кафедрой

 М.А. Митрохин

«26» апреля 2022 г.

**ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ**

1. Студент Халак Данил Сергеевич гр. 17BO2 факультета вычислительной техники.
2. Тема работы «Программа заполнения тактических карт в дополнительной реальности»

Тема утверждена приказом ПГУ №0357/о от "26" апреля 2022 г.

3. Руководитель работы д.т.н., заведующий кафедрой ВТ ПГУ Митрохин М.А.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

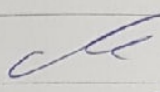
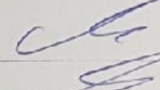
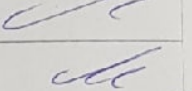
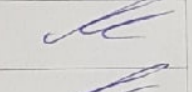
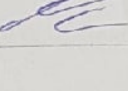
Программа заполнения тактических карт в дополнительной реальности предназначена для упрощения и ускорения заполнения тактических карт ответственными лицами.

Приложение должно обеспечивать добавление, изменение и удаление карт и тактических знаков из дополненной реальности; регистрацию и авторизацию пользователей в программе.

5. Перечень подлежащих разработке вопросов

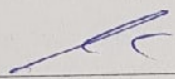
Требуется произвести анализ предметной области, сформировать техническое задание, выбрать инструменты для разработки, выполнить проектирование системы, разработать программу заполнения тактических карт в дополненной реальности. Программа должна выполнять функции: обеспечивать добавление, изменение и удаление карт и тактических знаков из дополненной реальности; регистрацию и авторизацию пользователей в программе. Также выполнить задание по БЖД и осуществить экономическое обоснование разработки системы

6. Календарный график выполнения работы

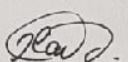
№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	8%	26.04.2022 – 29.04.2022	
2.	Составление технического задания	6%	30.04.2022 – 02.05.2022	
3.	Выбор инструментов разработки	18%	03.05.2022 – 06.05.2022	
4.	Проектирование системы	32%	07.05.2022 – 12.05.2022	
5.	Выполнение экономического обоснования	10%	02.06.2022 – 06.06.2022	
6.	Выполнение задания по БЖД	8%	07.06.2022 – 10.06.2022	
7.	Оформление пояснительной записки	18%	11.06.2022 – 14.06.2022	

Дата выдачи задания "26" апреля 2022 г.

Руководитель работы

 26.04.22
(подпись, дата)

Задание к исполнению принял студент

 26.04.22
(подпись, дата)

Работу к защите допустить
Декан факультета

 А. Р. Филимова
(подпись, дата)
23.06.2022

Реферат

Работа содержит 98 листов, 32 рисунков, 10 источников, 9 приложений.
API, AR, ARCORE, SDK, UI, UNITY, 2D, 3D, ПО.

Цель работы – разработать программу для редактирования карты путем добавления на нее тактических знаков в дополненной реальности.

Объектом разработки является программа, для заполнения тактических карт в дополненной реальности.

Разработка велась на движке *Unity 2021* с применением *ARCore*.

В процессе разработки был проведен анализ требований к разрабатываемому продукту, анализ существующих современных средств проектирования программных продуктов.

В результате проделанной работы была разработана программа заполнения тактических карт в дополненной реальности.

В экономической части дается технико-экономическое обоснование целесообразности разработки системы.

В разделе безопасности жизнедеятельности описаны психофизиологические факторы, воздействующие на оператора ЭВМ, а также рассмотрено воздействие работы с ПК на зрение человека.

					<i>ПГУ 1.090501.15.001 ПЗ</i>			
Изм.	Лист	№ докум.	№	Подпись	Дата			
Разраб.		Халак Д.С.				Лит.	Лист	Листов
Провер.		Митрохин М.А.					2	98
Н. Контр.		Бычков А.С.				ФВТ, каф. ВТ гр. 17В02		
Утверд.		Митрохин М.А.						
Программа заполнения тактических карт в дополненной реальности Пояснительная записка								

Содержание

Перечень сокращений и обозначений.....	5
Введение.....	6
1 Программа заполнения тактических карт в дополненной реальности	7
1.1 Общий анализ предметной области.....	7
1.2 Обзор AR технологий.....	8
1.2.1 Определение дополненной реальности.....	8
1.2.2 Классификация.....	9
1.2.3 Анализ популярности.....	9
1.2.4 Разработка диаграммы вариантов использования	10
1.2.5 Разработка диаграммы деятельности	14
1.3 Постановка задачи.....	17
1.4 Проектирование системы.....	17
1.4.1 Выбор аппаратного обеспечения.....	17
1.4.2 Выбор программного продукта для разработки.....	19
1.4.3 Приложения со схожим функционалом.....	21
1.5 Разработка приложения для работы в дополненной реальности.	22
1.6 Тестирование и отладка.....	37
2 Технико-экономическое обоснование разработки системы.....	46
2.1 Введение.....	46
2.2 Расчёт затрат на разработку программного обеспечения.....	47
2.3 Расчёт цены разработанной программы.....	49
2.4 Расчёт капитальных вложений.....	49
2.5 Расчёт эксплуатационных расходов.....	51
2.6 Определение расходов на интернет трафик	50
2.7 Техническо-экономические показатели разработки	52
3 Безопасность жизнедеятельности. Эргономика рабочего места.....	54
3.1 Характеристика условий труда программиста.....	54

3.2 Требования к параметрам воздушной среды.	55
3.3 Требования к уровню шума и вибрации.	56
3.4 Требования к монтажу производственного оборудования.	57
3.5 Требования к освещению рабочих мест.	57
3.6 Расчет освещения рабочего места программиста.	58
Заключение.	61
Список использованных источников	62
Приложение А – <i>UML</i> -диаграмма классов	64
Приложение Б – Схемы программ.	66
Приложение В – Листинг программы.	71
Приложение Г – Презентация	90

Перечень сокращений и обозначений

API (Application programming interface) – интерфейс приложения для программирования

AR (Augmented reality) – дополненная реальность

ARCore – это инструмент для разработки программного обеспечения для дополненной реальности

SDK (Software development kit) – набор средств разработки

UI (User interface) – это визуальная часть для передачи информации между пользователем и программой

2D (Two-dimensional) – компьютерная двумерная графика

3D (Three-dimensional) – компьютерная трехмерная графика

ПО – программное обеспечение

Введение

Географическая карта – это уменьшенное копия земной поверхности, перенесённая на плоскости с размещением природных и общественных объектов. Из-за многообразия объектов географические карты подразделяют на общегеографические и специальные. В них отражается одно или несколько физико-географических, социально-экономических или каких-либо других явлений.

Топографические карты в отличие от географических сохраняют постоянство масштаба. В военном деле используются именно этот тип карт. Не один род войск не может обойтись без подробного изучения местности при организации боя, при перемещении военнослужащих на небольшие расстояния по незнакомой местности, для управления подразделениями и частями, а также при выборе мест расположения каких-либо объектов. Для артиллерии эти карты крайне необходимы, они используются для определения противника, подготовки исходных данных для стрельбы, а также при планировании огня и перемещении артиллерии.

					<i>ПГУ 1.090501.15.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

1 Программа заполнения тактических карт дополненной реальности

1.1 Общий анализ предметной области

Рабочие карты ведутся всеми офицерами лично. Каждый офицер наносит только те данные обстановки, которые ему необходимы для работы по занимаемой должности. Офицер должен вести свою рабочую карту, чтобы в отображаемой обстановке мог свободно разбираться любой офицер (старший начальник, подчинённый).

Может возникнуть необходимость передавать обстановку старшему начальнику непосредственно с рабочей карты с помощью фототелеграфа, а также снимать с неё копии посредством использования множительной техники.

Очень часто случается, что районы боевых действий подразделений выходят за пределы одного листа карты. В таких случаях рабочая карта командира может состоять из одного и более листов. Для удобства пользования картой производится предварительное склеивание листов [1].

У бумажных карт есть большой недостаток. Они не могут отчетливо передают местность, для устранения слабого звена, и чтобы карта была более наглядной и на ней отчетливо выделялись нужные для выполнения задачи подробности, усиливается расцветка деталей, которые больше интересуют. Повышать расцветку карты рекомендуется подтушевкой или дополнительной штриховкой нужных участков тем же цветом, которым обозначены интересующие нас детали местности и местные предметы. Населенные пункты обводят по внешнему контуру, подчеркивая или подписывая более крупным шрифтом, ручьи и реки утолщают синим (голубым) цветом, дороги — коричневой линией непосредственно по самой дороге или параллельно ей.

Еще одним недостатком может стать выбор места для работы с картой.

					ПГУ 1.090501.15.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

Для заполнения тактической карты необходим свободный участок минимум 55 x 35 см, цветные карандаши, гелиевые ручки, линейка, ластик, точилка/нож. Порой эти вещи трудно найти в походных условиях. Эту проблему можно с лёгкостью решить установкой рабочего места, но это занимает какое-то время, а в некоторых ситуациях этого сделать просто невозможно.

По характеристикам электронные карты имеют множество преимуществ перед бумажными. В первую очередь – точность. Благодаря снимкам со спутника точность оказывается значительно выше, получаются более детализированными, что дает возможность без труда рассмотреть даже мелкие элементы местности. Благодаря этому ориентироваться по ним гораздо удобнее, проще и легче.

Немаловажным фактором подобных карт является и возможность беспроблемного масштабирования изображения, позволяющего получить всю необходимую информацию о конкретном участке. Кроме того, данные таких карт постоянно обновляются, чего нельзя сказать о бумажных версиях. Благодаря этому вы можете быть уверены, что данные, которые вы видите на экране, являются актуальными [2].

Таким образом, заполнение карт занимает достаточно много времени и внимания. Особенно это актуально в полевых условиях. Однако с использованием современных технологий появилась возможность уменьшить время, затрачиваемое на заполнение карт.

1.2 Обзор *AR* технологий

1.2.1 Определение дополненной реальности

Дополненная реальность - это технология расширенной реальности. Этот термин мы используем для технологий, которые добавляют к реальности или позволяют создавать совершенно новую реальность.

AR позволяет накладывать на мир текст, изображения, видео и даже 3D-модели. Другими словами, *AR* - дополняет или расширяет реальность. Два

					ПГУ 1.090501.15.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

популярных примера включают фильтры *Snapchat* (представьте, что у вас на лице появляются собачьи уши) и *Pokémon Go* (где геймеры используют свой телефон или планшет, чтобы ловить покемонов, которые, казалось бы, появляются перед ними).

1.2.2 Классификация

- распознавание изображения, видео, текст или 3D-объектов накладываются на реальное 2D-изображение;
- распознавание объектов - цифровая 3D-модель прикреплена к реальному 3D-объекту, который может захватывать и перемещать цифровую 3D-модель;
- обнаружение плоскости - цифровая 3D-модель прикреплена к реальной плоской поверхности, которая обычно неподвижна, поэтому вам нужно перемещаться по ней [3].

1.2.3 Анализ популярности

Часто пользователи не знают, что используют технологию *AR*. Чуть больше половины пользователей знакомы с технологией благодаря функциям дополненной реальности, реализованным в приложениях, таких как *Snapchat* и *Instagram*. Танцующие хот-доги, радуги изо рта пользователей и даже маски для лица (позже удаленные из приложений) возможны только благодаря дополненной реальности.

Это говорит о важном о распространении дополненной реальности. Но в данный момент не все новые смартфоны поддерживают *AR*. Это указывает на то, что мы все еще не считаем дополненную реальность незаменимой.

В 2014 году *Pepsi* стала популярна благодаря своей рекламной кампании с дополненной реальностью на автобусной остановке, и одно только видео собрало более 8 миллионов просмотров на *YouTube*. Экран, замаскированный под прозрачную стену автобусной остановки, «добавлял» в городской пейзаж гигантских роботов, НЛО и тигров, изумляя пассажиров.

Секрет успеха кампании заключался в доступности. Людям не нужно

					ПГУ 1.090501.15.001 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

было устанавливать приложение: чтобы испытать дополненную реальность, им просто необходимо было смотреть на экран.

Следующий пример использования технологии дополненной реальности - *AR-животные* от *Google*. Загрузка приложения не требуется. Неудивительно, что *3D-животные* стали популярными во время *COVID-19* [4].

В *App Store* таких приложений более 11 000, но лишь немногие из них известны обычному пользователю. Необходимость установки приложения является серьезным препятствием для внедрения, особенно если это не, то приложение, которое вы планируете использовать ежедневно.

Проблема заключается в том, что платформы, которые используют дополненную реальность, называют так: «фильтры», «линзы», «визуальный поиск» — все, кроме «дополненной реальности». Из-за этого пользователи не могут узнать данную технологию.

Еще один фактор заключается в том, что сначала нужно предоставить пользователям легкодоступность к дополненной реальности, прежде чем они зацепятся достаточно. Немногие компании имеют ресурсы, чтобы сделать то, что сделала *Pepsi*, но все еще есть много способов «обмануть» пользователей, заставив их сознательно попробовать дополненную реальность.

Некоторые компании предлагают продуктовые наборы или меню ресторана с *QR*-кодом, который пользователи могут отсканировать, чтобы увидеть появление *3D*-персонажа и рассказать о продукте.

1.2.4 Разработка диаграммы вариантов использования

Диаграммы вариантов использования описывают взаимоотношения и зависимости между группами вариантов использования и действующих лиц, участвующих, в процессе.

Диаграммы вариантов использования не предназначены для отображения проекта и не могут описывать внутреннее устройство приложения, они предназначены для упрощения взаимодействия с будущими пользователями системы, с клиентами.

Для приложения была разработана диаграмма вариантов использования, которая включает в себя: один актер (Пользователь) и 6 вариантов использования (Запрос на получение доступа к дополнительной реальности, Регистрация нового пользователя, Дополненная реальность, Постановка карты, Постановка тактических знаков, Удаление всех объектов).

При регистрации нового пользователя необходимо ввести пароль, после этого заполнить поля. Запрос на получение доступа представляет собой ввод логина и пароля и проверка этих данных с уже зарегистрированными. При выполнении дополненной реальности подключается камера и формируется сетка. Для постановки карты необходимо выбрать ее из списка предложенных. Для постановки тактических знаков необходимо выбрать его из списка предложенных. При удалении объектов удаляется и их координаты из дополненной реальности.

Диаграмма вариантов использования представлена на рисунке 1.

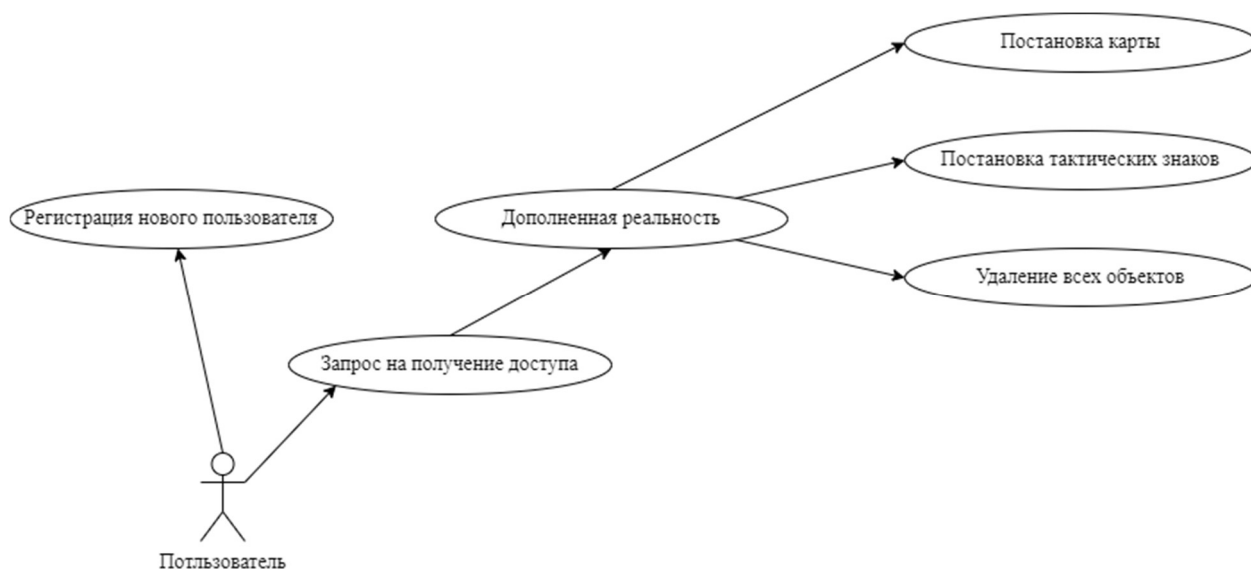


Рисунок 1 — Диаграмма вариантов использования

При запуске программы пользователя встречает окно доступа к программе. В этом окне необходимо авторизоваться или если нет учетной записи зарегистрироваться. После авторизации и входа в дополненную

реальность начинается запуск камеры и интерфейса. После чего система получает запрос от пользователя на создание поля. После запроса система определяет поверхность и создает на ней видимую сетку. Далее отображает данные пользователю. После чего пользователь начинает работать с картой. Выбирает карту из списка предложенных и выставляет ее на поверхность. После этого происходит передача данных об изменении поля в систему и отображения карты в дополненной реальности. Постановка тактических знаков таким же образом. Выбирается тактический знак из списка предложенных и выставляет ее на поверхность. После этого происходит передача данных об изменении поля в систему и отображения объектов в дополненной реальности.

Диаграмма последовательности представлена на рисунке 2.

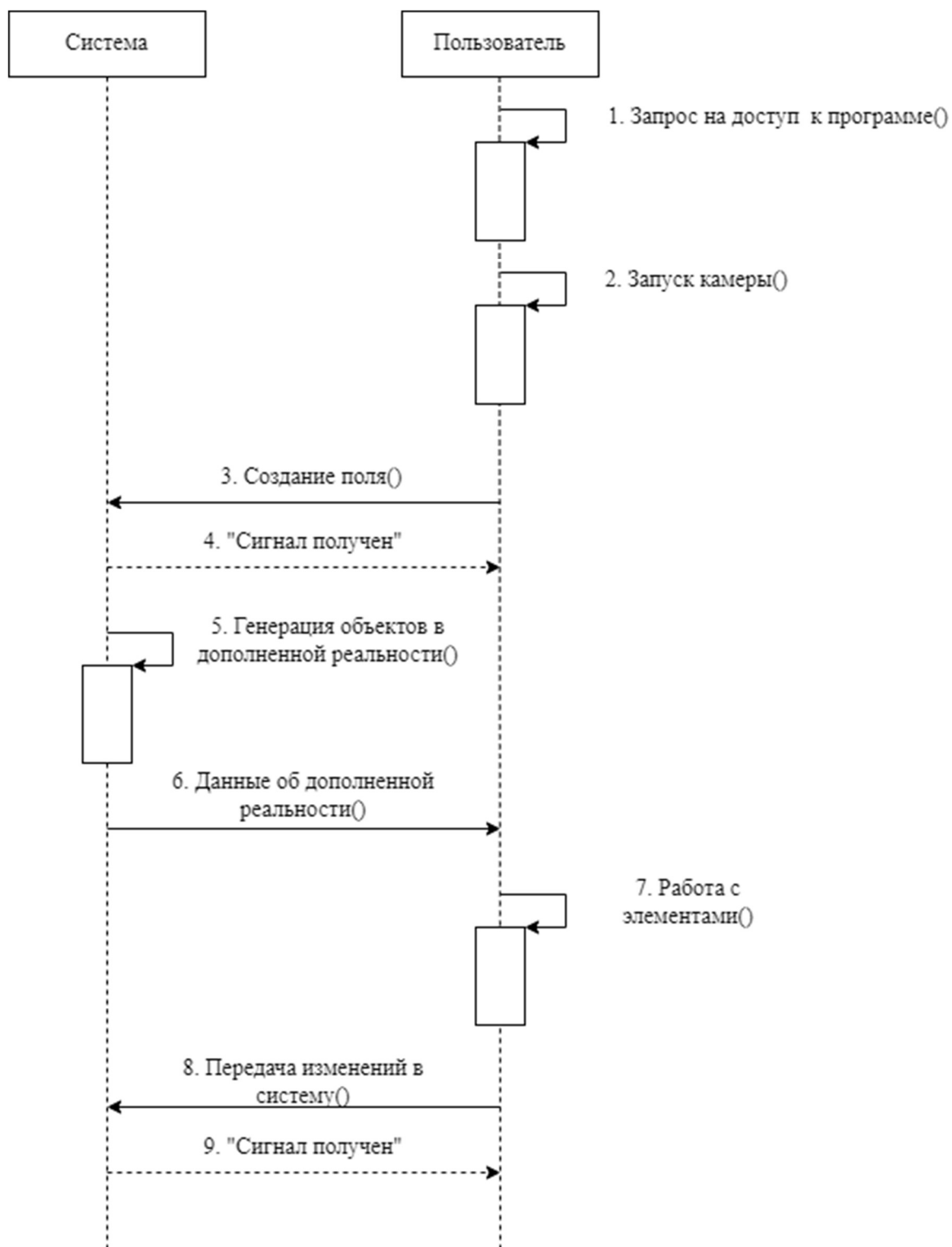


Рисунок 2 — Диаграмма последовательности

1.2.5 Разработка диаграмм деятельности

При входе в программу пользователь оказывается на окне авторизации. Если пользователь уже зарегистрирован, для входа необходимо ввести логин и пароль, после проверки пользователю представляется доступ к программе, при неверном вводе пароля или логина выходит сообщение об ошибке. Если пользователь, не зарегистрировав, ему предлагается зарегистрироваться, при регистрации пользователь должен ввести пароль для получения доступа к регистрации, если пароль не верен, то выводится ошибка и предлагается ввести пароль заново. Представлено на рисунке 3.

На рисунке 4 представлено включение камеры, где сначала загружается камера, потом отображается интерфейс. После включения камеры программа начинает распознавать поверхность. После того как поверхность распознана создается видимая сетка, на которую можно ставить объекты. Далее необходимо выбирать карту из списка, затем поставить ее на видимую сетку. Вовремя постановки объекта в дополненной реальности получается координата постановки объекта. Координаты объектов сохраняются, а также сохраняется и сам объект. Таким же образом происходит постановка тактических знаков.

Диаграммы деятельности представлены на рисунках 3 и 4

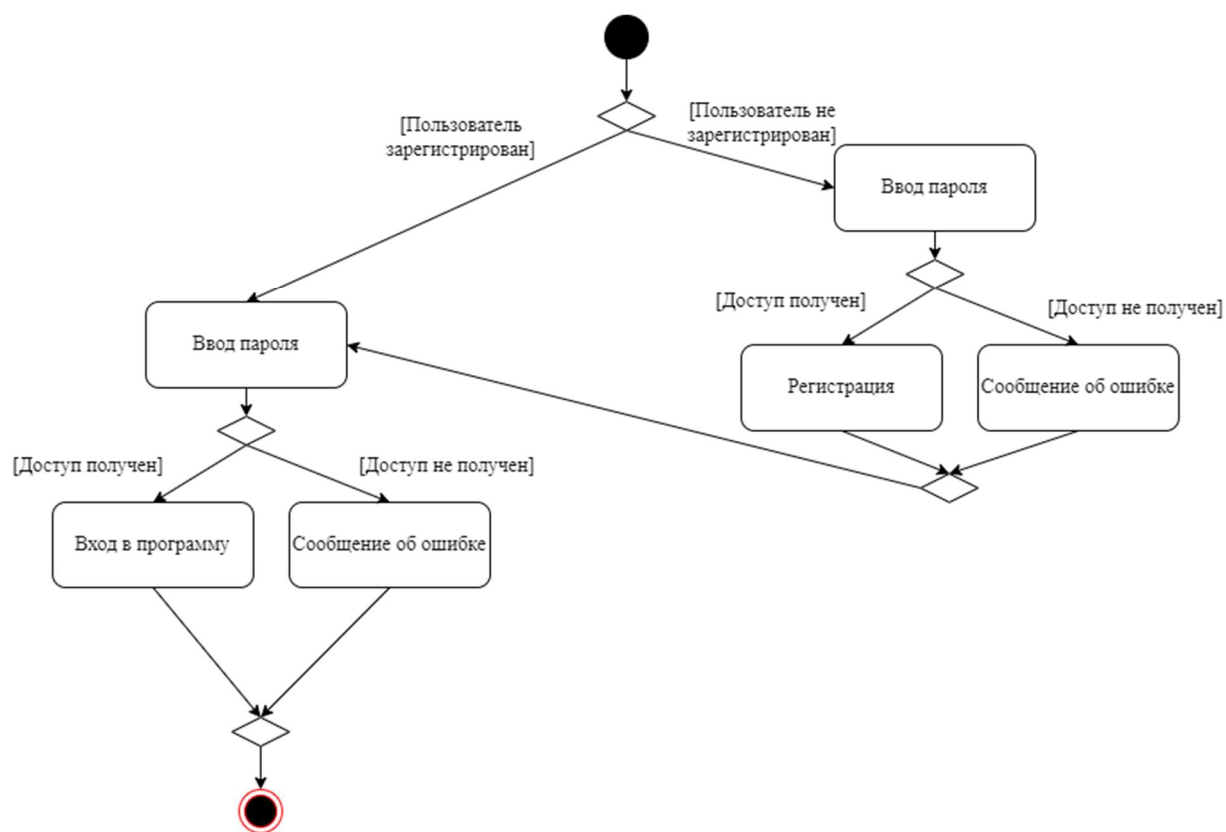


Рисунок 3 — Диаграмма деятельности «Запрос на доступ к программе»



Рисунок 4 — Диаграмма деятельности «Инициализация рабочего поля»

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.15.001 ПЗ

Лист

16

1.3 Постановка задачи

Необходимо разработать «Программа заполнения тактических карт в дополненной реальности».

В данной программе необходимо реализовать несколько частей: Авторизация/регистрация пользователя в приложении, заполнение тактических карт. Возможности программы должны предоставлять взаимодействие с объектами в *AR*, расстановка тактических карт в пространстве, наличие ряда тактических знаков, наличие карт. Для решения этих задач необходимо разработать и реализовать простой, удобный и понятный пользовательский интерфейс.

Программа должна быть многомодульной: необходимо поделить её на логические модули. Это способствует упрощению добавления в программу новых функций при дальнейшей разработке, а также упростит обнаружение и исправление ошибок при её отладке и тестировании.

Программа должна быть разработана для работы на телефонах с поддержкой *ARCore* под работай *OC Android 7* и выше.

1.4 Проектирование системы

1.4.1 Выбор аппаратного обеспечения

Для использования приложений дополненной реальности необходимы терминальные устройства со сложной композицией аппаратных модулей (Smartdevices). Исторически сложилось так, что смартфоны были первыми мобильными терминальными устройствами, использующими *AR*, особенно *Apple iPhone 3GS*. Позже за ним последовали планшеты на базе *Android* от *Samsung (Samsung Galaxy Tab)* или *iPad* от *Apple* на базе *IOS*. В 2012 году *Google* продвигал свой проект «*Google Glass*», хотя он был не единственным представителем смарт-очков, он привлек наибольшее внимание благодаря своему маркетингу и набору функций. С 2017 года смарт-очки и гарнитур

дополненной реальности (например, *HoloLens*, *Meta2*, *Magic Leap*) привлекли внимание быстрорастущих разработчиков из-за их более сложных технологий и обширных датчиков. После публикации *Apple (ARKit)* и *Google (ARCore)* новых фундаментальных технологий дополненной реальности в качестве части их репрезентативной операционной системы, обе компании начали разрабатывать аппаратное обеспечение, оптимизированное для дополненной реальности (смартфоны, планшеты, смарт-очки или даже более сложные гарнитуры).

Ниже выведены достоинства и недостатки очков для дополненной реальности и смартфонов.

Достоинства смартфонов:

- широкое распространение среди конечных пользователей;
- всегда присутствует с конечными пользователями;
- ежедневное использование приложений;
- мобильность.

Недостаток смартфонов:

- отсутствие полного погружение в дополненную реальность.

Достоинства специальных очков:

- специально изготовленные аппаратные профили;
- взаимодействие с помощью жестов и распознавания голоса;
- аппаратное обеспечение, оптимизированное для использования более сложные программные решения дополненной реальности;
- интуитивное восприятие дополненной реальности через наложенное зрение в собственном поле зрения.

Недостатки специальных очков:

- не распространённо;
- дорого;
- неудобны для повседневной жизни.

На основании всех вышеперечисленных плюсов и минусов для использования в данной работе был выбран смартфон, так как рассматривались простота обращения и удобство использования в повседневной жизни.

1.4.2 Выбор программного продукта для разработки

Наиболее популярное ПО для разработки *AR*-приложений:

- *Unity*;
- *Unreal Engine*.

Unity — межплатформенная среда разработки компьютерных игр, 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 идёт постоянное развитие.

Основными преимуществами *Unity* являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек.

Unreal был выпущен в 1998 году компанией *Epic Games*, а в 1999 году последовала известная игра *Unreal Tournament*. Движок *Unreal* всегда был известен своим потрясающим графическим качеством, широко используемым в большинстве лучших игр *AAA*. В отличие от *Unity*, язык кодирования *Unreal* (C++) более сложен, и за ним все еще нет большого сообщества.

Огромным плюсом *Unity* является интуитивно понятный интерфейс. В нем не трудно разобраться и начать экспериментировать с игровым движком (не говоря уже о большом количестве руководств и курсов, доступных в Интернете, а также о большом сообществе).

С другой стороны, *Unreal* использует C++, что довольно сложно для

разработчиков, незнакомых с этим языком программирования. Теоретически вы можете создать игру, не вводя ни одной строки кода, с помощью их недавней системы визуального скриптинга «*Blueprint*», которую *Unity* только сейчас пытается использовать с «*Bolt*». Тем не менее, без каких-либо знаний C++ вы все равно будете бороться. Интерфейс огромен, с бесконечными панелями инструментов, которые для некоторых пользователей могут быть довольно громоздкими.

Unity наполнен инструментами для физики, анимации, триггеров событий, звука и т. д. Однако после того, как вы освоитесь с ними, вы почувствуете, что им чего-то не хватает, и тогда магазин активов компенсирует это. С *Unreal* вы, скорее всего, испытаете обратное, поскольку большое разнообразие вариантов может показаться немного ошеломляющим [5-7].

Ниже представлены достоинства и недостатки инструментов.

Достоинства Unity:

- поддержка платформы;
- есть встроенная IDE;
- поддержка высококачественные звуковые и визуальные эффекты;
- детальная документация;
- отражение всего игрового мира при отладке.

Недостатки Unity:

- графика;
- физика;
- стоимость лицензии;
- трудно оптимизировать;
- захват памяти.

Достоинства Unreal Engine:

- лучшие характеристика графики;
- система программирования схема блоков, но большую часть

необходимо писать на C++;

- большая библиотека инструментов и функций;
- быстрая работа рендеринга (хорошая оптимизация).

Недостатки Unreal Engine:

- модель лицензирования;
- не подходит для разработки одним человеком;
- много данных, маленькие игры (несколько скриптов и 1 сцена)

занимает много места.

1.4.3 Приложения со схожим функционалом

Campaign Cartographer 3 (CC3) — это программа для фэнтезийной картографии от *ProFantasy*, которая используется для создания большинства карт.

Преимущество *CC3* заключается в том, что начать создавать карты очень просто. Существуют шаблоны для построения, а панели инструментов заполнены не только вспомогательными средствами для рисования, но и фактическими элементами отображения. Таким образом, в отличие от обычных графических программ, таких как *PhotoShop* или *Gimp*, вам не нужно быть художником, чтобы создать гору — просто необходимо выбрать тип горы, который вы хотите, а затем разместить его в любом месте на карте.

Несмотря на то, что с *Campaign Cartographer* легко начать работу и не требуется особых художественных способностей, не нужно рассчитывать, что с первой попытки можно создать профессионально выглядящие карты. Чтобы достичь этого уровня, потребуется значительный объем знаний и опыта. Нужно научиться использовать эффекты листа, чтобы смешивать, размывать, отбрасывать тени и все такое прочее.

Dungeon Painter Studio — это мощный инструмент для создания карт. Рассмотрим минусы и плюсы приложения.

Главный минус приложения — это невозможность долго работать в этом инструменте. При работе с верх 3 часов система замедляется.

Из плюсов стоит отметить, низкий порог вхождения. Карту возможно сделать довольно быстро, детализировано и красиво. В приложении есть функция добавления своих объектов, что является очень полезным.

1.5 Разработка приложения для работы в дополненной реальности

Программа заполнения тактических карт в дополнительной реальности состоит из совокупности двух частей – визуальной и программной:

- визуальная часть:

- 1) 3D модель окружающего мира и объектов в нем;
- 2) текстуры созданного мира;

- программная часть:

- 1) камера;
- 2) скрипты;
- 3) интерфейс.

Для создания нового проекта в *Unity* необходимо выполнить следующие действия:

- на стартовом окне приложения *Unity Hub* нажать на кнопку *New project* (рисунок 5);

- в открывшемся окне необходимо выбрать проект *ARCore* и нажать на кнопку *Create project* (рисунок 6).

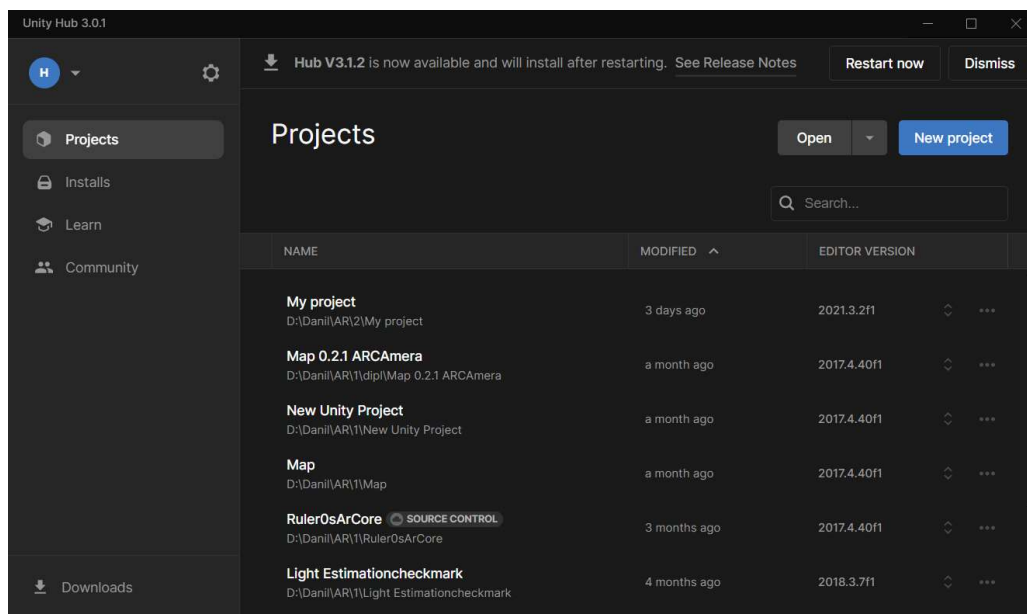


Рисунок 5 – Создания нового приложения

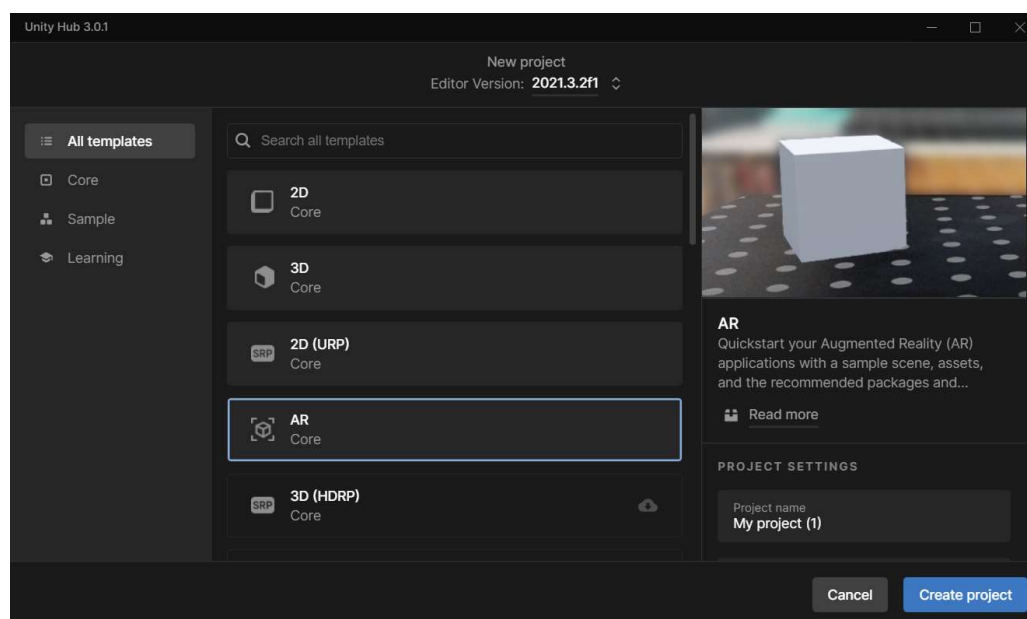


Рисунок 6 – Создание проекта типа AR

Для работы с дополненной реальностью необходимо изначально настроить проект. Настроить платформу разработки *Android* и подключить *XR-plugin ARCore*. Использование *API*-интерфейсы подключаемых модулей *ARCore XR* необходимо для использования функций *ARFoundation* в разработки.

Для настройки платформы программы необходимо:

- в правом верхнем углу нажать на кнопку *File* из открывшегося меню выбрать *Build Settings* (рисунок 7);
- открывшееся окно настроек в котором имеется платформы, поддерживаемые *Unity* (рисунки 8);
- необходимо выбрать платформу *Android* и нажать на кнопку *Switch Platform* (рисунок 9).

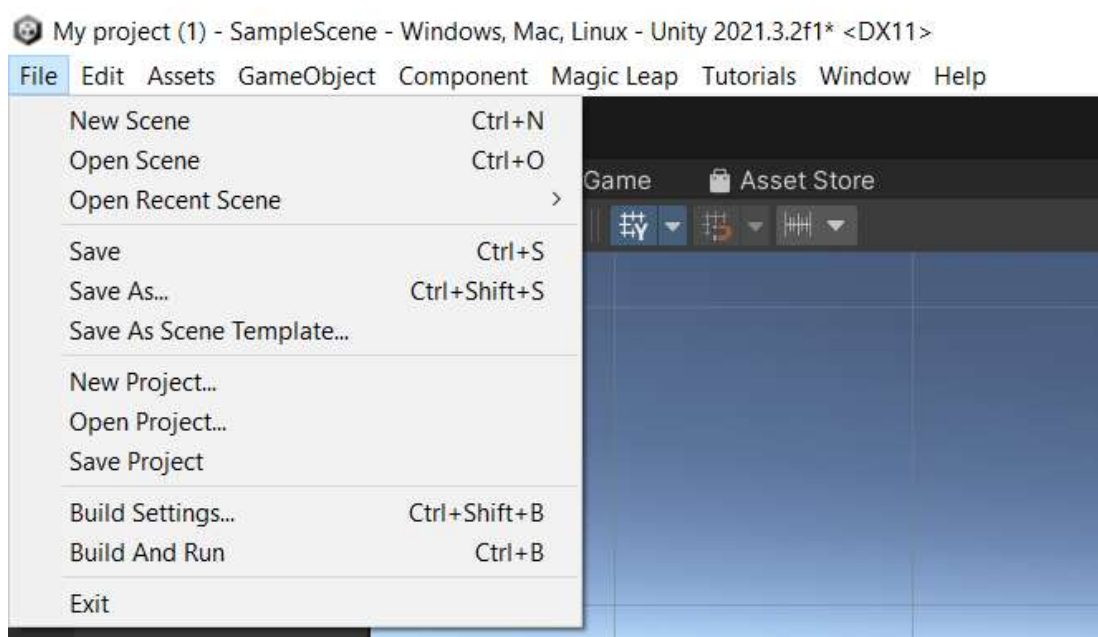


Рисунок 7 – Вкладка *File*

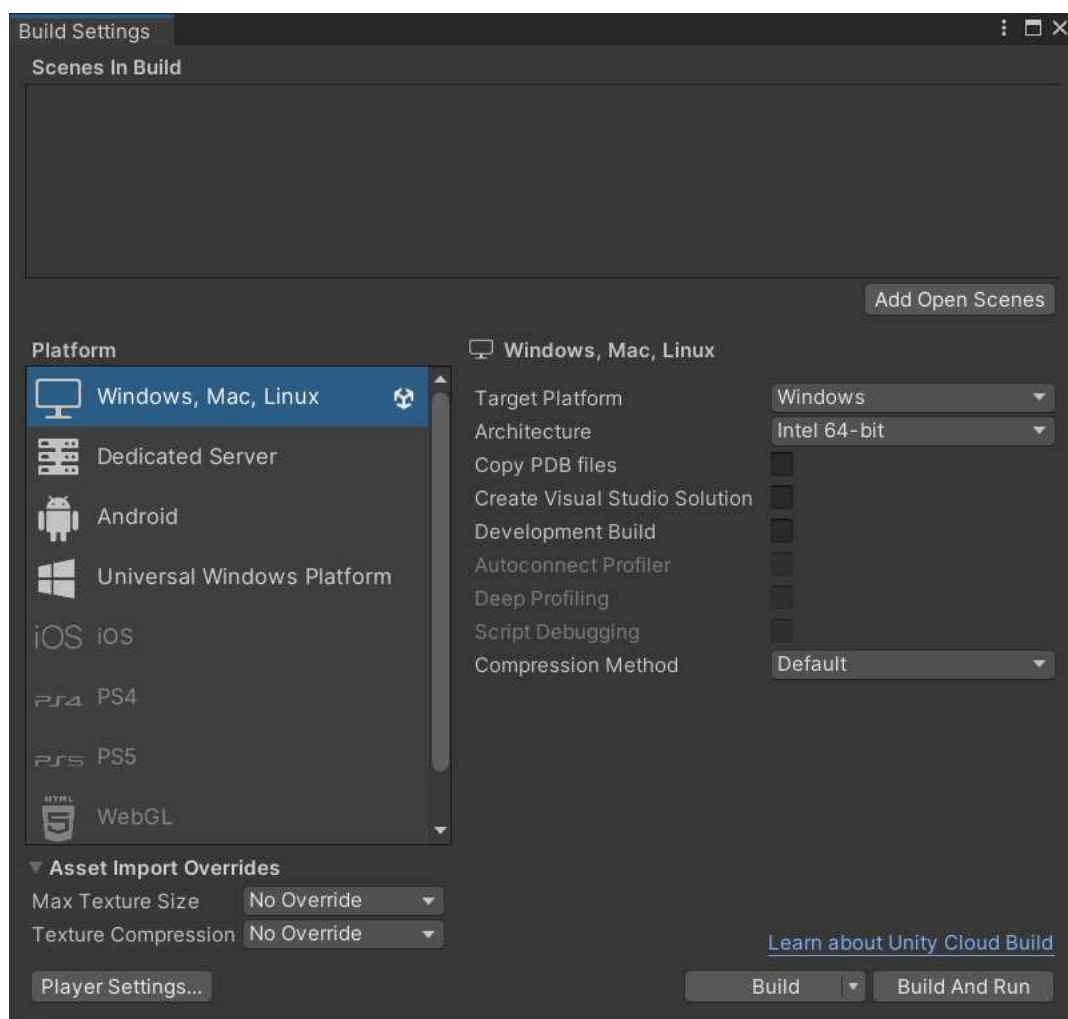


Рисунок 8 – Окно настроек проекта

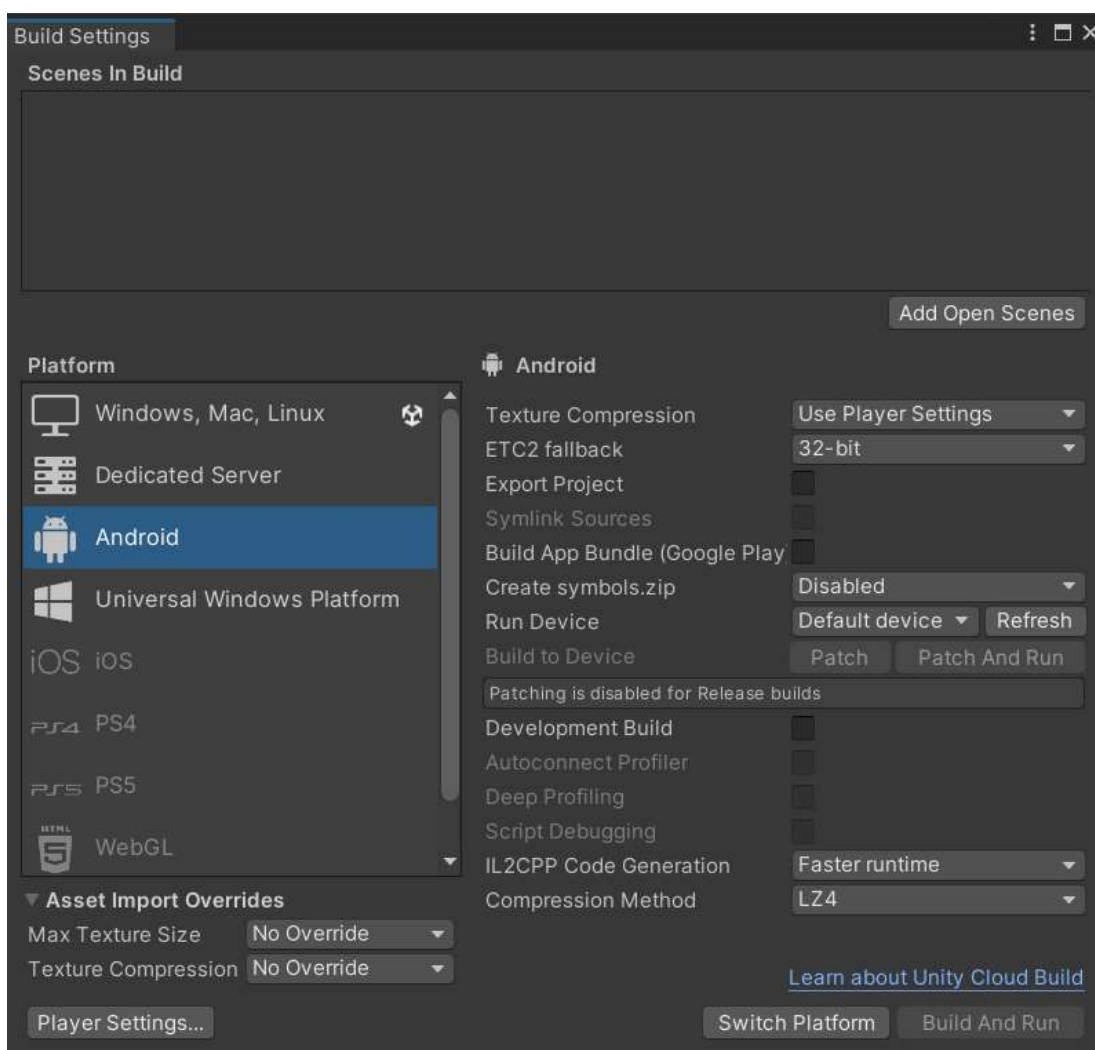


Рисунок 9 – Окно настроек проекта *Android*

После переключения платформы необходимо настроить настройки компиляции программы.

Для настройки компиляции программы необходимо:

- в левом нижнем углу настройки *Build Settings* нажать на кнопку *Player Setting* (рисунок 10);
- в открывшемся окне *Project Settings* открылась вкладка *Player*. В этой вкладке необходимо заполнить *Company Name* (название компании) и *Product Name* (название продукта). Эти поля обязательные для заполнения, даже если вы не работаете в компании поле названия компании необходимо заполнить. (рисунок 11).

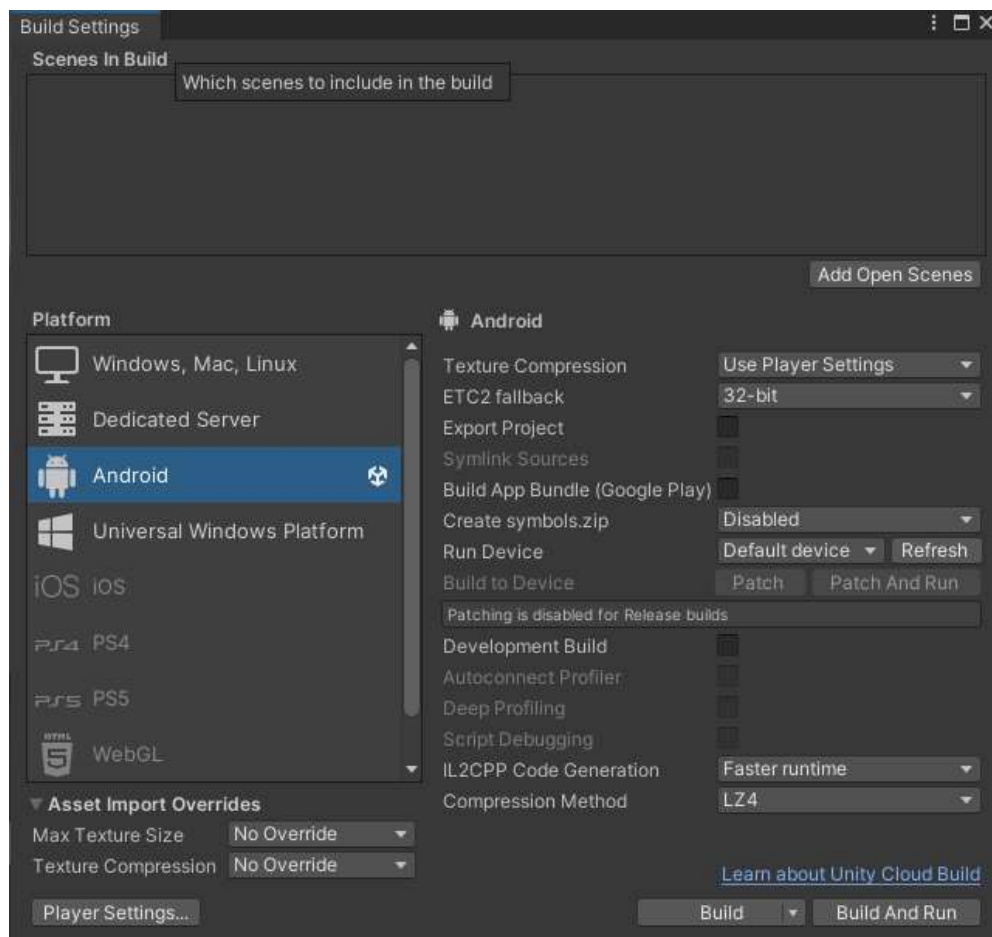


Рисунок 10 – Окно настроек платформ

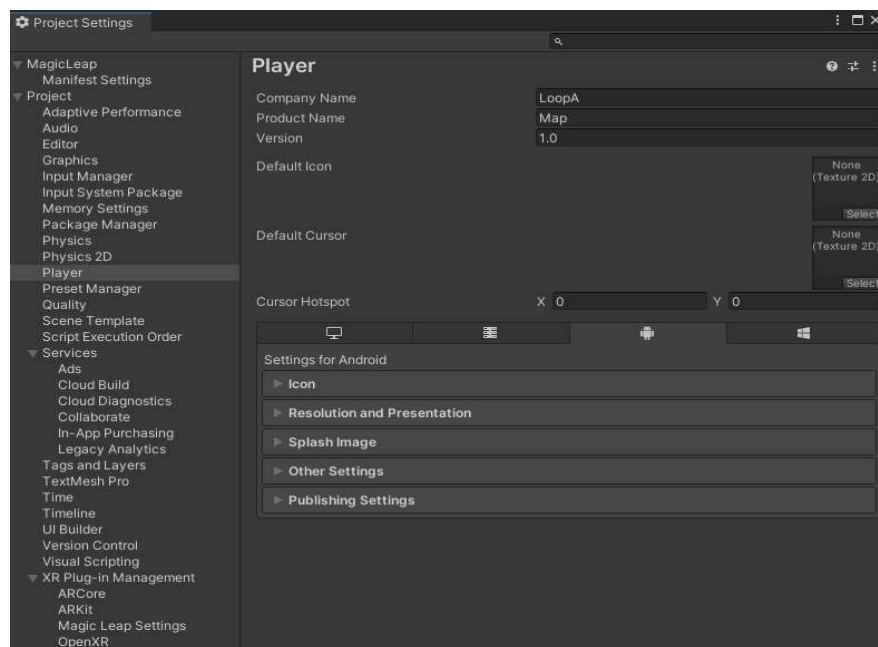


Рисунок 11 – Окно настройки проекта

Для настройки *ARCore* необходимо перейти на вкладку *XR Plug-in Management* и выбрать *ARCore* (рисунок 12).

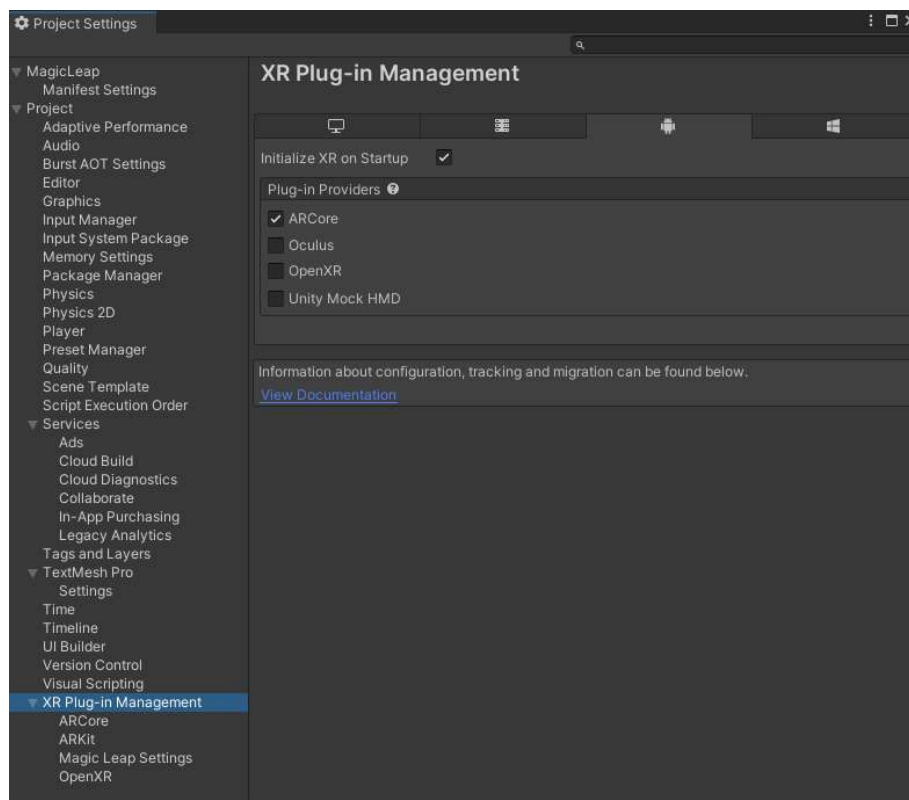


Рисунок 12 – Настройка *API*

Для создания картинок необходимо добавить *2D* компоненты в проект. Для этого в *Unity* имеется *Package Manager* (Диспетчер пакетов). Диспетчер пакетов предназначен для ускорения рабочих процессов и оптимизации размера среды выполнения за счет предоставления функций, разработанных *Unity*, в качестве дополнительных пакетов.

Для добавления пакетов *2D* необходимо:

- в верхней части приложения открыть пункт *Window* и выбрать пункт *Package Manager* (рисунок 13);
- в открывшемся окне *Package Manager* необходимо найти пакет *2D* и добавить его нажав кнопку *Install*. (рисунок 14).

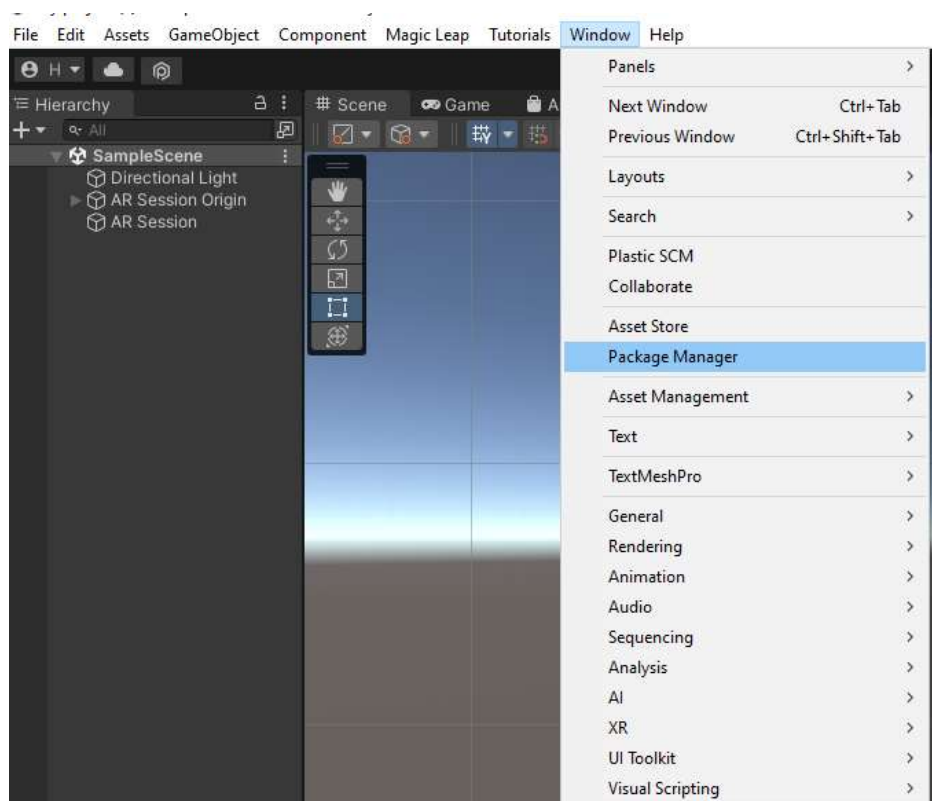


Рисунок 13 – Пункт меню *Window*

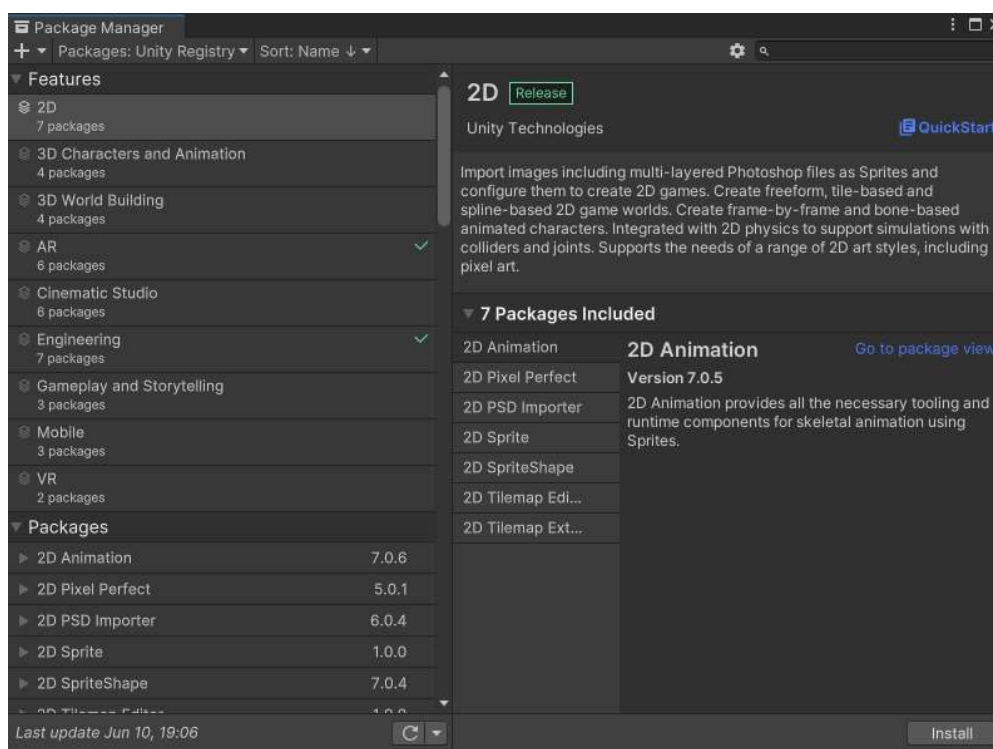


Рисунок 14 – Окно диспетчера пакетов

После проделывания всех настроек окно приложения будет выглядеть

					ПГУ 1.090501.15.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

следующим образом слева будет показываться все элементы, находящиеся на сцене, посередине сама сцена справа отражается свойства объекта, в низу папки и материалы, которые находятся в этом проекте (рисунок 15).

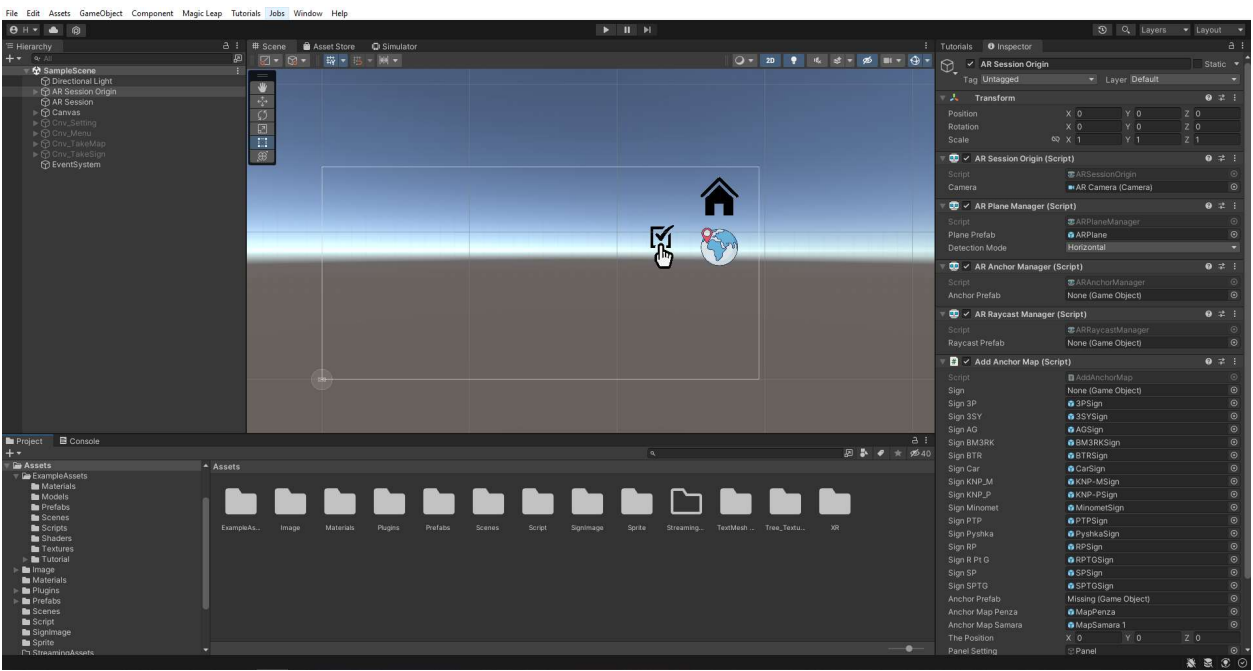


Рисунок 15 – Главная страница *Unity*

Редактировать объект можно в правой части экрана во вкладке *Inspector*. В ней добавляются свойства, скрипты, материалы, меняются размеры объекта. Также в этой вкладке можно отредактировать видимость объекта. (рисунок 16).

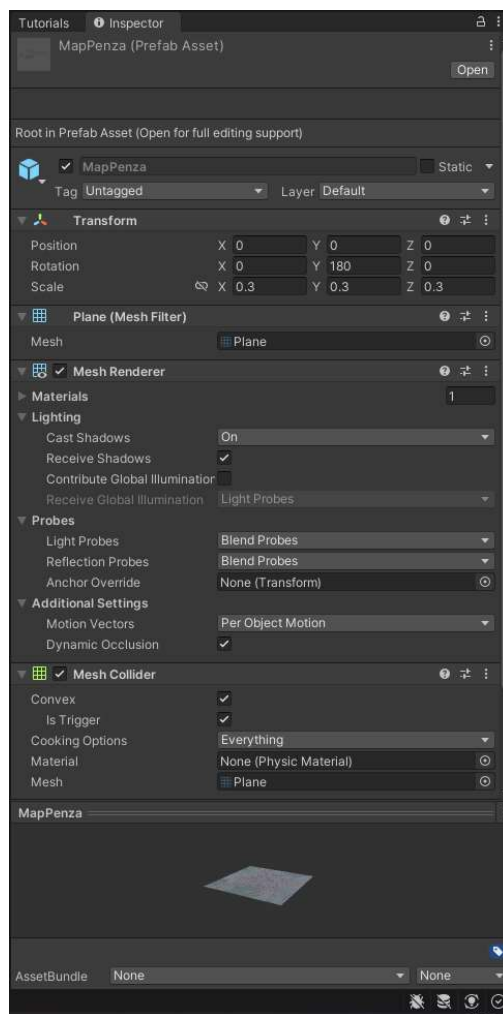


Рисунок 16 – Вкладка *Inspector*

При входе в приложение выполняется функция авторизации пользователя, если пользователь не существует выполняется функция регистрации. Авторизации/регистрации выполняется функция отображение объектов *UI* и скрывтия объектов *UI*. При успешной авторизации начинает выполняться функция перехода между сценами. Затем идет регистрация вхождения пользователя и вывод информации о пользователе на экран. При переходе в дополненную реальность выполняется функция перехода между сцен. Далее выполняется функция подключения камеры и отображения объектов *UI*. Следом идет функция анализа поверхности и вывод сетки поверхности. При постановке карты выполняется функция учета координат поставленных объектов и вывод объекта на экран. При постановке

тактических знаков выполняется функция учета координат поставленных объектов и вывод объекта на экран. При удалении карты выполняется функция удаления всех объектов и координат поставленных объектов.

На рисунке 17 представлена иерархия функций приложения.

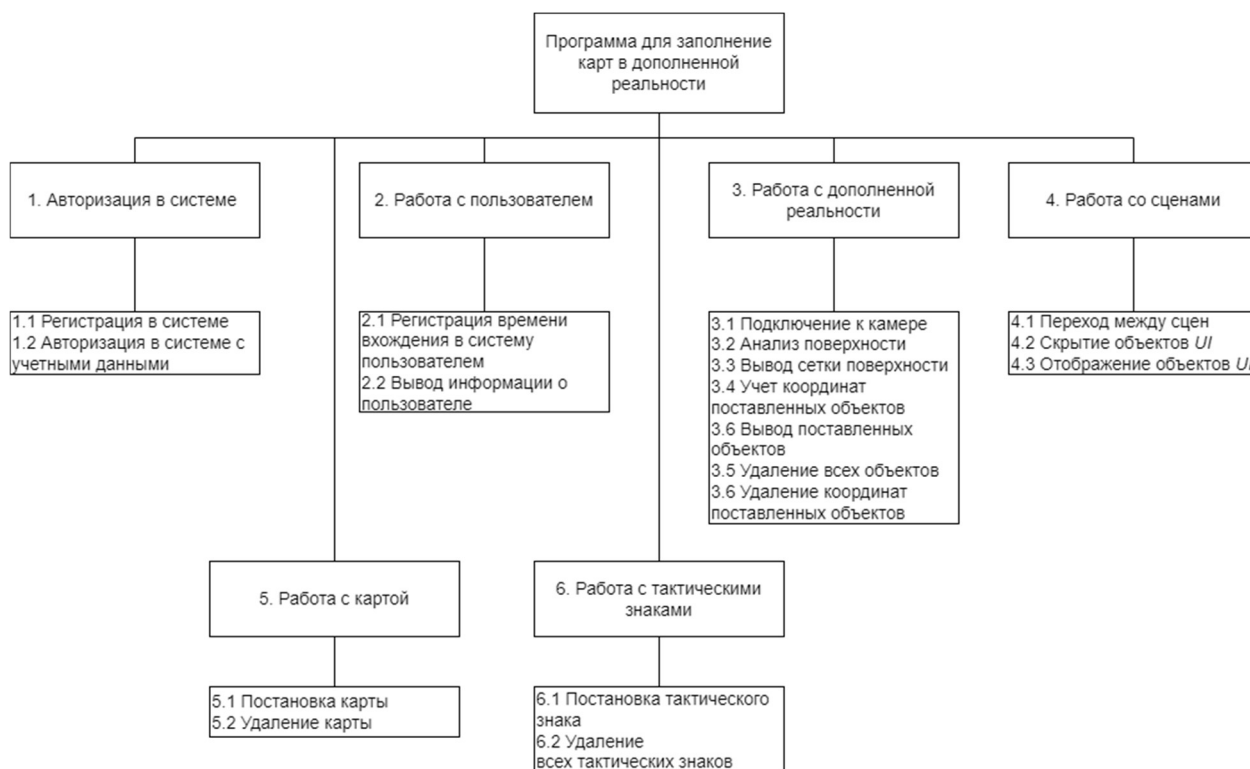


Рисунок 17 – Иерархия функций приложения

На рисунке 18 показана диаграмма классов. Она представляет собой упрощенную модель взаимодействия и иерархии элементов системы и их атрибутов [8-9].

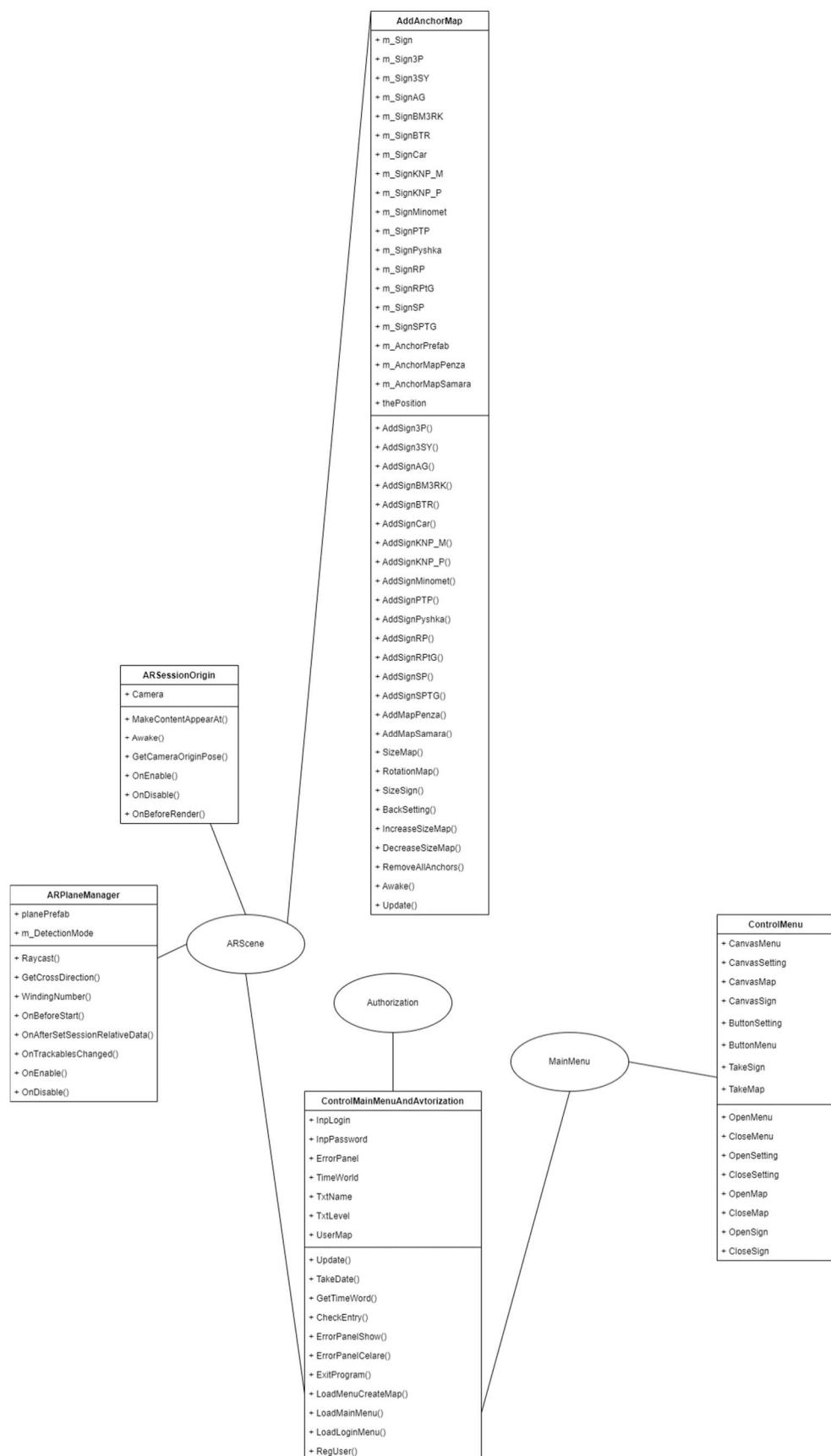


Рисунок 18 – Диаграмма классов

На диаграмме представлены следующие классы:

- *ARScene* – сцена, проецирующая через камеру дополненную реальность;
- *Authorization* – сцена, авторизации и регистрации пользователя в программе;
- *MainMenu* – сцена, после регистрации;
- *ARSessionOrigin* – *ARCore* класс для камеры, содержащая функции отображения внешнего мира:

- 1) *MakeContentatAt* – функция получения картинки с камеры;
 - 2) *Awake* – функция принимающая происходящее при движении камеры;
 - 3) *GetCameraOriginPose* – функция отслеживания позиции камеры;
 - 4) *OnEnable* – функция, отвечающая выполняющая при включении камеры;
 - 5) *OnDisable* – функция, отвечающая выполняющая при выключении камеры;
 - 6) *OnBeforeRender* – функция, отвечающая за отражении картинки полученных с камеры;
- *ARPlaneManager* – *ARCore* класс для распознавания поверхности:
 - 1) *Raycast* - функция, отвечающая падение луча на плоскость;
 - 2) *GetCrossDirection* - функция, отвечающая за получения направления;
 - 3) *WindingNumber* - функция, рассчитывающий алгоритм *Winding Number* (число витков);
 - 4) *OnBeforeStart* - функция, срабатывающая после старта сцены;
 - 5) *OnAfterSetSessionRelativeData*- функция, устанавливающая данные сеанса;
 - 6) *OnTrackablesChanged* - функция, изменения отслеживаемых

объектов;

8) *OnEnable* – функция, отвечающая выполняющая при включении камеры;

9) *OnDisable* - функция, отвечающая выполняющая при выключении камеры;

- *ControlMainMenuAndAvtorization* – класс для распознавания поверхности:

1) *Update* - функция, срабатывает после включения сцены к которой подключен скрипт и обновляется каждую секунду;

2) *TakeDate* - функция, отображения даты и времени устройства;

3) *TakeDate* - функция, получения даты и времени устройства;

4) *GetTimeWord* - функция, срабатывающая после старта сцены;

5) *CheckEntry* – функция авторизации;

6) *ErrorPanelShow* - функция, отображения панели ошибки;

8) *ErrorPanelCelare* – функция, скрытия панели ошибки;

9) *LoadMenuCreateMap* - функция, перехода на сцену с дополненной реальностью;

10) *LoadMainMenu*- функция, перехода на сцену главного меню;

11) *LoadLoginMenu*- функция, перехода на сцену авторизации;

12) *PanelRegShow* - функция, отображения панели регистрации;

13) *PanelRegCelare* – функция, скрытия панели регистрации;

14) *RegUser*– функция, регистрации пользователя;

- *ControlMenu* – класс для работы элементов меню на сцене *ARScene*:

1) *OpenMenu* - функция, отображения панели меню;

2) *CloseMenu* - функция, скрытия панели меню;

3) *OpenSetting* - функция, отображения панели настроек;

4) *CloseSetting* - функция, скрытия панели настроек;

5) *OpenMap* - функция, отображения панели выбора карт;

6) *CloseMap* - функция, скрытия панели выбора карт;

7) *OpenSign* - функция, отображения панели выбора тактических знаков;

8) *CloseSign* - функция, скрытия панели выбора тактических знаков;

- *AddAnchorMap* – класс для отображения объектов в дополненной реальности:

1) *AddSign3P*- функция, подтверждения выбора тактического знака;

2) *AddSign3SY* - функция, подтверждения выбора тактического знака;

3) *AddSignAG* - функция, подтверждения выбора тактического знака;

4) *AddSignBM3RK* - функция, подтверждения выбора тактического знака;

5) *AddSignBTR* - функция, подтверждения выбора тактического знака;

6) *AddSignCar* - функция, подтверждения выбора тактического знака;

7) *AddSignKNP_M* - функция, подтверждения выбора тактического знака;

8) *AddSignKNP_P* - функция, подтверждения выбора тактического знака;

9) *AddSignMinomet* - функция, подтверждения выбора тактического знака;

10) *AddSignPTP* - функция, подтверждения выбора тактического знака;

11) *AddSignPyshka* - функция, подтверждения выбора тактического знака;

12) *AddSignRP* - функция, подтверждения выбора тактического

знака;

13) *AddSignRPtG* - функция, подтверждения выбора тактического знака;

14) *AddSignSP* - функция, подтверждения выбора тактического знака;

15) *AddSignSPTG* - функция, подтверждения выбора тактического знака;

16) *AddMapPenza* - функция, подтверждения выбора карты;

17) *AddMapSamara* - функция, подтверждения выбора карты;

18) *BackSetting* - функция, закрытия панели настроек;

19) *RemoveAllAnchors* - функция, удаления карты и тактических знаков на ней;

20) *Awake* - функция, получение ссылки на компоненты;

21) *Update* - функция, срабатывает после включения сцены к которой подключен скрипт и обновляется каждую секунду.

1.6 Тестирование и отладка

Тестирование и отладка проводились с использованием смартфона Samsung Galaxy S8+ последнего поколения, имеющий следующие характеристики:

- система – *Android 9.0*;
- процессор – *Samsung Exynos 8895 Octa*;
- разрешение экрана – 2960x1440;
- тип дисплея – *Super AMOLED*;
- плотность пикселей – 529;
- фотокамера – 12 Мп;
- размер матрицы – 1/2.55 дюйм;
- диафрагма – f/1.7;
- вес – 200 г.

Данный телефона является мощными. Имея высокое разрешение и разрешение матрицы камеры, он дает лучшее изображение.

Для тестирования программы был использован ПК со следующими характеристиками:

- видеокарта: *NVIDIA GeForce 950*;
- процессор: *Intel Core i5-7200U 2.50GHz*;
- ОЗУ: *16 GB*;
- ОС: *Windows 10 X64*.

При включении приложения пользователь увидит окно авторизации. Изначально пользователь знает свой пароль и логин. Для ввода пароля необходимо заполнить поля ввода и нажать кнопку «Войти» (рисунок 19). При нажатии на кнопку «Выход» пользователь выйдет из программы

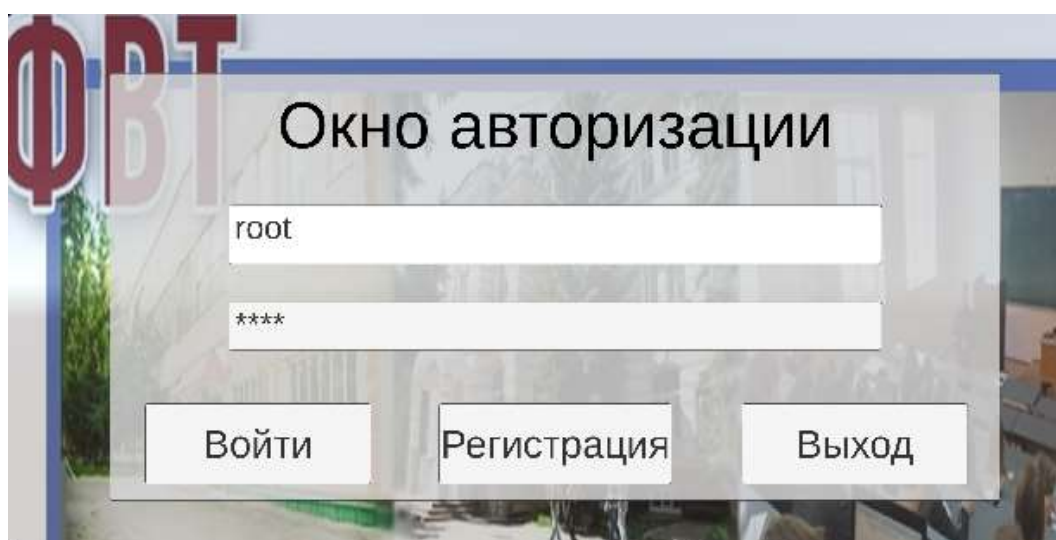


Рисунок 19– Окно авторизации

При введении неверного логина или пароля пользователь получает предупреждение. Для скрытия окна необходимо нажать на кнопку «Заккрыть» (рисунок 20)

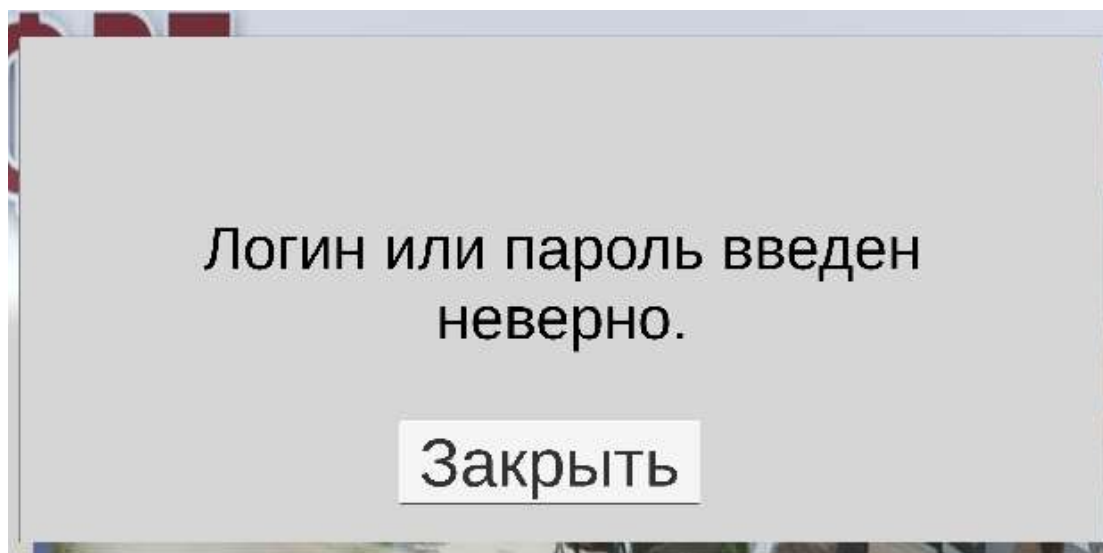


Рисунок 20 – Окно ошибки

Для создание нового пользователя необходимо нажать кнопку «Регистрация» (рисунок 19).

Далее пользователь увидит окно ввода пароля. Для введения пароля необходимо нажать на кнопку «Ввод» (рисунок 21).

При введении неверного пароля пользователь получит уведомление. Для скрытия окна необходимо нажать на кнопку «Заккрыть» (рисунок 22).

При введении верного пароля откроется окно регистрации. Для регистрации пользователя необходимо заполнить все поля и нажать на кнопку «Зарегистрировать» (рисунки 23). При нажатии на кнопку «Отмена» пользователь вернется к окну авторизации (рисунок 19). После регистрации пользователь вернется к окну авторизации (рисунок 19).

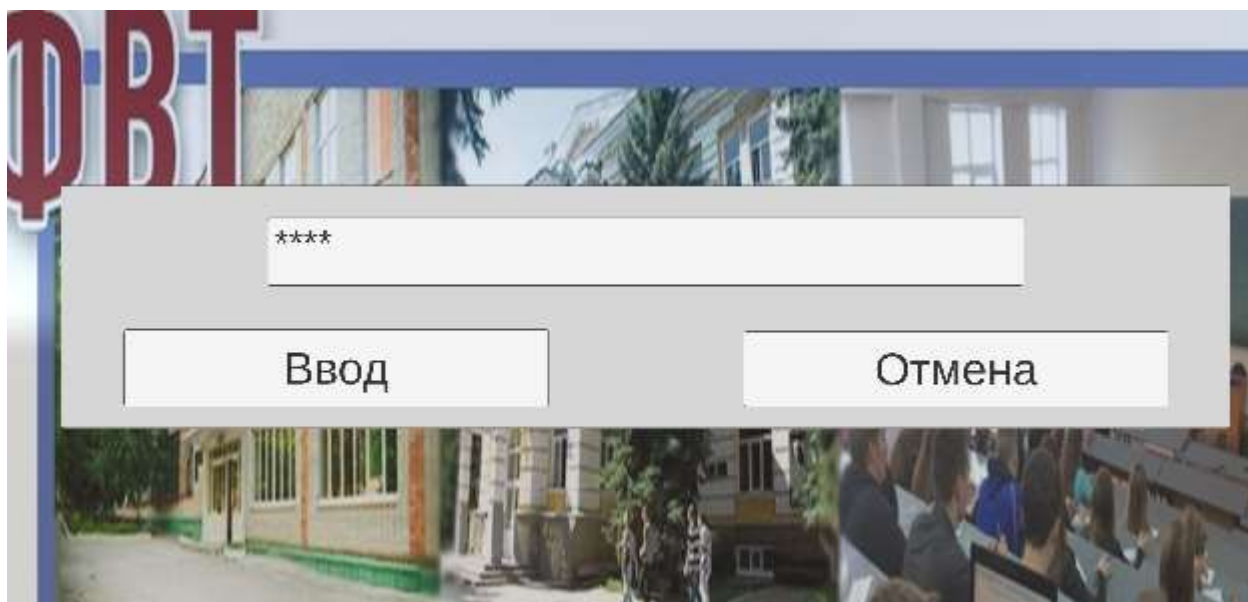


Рисунок 21 – Окно введения пароля

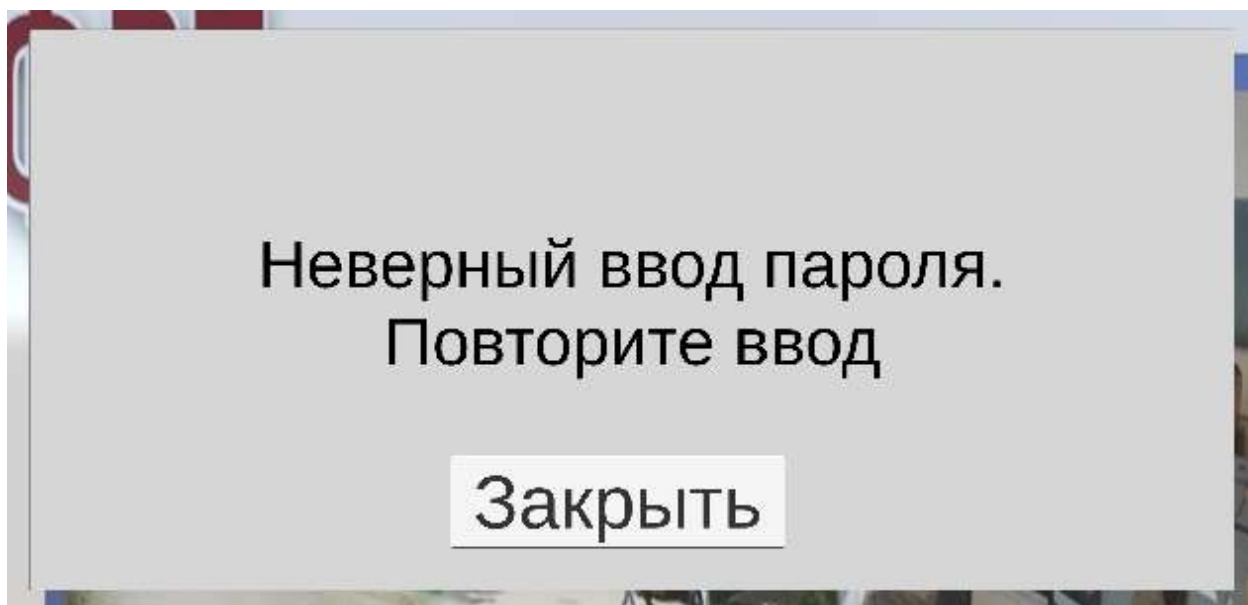


Рисунок 22 – Окно ошибки

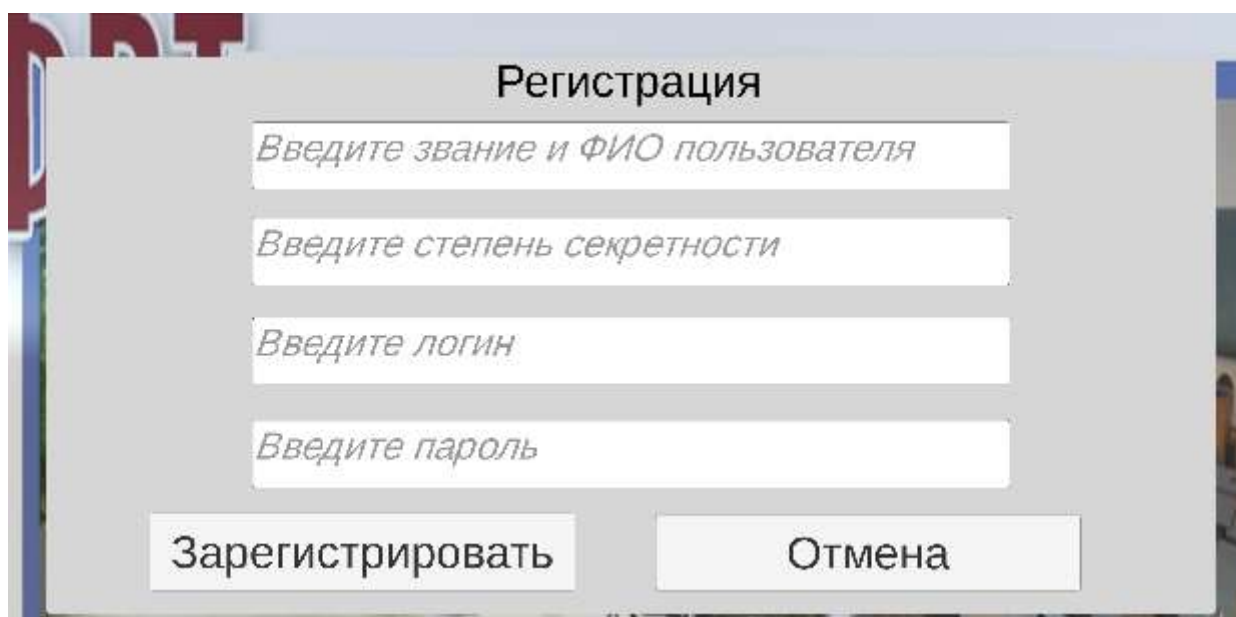


Рисунок 23 – Окно регистрации

После входа пользователь увидит окно главного меню (рисунок 24). Для перехода в дополненную реальность необходимо нажать на кнопку «Создание карты». При нажатии на кнопку «Выход» пользователь вернется к окну авторизации (рисунок 19).

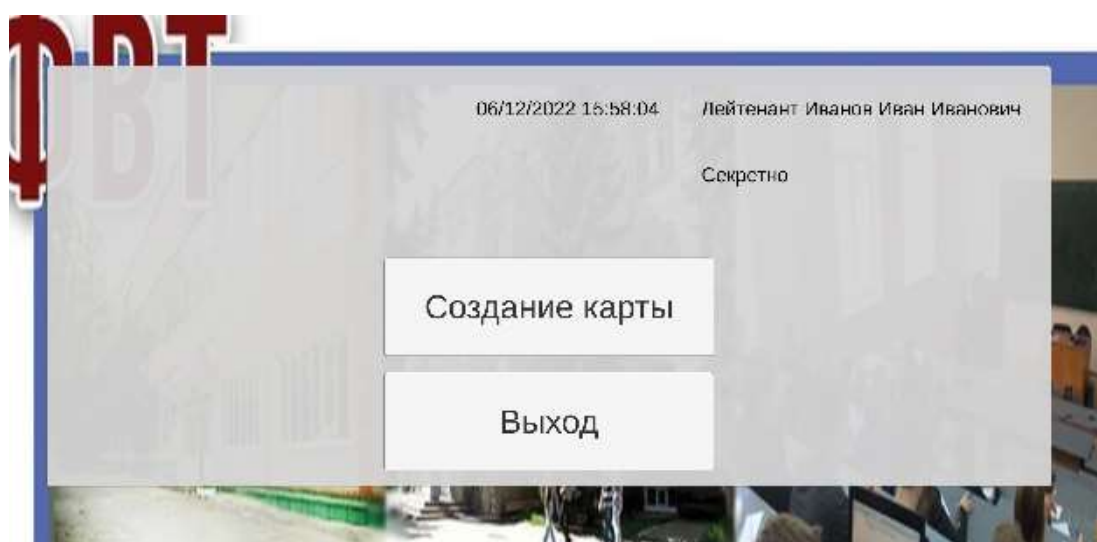


Рисунок 24 – Окно главного меню

После включения камеры пользователь видит (рисунок 25). В правом верхнем углу 3 иконки: главное меню (рисунок 26), выбор карты (рисунок 27)

и выбор тактических знаков (рисунок 28). Для начала работы необходимо выбрать карту из предложенного списка (рисунок 29). Для выбора карты необходимо нажать на кнопку с нужной картой. После выбора карты необходимо на поверхность установить карту. Это делается тыканьем на поверхность (рисунок 30).

Для расстановки тактических знаков необходимо нажать на кнопку (рисунок 28) выбрать из предложенного списка (рисунок 31). Для выбора необходимо нажать на знак. Далее необходимо на поверхность карты установить знак (рисунок 32).

Для удаления необходимо войти в главное меню. Для этого необходимо нажать на кнопку (рисунок 26). После этого нажать на кнопку «Удаление карты». При нажатии на кнопку «Назад» окно «Главное меню» закроется. При нажатии кнопки «Выход» пользователь попадет на окно главного меню (рисунок 24).

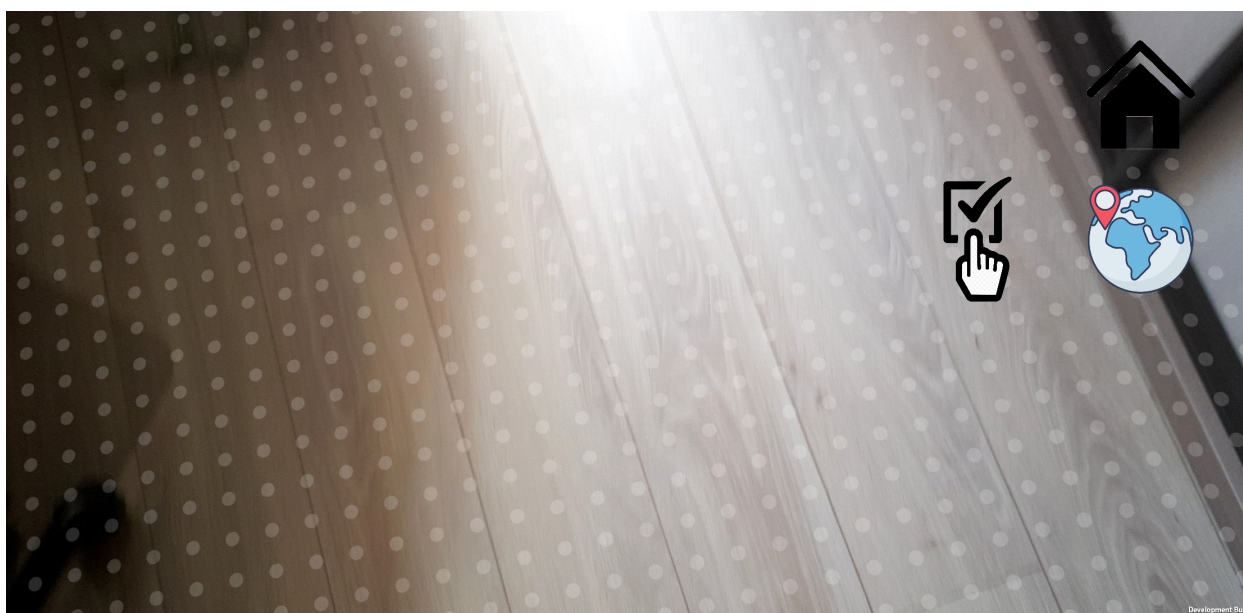


Рисунок 25 – Поле дополненной реальности



Рисунок 26 – Иконка главного меню дополненной реальности



Рисунок 27 – Иконка выбора карт



Рисунок 28 – Иконка выбора тактических знаков

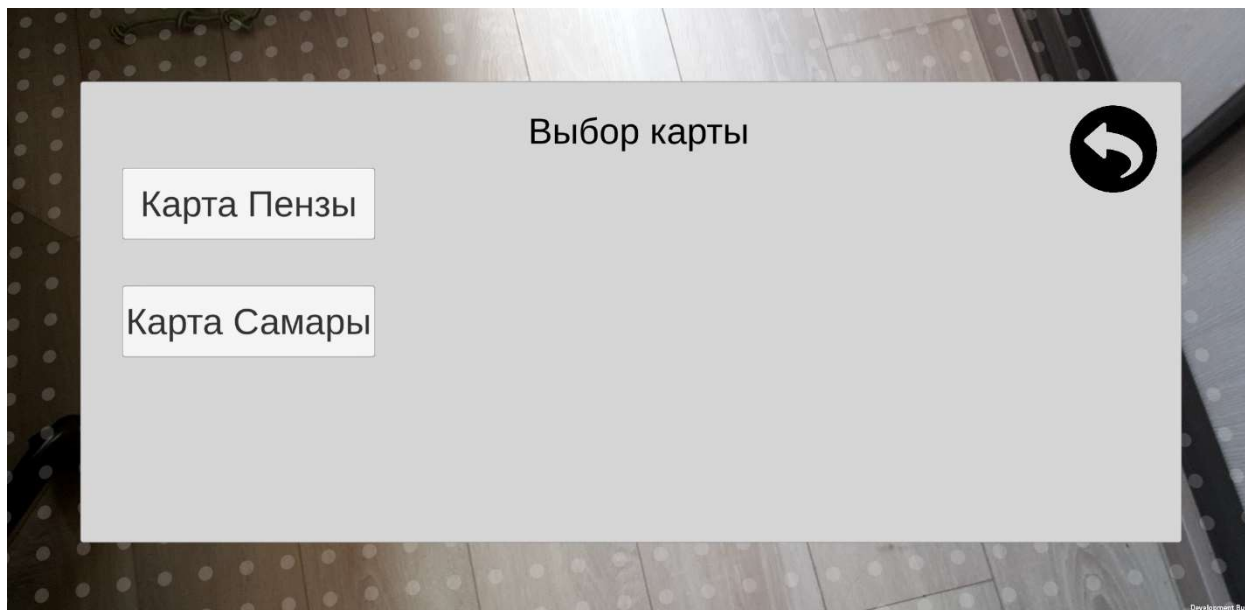


Рисунок 29 – Список карт

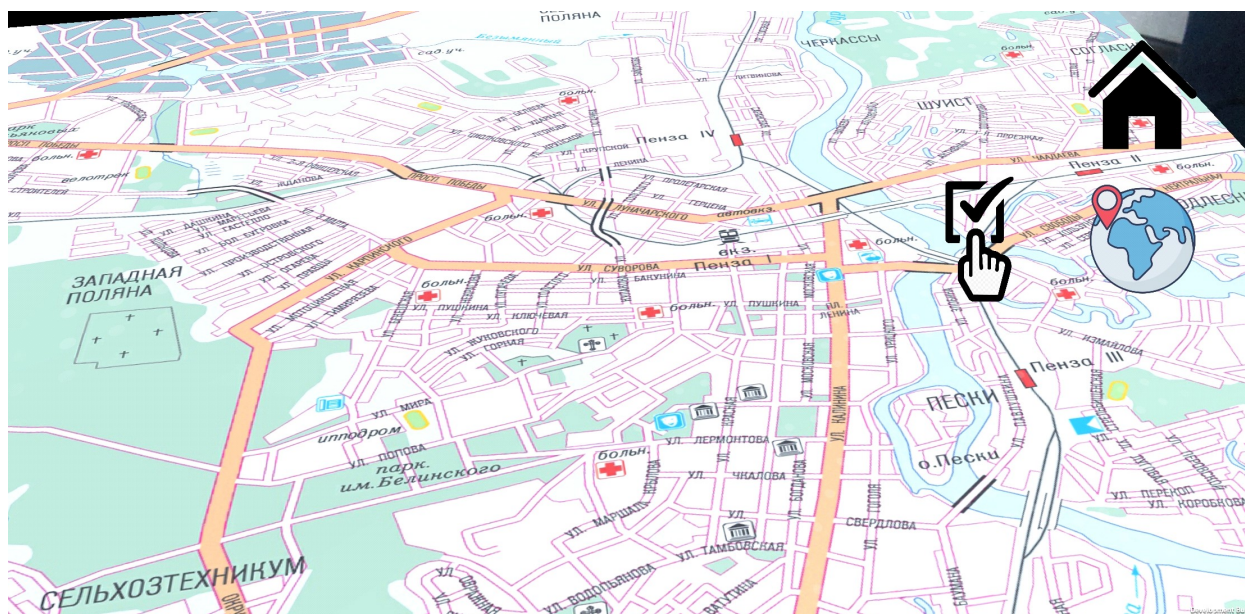


Рисунок 30 – Карта в дополненной реальности

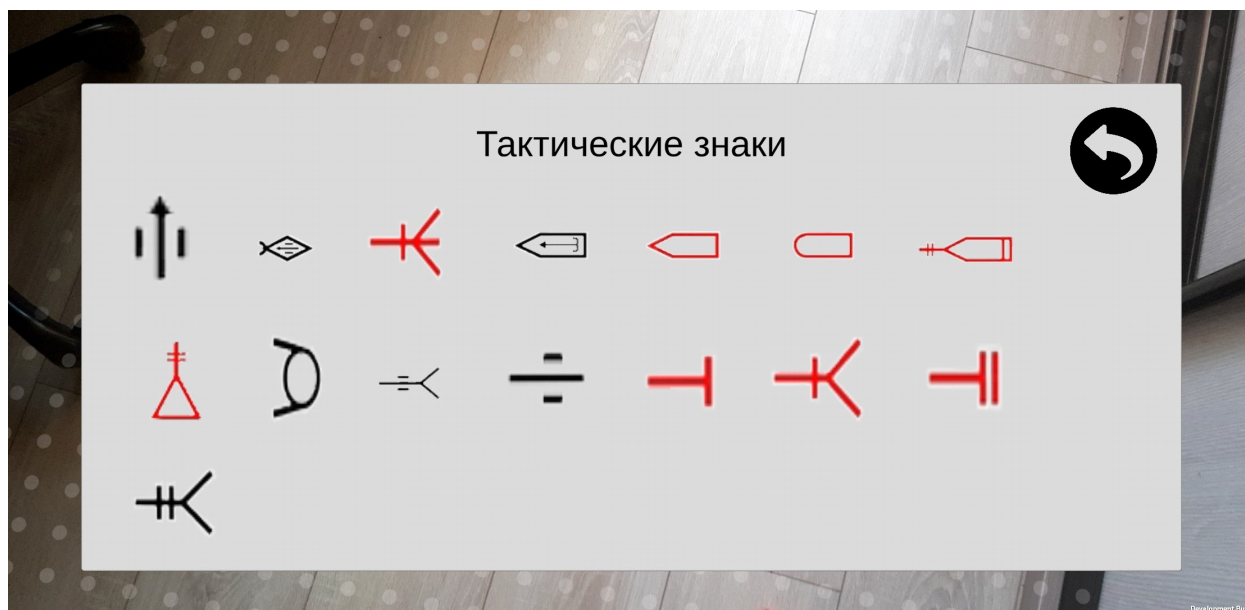


Рисунок 31 – Выбор тактических знаков

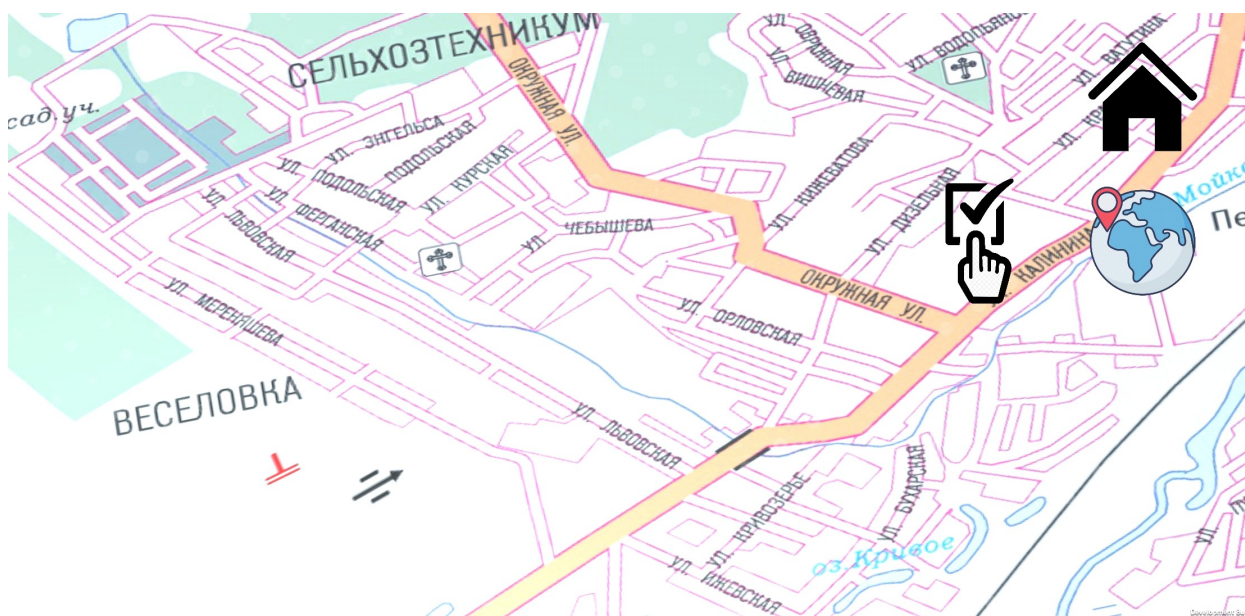


Рисунок 32 – Установка тактических знаков

Таким образом, программа выполняет все функции, описанные ранее, в полном объеме, картинка в программе плавная и с максимальным количеством кадров в секунду, что дает лучшую графику и удовлетворяет требованиям оптимизации.

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.15.001 ПЗ

Лист

45

2 Технико-экономическое обоснование разработки программы

2.1 Введение

Целью технико-экономического обоснования разработки является количественное и качественное доказательство экономической целесообразности разработки и усовершенствования программы, а также определение организационно-экономических условий ее эффективного функционирования. Эффективность программного модуля определяется его качеством и эффективностью процесса разработки и сопровождения [10]. Качество программного изделия определяется тремя составляющими:

- с точки зрения пользователя данного программного продукта;
- с позиции использования ресурсов и их оценки;
- по выполнению требований на программное изделие.

Программное изделие должно быть разработано так, чтобы оно выполняло свои функции без лишних затрат ресурсов (оперативной памяти ПК, машинного времени, пропускной способности каналов передачи данных и др. – на стадии функционирования; времени разработки и денежных ресурсов – на стадии использования программного изделия). Содержание технико-экономического обоснования разработки программы заключается в следующем:

- расчет технико-экономических показателей и выбор базы сравнения;
- определение трудоемкости и стоимости программного обеспечения;
- расчет цены ПО;
- расчет капитальных и эксплуатационных затрат на разработку;
- определение показателей финансово-экономической эффективности.

Приведенные в данном разделе дипломного проекта результирующие таблицы, перечни данных, позволяют сопоставить результаты разработки и затраты на нее, чтобы сделать вывод об эффективности проекта. Исходные

данные для расчета экономических показателей приведены в таблице 1.

Таблица 1 — Исходные данные для расчета экономических показателей

Обозначение	Наименование показателя	Единицы измерения	Значение показателя		
СЭВМ	Стоимость ЭВМ и телефона	тыс. руб	70		
ДМ	Среднее количество рабочих дней в месяце	дни	25		
ρ_n	Норматив рентабельности	-	0,24		
ω_d	Коэффициент, учитывающий дополнительную заработную плату разработчика программы	-	0,2		
ω_c	Коэффициент, учитывающий начисления органам социального страхования	-	0,27		
ω_n	Коэффициент, учитывающий накладные расходы организации	-	1,35		
qI	Количество I-задач, решаемых потребителем	Зад. год	60		
$tM.B.I$	Время решения I-ой задачи разработанной программой	Маш. час	1		
nn	Количество организаций, которые приобретут данную программу	шт.	1г	2г	3г
			4	5	8
$ZЭЛ$	Тариф за 1 кВт/час	Руб.	3,44		
$ТС$	Срок службы разработанной программы	год	5		
$НДС$	Налог на добавленную стоимость	%	20		
TP	Количество рабочих дней в году	дн	264		
NCM	Количество смен работы ЭВМ	-	1		
tCM	Продолжительность смены	ч	8		
α	Простои ЭВМ	%	10		
P	Мощность, потребляемая ЭВМ	кВт	0,4		
NCP	Среднее количество ремонтов в год	-	2		
SD	Стоимость деталей, заменяемых при ремонте	Руб.	2500		

2.2 Расчёт затрат на разработку программного обеспечения

Суммарные затраты на разработку проекта рассчитываются по следующей формуле:

$$S_{PI} = S_{3PI} + S_{HAK}, \quad (1)$$

где S_{3PI} – затраты по заработной плате инженера-программиста;

$S_{НАК}$ – накладные расходы.

Затраты по заработной плате инженера-программиста рассчитываются по формуле:

$$S_{ЗП} = O_{ЗП} * (1 + \omega_c) * (1 + \omega_o) * tp_i, \quad (2)$$

где $O_{ЗП}$ – основная заработная плата инженера-программиста за месяц (20000 руб.);

tp_i – время, необходимое для разработки программы программистом i -го разряда (чел.-мес.);

ω_o – коэффициент, учитывающий дополнительную заработную плату разработчика программы, в долях к сумме основной заработной платы;

ω_c – коэффициент, учитывающий начисления органам социального страхования на заработную плату разработчика программы, в долях к сумме основной заработной плате разработчика.

Система разрабатывалась 30 дней, если учесть, что в одном месяце 25 рабочих дней, то:

$$tp_{12} = 30/25 = 1,2 \text{ (чел./Мес)}. \quad (3)$$

Таким образом, затраты по заработной плате инженера-программиста составят:

$$S_{ЗП} = 10000 * (1 + 0,268) * (1 + 0,20) * 1,2 = 18258,52 \text{ (руб.)}$$

Накладные затраты рассчитываются с учетом ω_n – коэффициента, определяющего уровень накладных расходов организации по формуле:

$$S_{НАК} = O_{ЗП} * \omega_n * tp_{12}, \quad (4)$$

$$S_{НАК} = 20000 * 1,35 * 1,2 = 32400 \text{ (руб.)}.$$

Таким образом, суммарные затраты на разработку системы составляют:

$$S_{РП} = 18258,52 + 32400 = 50658,52 \text{ (руб.)}.$$

2.3 Расчет цены разработанной программы

Оптовая цена разработанной программы определяется по следующей формуле:

$$Z_{\Pi} = S_{\Pi\Pi} + \Pi, \quad (5)$$

где Z_{Π} – оптовая цена (цена разработчика) (руб.);

$S_{\Pi\Pi}$ – суммарные затраты на разработку программы (руб.);

Π – прибыль, рассчитанная по формуле:

$$\Pi = \rho_n * S_{\Pi\Pi}, \quad (6)$$

где ρ_n – норматив рентабельности, учитывающий прибыль организации, разрабатывающей данную программу в долях ко всем затратам данной организации на разработку программы.

Итак,

$$Z_{\Pi} = S_{\Pi\Pi} * (1 + \rho_n), \quad (7)$$

$$Z_{\Pi} = 50658,52 * (1 + 0,24) = 62815,92 \text{ (руб.)}.$$

Розничная цена программы рассчитывается с учетом налога на добавленную стоимость (НДС = 20%) по формуле:

$$Z_{\Pi p} = Z_{\Pi} + H_{\text{ДС}} = Z_{\Pi} + (Z_{\Pi} * 20) / 100 = Z_{\Pi} * (1 + 0,20) \quad (8)$$

$$Z_{\Pi p} = 62815,92 * (1 + 0,20) = 75379,1 \text{ (руб.)}.$$

Выручка от продаж при условии пп – количество организаций, желающих приобрести программу, составит:

$$B = Z_{\Pi p} * n_n \quad (9)$$

$$B1 = 75379,1 * 4 = 301516,41 \text{ (руб.)},$$

$$B2 = 75379,1 * 5 = 376895,52 \text{ (руб.)},$$

$$B3 = 75379,1 * 8 = 603032,83 \text{ (руб.)}.$$

2.4 Расчет капитальных вложений

Капиталовложения, связанные с работой ЭВМ рассчитываются по формуле:

					ПГУ 1.090501.15.001 ПЗ	Лист
						49
Изм.	Лист	№ докум.	Подпись	Дата		

$$K_{ЭВМ} = C_{ЭВМ} + S_3, \quad (10)$$

где $C_{ЭВМ}$ – стоимость ЭВМ (руб.);

S_3 – стоимость запасных частей (руб.).

Итак, произведем расчет коэффициентов, входящих в формулу расчета величины капиталовложений:

$$S_3 = 0,15 * C_{ЭВМ}; \quad (11)$$

$$S_3 = 0,15 * 70000 = 10500 \text{ (руб.)};$$

Капиталовложения в ЭВМ и телефона составляет:

$$K_{ЭВМ} = 70000 + 10500 = 80500 \text{ (руб.)}.$$

2.5 Расчет эксплуатационных расходов

Эксплуатационные расходы на ЭВМ рассчитываются по формуле:

$$E = (T_{MB} * E_ч) + Zn/Tc, \quad (12)$$

где T_{MB} – машинное время для решения задач с помощью разработанной программы, (маш.час/год);

$E_ч$ – эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ;

Zn – цена, по которой продается программа (руб.);

Tc – срок службы программы (г).

Полезный фонд времени работы ЭВМ рассчитывается по формуле:

$$T_{ПОЛ} = T_{ОБЩ} * t_{СМ} * N_{СМ} * (1 - \alpha/100), \quad (13)$$

где $T_{ОБЩ}$ – общий фонд времени работы ЭВМ (дни); $T_{ОБЩ} = T_P$; $N_{СМ}$ – количество смен работы ЭВМ;

$t_{СМ}$ – время одного рабочего дня (час);

α – простои ЭВМ (в % от общего фонда времени работы ЭВМ). Полезный фонд времени работы ЭВМ получим:

$$T_{ПОЛ} = 264 * 8 * 1 * (1 - 10/100) = 1900,8 \text{ (Маш. час /год)}.$$

Машинное время для решения задач с помощью данной программы рассчитывается по формуле:

$$T_{MB} = qI * t_{MBI}, \quad (14)$$

где qI – количество I -задач, решаемых потребителем в год(шт.);

t_{MBI} – время решения I -ой задачи, разработанной программой (Маш. час).

$$T_{MB} = 60 * 1 = 60 \text{ (Маш. час / год)}.$$

Эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ, оцениваются по формуле:

$$E_{\text{ч}} = (A_o + S_{3п} + S_{эл} + R_{pm}) / T_{пол}, \quad (15)$$

где A_o – амортизационные отчисления (руб.);

$S_{3п}$ – затраты по заработной плате инженера в год (руб./год);

$S_{эл}$ – стоимость потребляемой энергии (руб.);

R_{pm} – затраты на ремонт ЭВМ (руб.);

$T_{пол}$ – полезный годовой фонд работы ЭВМ, (маш.час/год).

Амортизационные отчисления рассчитываются с учетом нормы амортизации ($A_n = 12,5 \%$);

$$A_o = A_n * K_{ЭВМ} = 0,125 * 80500 = 10062 \text{ (руб.)}. \quad (16)$$

Затраты по заработной плате инженера за год рассчитывается по формуле:

$$S_{3п} = (1 + \alpha\sigma) * (1 + \alpha\omega) * O_{3п} * 12, \quad (17)$$

где $\alpha\sigma$ – коэффициент, учитывающий начисления органам социального страхования на заработную плату разработчика программы, в долях к сумме основной заработной плате разработчика;

$\alpha\omega$ – коэффициент, учитывающий дополнительную заработную плату разработчика программы, в долях к сумме основной заработной платы;

$O_{3п}$ – основная заработная плата инженера за месяц 8-го разряда (20000 руб.).

Рассчитываем годовые затраты по заработной плате и социальным отчислениям для инженера:

$$S_{3п} = (1 + 0,268) * (1 + 0,20) * 20000 * 12 = 365184 \text{ (руб./ год)}$$

Стоимость потребляемой энергии оценивается по формуле:

$$S_{эл} = P * T_{пол} * Z_{эл}, \quad (18)$$

где P – мощность, потребляемая ЭВМ (кВт);

$T_{пол}$ – полезный годовой фонд работы ЭВМ (маш.час/год);

$Z_{эл}$ – тариф за 1 кВт/час (руб. /кВт).

Итак, произведем расчет стоимости потребляемой энергии:

$$S_{эл} = 0,4 * 1900,8 * 3,44 = 2202,7 \text{ (руб.)}.$$

Затраты на ремонт ЭВМ вычисляются по формуле:

$$R_{PM} = N_{CP} * S_D, \quad (19)$$

где N_{CP} – среднее количество ремонтов в год;

S_D – стоимость деталей, заменяемых при одном ремонте, в среднем.

$$R_{PM} = 2 * 2500 = 5000 \text{ (руб.)}.$$

Произведем вычисление эксплуатационных расходов, приходящихся на 1 час работы ЭВМ:

$$E_q = (10062 + 365184 + 2202,7 + 5000) / 1900,8 = 201 \text{ (руб./час)}.$$

Далее вычислим эксплуатационные расходы на ЭВМ:

$$E = (60 * 201) + 62815,92 / 5 = 24653,184 \text{ (руб.)}.$$

2.6 Определение расходов на интернет трафик

Для облегчения работы программиста и повышения его производительности в проекте предусмотрено использование глобальной сети интернет.

В настоящее время на пензенском рынке интернет очень много провайдеров предлагающие свои услуги. Цены, предлагаемые основными провайдерами, сведем в таблицу 2, для анализа и экономически целесообразного выбора поставщика услуги по интернет трафику.

Таблица 2 — Таблица интернет трафиков и провайдеров

Название провайдера	Предлагаемый тариф	Стоимость тарифа(руб.)	Скорость доступа(Мбит/с)
Ростелеком	Пакет-Для Впечатлений	699	100
МТС	Безлимитище	450	100
Дом.РУ	Скорость света	550	100

Т.к. вышеперечисленные провайдеры предлагают безлимитные интернет трафики с одинаковой скоростью доступа в сеть интернет, то экономически целесообразно выбрать трафик с самой низкой стоимостью. Выбираем ПАО "МТС", тариф "Безлимитище", с абонентской платой в месяц 450 рублей.

2.7 Технико-экономические показатели разработки

Обобщенные технико-экономические показатели разработки программы сведены в таблицу 3

Таблица 3 — Технико-экономические показатели разработки

Показатель	Значение
Капитальные вложения (руб.)	80500
Эксплуатационные расходы (руб.)	24653,184
Затраты на разработку ПО (руб.)	50658,52
Затраты на интернет трафик (руб.)	450
Затраты на заработную плату программисту (руб.)	30432
Суммарно (руб.)	186693,704

3 Безопасность жизнедеятельности. Эргономика рабочего места программиста

3.1 Характеристика условий труда программиста

В наше время программистам приходится работать почти со всей электроникой, окружающей нас, начиная от компьютеров и телефонов и заканчивая умными холодильниками и беспилотными машинами. Из-за этого всего от программистов требуется увеличенная трата физической и умственной энергии. Что бы не допускать травм или других калечащих эффектов. Необходимо комплексно решать вопросы в сферах оборудования рабочих мест, контроля уровня шума, светом и воздушной средой.

В наши дни работа с ПК и телефонами интенсивно применяется во всех сферах работы человека. Такая деятельность подвергает человека влиянию ряда опасных электромагнитных излучений (СВЧ, УВЧ, ВЧ), шума, статики и других.

Взаимодействие с ПК и телефонами сопровождается увеличенной умственной нагрузкой на пользователя, постоянной зрительной работой и работой с клавиатурой. Несомненная важность имеет правильное оборудование и размещение рабочих мест пользователей и разработчиков.

Также в процессе своей деятельности работникам следует помнить о правильном соблюдении режимов труда и отдыха. При пренебрежении данного режима у рабочих могут возникать различные симптомы, выражающиеся в головных болях, нарушениях сна, раздражительности, усталости, болях в пояснице, шее и руках.

Одними из критериев, по которым определяется пригодность рабочих пространств работников, являются требования к параметрам воздушной среды, требования к уровню шума и вибрации, требования к освещенности рабочих мест, требования к производственному оборудованию и требования к использованию телефонами.

3.2 Требования к параметрам воздушной среды

Микроклимат может меняться в широких диапазонах, необходимым условием для работы является поддержание постоянной температуры тела благодаря свойству терморегуляции, способности организма регулировать отдачу тепла в окружающей среде.

Для рабочих помещений допустимые и оптимальные значения параметров микроклимата приведены в таблице 4.

Таблица 4 — Климат рабочих помещений

Период года	Зона	Температура воздуха, С	Скорость движения воздуха, м/с	Относительная влажность воздуха, %
Холодный	Оптимальная	18-21	до 0,1	60-40
Переходный	Допустимая	17-21	до 0.3	до 75
Теплый ($t > 100$ С)	Оптимальная	20-25	до 0.3	60-40
Жаркий	Допустимая	До 28 в 13 часов самого жаркого месяца	до 0.5	до 75

Воздух, находящийся в рабочих помещениях работников ЭВМ, должен быть профильтрован от пыли, загрязнений и микроорганизмов.

Температура воздуха в помещении должна регулироваться с учетом теплового потока от оборудования. Предпочтительным выбором оборудования должна быть техника с малой электрической мощностью. Оборудование техники должно быть установлено так, чтобы тепловой поток не был направлен на рабочих. Следует ограничивать количество техники, находящейся в помещении.

Пол в помещениях должен быть ровный, из нескользкого материала, удобным для чистки и влажной уборки. Покрытия рабочих пространств должны обладать антистатическими свойствами и сохранять свои параметры в процессе эксплуатации. В помещениях должна проводиться влажная уборка.

3.3 Требования к уровню шума и вибрации

Шум оказывает не благоприятные последствия на труд человека. При длительной работе при шуме человек может начать испытывать раздражительность, головные боли, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах.

Так же под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением, ухудшается речевая коммуникация.

Главными источниками шума в рабочих комнатах с установленными в них ЭВМ являются принтеры, вентиляторы, кондиционеры, вентиляционные системы, трансформаторы, множительная техника.

Уровень шума на рабочем месте программистов не должен превышать 50 дБА. Для снижения уровня шума стены и потолок помещений, где установлены компьютеры, могут быть облицованы звукопоглощающими материалами. Уровень вибрации в помещениях вычислительных центров может быть снижен путем установки оборудования на специальные

виброизоляторы. Существует возможность использования акустических подвесных потолков.

Оборудование, уровень издаваемого шума которого превышает допустимые нормы, необходимо монтировать вне рабочих помещений.

3.4 Требования к монтажу производственного оборудования

Для обеспечения электропитанием ЭВМ и компьютеров в комнатах должна быть смонтирована отдельная сеть электроснабжения.

Периферийные устройства компьютеров Должны подключаться исключительно к линии электроснабжения компьютерной сети.

Запрещается включение в линии электроснабжения компьютерной сети энергопотребителей, не относящихся к компьютерам (вентиляторы, лампы и т.п.).

Запрещается использовать оборудование, не подключенное к защитному заземлению, если оно предусмотрено конструкцией оборудования.

Запрещается использовать технику с открытыми корпусами, если это не предусмотрено правилами работы этого оборудования.

Проведение обслуживания рабочих мест программистов должно проводиться специально подготовленным персоналом, имеющим необходимую квалификацию техника (инженера).

Электрические розетки системы электропитания необходимо располагать так, чтобы кабели электрооборудования на рабочих местах не пересекали его, мешая работе.

3.6 Требования к освещению рабочих мест

Одним из важнейших параметров для оптимальных условий труда является уровень световой среды помещения. Это рационализация организации совокупности естественного и искусственного освещения на рабочих местах.

					<i>ПГУ 1.090501.15.001 ПЗ</i>	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

Осуществление искусственного освещения комнат с рабочими местами должно осуществляться с помощью световых проемов, ориентированных на север, либо северо-восток.

В том случае, если ориентация производится в другие стороны, необходимо предусмотреть наличие средств регулирования интенсивности поступающего естественного освещения.

3.6 Расчет освещения рабочего места программиста

Расчет освещенности рабочего места сводится к выбору системы освещения, определению необходимого числа светильников, их типа и размещения. Исходя из этого, рассчитаем параметры искусственного освещения. Обычно искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые по сравнению с лампами накаливания имеют ряд существенных преимуществ:

- по спектральному составу света они близки к дневному, естественному свету;
- обладают более высоким КПД (в 1,5-2 раза выше, чем КПД ламп накаливания);
- обладают повышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);
- более длительный срок службы.

Расчет освещения производится для комнаты площадью 15м², ширина которой 5м, высота - 3 м. Воспользуемся методом светового потока. Для определения количества светильников определим световой поток, падающий на поверхность по формуле:

$$F = E * K * S * Z / n, \quad (20)$$

где F - рассчитываемый световой поток, Лм;

E - нормированная минимальная освещенность, Лк (определяется по

таблице). Работу программиста, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная освещенность будет $E = 300 \text{ Лк}$;

S - площадь освещаемого помещения (в нашем случае $S = 20 \text{ м}^2$);

Z - отношение средней освещенности к минимальной (обычно принимается равным $1,1 \dots 1,2$, пусть $Z = 1,2$); K - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае $K = 1,5$);

n - коэффициент использования, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; характеризуется коэффициентами отражения от стен (PC) и потолка (PP), значение коэффициентов PC и PP были указаны выше: $PC = 40\%$, $PP = 60\%$. Значение n определим по таблице коэффициентов использования различных светильников. Для этого вычислим индекс помещения по формуле:

$$I = S / h * (A + B), \quad (21)$$

где S - площадь помещения, $S = 20 \text{ м}^2$;

h - расчетная высота подвеса, $h = 3 \text{ м}$;

A - ширина помещения, $A = 4 \text{ м}$;

B - длина помещения, $B = 5 \text{ м}$.

Подставив значения получаем:

$$I = 20 / 3 * (4 + 5) = 60$$

Зная индекс помещения I , по таблице находим $n = 0,22$

Подставим все значения в формулу для определения светового потока F :

$$F = 300 * 1,5 * 20 * 1,2 / 0,22 = 49090 \text{ Лм}$$

Для освещения выбираем люминесцентные лампы типа ЛБ40-1, световой поток которых $F = 4320 \text{ Лк}$.

Рассчитаем необходимое количество ламп по формуле:

$$N = F/F_{\text{л}}, \quad (22)$$

где N - определяемое число ламп;

F - световой поток, $F = 49090$ Лм;

$F_{\text{л}}$ - световой поток лампы, $F_{\text{л}} = 4320$ Лм.

$$N = 49090/4320 = 11 \text{ штук.}$$

					ПГУ 1.090501.15.001 ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В процессе выполнения данной выпускной квалификационной работы проведены анализ требований, проектирование и разработка, тестирование программы для заполнения тактических карт в дополненной реальности.

В процессе анализа требований, были сформированы требования к системе. Проведён анализ вариантов использования. Разработана модель предметной области, представленная диаграммой классов. В результате выполнения выпускной квалификационной работы реализована программа для заполнения тактических карт в дополненной реальности.

Проект реализован на движке *Unity 2021* с применением *ARCore*, в качестве технологии программирования объектов системы. На основе анализа, полученных результатов, можно сделать вывод, что все функции приложения отрабатываются корректно в штатном режиме.

Таким образом, требования технического задания были выполнены в полном объёме и в установленные сроки.

					ПГУ 1.090501.15.001 ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

Список используемых источников

1. Основы военной топографии [Электронный ресурс]: Режим доступа: <https://ppt-online.org/25226> (дата обращения 01.05.22)
2. Преимущество электронных карт [Электронный ресурс]: Режим доступа: <https://providers.by/2017/04/press-release/preimushhestva-elektronnykh-kart/> (дата обращения 11.05.22)
3. *Augmented Reality (AR)*: Что это такое, как она работает, Типы и использование [Электронный ресурс] URL: <https://research.aimultiple.com/augmented-reality/> (дата обращения 15.05.22)
4. Примеры использования *AR* технологий [Электронный ресурс] URL: <https://wellsoft.pro/blog/primery-ispolzovaniya-ar-tehnologii-v-roznitse> (дата обращения 17.05.22)
5. Что такое *Unity* [Электронный ресурс]: Режим доступа: <http://web.spt42.ru/index.php/chto-takoe-unity-3d> (дата обращения 20.02.22)
6. *Unreal vs Unity* для разработки виртуальной реальности [Электронный ресурс] URL: <https://arvrjourney.com/unreal-vs-unity-for-vr-development-9e3ce634309b> (дата обращения 20.02.22)
7. *Unity vs Unreal Engine* для разработки *XR*: Какой из них Лучше? [Электронный ресурс] URL: <https://circuitstream.com/blog/unity-vs-unreal/> (дата обращения 21.02.22)
8. Архитектура приложения [Электронный ресурс]: https://codernet.ru/articles/drugoe/arxitektura_prilozhenij_opredelenie_opisanie_i_rukovodstvo/ (дата обращения 25.05.22)
9. Документирование архитектуры программного обеспечения [Электронный ресурс]: https://codernet.ru/articles/drugoe/arxitektura_prilozhenij_opredelenie_opisanie_i_rukovodstvo/ (дата обращения 01.06.22)

10. Техническо-экономическое обоснование [Электронный ресурс]:
Режим доступа: <https://studfile.net/preview/4235502/> (дата обращения
03.06.22)

Приложение А

(обязательное)

Программа заполнения тактических карт в дополненной реальности

UML-диаграмма классов

					ПГУ 1.090501.15.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

Приложение Б

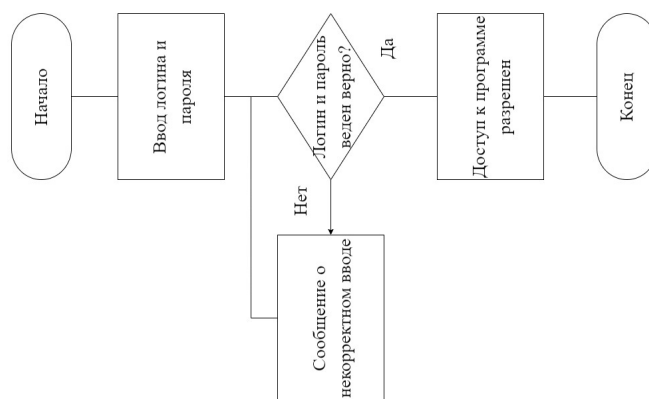
(обязательное)

Программа заполнения тактических карт в дополненной реальности

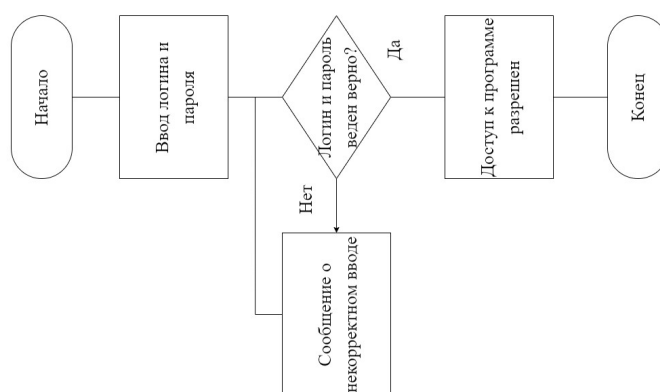
Схемы программы

					ПГУ 1.090501.15.001 ПЗ	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		

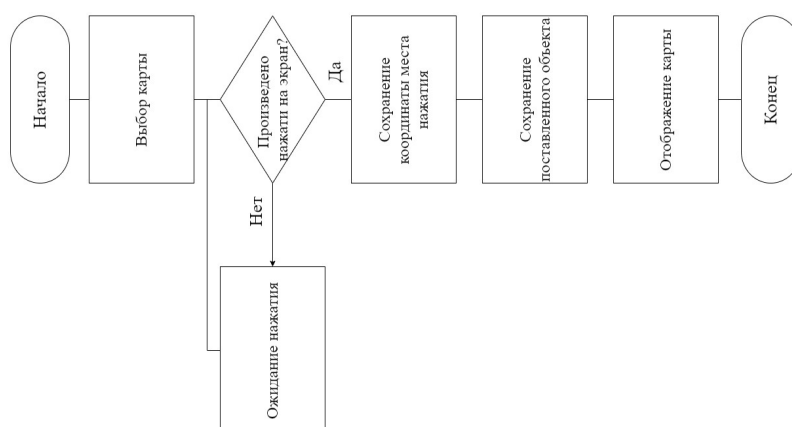
Приложение Б.1 – Окно авторизации

[illegible]

Приложение Б.2 – Окно регистрации

[illegible]

Приложение Б.3 – Постановка карты в дополненной реальности)

[illegible]

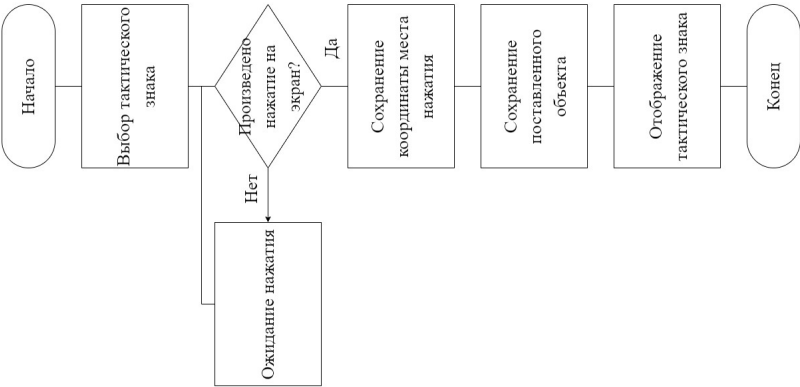
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.15.001 ПЗ

Лист

69

Приложение Б.4 – Постановка тактических знаков в дополненной реальности



ПГУ 1.090501.15.001 ПЗ									
№ п/п	№ докум.	Дата	Подпись	Инициалы	Фамилия	Имя	Отчество	Подпись	Инициалы
1	1	2020							
2	2	2020							
3	3	2020							
4	4	2020							
5	5	2020							
6	6	2020							
7	7	2020							
8	8	2020							
9	9	2020							
10	10	2020							
11	11	2020							
12	12	2020							
13	13	2020							
14	14	2020							
15	15	2020							
16	16	2020							
17	17	2020							
18	18	2020							
19	19	2020							
20	20	2020							
21	21	2020							
22	22	2020							
23	23	2020							
24	24	2020							
25	25	2020							
26	26	2020							
27	27	2020							
28	28	2020							
29	29	2020							
30	30	2020							
31	31	2020							
32	32	2020							
33	33	2020							
34	34	2020							
35	35	2020							
36	36	2020							
37	37	2020							
38	38	2020							
39	39	2020							
40	40	2020							
41	41	2020							
42	42	2020							
43	43	2020							
44	44	2020							
45	45	2020							
46	46	2020							
47	47	2020							
48	48	2020							
49	49	2020							
50	50	2020							
51	51	2020							
52	52	2020							
53	53	2020							
54	54	2020							
55	55	2020							
56	56	2020							
57	57	2020							
58	58	2020							
59	59	2020							
60	60	2020							
61	61	2020							
62	62	2020							
63	63	2020							
64	64	2020							
65	65	2020							
66	66	2020							
67	67	2020							
68	68	2020							
69	69	2020							
70	70	2020							
71	71	2020							
72	72	2020							
73	73	2020							
74	74	2020							
75	75	2020							
76	76	2020							
77	77	2020							
78	78	2020							
79	79	2020							
80	80	2020							
81	81	2020							
82	82	2020							
83	83	2020							
84	84	2020							
85	85	2020							
86	86	2020							
87	87	2020							
88	88	2020							
89	89	2020							
90	90	2020							
91	91	2020							
92	92	2020							
93	93	2020							
94	94	2020							
95	95	2020							
96	96	2020							
97	97	2020							
98	98	2020							
99	99	2020							
100	100	2020							

Приложение В

(обязательное)

Программа заполнения тактических карт в дополненной реальности

Листинг программы

					ПГУ 1.090501.15.001 ПЗ	Лист
						71
Изм.	Лист	№ докум.	Подпись	Дата		

Скрипт добавления объектов

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;

[RequireComponent(typeof(ARAnchorManager))]
[RequireComponent(typeof(ARRaycastManager))]
[RequireComponent(typeof(ARPlaneManager))]
public class AddAnchorMap : MonoBehaviour
{
    bool AddMap = false, AddSign = false;
    //Добавление данных
    [SerializeField]
    GameObject m_Sign;
    [SerializeField]
    GameObject m_Sign3P;
    [SerializeField]
    GameObject m_Sign3SY;
    [SerializeField]
    GameObject m_SignAG;
    [SerializeField]
    GameObject m_SignBM3RK;
    [SerializeField]
    GameObject m_SignBTR;
    [SerializeField]
    GameObject m_SignCar;
    [SerializeField]
    GameObject m_SignKNP_M;
    [SerializeField]
    GameObject m_SignKNP_P;
    [SerializeField]
    GameObject m_SignMinomet;
    [SerializeField]
    GameObject m_SignPTP;
    [SerializeField]
    GameObject m_SignPyshka;
    [SerializeField]
    GameObject m_SignRP;
```

```

[SerializeField]
GameObject m_SignRptG;
[SerializeField]
GameObject m_SignSP;
[SerializeField]
GameObject m_SignSPTG;
[SerializeField]
GameObject m_AnchorPrefab;
[SerializeField]
GameObject m_AnchorMapPenza;
[SerializeField]
GameObject m_AnchorMapSamara;
public Vector3 thePosition;

[SerializeField]
GameObject PanelSetting;
[SerializeField]
Button ButtonBack;
[SerializeField]
Button ButtonPlus;
[SerializeField]
Button ButtonMinus;

private GameObject Delite_AnchorPrefab, del_signTank;

public GameObject Sign
{
    get => m_Sign;
    set => m_Sign = value;
}

public GameObject Sign3P
{
    get => m_Sign3P;
    set => m_Sign3P = value;
}

public GameObject Sign3SY
{
    get => m_Sign3SY;
    set => m_Sign3SY = value;
}

```

```

    }

    public GameObject SignAG
    {
        get => m_SignAG;
        set => m_SignAG = value;
    }

    public GameObject SignBM3RK
    {
        get => m_SignBM3RK;
        set => m_SignBM3RK = value;
    }

    public GameObject SignBTR
    {
        get => m_SignBTR;
        set => m_SignBTR = value;
    }

    public GameObject SignCar
    {
        get => m_SignCar;
        set => m_SignCar = value;
    }

    public GameObject SignKNP_M
    {
        get => m_SignKNP_M;
        set => m_SignKNP_M = value;
    }

    public GameObject SignKNP_P
    {
        get => m_SignKNP_P;
        set => m_SignKNP_P = value;
    }

    public GameObject SignMinomet
    {
        get => m_SignMinomet;
    }

```

```

        set => m_SignMinomet = value;
    }

    public GameObject SignPTP
    {
        get => m_SignPTP;
        set => m_SignPTP = value;
    }

    public GameObject SignPyshka
    {
        get => m_SignPyshka;
        set => m_SignPyshka = value;
    }

    public GameObject SignRP
    {
        get => m_SignRP;
        set => m_SignRP = value;
    }

    public GameObject SignRPtG
    {
        get => m_SignRPtG;
        set => m_SignRPtG = value;
    }

    public GameObject SignSP
    {
        get => m_SignSP;
        set => m_SignSP = value;
    }

    public GameObject SignSPTG
    {
        get => m_SignSPTG;
        set => m_SignSPTG = value;
    }

    public GameObject anchorPrefab
    {

```

```

        get => m_AnchorPrefab;
        set => m_AnchorPrefab = value;
    }

    public GameObject anchorMapPenza
    {
        get => m_AnchorMapPenza;
        set => m_AnchorMapPenza = value;
    }

    public GameObject anchorMapSamara
    {
        get => m_AnchorMapSamara;
        set => m_AnchorMapSamara = value;
    }

    public void AddSign3P()
    {
        m_Sign = m_Sign3P;
        AddSign = true;
    }

    public void AddSign3SY()
    {
        m_Sign = m_Sign3SY;
        AddSign = true;
    }

    public void AddSignAG()
    {
        m_Sign = m_SignAG;
        AddSign = true;
    }

    public void AddSignBM3RK()
    {
        m_Sign = m_SignBM3RK;
        AddSign = true;
    }

    public void AddSignBTR()

```

```

{
    m_Sign = m_SignBTR;
    AddSign = true;
}

public void AddSignCar()
{
    m_Sign = m_SignCar;
    AddSign = true;
}

public void AddSignKNP_M()
{
    m_Sign = m_SignKNP_M;
    AddSign = true;
}

public void AddSignKNP_P()
{
    m_Sign = m_SignKNP_P;
    AddSign = true;
}

public void AddSignMinomet()
{
    m_Sign = m_SignMinomet;
    AddSign = true;
}

public void AddSignPTP()
{
    m_Sign = m_SignPTP;
    AddSign = true;
}

public void AddSignPyshka()
{
    m_Sign = m_SignPyshka;
    AddSign = true;
}

```

```

public void AddSignRP()
{
    m_Sign = m_SignRP;
    AddSign = true;
}

public void AddSignRPtG()
{
    m_Sign = m_SignRPtG;
    AddSign = true;
}

public void AddSignSP()
{
    m_Sign = m_SignSP;
    AddSign = true;
}

public void AddSignSPTG()
{
    m_Sign = m_SignSPTG;
    AddSign = true;
}

public void AddMapPenza()
{
    m_AnchorPrefab = m_AnchorMapPenza;
    AddMap = true;
}

public void AddMapSamara()
{
    m_AnchorPrefab = m_AnchorMapSamara;
    AddMap = true;
}

// Изменение размера карты
public void SizeMap()
{
    PanelSetting.SetActive(false);
    ButtonBack.gameObject.SetActive(true);
    ButtonPlus.gameObject.SetActive(true);
    ButtonMinus.gameObject.SetActive(true);
}

```

```

    }

    // Поворот карты
    public void RotationMap()
    {
        PanelSetting.SetActive(false);
        ButtonBack.gameObject.SetActive(true);
        ButtonPlus.gameObject.SetActive(true);
        ButtonMinus.gameObject.SetActive(true);
    }

    // Изменение размера значков
    public void SizeSign()
    {
        PanelSetting.SetActive(false);
        ButtonBack.gameObject.SetActive(true);
        ButtonPlus.gameObject.SetActive(true);
        ButtonMinus.gameObject.SetActive(true);
    }

    // Назад к меню настроек
    public void BackSetting()
    {
        PanelSetting.SetActive(true);
        ButtonBack.gameObject.SetActive(false);
        ButtonPlus.gameObject.SetActive(false);
        ButtonMinus.gameObject.SetActive(false);
    }

    // Увеличить размер карты
    public void IncreaseSizeMap()
    {
        var originSizeX = Delite_AnchorPrefab.transform.localScale.x;
        var originSizeZ = Delite_AnchorPrefab.transform.localScale.z;

        Delite_AnchorPrefab.transform.localScale = new Vector3(originSizeX *
2, Delite_AnchorPrefab.transform.localScale.y, originSizeZ * 2);

        //Сделать на все знаки, добавить вращение карты со знаками, добавить
размер только знаков
        var originSizeSignX = del_signTank.transform.localScale.x;
        var originSizeSignZ = del_signTank.transform.localScale.z;

        del_signTank.transform.localScale = new Vector3(originSizeSignX * 2,
del_signTank.transform.localScale.y, originSizeSignZ * 2);

```

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.15.001 ПЗ

Лист

79

```

    }

    //Уменьшить размер карты
    public void DecreaseSizeMap()
    {
        var originSizeX = Delite_AnchorPrefab.transform.localScale.x;
        var originSizeZ = Delite_AnchorPrefab.transform.localScale.z;

        Delite_AnchorPrefab.transform.localScale = new Vector3(originSizeX /
2, Delite_AnchorPrefab.transform.localScale.y, originSizeZ / 2);
        //Сделать на все знаки, добавить вращение карты со знаками, добавить
размер только знаков
        var originSizeSignX = del_signTank.transform.localScale.x;
        var originSizeSignZ = del_signTank.transform.localScale.z;

        del_signTank.transform.localScale = new Vector3(originSizeSignX / 2,
del_signTank.transform.localScale.y, originSizeSignZ / 2);
    }

    // Удалить карту
    public void RemoveAllAnchors()
    {
        foreach (var anchor in m_AnchorPoints)
        {
            Destroy(anchor);
        }
        m_AnchorPoints.Clear();
        Destroy(Delite_AnchorPrefab);
        Destroy(del_signTank);
    }

    // В Awake() мы получаем ссылку на все необходимые компоненты.
    // ARRaycastManager позволяет нам выполнять рейкасты, чтобы мы знали, где
разместить якорь.
    // ARPlaneManager обнаруживает поверхности, на которых мы можем
разместить наши объекты.
    // ARAnchorManager обрабатывает все привязки и обновляет их положение и
поворот.
    void Awake()
    {

```

```

        m_RaycastManager = GetComponent<ARRaycastManager>();
        m_AnchorManager = GetComponent<ARAnchorManager>();
        m_PlaneManager = GetComponent<ARPlaneManager>();
        m_AnchorPoints = new List<ARAnchor>();
    }

    void Update()
    {
        // Если тапа нет, то просто ничего не делайте до следующего вызова
        Update().

        if (Input.touchCount == 0)
            return;

        var touch = Input.GetTouch(0);
        if (touch.phase != TouchPhase.Began)
            return;

        if (AddMap == true){
            if (m_RaycastManager.Raycast(touch.position, s_Hits,
TrackableType.PlaneWithinPolygon))
            {
                // Попадания Raycast сортируются по расстоянию, поэтому
первое

                // будет ближайшим хитом.
                var hitPose = s_Hits[0].pose;
                var hitTrackableId = s_Hits[0].trackableId;
                var hitPlane = m_PlaneManager.GetPlane(hitTrackableId);

                // Это прикрепляет якорь к области на плоскости,
соответствующей попаданию raycast,
                // а затем создает экземпляр выбранного вами префаба в этот
момент.

                // Этот экземпляр префаба связан с привязкой, чтобы
убедиться, что положение префаба согласовано.

                // с якорем, так как якорь, прикрепленный к ARPlane, будет
автоматически обновляться ARAnchorManager по мере уточнения точного положения
ARPlane.

                var anchor = m_AnchorManager.AttachAnchor(hitPlane, hitPose);
                Delite_AnchorPrefab = Instantiate(m_AnchorPrefab,
anchor.transform);

```

```

        if (anchor == null)
        {
            Debug.Log("Error creating anchor.");
        }
        else
        {
            // Сохраняет якорь, чтобы его можно было удалить позже.
            m_AnchorPoints.Add(anchor);
        }
    }
    AddMap = false;
}
if (AddSign == true)
{
    if (m_RaycastManager.Raycast(touch.position, s_Hits,
TrackableType.PlaneWithinPolygon))
    {
        // Попадания Raycast сортируются по расстоянию, поэтому
первое

        // будет ближайшим хитом.
        var hitPose = s_Hits[0].pose;
        var hitTrackableId = s_Hits[0].trackableId;
        var hitPlane = m_PlaneManager.GetPlane(hitTrackableId);

        // Это прикрепляет якорь к области на плоскости,
соответствующей попаданию raycast,
        // а затем создает экземпляр выбранного вами префаба в этот
момент.

        // Этот экземпляр префаба связан с привязкой, чтобы
убедиться, что положение префаба согласовано.

        // с якорем, так как якорь, прикрепленный к ARPlane, будет
автоматически обновляться ARAnchorManager по мере уточнения точного положения
ARPlane.

        var anchor = m_AnchorManager.AttachAnchor(hitPlane, hitPose);
        del_signTank = Instantiate(m_Sign, anchor.transform);

        if (anchor == null)
        {
            Debug.Log("Error creating anchor.");
        }
        else

```

```

        {
            // Сохраняет якорь, чтобы его можно было удалить позже.
            m_AnchorPoints.Add(anchor);
        }
    }
    AddSign = false;
}

static List<ARRaycastHit> s_Hits = new List<ARRaycastHit>();

List<ARAnchor> m_AnchorPoints;

ARRaycastManager m_RaycastManager;

ARAnchorManager m_AnchorManager;
ARPlaneManager m_PlaneManager;
}

```

Скрипт меню авторизации

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using TMPro;
using System;
using System.Globalization;

public class ControlMainMenuAndAvtorization : MonoBehaviour
{
    [SerializeField]
    TMP_InputField InpLogin;
    [SerializeField]
    TMP_InputField InpPassword;
    [SerializeField]
    GameObject ErrorPanel;
    [SerializeField]
    TextMeshProUGUI TimeWorld = null;
    [SerializeField]
    TextMeshProUGUI TxtName = null;
    [SerializeField]

```

```

TextMeshProUGUI TxtLevel = null;

DateTime TimeWordModile;
//Номер зашедшего пользователя
int UserInt;

struct User
{
    public String Login;
    public String Password;
    public String Name;
    public String Level;
}

List<User> UserMap = new List<User>()
{
    new User { Login = "root", Password = "root", Name = "Лейтенант
Иванов Иван Иванович", Level = "Секретно"}

};

// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{
    TimeWordModile = DateTime.Now;
    TakeDate();
}

public void TakeDate()
{
    if (TxtName != null)
    {
        TxtName.text = UserMap[UserInt].Name;
    }
    if (TxtLevel != null)

```

```

    {
        TxtLevel.text = UserMap[UserInt].Level;
    }
    if (TimeWorld != null)
    {
        GetTimeWord();
    }
}

//Выводим время на экран
/** Output.
 *
 * d :08/17/2000
 * D :Thursday, August 17, 2000
 * f :Thursday, August 17, 2000 16:32
 * F :Thursday, August 17, 2000 16:32:32
 * g :08/17/2000 16:32
 * G :08/17/2000 16:32:32
 * m :August 17
 * r :Thu, 17 Aug 2000 23:32:32 GMT
 * s :2000-08-17T16:32:32
 * t :16:32
 * T :16:32:32
 * u :2000-08-17 23:32:32Z
 * U :Thursday, August 17, 2000 23:32:32
 * y :August, 2000
 * dddd, MMMM dd yyyy :Thursday, August 17 2000
 * ddd, MMM d "'yy :Thu, Aug 17 '00
 * dddd, MMMM dd :Thursday, August 17
 * M/yy :8/00
 * dd-MM-yy :17-08-00
 */
public void GetTimeWord()
{
    TimeWorld.text = TimeWordModile.ToString("G",
DateTimeFormatInfo.InvariantInfo);
}

//Проверка данных для авторизации
public void CheckEntry()
{
    UserInt = 0;

```

```

        foreach (User user1 in UserMap)
        {
            if (InpLogin.text == user1.Login && InpPassword.text ==
user1.Password)
            {
                SceneManager.LoadScene("MainMenu");
            }
            else
            {
                ErrorPanelShow();
            }
            UserInt++;
        }
    }

    //Показать панель ошибки
    public void ErrorPanelShow()
    {
        ErrorPanel.SetActive(true);
    }
    //Скрыть панель ошибки
    public void ErrorPanelCelare()
    {
        ErrorPanel.SetActive(false);
    }
    //Завершение программы
    public void ExitProgram()
    {
        Application.Quit();
    }
    //Переход на сцену создание карты
    public void LoadMenuCreateMap()
    {
        SceneManager.LoadScene("SampleScene");
    }
    //Переход на сцену главное меню
    public void LoadMainMenu()
    {
        SceneManager.LoadScene("MainMenu");
    }

```

```

        //Переход на сцену авторизация
        public void LoadLoginMenu()
        {
            SceneManager.LoadScene("Authorization");
        }

    }

```

Скрипт главного меню и UI дополненной реальности

```

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

public class ControlMenu : MonoBehaviour
{
    [SerializeField]
    GameObject CanvasMenu;
    [SerializeField]
    GameObject CanvasSetting;
    [SerializeField]
    GameObject CanvasMap;
    [SerializeField]
    GameObject CanvasSign;
    [SerializeField]
    Button ButtonSetting;
    [SerializeField]
    Button ButtonMenu;
    [SerializeField]
    Button TakeSign;
    [SerializeField]
    Button TakeMap;

    // Открытие меню
    public void OpenMenu()
    {
        CanvasMenu.SetActive(true);
        ButtonSetting.gameObject.SetActive(false);
        ButtonMenu.gameObject.SetActive(false);
        TakeSign.gameObject.SetActive(false);
        TakeMap.gameObject.SetActive(false);
    }
}

```

```

// Закрытие меню
public void CloseMenu()
{
    CanvasMenu.SetActive(false);
    ButtonSetting.gameObject.SetActive(true);
    ButtonMenu.gameObject.SetActive(true);
    TakeSign.gameObject.SetActive(true);
    TakeMap.gameObject.SetActive(true);
}

// Открытие настроек
public void OpenSetting()
{
    CanvasSetting.SetActive(true);
    ButtonSetting.gameObject.SetActive(false);
    ButtonMenu.gameObject.SetActive(false);
    TakeSign.gameObject.SetActive(false);
    TakeMap.gameObject.SetActive(false);
}

// Закрытие настроек
public void CloseSetting()
{
    CanvasSetting.SetActive(false);
    ButtonSetting.gameObject.SetActive(true);
    ButtonMenu.gameObject.SetActive(true);
    TakeSign.gameObject.SetActive(true);
    TakeMap.gameObject.SetActive(true);
}

// Открытие выбора карт
public void OpenMap()
{
    CanvasMap.SetActive(true);
    ButtonSetting.gameObject.SetActive(false);
    ButtonMenu.gameObject.SetActive(false);
    TakeSign.gameObject.SetActive(false);
    TakeMap.gameObject.SetActive(false);
}

// Закрытие выбора карт
public void CloseMap()
{
    CanvasMap.SetActive(false);
    ButtonSetting.gameObject.SetActive(true);
}

```

```

        ButtonMenu.gameObject.SetActive(true);
        TakeSign.gameObject.SetActive(true);
        TakeMap.gameObject.SetActive(true);
    }
    // Открытие выбора знака
    public void OpenSign()
    {
        CanvasSign.SetActive(true);
        ButtonSetting.gameObject.SetActive(false);
        ButtonMenu.gameObject.SetActive(false);
        TakeSign.gameObject.SetActive(false);
        TakeMap.gameObject.SetActive(false);
    }
    // Закрытие выбора знака
    public void CloseSign()
    {
        CanvasSign.SetActive(false);
        ButtonSetting.gameObject.SetActive(true);
        ButtonMenu.gameObject.SetActive(true);
        TakeSign.gameObject.SetActive(true);
        TakeMap.gameObject.SetActive(true);
    }
}

```

Приложение Г

(обязательное)

Программа заполнения тактических карт в дополненной реальности

Презентация

					ПГУ 1.090501.15.001 ПЗ	Лист
						90
Изм.	Лист	№ докум.	Подпись	Дата		

Программа заполнения тактических карт в дополненной реальности

Студент – Халак Д.С.
Руководитель – Митрохин М.А.

- ▶ Географическая карта – это уменьшенное копия земной поверхности, перенесённая на плоскости с размещением природных и общественных объектов.
- ▶ Топографические карты в отличие от географических сохраняют постоянство масштаба. В военном деле используются именно этот тип карт. Не один род войск не может обойтись без подробного изучения местности при организации боя, при перемещении военнослужащих на небольшие расстояния по незнакомой местности, для управления подразделениями и частями, а также при выборе мест расположения каких-либо объектов.



Изм.	Лист	№ докум.	Подпись	Дата

- Дополненная реальность (AR) — это технология, которая помещает в наш с вами реальный мир несуществующие в действительности, виртуальные, объекты. Чтобы увидеть эти объекты, понадобится смартфон. Так виртуальный контент и физический мир взаимодействуют в режиме реального времени.



Классификации:

- - Распознавание изображения
- - Распознавание объектов
- - Обнаружение плоскости

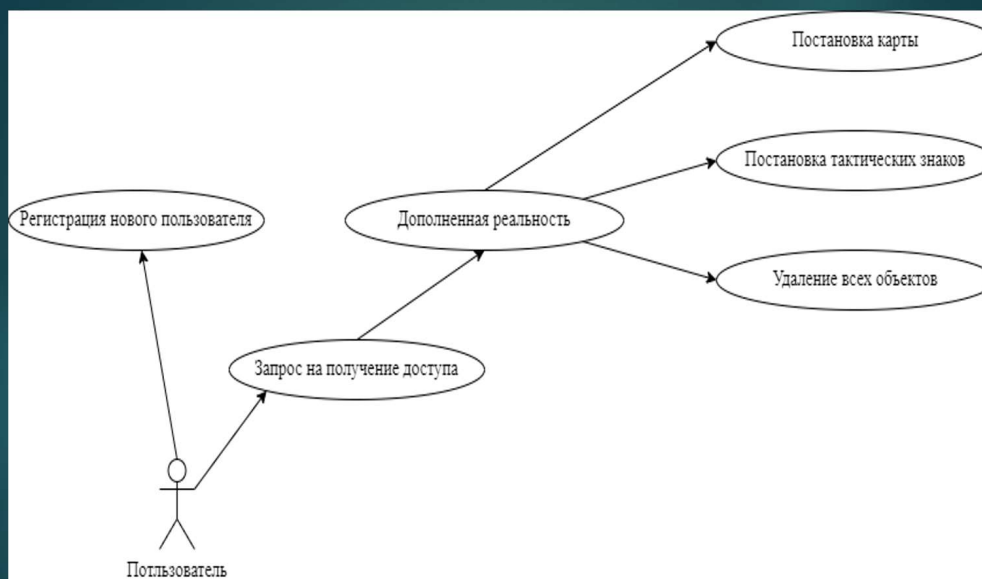
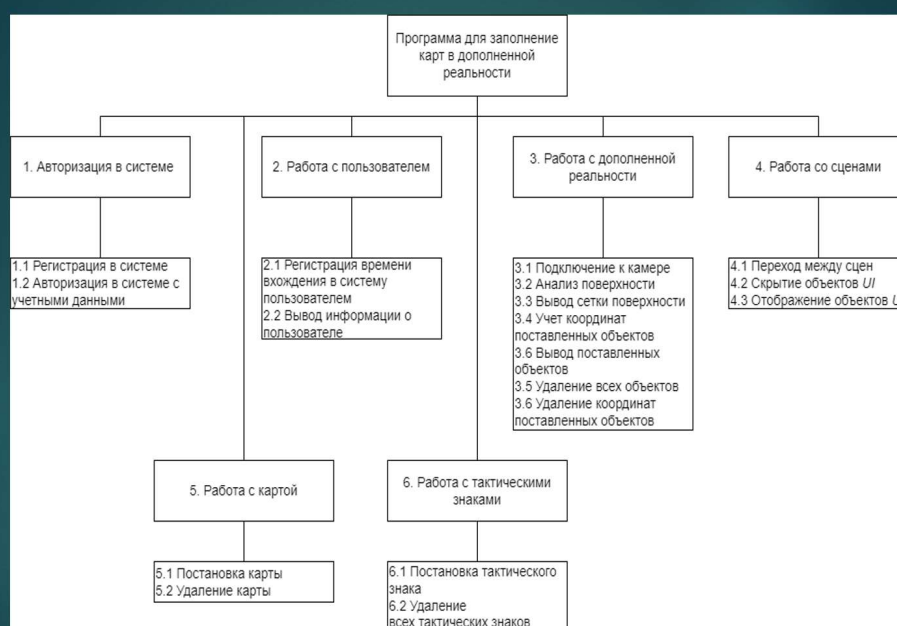


Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.15.001 ПЗ

Лист

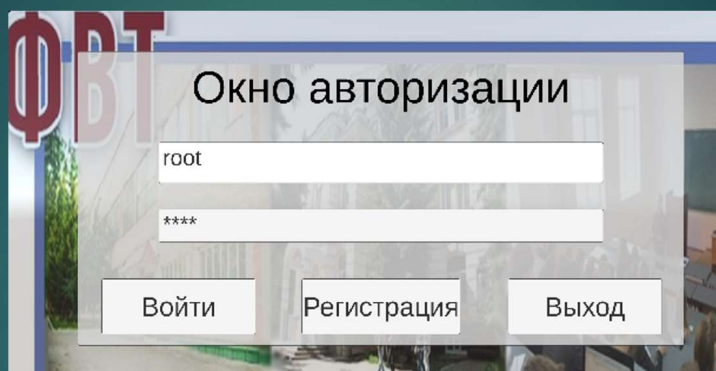
92



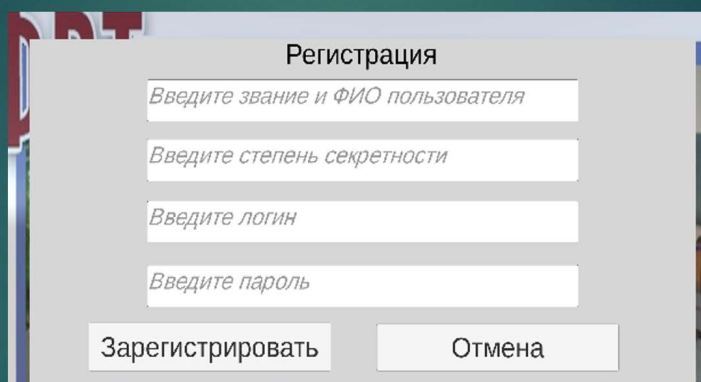
Среда разработки Почему Unity?



Окно авторизации

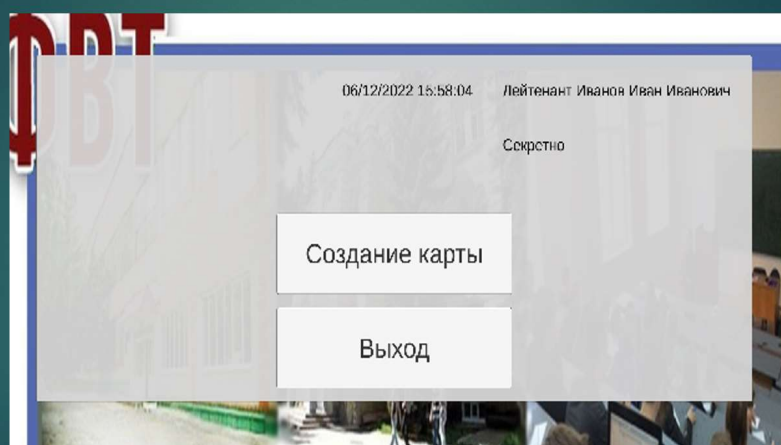


Окно регистрации



The screenshot shows a registration dialog box titled "Регистрация". It contains four input fields with placeholder text: "Введите звание и ФИО пользователя", "Введите степень секретности", "Введите логин", and "Введите пароль". At the bottom, there are two buttons: "Зарегистрировать" and "Отмена".

Главное меню



The screenshot shows a main menu dialog box. At the top, it displays the date and time "06/12/2022 15:58:04" and the user name "Лейтенант Иванов Иван Иванович". Below this, the word "Секретно" is displayed. The menu contains two buttons: "Создание карты" and "Выход".

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.15.001 ПЗ

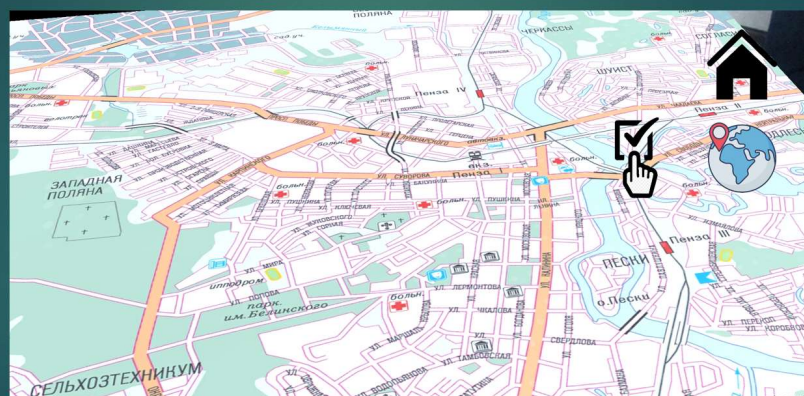
Лист

96

Поле дополненной реальности



Расставление карт



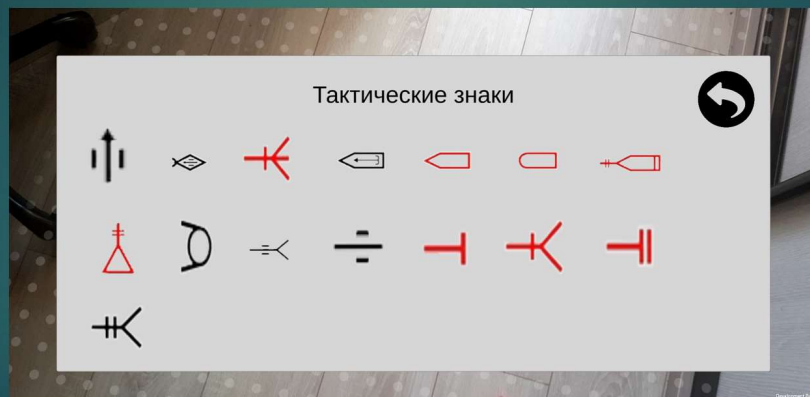
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.15.001 ПЗ

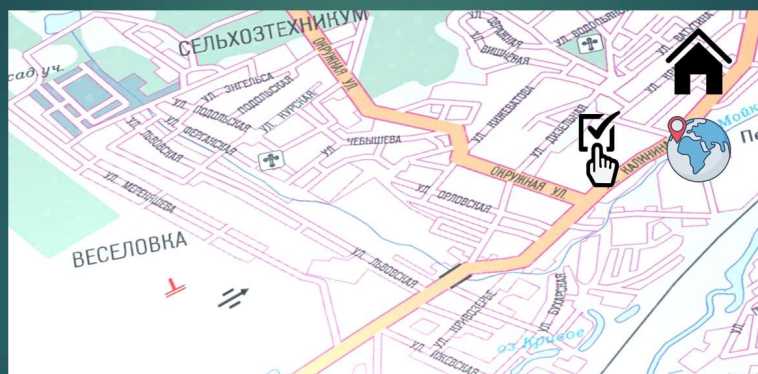
Лист

97

Выбор тактических знаков



Расставление тактических знаков



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.15.001 ПЗ

Лист

98