

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Программное обеспечение средств вычислительной техники и автоматизированных систем»

БАКАЛАВРСКАЯ РАБОТА

на тему

ИГРОВОЕ ПРИЛОЖЕНИЕ «ЛАБИРИНТ» НА
UNITY

Студент

31.05.2023 Н.Д. Гуськова
(подпись, дата)

Руководитель

31.05.2023 О.В. Юрова
(подпись, дата)

Нормоконтролёр

31.05.2023 А.С. Бычков
(подпись, дата)

Работа допущена к защите (протокол зас. кафедры от 02.06.23 г. № 14)

Заведующий кафедрой

(подпись)

(фамилия, инициалы)

М.А. Митрохин

Работа защищена с отметкой

удовлетворительно

(протокол заседания ГЭК от

21.09.2023 № 35)

Секретарь ГЭК

(подпись)

Е.И. Гурин

(фамилия, инициалы)

Пенза 2023

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

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
« 26 » апреля 20 23 г.

ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ

1. Студент Гуськова Н.Д. гр 19ВВП1 факультета вычислительной техники.
2. Тема работы Образовательное приложение «Игровое приложение «Лабиринт» на Unity»

Тема утверждена приказом ПГУ № 458/о от "26" апреля 2023 г.

3. Руководитель бакалаврской работы к.т.н., доцент, каф. ВТ ПГУ Юрова О.В.

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

Реализовать игровое приложение "Лабиринт" на Unity, помогающее в развитии логики.
Реализовать лабиринт из нескольких уровней, возможность сохранения и графический интерфейс. Провести тестирование и отладку каждого элемента приложения.

5. Перечень подлежащих разработке вопросов

Требуется произвести анализ предметных областей: изучить существующие игры, выделить их достоинства и недостатки. Выбрать язык программирования, который подойдет для написания логики приложения. Определить архитектуру приложения и вспомогательные средства разработки.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Изучение предметной области	50	26.04.2023-30.04.2023	<i>ф.и.о.</i>
2.	Анализ требований	10	01.05.2023-02.05.2023	<i>ф.и.о.</i>
3.	Выбор технологий, применяемых при проектировании и разработке	50	03.05.2023-07.05.2023	<i>ф.и.о.</i>
4.	Описание алгоритмов	40	08.05.2023-11.05.2023	<i>ф.и.о.</i>
5.	Реализация игрового приложения	150	12.05.2023-29.05.2023	<i>ф.и.о.</i>
6.	Оформление пояснительной записки	24	30.05.2023-31.05.2023	<i>ф.и.о.</i>

Дата выдачи задания "26" апреля 2023 г.

Руководитель бакалаврской работы _____ *ф.и.о.* Юрова О.В.
(подпись, дата)

Задание к исполнению принял студент _____ *ф.и.о.* Гуськова Н.Д.
(подпись, дата)

Работу к защите допустить
Декан факультета _____ *ф.и.о.* Фионова Л.Р.
(подпись, дата)

Реферат

Бакалаврская работа содержит 80 с., 23 рис., 4 источ. и 3 прил.

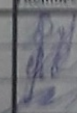
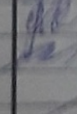
ИГРОВОЕ ПРИЛОЖЕНИЕ, ЛАБИРИНТ, *ANDROID*, *UNITY*, *VISUALSTUDIO*, *C#*

Цель работы – реализовать игровое приложение «Лабиринт» на *Unity*.

Объектом разработки является игровое приложение, представляющие собой лабиринт с перемещением объектов.

Результатом проделанной работы является проект, удовлетворяющий поставленную задачу. А именно, разработано и протестировано игровое приложение. Приложение совместимо с актуальными версиями *Android* смартфонов.

Разработка производилась с использованием широкого стека технологий, к которым относятся: язык программирования *C#*, интегрированная среда разработки *Visual Studio 2022* для работы со скриптами и межплатформенная среда *Unity* для разработки игрового приложения.

					ПГУ1.090301.5.001 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	Игровое приложение "Лабиринт" на Unity Пояснительная записка	Лист	Лист	Листов	
Разраб		Гуськова Н.Д.							
Провер.		Юрова О.В.					2	80	
Н. Контр		Бичков А.С.				ФВТ, каф. ВТ зр. 19ВВН11			
Утверд		Митрохин М.А.							

Содержание

Содержание	1
Перечень сокращений и обозначений	3
Введение	4
1 Игровое приложение «Лабиринт» на <i>UNITY</i>	6
1.1 Разработка игр	6
1.2 Актуальность	8
1.2.1 Статистика доходы от мобильных игр	8
1.2.2 Лучшие жанры по возрастным группам	9
1.2.3 Размер аудитории мобильных игр	10
1.2.4 Аудитория для игр головоломок	11
1.3 Классификация мобильных игр	12
1.4 Аналоги	15
1.4.1 <i>Sokoban</i>	15
1.4.2 <i>Unblock Me</i>	16
1.4.3 <i>Roll the Ball</i>	18
1.5 Постановка задач	19
1.6 Выбор среды разработки	22
1.6.1 <i>Unity</i>	22
1.6.2. <i>Unreal Engine</i>	24
1.6.3 <i>Construct</i>	24
1.6.4 <i>GameMaker Studio</i>	25
1.6.5 <i>Godot</i>	25
1.7 Выбор языка программирования	26
1.7.1 Язык программирования <i>C#</i>	26
1.7.2 Язык программирования <i>Java</i>	27
1.7.3 Язык программирования <i>C++</i>	28
1.7.4 Язык программирования <i>JavaScript</i>	29
1.8 Вывод	29

1.9 Инструменты разработки	30
1.10 Архитектура игрового приложения лабиринта на <i>Unity</i>	30
1.11 Разработка игрового приложения.....	37
1.11.1 Изучение технологий разработки игр на платформе <i>Android</i> и игрового движка <i>Unity</i>	38
1.11.2 Создание игрового поля.	38
1.11.3 Реализация игровой логики и механики.	39
1.11.4 Создание пользовательского интерфейса и дизайн.....	42
1.11.5 Создание основного игрового экрана	44
1.9 Тестирование приложения	48
2 Охрана труда.....	51
2.1 Вредоносные факторы	51
2.2 Требования к помещению	52
2.3 Требования к освещению	53
2.4 Требования по организации рабочего места	54
2.5 Требования к организации работы	55
Заключение.....	57
Список использованных источников	58
Приложение А – <i>UML</i> – диаграммы.....	59
Приложение А. 1 – <i>UML</i> – диаграмма прецедентов.....	60
Приложение А.2 – <i>UML</i> – диаграмма классов.....	61
Приложение А.3 – <i>UML</i> – диаграмма деятельности.....	62
Приложение Б – Листинг программы	63
Приложение В – Презентация.....	69

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

Веб — система доступа к связанным между собой документам на различных компьютерах, подключённых к Интернету

МБ (Мегабайт) — единицы измерения количества информации

ПК – персональный компьютер

ПО — программное обеспечение

ПЭВМ – персональная электронно-вычислительная машина

APK (Android Package) — формат файла приложения *Android*

CSS (Cascading Style Sheets) — формальный язык декорирования и описания внешнего вида документа , написанного с использованием языка разметки

Framework – программная платформа, которая упрощает разработку программного продукта, определяет структуру проекта

GUI (graphical user interface) — графический интерфейс пользователя (ГИП)

HTML(HyperText Markup Language) — стандартизированный язык гипертекстовой

разметки документов для просмотра веб-страниц в браузере

IDE (Integrated Development Environment) — интегрированная среда разработки

JVM (Java Virtual Machine) — виртуальная машина *Java*

.NET — кроссплатформенная платформа для создания различных типов приложений

NDK (Native Development Kit) — набор инструментов, которые помогают операционной системе *Android* работать с кодом, написанным на языках *C* и *C++*.

SDK (software development) — набор средств разработки, позволяющий специалистам по программному обеспечению создавать приложения для определённого пакета программ

RPG (Role-Playing Game) — компьютерная ролевая игра

UI (User interface) — пользовательский интерфейс

UML (Unified Modeling Language) – унифицированный язык моделирования

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

Введение

В настоящее время игры на мобильных платформах очень популярны. Это связано с тем, что игры на мобильных устройствах доступны каждому, они не требуют специальных навыков и знаний, люди могут играть в любое удобное время и в любом месте. Пользователям предлагается широкий выбор игровых жанров, от стратегий и головоломок до спортивных и аркадных игр.

Рынок игр на *Android*-приложениях — один из крупнейших и динамично развивающихся. С каждым годом игры для *Android* становятся все разнообразнее и качественнее.

Игры уже давно перестали быть просто дополнительной функцией телефона и стали целым миром, в котором существуют тысячи игр. Игры эволюционировали постепенно — каждый новый жанр чем-то напоминал предшествующие, и попутно расширялось значение слова игра, охватывавшее их [1]. Игры — это не только развлечение, но и отличный способ тренировки мозга, развития логического мышления, а также способ интересно провести время. Они являются важным элементом в нашей жизни, так как позволяют отдохнуть, расслабиться и насладиться особенным моментом.

Одной из особенностей *Android* является то, что платформа имеет разные версии операционной системы. Это означает, что разработчики должны учитывать совместимость приложения с разными версиями *Android*, чтобы оно корректно работало на всех устройствах. Кроме того, в *Android* разные разрешения экрана, что тоже нужно учитывать при разработке приложения.

Еще одна особенность *Android* — ограничение ресурсов. Мобильные устройства имеют ограниченные ресурсы, такие как процессор, оперативная память и батарея. Поэтому приложение необходимо оптимизировать для работы на мобильных устройствах, чтобы не нагружать их и не снижать производительность.

Также игровые приложения должны быть безопасными и не нарушать авторские права. Разработчики должны учитывать эти требования, чтобы их

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

приложение могло быть опубликовано в магазине.

В целом, анализ предметной области *Android* является важным этапом разработки игрового приложения. Разработчики должны учитывать функции, возможности и ограничения платформы, чтобы создать приложение, которое будет корректно работать на устройствах с операционной системы *Android*.

					ПГУ1.090301.5.001 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

1 Игровое приложение «Лабиринт» на *UNITY*

1.1 Разработка игр

Разработка игр для *Android* — это процесс создания игры для устройств на базе операционной системы *Android*. Эта тема очень популярна в настоящее время, так как *Android* сегодня является одной из самых используемых мобильных платформ в мире. Количество смартфонов и планшетов на этой платформе постоянно растет, и многие пользователи используют для игры *Android*-устройства.

Разработка игровых приложений для *Android* имеет свои особенности, которые могут повлиять на успех игры. В частности, приложение должно быть легковесными и совместимым с широким спектром *Android*-устройств, включая бюджетные и старые модели. Пользовательский интерфейс и управление игровыми элементами также могут иметь большое значение.

Сегодня игровые приложения занимают большую часть рынка мобильных приложений и являются одной из самых популярных категорий приложений. Несмотря на некоторые трудности, разработка игрового приложения на *Android* открывает множество возможностей для разработчиков игр. Разработка игровых приложений на *Android* — высокотехнологичный процесс, требующий знаний и навыков программирования, аналитики и дизайна. Привлекательность этой темы в том, что наряду с инновационными технологиями и функциями смартфонов разработчикам игр доступны новые инструменты для создания более интересных и захватывающих игр.

Такие приложения могут варьироваться от простых игр, таких как пазлы и головоломки, до сложных и многопользовательских онлайн-игр. Разработка игрового приложения на *Android* подразумевает создание игровой механики, игрового интерфейса и графических эффектов.

Важность игровых приложений для *Android* заключается в том, что такие

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

приложения могут обеспечить большой потенциал для привлечения пользователей и получения высокой прибыли для разработчиков. При правильном подходе к разработке мобильное приложение на базе *Android* может стать не только развлекательной площадкой, но и средством продвижения бренда, рекламы товаров, услуг и т.д.

При создании игровых приложений на *Android* разработчикам приходится иметь дело с рядом особенностей платформы. Одной из главных особенностей является большое количество устройств на базе *Android*, каждое из которых имеет свои характеристики, поэтому разработчикам необходимо убедиться, что игры будут хорошо работать на как можно большем количестве устройств.

Кроме того, стоит учесть, что большинство игр для *Android* бесплатны с возможностью дополнительных внутриигровых покупок. Эта политика монетизации позволяет разработчикам получать доход, а также собирать данные о поведении пользователей в игре.

В целом, тема игровых приложений на *Android* интересна как пользователям, так как предлагает большое разнообразие игр на любой вкус, так и разработчикам, у которых есть возможность зарабатывать на создании и продвижении игр.

Безопасность не менее важна: вредоносные атаки на мобильные устройства становятся все более распространенными. Разработчики должны уделять большое внимание безопасности приложений и защите пользовательских данных.

Одним из важнейших элементов успешного игрового приложения является уникальность и оригинальность игрового процесса. В современном мире существует огромное количество уже существующих игр и приложений, поэтому создание чего-то нового и интересного может стать залогом успеха.

Игры для *Android* любят пользователи разных возрастных категорий, поэтому такие приложения могут быть ориентированы на разный контингент пользователей. Например, игры с простым и интересным геймплеем могут

					ПГУ1.090301.5.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

понравиться детям и подросткам, а более сложные и многогранные игры понравятся пользователям постарше.

Дизайн интерфейса и яркая графика являются важными аспектами игровых приложений для *Android*. Игры с привлекательным дизайном, отличной графикой и звуком способны привлечь внимание пользователей и увеличить количество загрузок.

1.2 Актуальность

Игры на мобильных устройствах — это одна из самых популярных и востребованных категорий приложений.

Актуальность разработки игрового приложения на платформе *Android* обусловлена растущей популярностью мобильных устройств и развитием мобильной игровой индустрии. Сейчас игры на смартфонах и планшетах уже не являются просто развлечением, а могут стать полноценным способом заработка

Ознакомившись с отчетом *Sensor Tower* по рынку мобильных игр за 2022 год, написал, что количество загрузок мобильных игр в России снизилось на 11% год к году, но в четвертом квартале составило 673 млн, выйдя на уровень третьего квартала 2021 года. Весной и летом был спад, но даже с ним Россия остается самым крупным мобильным рынком Европе с 24% загрузок в четвертом квартале прошлого года [2].

1.2.1 Статистика доходы от мобильных игр

Мировой рынок игр, включая мобильные и другие платформы, заработал более 184 миллиардов долларов в 2022 году. Выручка от мобильных игр в 2022 году составила 92 миллиарда долларов, снизившись на 6,4% по сравнению с аналогичным периодом прошлого года. Несмотря на снижение доходов, произошел всплеск загрузок мобильных игр. В 2022 году они достигли почти 90 миллиардов, что на 6 миллиардов больше, чем в 2021 году. Гипер-казуальные игры были основным драйвером загрузок.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

Наш обширный список статистики мобильных игр был бы неполным без ведущих жанров и поджанров. Они были лидерами в 2022 году с точки зрения потребительских расходов и загрузок [2].

По данным *Sensor Tower*, это были лучшие жанры в 2022 году по валовому доходу во всем мире:

- *RPG* (20,2 миллиарда долларов);
- стратегия (16,3 миллиарда долларов);
- головоломка (9,1 миллиарда долларов);
- казино (8 миллиардов долларов);
- шутер (4,6 миллиарда долларов);
- симуляция (4,5 миллиарда долларов);
- спорт (2,7 миллиарда долларов);
- аркада (2,1 миллиарда долларов);
- настольный (1,7 миллиарда долларов).

Лучшие жанры мобильных игр по количеству загрузок согласно отчету *State of Mobile Gaming om Sensor Tower* :

- гипер-казуальные игры — 12,3 млрд;
- аркады — 5,9 млрд;
- симуляторы — 4,8 млрд;
- головоломки — 4,5 млрд;
- лайфстайл-игры — 3 млрд [2].

1.2.2 Лучшие жанры по возрастным группам

Что касается жанров, то существуют определенные предпочтения у игроков в зависимости от возраста. Прослеживаются некоторые общие тенденции и предпочтения аудитории в зависимости от возраста и жанра игры.

Например, игры в жанре экшн и шутеры обычно более популярны среди молодых людей в возрасте от 18 до 24 лет, чем у других возрастных групп. Это

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

может быть связано с тем, что молодые люди часто ищут более интенсивные и динамичные игры, которые могут представлять собой большой вызов для них.

Игры в жанре стратегии и головоломки обычно более популярны среди людей старше 25 лет. Это может быть связано с тем, что эти игры обычно требуют большего внимания к деталям и более глубокого понимания механики игры, что

может быть более интересно для взрослых.

Игры для детей, такие как пазлы и аркады, обычно более популярны среди детей и подростков в возрасте от 6 до 17 лет. Это связано с тем, что такие игры часто более простые и доступные для детей.

Кроме того, следует отметить, что многие игры имеют возрастные ограничения, которые определяются рейтинговыми организациями, такими как *ESRB (Entertainment Software Rating Board)* и *PEGI (Pan European Game Information)*. Эти ограничения могут варьироваться от "для всех возрастов" до "только для взрослых", что может ограничивать доступ к играм в зависимости от возраста [2].

1.2.3 Размер аудитории мобильных игр

Если мы посмотрим на общий мировой игровой рынок, то в 2022 году на нем было 2 миллиарда игроков.

Согласно исследованию *Entertainment Software Association*, наибольшее количество игроков (35%) приходится на возрастную группу от 18 до 35 лет. За ними следуют игроки в возрасте от 36 до 49 лет (29%), а также игроки в возрасте от 50 до 64 лет (17%). Менее всего игроков в возрасте до 18 лет (15%) и старше 65 лет (4%).

Мобильные игры почти одинаково популярны среди мужчин и женщин. Более конкретно, 53,6% от общей аудитории мобильных игр составляют мужчины и 46,4% - женщины.

В мобильные игры играют люди всех возрастов. 5% мобильных игроков составляют 25-34 года, а 28,3% - 16-24 года. Играми также увлекаются

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

представители старшего поколения – 23,1% аудитории мобильных игр составляют 35-44 года, 12,8% - 45-54 года и 6,3% - 55 лет и старше.

Средний возраст мобильного геймера составляет 36 лет.

Люди старше 35 лет составляют почти половину аудитории мобильных игроков.

Интересы и поведение мобильных игроков:

- поколение Z и миллениалы проводят за играми больше времени, чем за любым другим видом развлечений;
- 85% мобильных игроков не идентифицируют себя как геймеры;
- в среднем сеансы мобильных игр у женщин длятся на 25% дольше, чем у мужчин;
- 43% женщин-геймеров играют более пяти раз в неделю;
- 38% геймеров мужского пола играют более пяти раз в неделю;
- пользователи тратят 90% своего использования смартфонов на приложения и игры;
- в среднем геймеры играют в две-пять мобильных игр в месяц;
- пользователи проводят за мобильными устройствами три с половиной часа каждый день, и 11% этого времени тратится на мобильные игры;
- в среднем мобильные геймеры, увлекающиеся казуальными играми, загружают в десять раз больше игр, чем другие игроки [2].

1.2.4 Аудитория для игр головоломок

Аудиторией игрового приложения на *Android* являются люди, использующие мобильные устройства на базе этой операционной системы. Это круг пользователей может быть широким, так как игра подходит для любого возраста. Она может быть интересна как детям, так и взрослым, которые любят решать головоломки. А также может быть интересна тем, кто ищет отдых от повседневной суеты и хочет провести время в увлекательной игре.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

Разработка игрового приложения на *Android* может иметь значительное значение в различных аспектах. Например, оно может быть использовано в образовательных целях, в качестве инструмента для улучшения когнитивных навыков, для развлечения, для социальной взаимодействия и т.д.

1.3 Классификация мобильных игр

Классификация мобильных игр основана на различных критериях, таких как жанр, тип геймплея, аудитория и т.д. Ниже представлены основные виды мобильных игр:

- платформеры: игры, в которых игрок управляет персонажем, должен проходить уровни, собирать бонусы и уничтожать врагов;
- головоломки: игры, в которых игроку необходимо решать сложные головоломки и преодолевать логические препятствия. Примеры: *Candy Crush Saga*, *Angry Birds* ;
- спортивные игры: игры, в которых игрок соревнуется с другими онлайн-игроками в различных спортивных дисциплинах, таких как футбол, баскетбол, теннис или бег;
- казуальные игры: игры, которые не требуют от игрока много времени и усилий и могут быть запущены и остановлены в любой момент. К ним относятся такие игры, "Крестики-нолики", "Одевалки" и "Собери слова";
- ролевые игры: Игры, в которых игрок управляет персонажем и переживает его приключения. Ролевые игры обычно включают в себя элементы боя, экспериментов и сюжета. Примеры: *Final Fantasy*, *Pokemon Go*;
- стратегии: игры, в которых игрок сражается на поле боя или строит свою базу. В этих играх игрок должен использовать свои навыки планирования и стратегии, чтобы победить своих врагов;

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

- симуляторы: это игры, которые позволяют игроку получить виртуальный опыт, будь то управление самолетом или управление городом. Примеры: *The Sims, Farming Simulator*;

- экшн: эти игры не требуют от игрока много времени на подготовку, они ориентированы на быстрый и динамичный геймплей, в котором игрок сражается с врагами и выполняет миссию;

- аркады — это игры с простым геймплеем, в которых игроки управляют персонажами и выполняют различные задания. Примеры: *Subway Surfers, Temple Run*;

- шутеры — игры, в которых игрок управляет персонажем и стреляет в противников. Примеры: *Call of Duty, PUBG*.

Виды мобильных игр:

- одиночная: игры, в которые можно играть в одиночку без подключения к Интернету или другими игроками;

- многопользовательские: игры, в которых игроки играют вместе в режиме реального времени. Они могут играть как в команде, так и друг против друга;

- бесплатные: игры, которые можно загрузить и играть бесплатно. Как правило, эти игры предоставляют платные функции, которые можно приобрести с помощью микротранзакций;

- платные: Игры, которые требуют покупки перед использованием. Обычно такие игры предоставляют больше возможностей и лучшего качества, чем бесплатные игры;

- рекламные: игры, предоставляющие игрокам бесплатные функции в обмен на просмотр рекламы;

- гибридные: игры, сочетающие в себе несколько типов мобильных игр, таких как платформеры и головоломки;

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

- инди-игры — это игры, созданные независимыми разработчиками и часто отличающиеся уникальным стилем и геймплеем. Примеры: *Limbo*, *Monument Valley*.

Классификация мобильных игр по типу устройства:

- игры для *iOS*. Игры, доступные только для устройств *iOS*, такие как *iPhone* и *iPad*;

- игры для *Android*. Игры, доступные только для устройств *Android*, таких как телефоны и планшеты *Samsung* и *LG*;

- игры на мобильной веб-платформе. Игры, доступные через мобильные

браузеры на любой мобильной платформе;

- игры на гибридной платформе. Игры, предназначенные для работы на разных мобильных устройствах с использованием гибридных технологий разработки.

Описание жанра игры головоломки.

Головоломки — это игры, которые требуют от игрока решения различных задач и головоломок. Они могут быть различной сложности и включать в себя различные элементы, такие как логика, математика, физика и т.д. Головоломки могут быть как одиночными, так и многопользовательскими.

Игры Лабиринты — являются одними из самых популярных и давно известных игр данного жанра, которые требуют от игрока логического мышления, скорости реакции и умения быстро принимать решения. В этой игре нужно найти выход из лабиринта, преодолевая препятствия и находя выход.

Аудитория такого приложения может быть достаточно широкой, включая людей разных возрастов и интересов. Возможно, наибольший интерес игру "Лабиринт" могут проявлять дети и подростки, но она также может быть интересна и для взрослых пользователей.

Значимость данного игр данного жанра заключается в том, что в мире мобильных игр очень важно создавать игры, которые будут привлекать пользователей и предоставлять им увлекательное развлечение. Также игры могут

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

использоваться для развития умственных способностей, улучшения реакции и повышения концентрации внимания, что делает их еще более значимыми для пользователей.

Игра "Лабиринт" на *Android* может являться актуальной в настоящее время и привлечь широкую аудиторию и имеет важное значение для развития креативных и когнитивных навыков игроков.

1.4 Аналоги

Существует множество игр на *Android*, которые могут быть аналогами игры "Лабиринт". Некоторые из них могут быть направлены на различные возрастные группы, иметь разные уровни сложности и геймплеи.

Среди аналогов можно выделить:

- *Sokoban*;
- *Unblock Me*;
- *Roll the Ball*.

1.4.1 *Sokoban*

Sokoban — это игра-головоломка, в которой игрок должен перемещать ящики по лабиринту, чтобы доставить их на определенные места. Игра имеет несколько уровней сложности, начиная от легкого уровня, который позволяет новичкам быстро освоиться с игрой, и заканчивая сложным уровнем. Пример уровня из игры представлен на рисунке 1.

Цель игры — переместить все ящики по лабиринту на нужные позиции за минимальное количество ходов. Игрок может перемещать ящики только в том случае, если рядом с ними есть свободное место. Если ящик оказывается заблокированным другим ящиком или стеной, то игрок не сможет его переместить.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

Игра *Sokoban* на *Android* имеет несколько уровней сложности, которые позволяют игрокам выбирать наиболее подходящий уровень для своих способностей. Кроме того, игра имеет простой и интуитивно понятный интерфейс, который позволяет быстро освоиться с игрой.

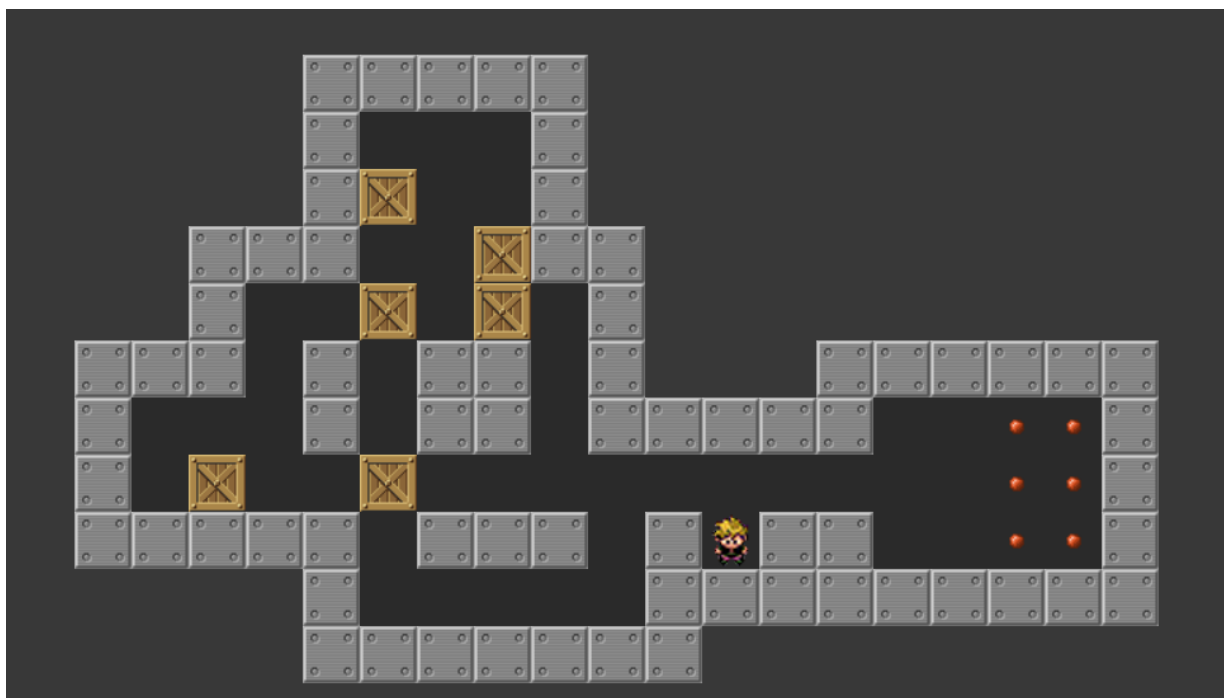


Рисунок 1 – Игра *Sokoban*

1.4.2 *Unblock Me*

Unblock Me — это игра-головоломка, в которой игрок должен перемещать блоки по игровому полю, чтобы освободить красный блок и выйти из уровня. Игра имеет несколько уровней сложности, начиная от легкого уровня и заканчивая сложным уровнем. Пример уровня из игры представлен на рисунке 2.

Цель игры - переместить блоки таким образом, чтобы красный блок мог свободно выйти из лабиринта. Игрок может перемещать блоки только в том случае, если рядом с ними есть свободное место. Если блок оказывается заблокированным другим блоком или стеной, то игрок не сможет его переместить.



1.4.3 *Roll the Ball*

Roll the Ball — это головоломка, в которой игрок должен перемещать блоки, чтобы создать путь для мяча и доставить его до цели. Игра имеет несколько уровней сложности. Пример уровня из игры представлен на рисунке 3.

Цель игры — создать путь для мяча, перемещая блоки по игровому полю. Игрок может перемещать блоки только в том случае, если рядом с ними есть свободное место. Если блок оказывается заблокированным другим блоком или стеной, то игрок не сможет его переместить.

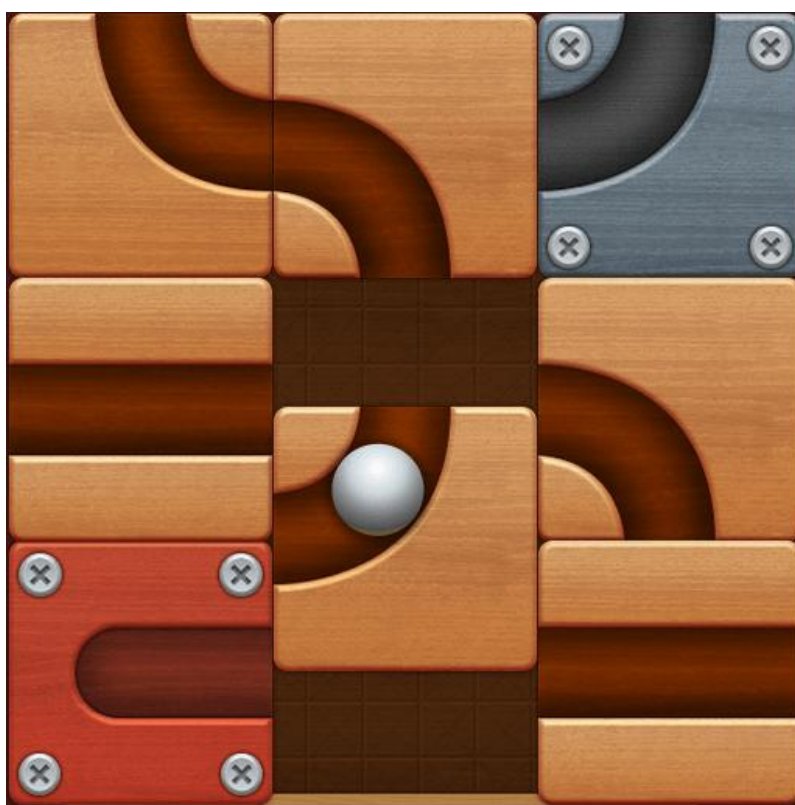


Рисунок 3 – Игра *Roll the Ball*

Достоинства и недостатки :

Достоинства :

- интуитивно понятный и простой интерфейс. Игроки должны легко понимать, как управлять блоками и как проходить уровни ;
- развивают логическое мышление и умение планировать, что полезно для улучшения когнитивных функций ;
- помогает улучшить концентрацию и внимательность ;
- может быть интересной и занимательной развлекательной игрой, что есть хороший способ провести время и расслабиться.

Недостатки :

- может стать зависимостью и отнимать много времени ;
- могут быть слишком простыми или сложными для некоторых игроков ;
- могут быть монотонными или скучными для некоторых игроков ;
- не подходят для тех, кто предпочитает более динамичные игры;
- не всегда может быть полезной для развития других навыков и умений.

1.5 Постановка задач

Основными функциями игры должны быть:

- уровни сложности лабиринта: расстановка блоков различна на каждом уровне с учетом возможности прохождения игроком уровня;
- перетаскивание блоков: игрок должен иметь возможность перетаскивать блоки, чтобы создавать путь к выходу;
- выигрыш: реализовать финиш игры, когда игрок пошёл уровень и может перейти на следующий;
- пройти уровень заново: игрок в любой момент должен иметь возможность перезапустить уровень;
- меню игры: игрок должен иметь возможность начать игру, продолжить, выбрать уровень, выйти из игры;
- переход в меню: игрок должен иметь возможность перейти в меню из игровой сцены;
- интуитивно понятный графический интерфейс;

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

- возможность сохранения прогресса: игрок должен иметь возможность сохранять свой прогресс пройденных уровней в игре и продолжать играть с того же.

Сюжет игры. Сюжет игры заключается в том, что игрок должен помочь розовому блоку выйти из лабиринта, перемещая другие блоки, которые мешают ему пройти. Игрок играет за исследователя, который должен найти выход и освободить определенный блок.

Игрок должен использовать свои логические навыки, чтобы перемещать блоки таким образом, чтобы создать путь для определенного блока. Игра состоит из различных уровней сложности, где игрок должен решать все более сложные головоломки, чтобы продвигаться дальше.

Геймплей. В начале каждого уровня игрок видит лабиринт, в котором определенный блок находится на одной из клеток. Остальные блоки различных

форм и размеров занимают оставшееся пространство. Цель игрока - переместить определенный блок на выход, который находится на противоположной стороне лабиринта.

Для этого игрок может перемещать другие блоки, которые мешают определенному блоку пройти. Блоки можно двигать только вдоль их оси. Игрок может перемещать блоки только тогда, когда между ними есть свободное место.

Движение объектов осуществляется путем перетаскивания блоков пальцем по экрану (жесты, "touchscreen controls" (сенсорное управление)). Выбрав блок, который нужно переместить, вы можете двигать его вверх, вниз, влево или вправо, чтобы освободить путь для определенному блоку.

Движение блоков реализовано довольно просто и интуитивно понятно. При перемещении блока он двигается в выбранном направлении, если на этом пути нет других блоков или стен. Если на пути блока находятся другие блоки, то они также будут двигаться вместе с ним, пока встретятся со стеной или другим блоком.

Аудитория.

					ПГУ1.090301.5.001 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

Игра может заинтересовать широкую аудиторию, включая детей и взрослых. Она подходит для людей, которые любят решать головоломки и игры на логику. Так же подходит для тех, кто хочет провести время с пользой и развить свои когнитивные навыки. Игра не имеет возрастных ограничений, так как она не содержит никакого насилия или контента, который мог бы оскорбить кого-то. Кроме того, игра может быть полезной для людей, которые хотят улучшить свои навыки решения задач и принятия решений. Она помогает развивать логическое мышление и улучшать координацию движений.

В целом, игра может заинтересовать широкую аудиторию людей, желающих провести время с пользой и развить свои когнитивные навыки.

Технические требования к устройству.

Сенсорные устройства *Android*. Для более плавной работы рекомендуется использовать более новые версии операционной системы. Также необходимо наличие свободного места на устройстве - минимальный объем свободного места

от 20 до 50 МБ. В целом, для игры не требуется мощное устройство, и она может работать на большинстве современных смартфонов и планшетах. Однако для более

комфортной игры рекомендуется выбирать устройства с более крупными экранами и более быстрыми процессорами.

Ниже, на рисунке 4 представлена *UML*-диаграмма вариантов использования. Она позволяет описать функциональные требования и определить, какие задачи должно выполнять приложение. На данной диаграмме показаны различные варианты использования приложения. Например, пользователь может начать новую игру, выбрать уровень, перейти на уровень и т.д.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

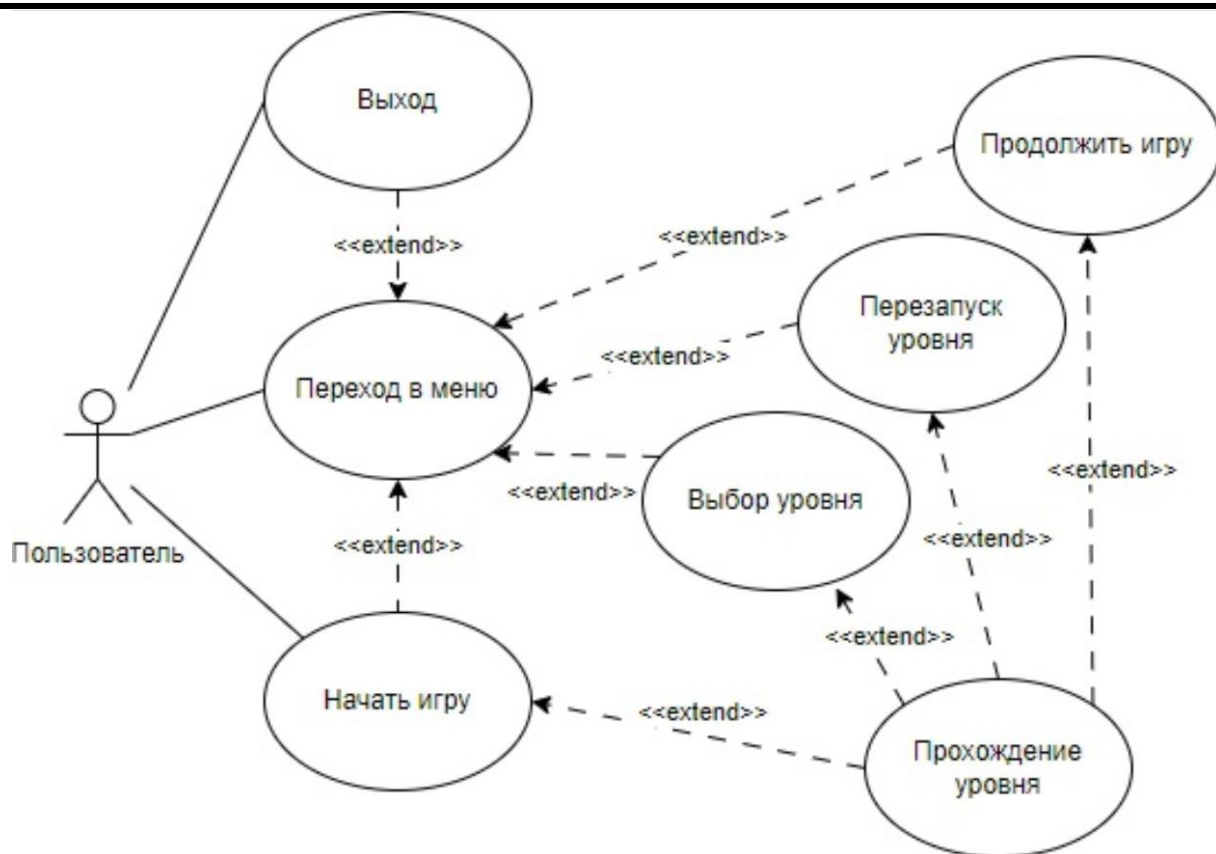


Рисунок 4 — UML-диаграмма вариантов использования

1.6 Выбор среды разработки

1.6.1 Unity

Unity представляет собой профессиональный игровой движок, который используется в создании видеоигр для различных платформ. Это инструмент, которым ежедневно пользуются опытные разработчики, а также один из наиболее доступных инструментов для новичков [3]. *Unity* позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-

приложения и другие [4]. И создавать 2D и 3D игры. Он обладает мощными инструментами для создания игровых объектов, анимации, физики, света и звука.

Разработка игры на *Unity* представляет собой высокотехнологичный и мультидисциплинарный процесс, который включает в себя работу с 3D/2D-графикой, специальными эффектами, звуковым оформлением, создание игровых объектов, физикой, создание интерфейса пользователя многими другими аспектами. Так же предоставляет мощные инструменты для создания и управления анимациями, что позволяет создавать более реалистичные и живые игровые персонажи и обеспечивает поддержку различных физических движков, что позволяет создавать более реалистичную физику в играх [1].

Кроме того, создание игры на *Unity* требует от разработчика обширных знаний в области программирования на C#, написания скриптов и управления кодом. Также, важно иметь опыт работы с редактором *Unity* и знание принципов геймдизайна.

Любой игровой движок предоставляет множество функциональных возможностей, которые задействуются в различных играх. Реализованная на этом движке игра получает все эти функциональные возможности, кроме того, добавляются ее собственные игровые ресурсы и код игрового сценария. *Unity* предлагает моделирование физических сред, карты нормалей, преграждение окружающего света в экранном пространстве. Подобным набором

функциональных возможностей могут похвастаться многие игровые движки, но у

Unity есть два основных преимущества перед другими передовыми инструментами разработки игр: чрезвычайно производительный визуальный рабочий процесс и мощная межплатформенная поддержка [3].

Преимуществом *Unity* является простота использования. Среда разработки имеет интуитивно понятный интерфейс и предоставляет множество готовых компонентов и ресурсов, которые вы можете использовать в своих проектах. *Unity* также имеет мощный движок, позволяющий создавать качественные игры с

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

хорошей производительностью. Разнообразие инструментов и расширений позволяет разработчикам создавать различные типы игр и приложений.

1.6.2 Unreal Engine

Unreal Engine — это еще одна мощная среда разработки игр с открытым исходным кодом, использующая язык программирования C++. *Unreal Engine* имеет отличную графику и поддерживает множество платформ, включая *Android*. Он также содержит множество инструментов и ресурсов, которые могут помочь вам в разработке вашей игры. Он имеет мощный графический движок, который позволяет создавать реалистичные 3D-модели и эффекты. Система сценариев позволяет разработчикам создавать собственные функции и поведение игровых объектов. *Unreal Engine* имеет мощную систему материалов, позволяющую создавать высококачественные текстуры и эффекты. Это может быть очень полезно для создания реалистичных игровых миров. *Unreal Engine* имеет встроенный графический редактор *Blueprints*, который позволяет создавать игровую логику без необходимости знания программирования. *Unreal Engine* имеет мощный редактор уровней, который позволяет создавать игровые миры из различных объектов, таких как стены, двери, окна и т. д. Редактор уровней также позволяет настраивать освещение, атмосферу и другие аспекты игрового мира.

1.6.3 Construct

Construct — относительно новая среда разработки игр, интуитивно понятный игровой движок для создания 2D-игр. Этот движок позволяет создавать игры без

программирования, используя графический интерфейс и систему поведения объектов.

Игры написаны на *JavaScript*. Он имеет простой пользовательский интерфейс и может помочь вам быстро и эффективно создавать игры. Одним из основных преимуществ *Construct* является простота использования. Создавать игры на этом движке могут даже люди без опыта программирования благодаря интуитивно понятному интерфейсу и системе поведения объектов. *Construct*

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

также имеет богатый набор инструментов для создания 2D-игр, включая возможность создания анимации, редактирования уровней, создания пользовательских объектов и многого другого. Кроме того, *Construct* поддерживает экспорт игр для нескольких платформ, включая *Windows*, *Mac*, *Linux*, *Android* и *iOS*, что делает его универсальным инструментом разработки игр. Недостатком *Construct* является то, что он не такой мощный, как некоторые другие игровые движки, такие как *Unity* или *Unreal Engine*. Это может ограничить возможность создания более сложных игр с продвинутыми функциями.

1.6.4 GameMaker Studio

GameMaker Studio — эта среда разработки игр имеет мощный движок для создания 2D-игр и поддерживает множество платформ, включая *Android*. *GameMaker Studio* использует собственный язык программирования под названием *GameMaker Language*. Он также имеет простой пользовательский интерфейс и множество инструментов, которые помогут вам создать свою игру. *GameMaker Studio* предоставляет множество инструментов для создания игр, в том числе возможность создавать пользовательских сценариев, персонажей и объекты, а также редактировать уровни. Этот движок также имеет встроенные инструменты анимации и звуковых эффектов. Одним из основных преимуществ *GameMaker Studio* является простота использования. Создавать игры на этом движке могут даже люди без опыта программирования благодаря интуитивно понятному интерфейсу и системе поведения объектов. Поддерживает игры для различных платформ, включая: *Windows*, *macOS*, *Linux*, *Android*, *iOS*, *Xbox*, *PlayStation*, что

делает его универсальным инструментом разработки игр.

1.6.5 Godot

Godot — это открытая среда разработки игр, предоставляющая множество инструментов для создания 2D- и 3D-игр. Он имеет мощный движок и поддерживает множество платформ, включая *Windows*, *Linux*, *macOS*, *Android*, *iOS* и веб-браузеры. У него также есть отличное сообщество разработчиков и

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

множество ресурсов, которые помогут вам в разработке вашей игры. Он использует собственный язык программирования под названием *GDScript*. *Godot* также поддерживает другие языки программирования, такие как *C#*, *C++* и *VisualScript*. *Godot* имеет мощный визуальный редактор, позволяющий создавать игровые объекты, задавать их свойства, создавать анимации и управлять сценами. Он также имеет множество инструментов для создания спецэффектов, редактирования звука и создания пользовательских редакторов. *Godot* также имеет встроенную систему физики, которая позволяет создавать реалистичные физические эффекты, такие как гравитация, столкновения и деформация объектов. Кроме того, *Godot* поддерживает множество плагинов и расширений, которые позволяют создавать игры с различными функциями и возможностями.

1.7 Выбор языка программирования

1.7.1 Язык программирования *C#*.

C# — это язык программирования, который используется для создания приложений, веб-сайтов и игр для платформ *Windows*, *Linux*, *macOS*, *Android* и *iOS*. *C#* является одним из наиболее популярных языков программирования и имеет обширную документацию и сообщество разработчиков.

C# является объектно-ориентированным языком программирования, что означает, что он позволяет создавать классы, объекты, методы, свойства и события для организации данных и функций в логические блоки. Он также имеет множество встроенных типов данных, таких как целые числа, дробные числа, логические значения и строки, которые упрощают работу с данными. *C#* также

поддерживает множество функций, таких как управление памятью, обработка исключений, многопоточность и асинхронность, которые позволяют создавать эффективные и надежные приложения. *C#* имеет простой и понятный синтаксис, что делает его доступным для начинающих программистов. Однако, он также имеет множество продвинутых функций и возможностей, что делает его популярным среди опытных разработчиков.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

В целом, C# — это мощный и гибкий язык программирования, который позволяет создавать широкий спектр приложений и игр для различных платформ.

1.7.2 Язык программирования *Java*.

Java - это основной язык программирования для приложений *Android*. Имеет широкую поддержку и множество библиотек и инструментов, которые можно использовать для создания игровых приложений.

Основные возможности *Java*:

- объектно-ориентированный подход: *Java* основана на концепции объектов, что позволяет разработчикам создавать модульные приложения и повторно использовать код;
- переносимость: благодаря *JVM*, приложения *Java* могут работать на любой платформе, где установлена соответствующая версия *JVM*;
- безопасность: *Java* имеет встроенные механизмы безопасности, которые позволяют защитить ваше приложение от вредоносного кода и утечек данных;
- многопоточность: *Java* поддерживает многопоточность, что позволяет одновременно выполнять несколько задач в рамках одного приложения;
- простота: язык *Java* имеет простой и лаконичный синтаксис, что облегчает его чтение и понимание разработчиками.

Java имеет большое количество стандартных библиотек, содержащих множество классов и методов для решения различных задач. В их состав входят библиотеки для работы с файлами, сетевыми протоколами, графическими интерфейсами и многие другие.

Java также имеет множество интегрированных сред разработки (*IDE*), таких как *Eclipse*, *NetBeans* и *IntelliJ IDEA*, которые упрощают разработку приложений *Java*. Они предоставляют различные инструменты для создания, отладки и тестирования приложений.

В целом, *Java* — это мощный и гибкий язык программирования, который можно использовать для разработки самых разных приложений. Он пользуется

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

широкой поддержкой сообщества разработчиков и продолжает развиваться, чтобы оставаться актуальным и востребованным в наши дни.

1.7.3 Язык программирования C++.

C++ — это язык программирования, который часто используется для разработки игровых приложений. C++ имеет множество функций, которые делают его подходящим для разработки различных типов приложений, включая игры, системное программное обеспечение, веб-приложения и многое другое. Вот некоторые из его основных особенностей:

- объектно-ориентированное программирование (ООП): C++ позволяет создавать классы и объекты, что делает код более организованным, легким для чтения и обслуживания;
- низкоуровневые возможности: C++ позволяет работать напрямую с памятью и битами, что делает его очень гибким для написания системного ПО;
- высокая производительность: C++ — компилируемый язык программирования, что означает, что он компилируется в машинный код, который выполняется непосредственно на процессоре. Это делает его очень быстрым и эффективным;
- переносимость: C++ можно написать один раз и скомпилировать для разных операционных систем и аппаратных платформ, что делает его подходящим для разработки кросс-платформенных приложений;
- обширная стандартная библиотека: C++ имеет обширную стандартную библиотеку, содержащую множество полезных функций и классов для работы с файлами, строками, потоками, контейнерами и т. д.

C++ имеет множество расширений и дополнительных библиотек, которые делают его еще более мощным и простым в разработке. Некоторые из них включают *Boost*, *Qt* и *OpenGL*.

C++ — это язык программирования, который обладает высокой производительностью и возможностями низкоуровневого программирования, а

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

также поддерживает объектно-ориентированное программирование и имеет обширную стандартную библиотеку.

1.7.4 Язык программирования *JavaScript*.

JavaScript — это язык программирования, который может быть использован для создания интерактивных веб-страниц и приложений.

JavaScript является интерпретируемым языком, что означает, что его код выполняется непосредственно в браузере пользователя, а не компилируется в машинный код. Это позволяет быстро разрабатывать и тестировать приложения, а также облегчает их распространение.

JavaScript поддерживает объектно-ориентированное программирование (ООП), функциональное программирование и асинхронное программирование. Он также имеет множество встроенных функций и методов, облегчающих разработку приложений.

JavaScript используется во многих областях, включая создание веб-страниц, разработку мобильных приложений, игр, серверных приложений и многое другое. Он также широко используется в различных фреймворках и библиотеках, таких как *React*, *Angular* и *jQuery*.

Разработка *JavaScript* требует знания основ языка, таких как переменные, операторы, условные операторы, циклы и функции. Также полезно знать основы *HTML* и *CSS* для создания интерактивных веб-страниц.

1.8 Вывод

Хотя каждый язык программирования имеет свои преимущества и недостатки, и выбор языка зависит от конкретных задач, которые необходимо

решить. Однако для реализации игрового приложения был выбран язык программирования *C#*. Он более популярный язык программирования в *Unity*, и большая часть документации и руководств написана на нем. Кроме того, *C#* более строго типизирован, что делает его более безопасным и удобным для больших проектов.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

C# также упрощает взаимодействие с платформой Android, что может быть важным фактором при разработке игры для этой платформы.

По итогу проведения анализа для разработки мобильного приложения под ОС Android был выбран язык программирования C# .

1.9 Инструменты разработки

Касательно остальных инструментов разработки, для создания игрового приложения можно использовать следующие инструменты:

Visual Studio - интегрированная среда разработки (IDE), которая позволяет нам легко создавать и отлаживать код, используется для создания приложений, веб-сайтов, игр и других программных продуктов для платформ *Windows, Linux, macOS, Android и iOS*.

Visual Studio предоставляет широкий спектр инструментов и функций для разработки приложений на разных языках программирования, таких как C#, C++, *Visual Basic, JavaScript, Python* и многих других. Он также имеет множество продвинутых функций, таких как отладчик кода, система контроля версий, инструменты тестирования и оптимизации производительности.

Visual Studio также имеет интегрированный дизайнер пользовательского интерфейса, который позволяет создавать и настраивать элементы управления, макеты и стили для приложений. Он также имеет инструменты для создания баз данных, веб-сайтов и мобильных приложений.

Visual Studio также поддерживает различные плагины и расширения, которые позволяют настраивать и расширять функциональность IDE. Эти плагины и

расширения могут помочь улучшить производительность, улучшить качество кода и расширить возможности разработки.

1.10 Архитектура игрового приложения лабиринта на Unity

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

Архитектура разбита на несколько основных компонентов:

- сцена;
- объекты;
- ресурсы;
- скрипты.

Сцены — это основные единицы разработки в *Unity*. Они содержат все объекты, компоненты и ресурсы, необходимые для создания игры. Ниже на рисунке 5 представлена папка со всеми сценами, что есть в данном проекте.

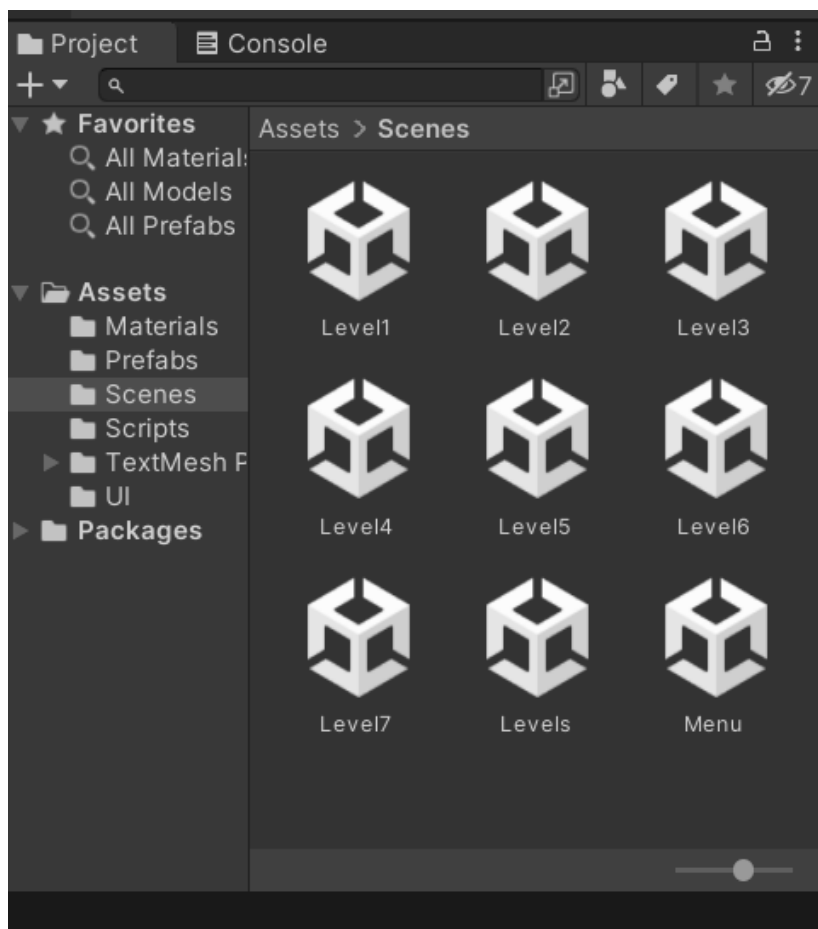


Рисунок 5 — Папки с сценами

Объекты — это базовые элементы сцены. Они могут содержать компоненты, которые определяют их поведение и взаимодействие с другими объектами.

На рисунке 6 представлена иерархия объектов на сцене с главным меню. На ней содержится :

- *Main Camera*: это компонент, который определяет точку зрения игрока в игровом мире. Он может быть настроен для изменения различных параметров, таких как поле зрения, глубина резкости и т.д.;

- *Direction Light*: это компонент, который определяет направление и интенсивность освещения в игровом мире. Он может быть настроен для изменения цвета, интенсивности и направления света;

- *Canvas*: это компонент, который определяет пользовательский интерфейс (UI) в игровом мире. Он содержит элементы управления, такие как кнопки – *Levels*, *Exit*, *Continue*, *NewGame* и *BG* представляющий собой фон;

- *Event System*: это компонент, который отвечает за обработку событий в пользовательском интерфейсе. Он может обрабатывать нажатия кнопок, перемещения и другие действия пользователя.

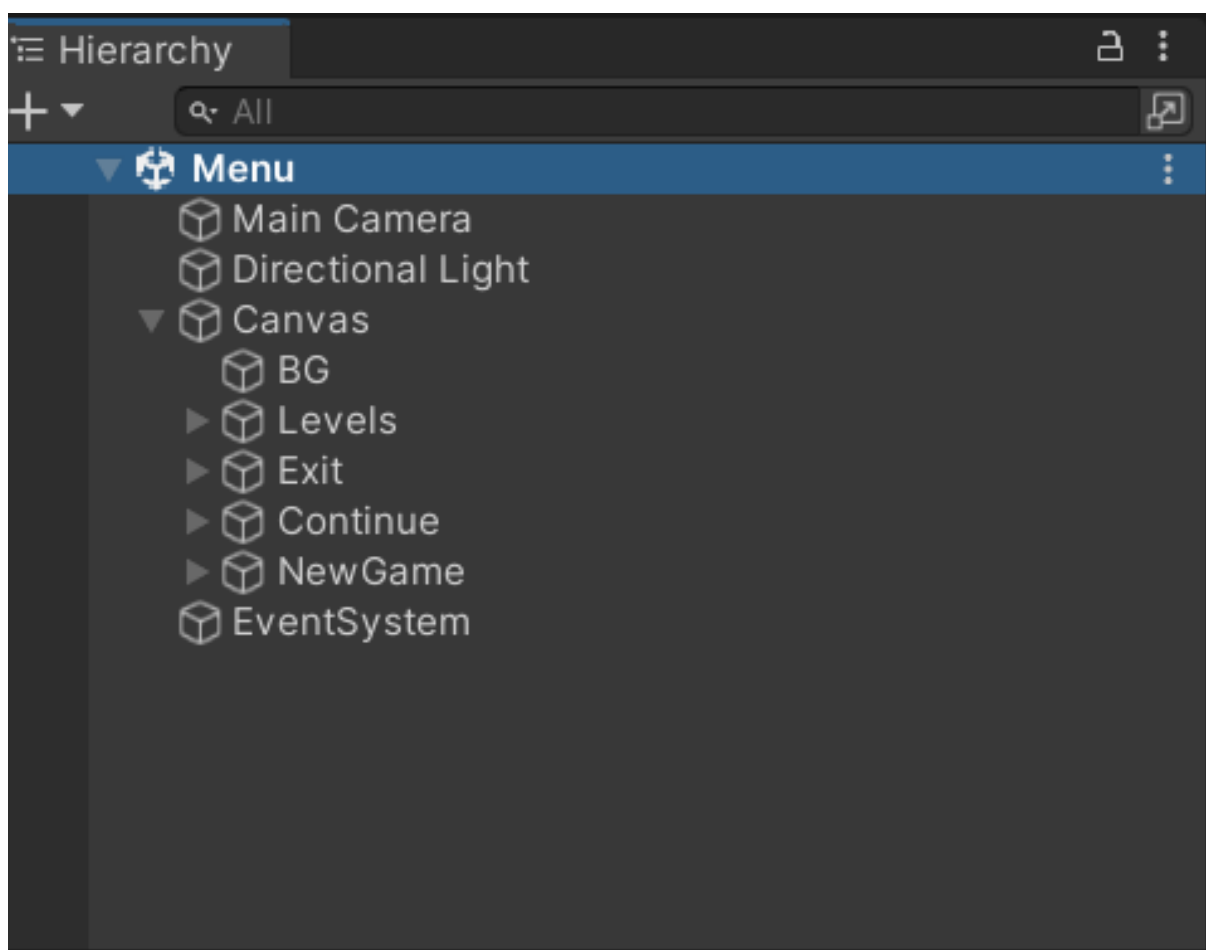


Рисунок 6 —Иерархия объектов

На рисунке 7 представлена иерархия объектов на сцене с уровнем. На ней содержится так же *Main Camera*, *Directional Light* и *Canvas* ,в котором находится *NumberLevel*.

NumberLevel— текстовое поле с номером уровня, кнопки *Reset* для перезагрузки уровня и *Menu* для перехода в главное меню.

Так же здесь содержатся: *Plane* , в котором хранятся объекты для стен и пола,

Player— объект, который должен пройти к финишу, *Finish*— объект обозначающий место конца игры .*Cubes* хранит ,объекты – препятствия ,которые нужно передвинуть чтобы пройти уровень. *WinPanel* – это панель, которая появляется когда уровень пройден.

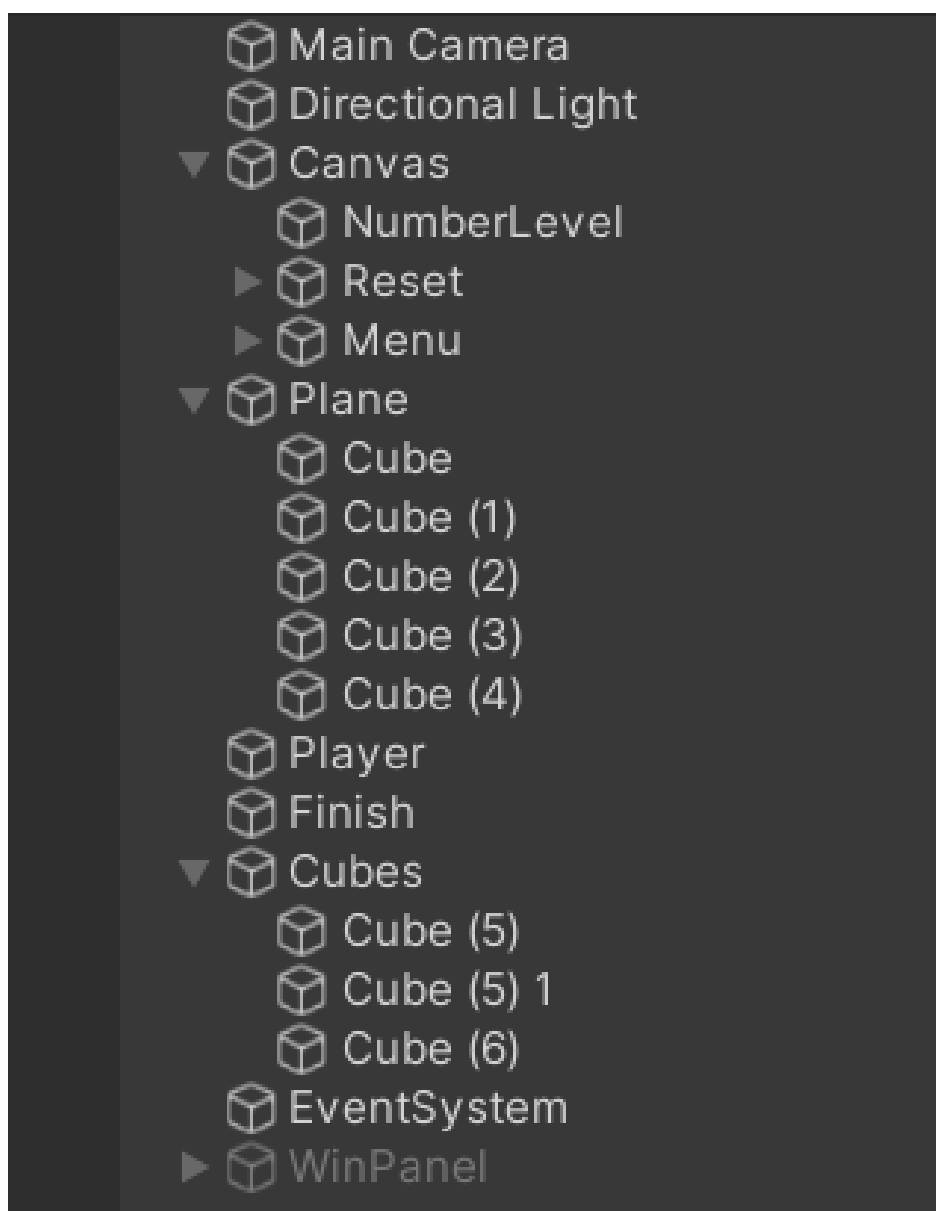


Рисунок 7 — Папки с сценами

Ресурсы — это файлы, используемые для создания объектов и сцен. Это могут быть текстуры, модели и другие типы файлов.

На рисунках 8 и 9 представлены папки, где хранились материалы с текстурами и спрайты для кнопок и фона .

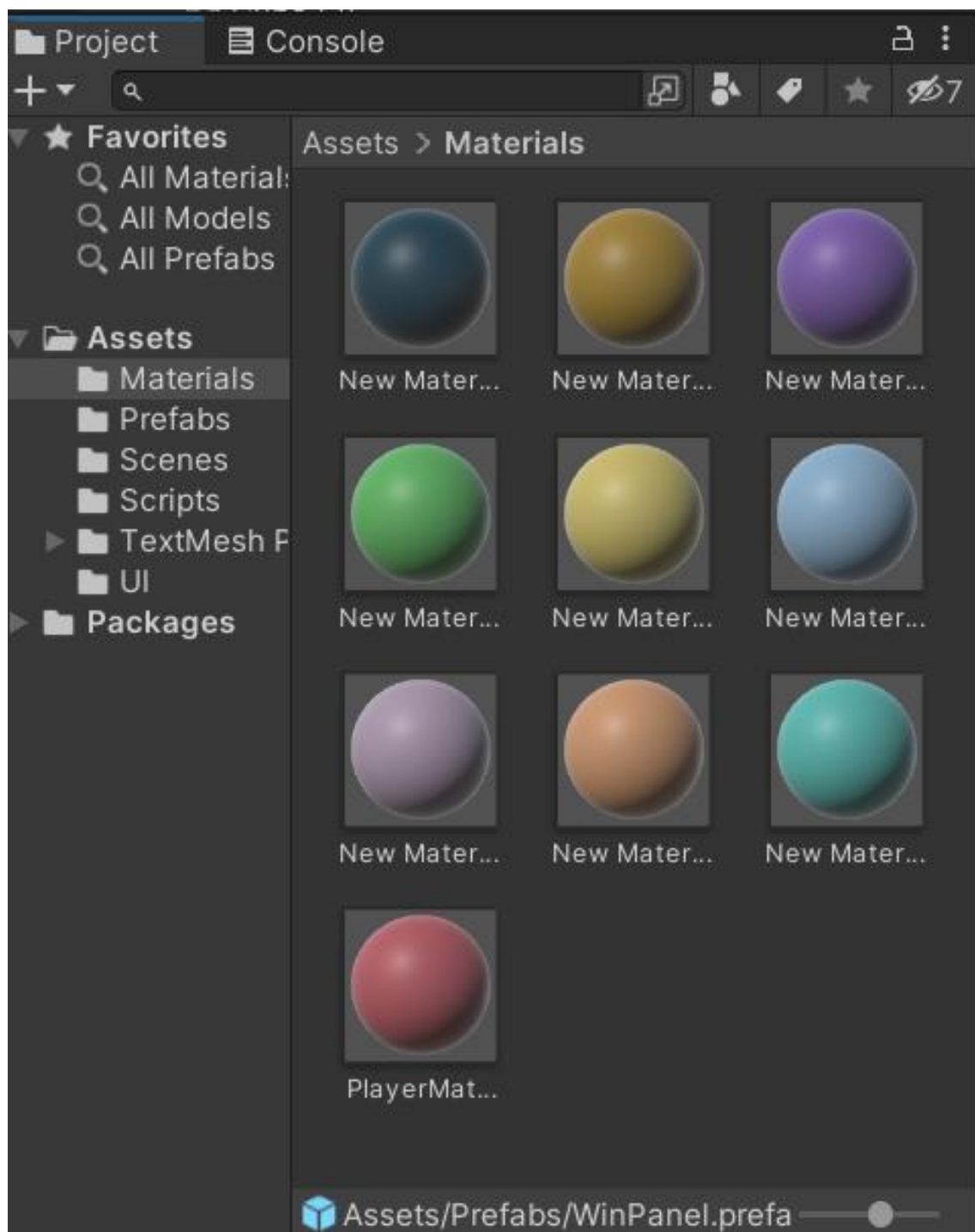


Рисунок 8 — Папки с текстурами

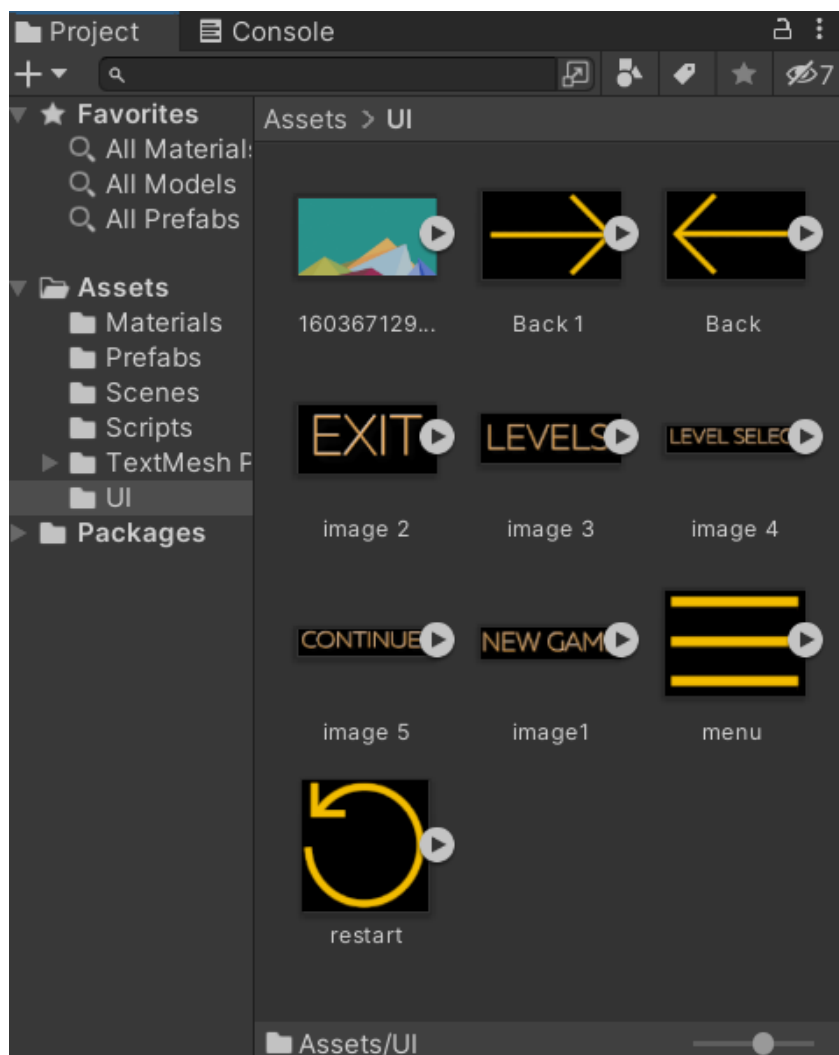


Рисунок 9 — Папки с спрайтами

Скрипты — это программные модули, написанные на языке C#. Они используются для определения поведения объектов и сцен, а также для взаимодействия с другими компонентами.

На рисунке 10 представлена папка – в ней расположены все скрипты.

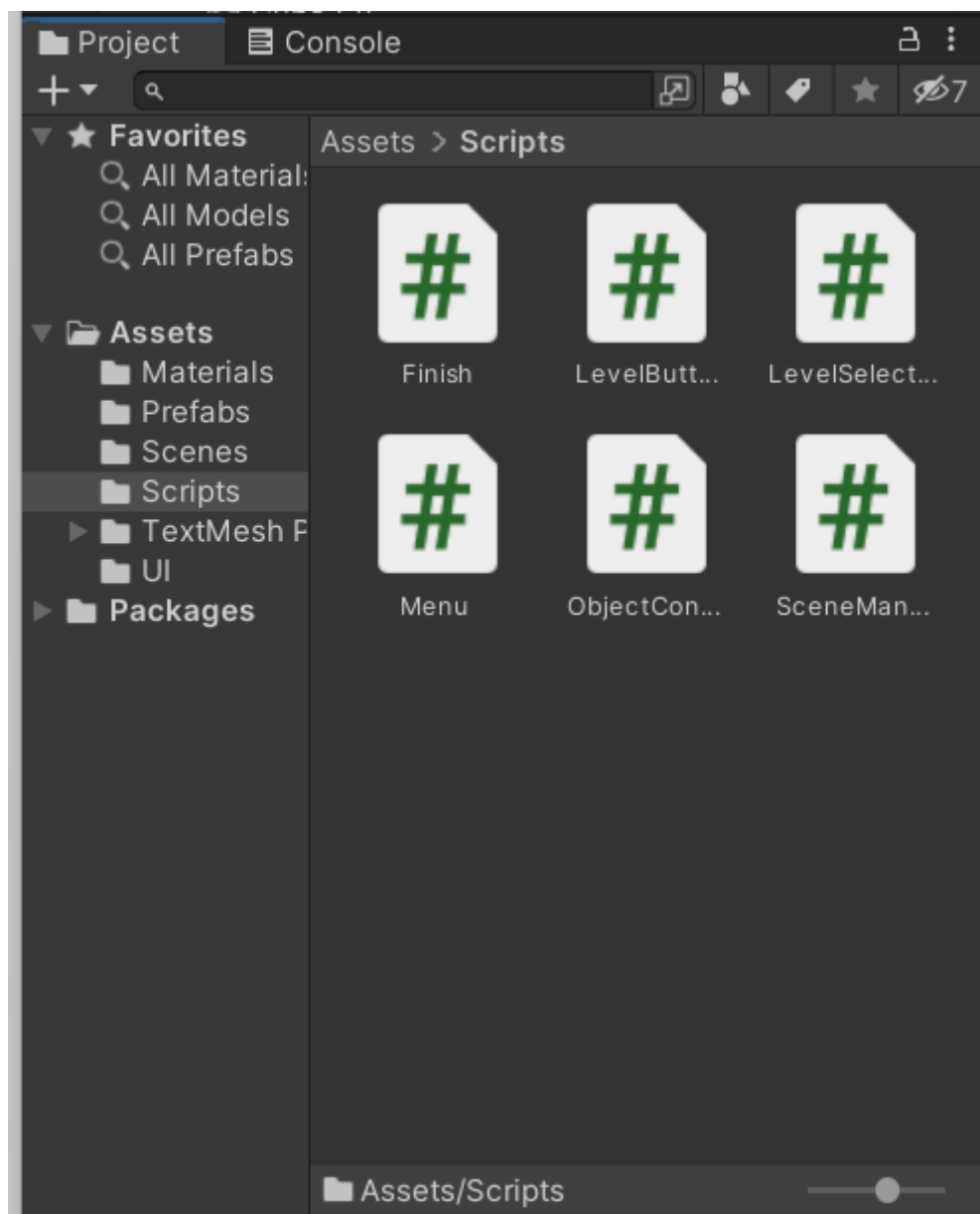


Рисунок 10 — Папки с скриптами

Так же в проекте есть папки. Папка *Prefabs* в *Unity* нужна для создания и хранения готовых объектов, которые можно использовать повторно в игре. Это удобно, когда нужно многократно использовать один и тот же объект.

TextMesh Pro — это пакет для *Unity*, который предоставляет расширенные возможности работы с текстом в игре. Он позволяет создавать более качественные и настраиваемые текстовые элементы, такие как шрифты, текстовые метки и т.д.

Packages — это папка, которая содержит все установленные пакеты в *Unity*. Это может быть как стандартные пакеты, так и сторонние пакеты, которые можно

установить из Asset Store. Они предоставляют различные функциональные возможности для разработки игр.

1.11 Разработка игрового приложения

1.11.1 Изучение технологий разработки игр на платформе *Android* и игрового движка *Unity*.

Первым шагом в разработке игрового приложения “Лабиринта” *Unity* для *Android* является изучение основ *Unity* и создание проекта. Для этого необходимо выполнить задачи описанные ниже.

Изучение основных элементов интерфейса *Unity*. Прежде чем приступить к работе с *Unity*, необходимо ознакомиться с его интерфейсом. Он состоит из нескольких окон, каждое из которых выполняет свою функцию. Основные окна, которые будут использоваться при разработке игры:

- сцена (*Scene*): окно, в котором отображается игровая сцена;
- иерархия (*Hierarchy*): окно , отображающее все объекты на сцене;
- инспектор (*Inspector*): окно, в котором можно задать свойства выбранного объекта;
- проект (*Project*): окно, в котором хранятся все игровые ресурсы (модели, текстуры, звуки и т.д.).

Создайте новый проект в *Unity*. После того, как вы изучили основные элементы интерфейса *Unity*, можно приступить к созданию нового проекта. Для этого выберите пункт «*New Project*» в меню «*File*» и укажите имя проекта, путь для сохранения и настройки проекта.

Настройка проекта для разработки игр для *Android*. После создания проекта его необходимо настроить для разработки игр для *Android*. Для этого необходимо выполнить следующие шаги:

- установите пакеты для разработки игр для *Android* в *Unity* (поддержка сборки *Android* и инструменты *Android SDK* и *NDK*);
- настройте параметры проекта для совместимости с *Android* (установите разрешение экрана, ориентацию экрана и т. д.);

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

- настройте параметры экспорта приложений *Android* (имя приложения,

значок, версия и т. д.).

После выполнения этих заданий можно приступить к созданию игры.

1.11.2 Создание игрового поля

После настройки проекта для разработки игры на *Android* необходимо приступить к созданию игрового мира. Для игры в лабиринт нужно создать игровое поле, на котором будут располагаться стены и объекты, а также объект, который будет перемещаться по лабиринту до финиша.

Игровое поле — это область, в которой игрок сможет перемещать блоки, чтобы достичь цели.

Для создания игрового поля необходимо создать объекты, который будет использоваться для создания стен и пола на игровом поле. Для этого можно использовать стандартные модели из *Unity*.

Создание блоков. Блоки — это элементы, которые игрок должен сдвинуть, чтобы достичь цели (рисунок 11). Блоки могут быть различных форм и размеров.

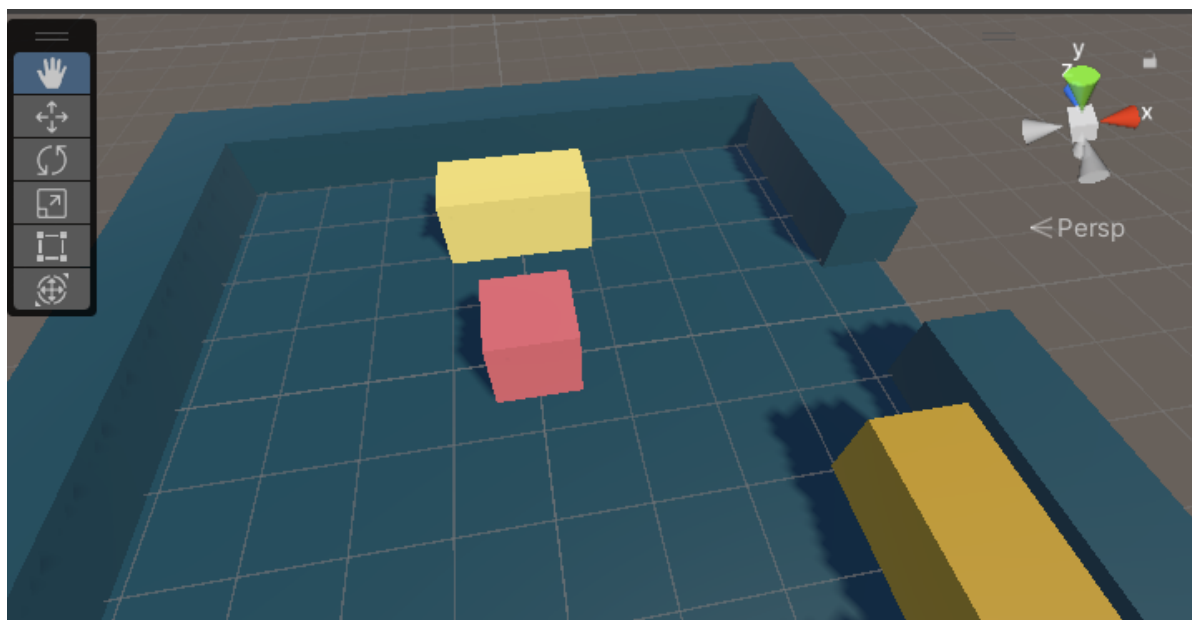


Рисунок 11– Блоки

Для создания блоков необходимо создать объекты для чего так же можно использовать стандартные модели из *Unity* или создать свою собственную модель в 3D-редакторе .

Затем нужно добавить к объектам компоненты, которые будут отвечать за их поведение на игровом поле (например, компонент "*Rigidbody*" для объектов, которые должны падать под действием гравитации). Для создания игровых объектов используются различные средства *Unity*, такие как *Colliders* для назначения коллизий и т.д.

В результате выполнения всех этих задач будет создан игровой мир для игры в лабиринт на *Unity* для *Android*.

1.11.3 Реализация игровой логики и механики

Один из ключевых аспектов при разработке игры является проектирование геймплея и механик игры.

Геймплей — это основная часть игры, которая будет управлять всеми аспектами игрового процесса. Геймплей будет включать в себя управление персонажем, уровнями лабиринта, интерактивными объектами и т.д.

Логика игры — это та часть приложения, которая будет управлять всеми правилами игры, такими как условия победы и поражения. Для реализации логики игры можно использовать скрипты *C#* в *Unity*.

Цель — это розовый блок, который нужно освободить, переместив другие блоки (рисунок 12). Она может находиться в любом месте на игровом поле.

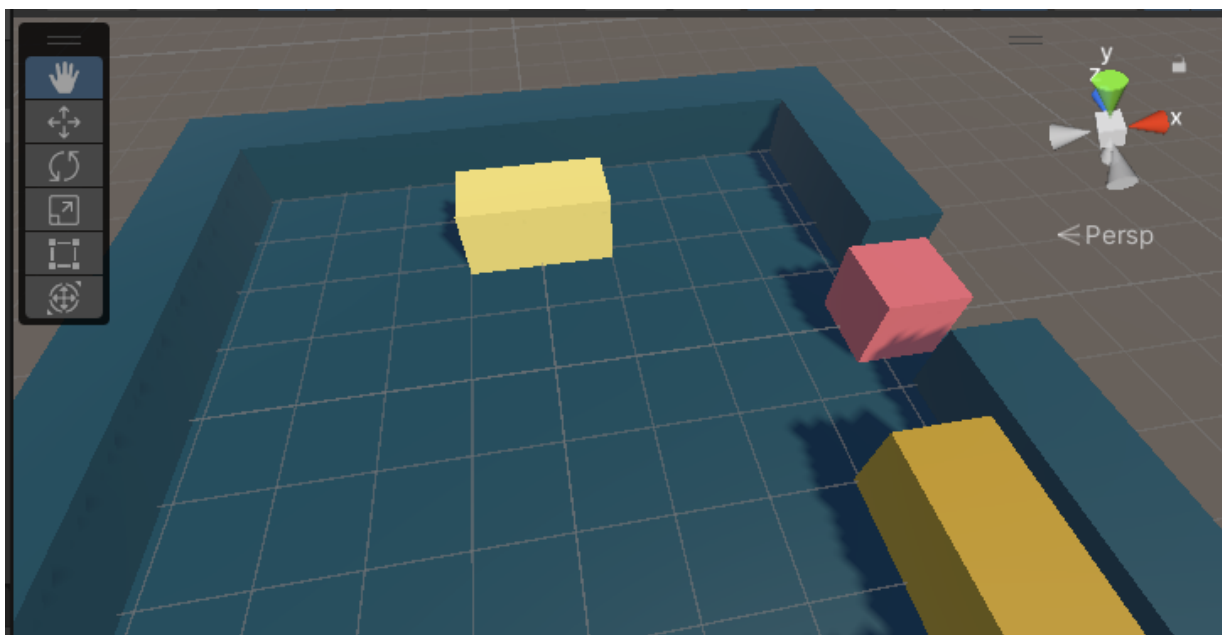


Рисунок 12 – Цели игры прохождение уровня

В классе *Finish* .cs реализуется цель игры -конец уровня и в нем есть следующие методы :

- *Start()*. В данном случае, метод определяет индекс следующей сцены (уровня), который будет загружен при прохождении текущего уровня.

- *OnTriggerEnter*. Используется для обработки столкновений объектов. Метод проверяет, что столкнувшийся объект нужный розовый блок. через тег , и если это так, то выводится панель и сохраняет его индекс в настройках игры. Если текущий уровень является последним, то выводится панель о том, что все уровни пройдены.

- *NextLevel()* . Загружает следующий уровень

Меню выигрышного уровня содержит имеет 2 кнопки: для перехода на следующий уровень и для запуска новой сцены, переходит в основное меню (рисунок 13).



Рисунок 13 – Меню победы

Для игры лабиринт были реализованы механики, описанные ниже.

Управление блоками. Блоки были созданы необходимо создать скрипт, который будет отвечать за управление объектами. В этом скрипте нужно определить, каким образом будут перемещаться блоки по игровому полю, а также определить его скорость и поведение при столкновении со стенами. Так же нужно определить, каким образом будут взаимодействовать объекты друг с другом.

Блоки могут быть перемещены только в том направлении, в котором нет препятствий вдоль своей оси. Движение объектов осуществляется путем перетаскивания блоков пальцем по экрану (жесты, "touchscreen controls" (сенсорное управление)).

В классе реализуется движение объектов *ObjectController.cs* и в нем есть следующие методы :

- *Start()*: метод используется для получения ссылки на физический компонент *Rigidbody*, который является необходимым для выполнения перемещения объекта и присваивает ему значение массы;

- *Update()*: вызывает метод, который отвечает за обработку движения;

- *Swipe()*: метод, который обрабатывает взаимодействие пользователя и перемещение объекта. Если пользователь нажимает на объект, он может начать перетаскивание объекта. Когда пользователь отпускает его, объект перемещается в заданном направлении на основе движения пальца пользователя.

Реализация системы коллизий. Для реализации системы коллизий мы будем использовать физический движок *Unity*. Мы будем размещать коллайдеры на препятствиях и персонаже, чтобы определить, когда они сталкиваются друг с другом.

Препятствия на пути к финишу. Препятствия осуществляются в виде блоков, расположенных на поле, которые нужно передвинуть для прохождения уровня.

1.11.4 Создание пользовательского интерфейса и дизайн.

Графический интерфейс пользователя (*GUI*) — это то, что пользователь увидит на экране. *GUI* будет содержать элементы, такие как кнопки меню, изображения и текст. Для создания *GUI* можно использовать встроенные средства *Unity* или создать собственную пользовательскую оболочку с помощью сцен и префабов.

Интерфейс пользователя должен быть интуитивно понятен и удобен для использования на мобильных устройствах. Это может включать в себя :

- создание главного меню;
- создание меню выбора уровней;
- создание кнопок.

Главное меню игры является первой сценой отображается при запуске игры и позволяет игрокам начать игру, продолжить, выбрать уровень или выйти

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

из игры. Имеет 4 кнопки: *New game* –запускает новую сцену, *Continue* – продолжить игру, *Levels* — переход на сцену с выбором уровня . *Quit*–выход из игры (рисунок 14).

Реализовано в классе *Menu.cs* .В нем есть следующие методы :

- *Level()*: активирует сцену с выбором уровня;
- *ExitGame()*: выход из игры;
- *Continue()*: продолжение игры , переход на сцену с не пройденным уровнем;
- *NewGame()*: начало игры ,удаление данных о пройденных уровнях.

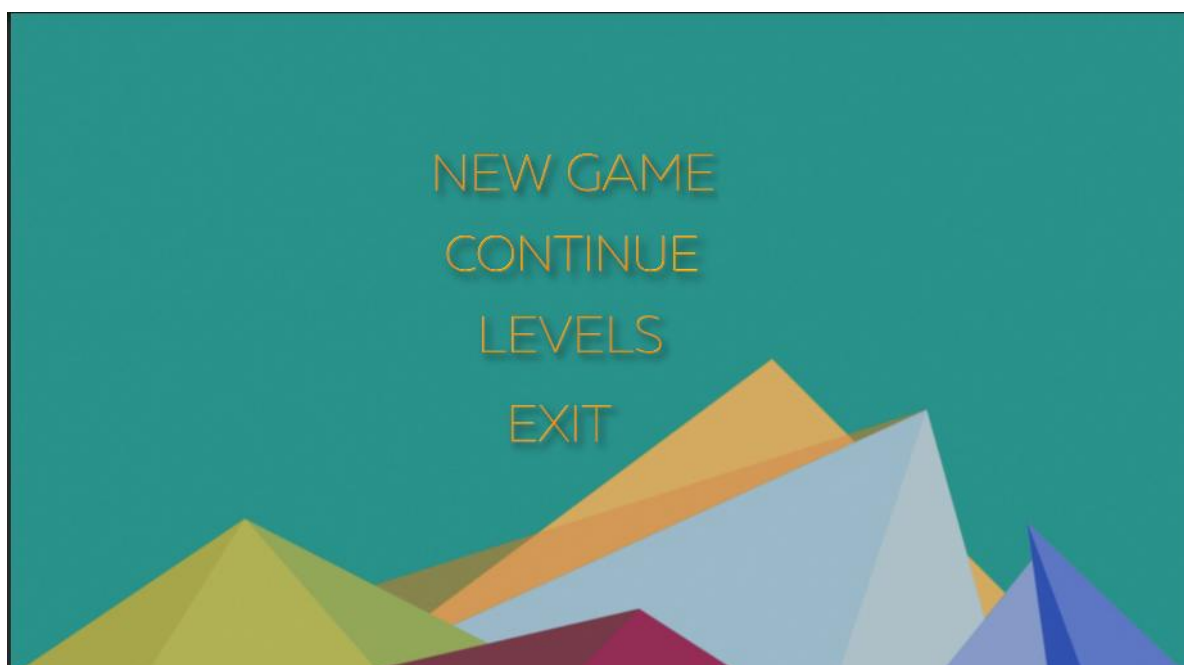


Рисунок 14 – Главное меню игры

Создание меню выбора уровня .Меню выбора уровня игры содержит кнопки для выбора уровня из пройденных и возвращение в меню (рисунок 15).



Рисунок 15 – Меню выбора уровня

Реализовано в классе *SceneManaging.cs*. В нем есть следующие методы :

- *ChangeScene*: выбор уровня и переход на сцену в зависимости от номера сцены.

Класс *LevelSelection.cs* :

- *Start()*: используется для управления кнопками уровней в игре : из сохраненных настроек игры и запрещает доступ к кнопкам уровней, которые еще не были достигнуты игроком.

Создание дизайна .Может включать в себя графические элементы, такие как фоны, текстуры и т.д.

1.11.5 Создание основного игрового экрана

Основной игровой экран — отображает игровое поле, объекты, информацию о текущем уровне , кнопки перезагрузки сцены и выход в меню (рисунок 16).

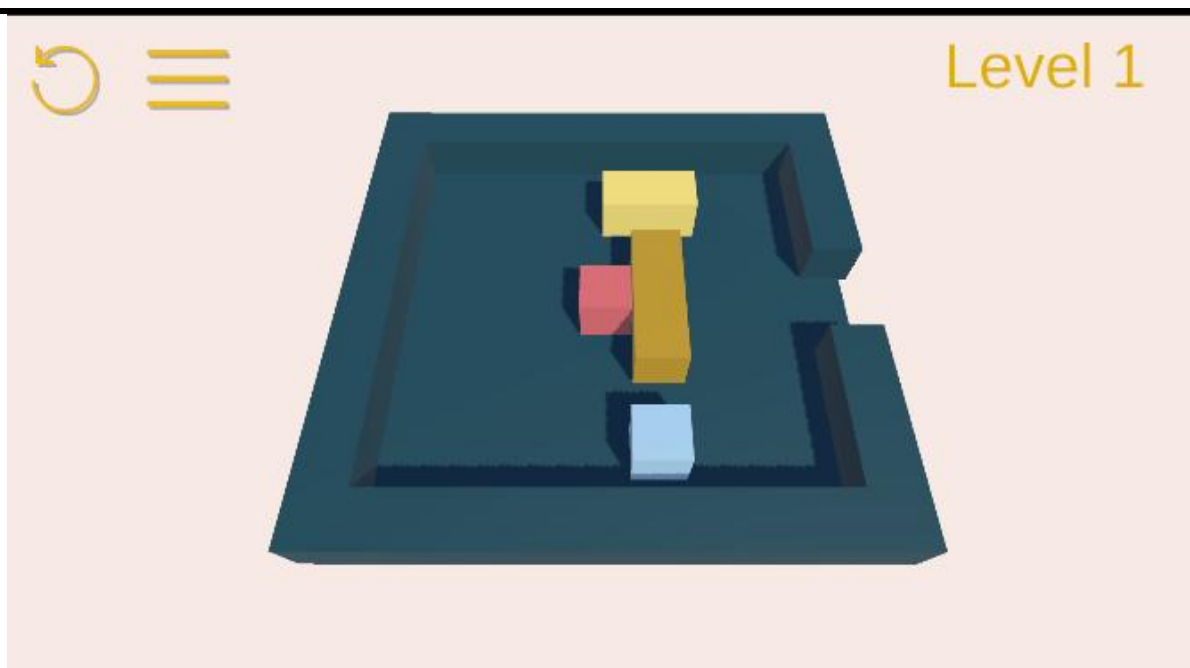


Рисунок 16 – основной игровой экран

Класс *LevelButton.cs* :

- *ResetLevel()*: перезапуск сцены;
- *Menu()*: переход на сцену меню.

1.11.6 Уровни

Уровни — это наборы игровых полей, которые игрок должен пройти, чтобы завершить игру. Они могут иметь различную сложность и размеры игровых полей и быть созданы в *Unity* с помощью инструментов редактирования сцен.

Ниже на рисунке 17 представлена *UML*—диаграмма классов. Она показывает структуру классов. В ней присутствуют классы для игрового поля, блоков, уровней, конца уровня, а также классы для управления игрой и сохранения прогресса.

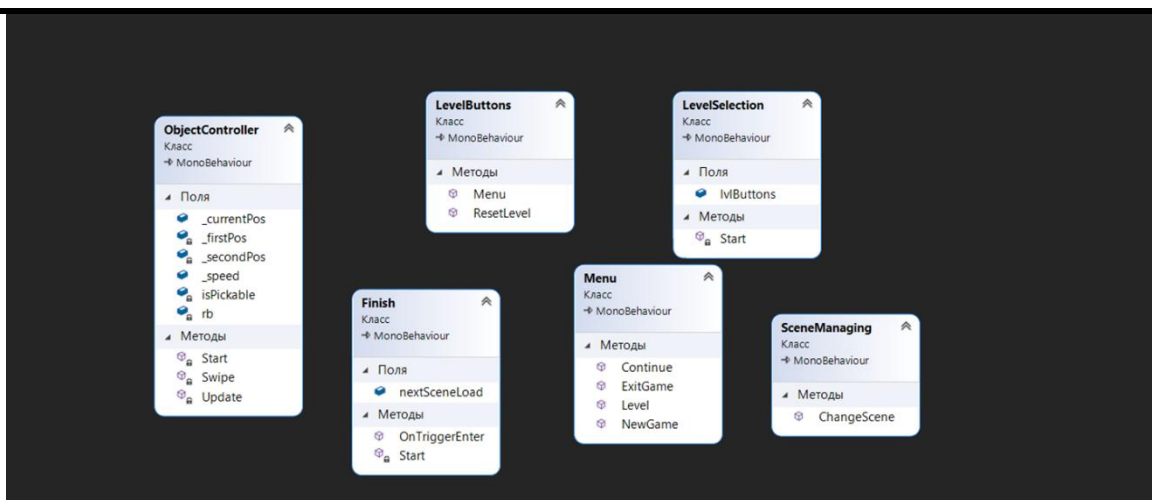


Рисунок 17 – *UML*-диаграмма классов

Ниже на рисунке 18 представлена *UML*-диаграмма деятельности, она позволяет представить последовательность действий, необходимых для решения определенных задач.

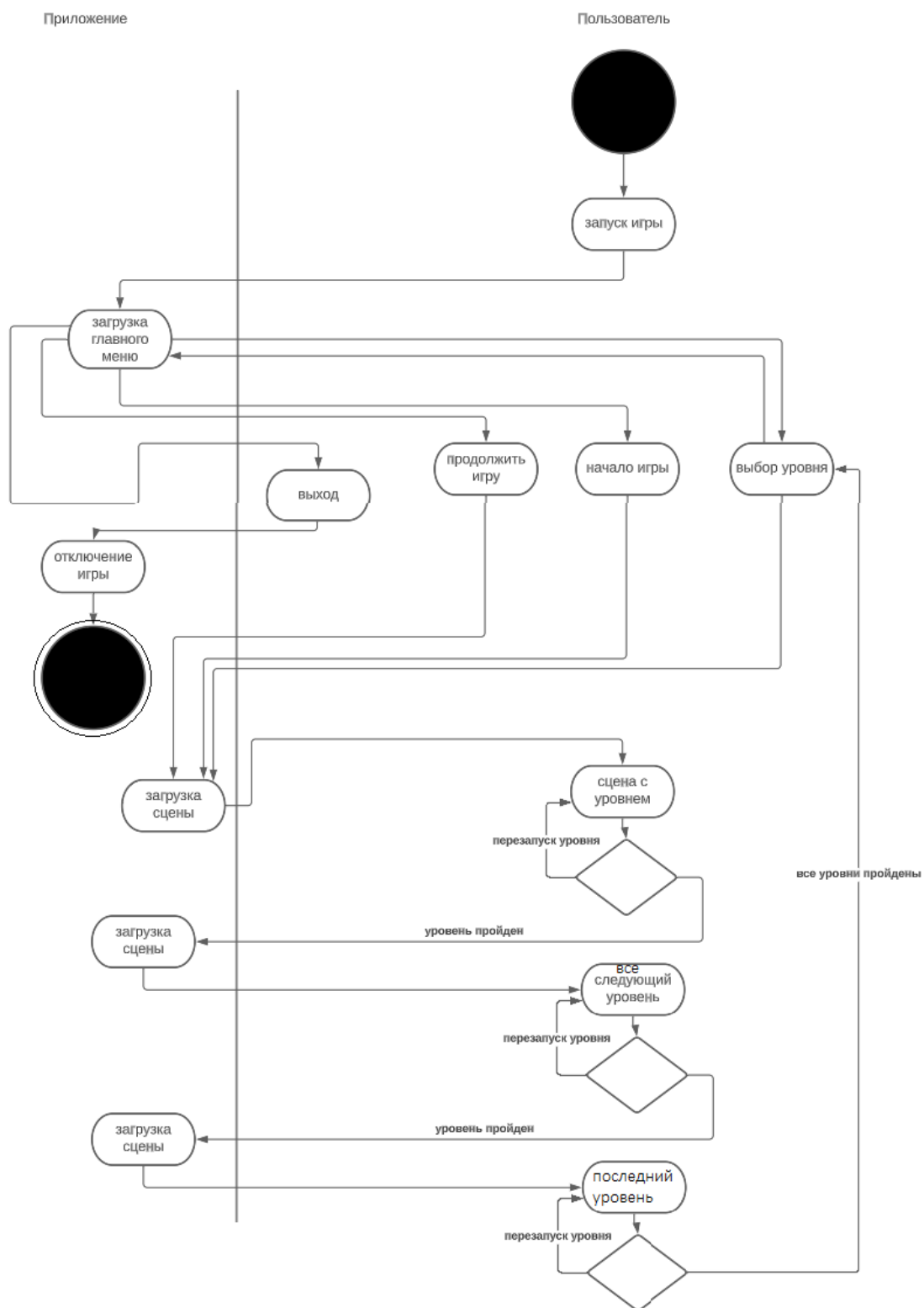


Рисунок 18 – UML-диаграмма деятельности

1.9 Тестирование приложения

Для проверки корректности работы игрового приложения оно было протестировано напрямую в системе *Unity*, через специальные модули. После этого был выполнен *build* проекта для платформы *Android*, сформировали *apk* файл, который можно загружать на мобильное устройство. В ходе проверки ошибок и недочетов выявлено не было. На рисунках 19 - 23 представлены сцены из игры.

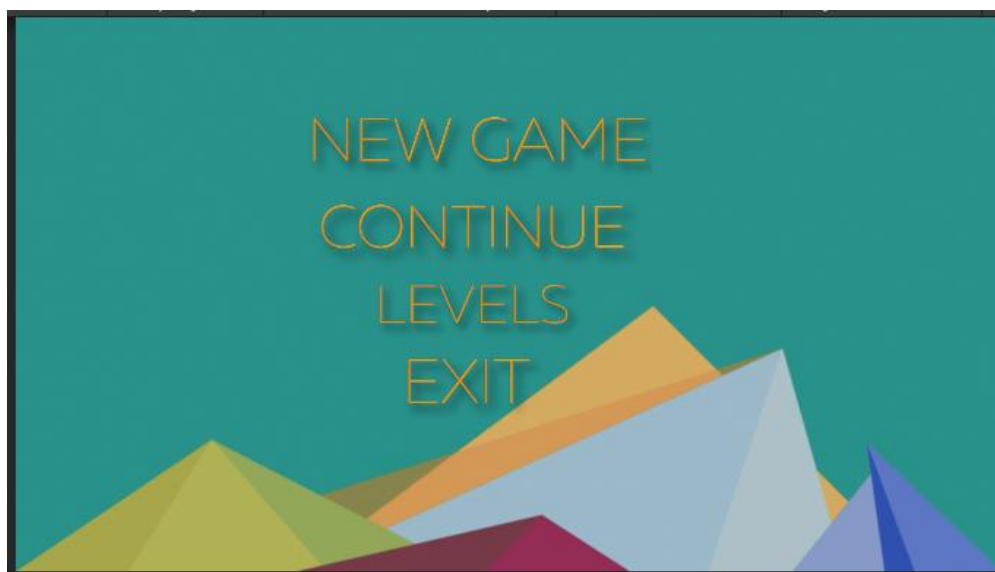


Рисунок 19 – Главное меню приложения

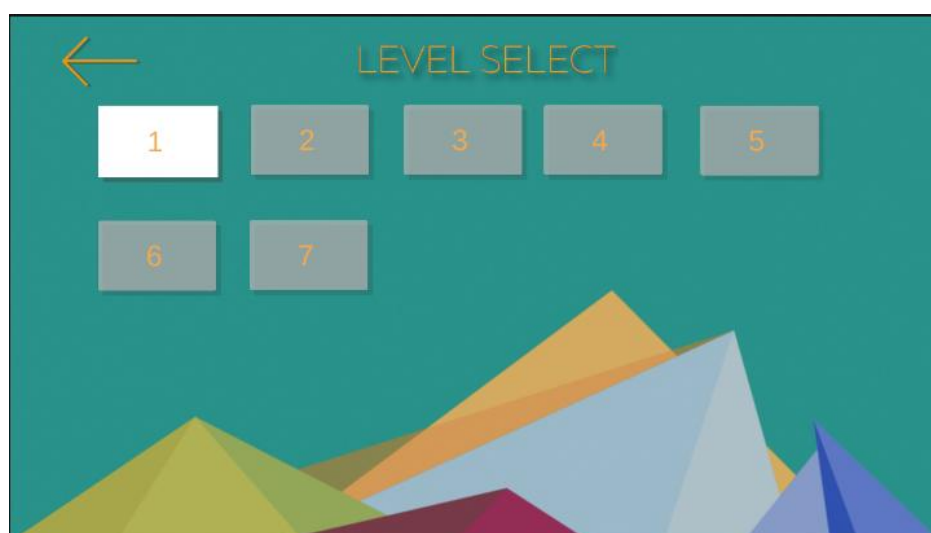


Рисунок 20 – Меню выбора игрового уровня

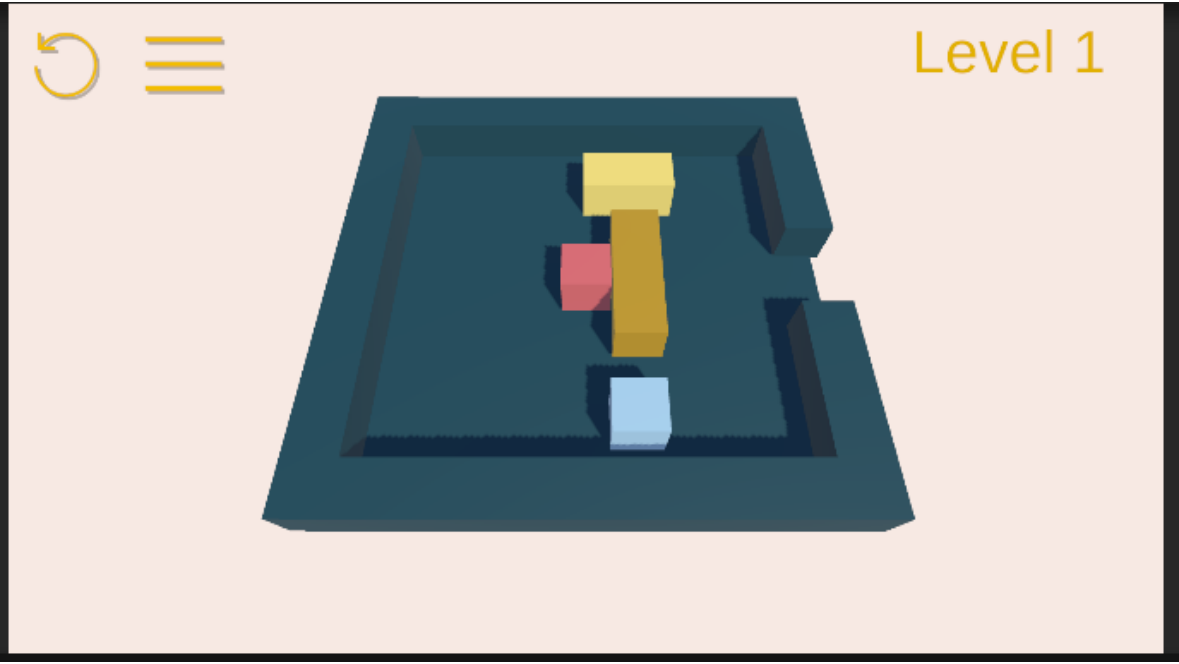


Рисунок 21 –Игровой процесс

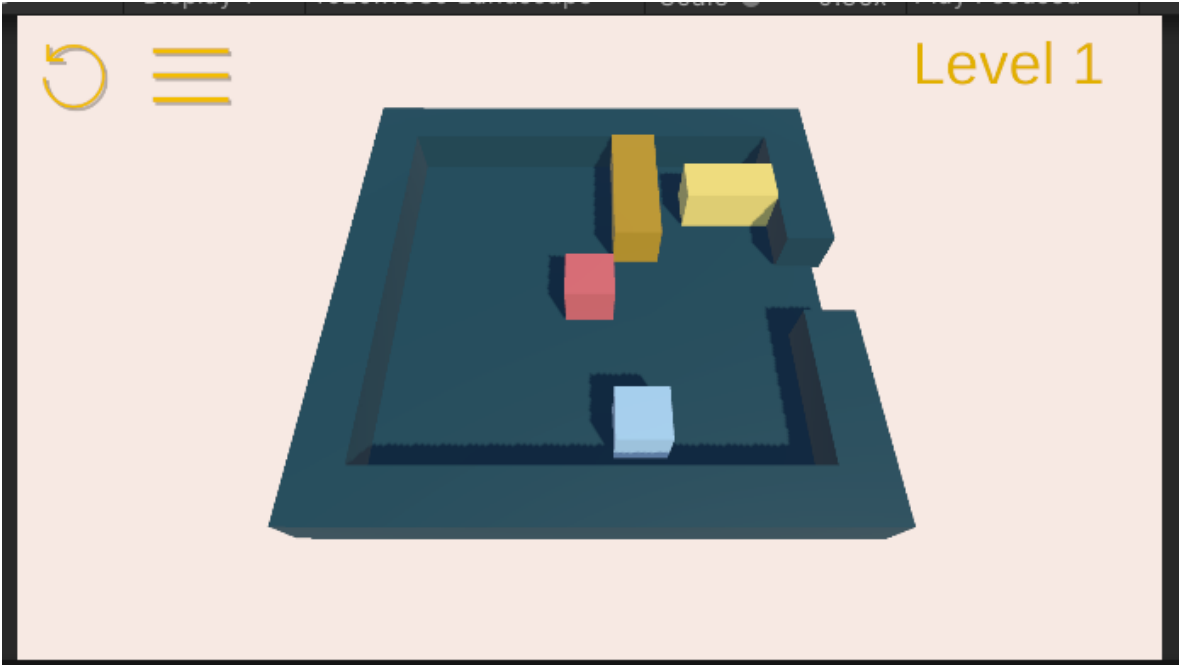


Рисунок 22 –Игровой процесс



Рисунок 23 –Конец уровня. Победа.

Использование игрового приложения “Лабиринт” на виртуальных и физических устройствах не выявило ошибок и проблем во время работы. Тесты показали ,что приложение работает корректно и может использоваться игроком.

					ПГУ1.090301.5.001 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

2 Охрана труда

2.1 Вредоносные факторы

При разработке приложений необходимо учитывать не только функциональные и эстетические требования, но и требования безопасности жизнедеятельности.

Организация рабочего места является одним из важных аспектов безопасности работы за компьютером. Разработчики должны убедиться, что рабочее место соответствует эргономическим требованиям, таким как правильное положение тела, использование комфортной мебели и оборудования. Регулярные перерывы для отдыха и гимнастики для глаз также являются важными мерами для предотвращения физических рисков.

Требования безопасности для работы за компьютером – это обязательные правила, необходимые для минимизации вредного воздействия офисного оборудования на здоровье и работоспособность сотрудников. Использование персонального компьютера не так безопасно, как можно подумать. В список основных вредных и опасных факторов трудового процесса входят:

- высокая температура, характерная для отдельных элементов компьютерной техники, влияющая на общий температурный фон в помещении;
- повышенная или пониженная влажность воздуха;
- высокая степень монотонности рабочего процесса;
- значительный уровень зрительный нагрузок;
- недостаточная освещенность рабочей зоны;
- риск поражения статическим электричеством при случайном контакте с отдельными элементами техники;
- повышенный уровень напряженности электрического и магнитного полей, генерируемых работающей техникой;

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

- повышенная яркость света, прямая и отраженная блескость, повышенный уровень контрастности рабочего экрана, негативно влияющий на зрение работника в длительной перспективе.

2.2 Требования к помещению

Одним из основных аспектов безопасности является организация рабочего места.

При создании рабочего места необходимо учитывать эргономические требования, такие как правильное положение тела, использование комфортной мебели и оборудования. Например, стул должен иметь регулируемую высоту и спинку, а монитор должен быть настроен на оптимальную яркость и контрастность.

Кроме того, важно учитывать требования к помещению. Рабочее место должно быть хорошо освещено и проветриваемо. Также необходимо предусмотреть возможность регулировки температуры в помещении.

Основными требованиями к рабочему месту являются:

- эргономика. Правильное положение тела, использование комфортной мебели и оборудования. Стул должен иметь регулируемую высоту и спинку, а монитор должен быть настроен на оптимальную яркость и контрастность;

- освещение. Рабочее место должно быть хорошо освещено. Необходимо учитывать индивидуальные особенности зрения каждого сотрудника;

- вентиляция. Рабочее место должно быть хорошо проветриваемым. Необходимо предусмотреть возможность регулировки температуры в помещении;

- регулярные перерывы для отдыха и гимнастики для глаз. Разработчики могут рекомендовать пользователям использовать специальные упражнения для глаз и рук, которые помогут предотвратить повреждения;

- микроклимат. Оптимальный микроклимат на рабочем месте — это температура воздуха от 20 до 24 градусов, влажность от 40 до 60%, скорость движения воздуха не более 0,2 м/с;

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		52

- правильное расположение оборудования. Клавиатура и мышь должны быть расположены на уровне локтей, монитор — на уровне глаз;
- чистота и порядок на рабочем месте. Рабочее место должно быть чистым и аккуратным.

2.3 Требования к освещению

Освещение является одним из важных факторов при работе за компьютером. Недостаточное освещение может привести к ухудшению зрения, быстрой утомляемости глаз и головной боли. Следующие требования должны быть соблюдены:

- освещение должно быть равномерным и достаточным. Необходимо обеспечить равномерное освещение всего рабочего места, включая клавиатуру и монитор;
- использовать источники света с нейтральным цветом. Источники света с желтым или синим оттенком могут привести к ухудшению зрения и быстрой утомляемости глаз;
- избегать бликов на экране монитора. Освещение не должно создавать бликов на экране монитора, так как это может привести к ухудшению зрения и быстрой утомляемости глаз;
- освещенность на поверхности рабочего стола должна быть в пределах от 300 до 500 лк. Светильники местного освещения не должны быть слишком яркими и не должны создавать бликов на поверхности экрана, так как ее освещенность не должна превышать 300 лк. Кроме этого, настольные лампы должны иметь непросвечивающий отражатель с защитным углом не менее 40 градусов;
- регулировать освещение в зависимости от условий окружающей среды. Освещение должно быть регулируемым, чтобы можно было изменять яркость света в зависимости от условий окружающей среды.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		53

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

2.4 Требования по организации рабочего места

Санитарными правилами регламентирована и площадь на одно рабочее место:

- при использовании компьютеров с мониторами на базе электронно-лучевой трубки площадь должна быть не менее 6 кв. м;
- при использовании компьютеров с жидкокристаллическими или плазменными мониторами - не менее 4,5 кв. м.

Рабочие столы желательно размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам. Расставлять столы желательно так, чтобы естественный свет падал преимущественно слева.

При размещении рабочих мест с компьютерами расстояние между рабочими столами должно быть не менее 2 метров, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 метра. Если с помощью компьютеров выполняется творческая работа, требующая значительного умственного напряжения или высокой концентрации внимания, рабочие места рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2 метра.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Оптимальными размерами поверхности рабочего стола для ПЭВМ являются: ширина - от 800 до 1400 мм, глубина — 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

Монитор на столе желательно располагать на расстоянии 60 - 70 см от глаз пользователя, но не ближе 50 см с учетом размеров алфавитно-цифровых знаков и символов, клавиатуру — на расстоянии 100 - 300 мм от края стола, обращенного к пользователю, или на специальной регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

Стул должен обеспечивать поддержание рациональной рабочей позы, позволять изменять ее с целью снижения напряжения мышц спины и шейно-плечевой области. Более приемлемым является подъемно-поворотное кресло, которое регулируется по высоте и углам наклона спинки. Регулировка каждого параметра при этом должна быть независимой, легко осуществимой и иметь надежную фиксацию.

Рабочее место пользователя ПЭВМ предлагается оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов. Желательно, чтобы при этом поверхность подставки была рифленой и имела по переднему краю бортик высотой 10 мм

2.5 Требования к организации работы

Организация работы осуществляется в зависимости от вида и категории трудовой деятельности. Виды трудовой деятельности подразделяются на три группы:

- работа по считыванию информации с экрана монитора с предварительным запросом отнесена к группе А;
- работа по вводу информации – к группе Б;
- творческая работа в режиме диалога – к группе В.

При выполнении в течение рабочей смены работ, относящихся к разным видам трудовой деятельности, за основную работу следует принимать такую, которая занимает не менее 50% времени в течение рабочей смены или рабочего дня.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		55

Для видов трудовой деятельности устанавливается три категории тяжести и напряженности работы. Максимальные значения соответствуют III категории работы, их значения определяются:

- для группы А по суммарному числу считываемых знаков за рабочую смену, но не более 60 000 знаков за смену;
- для группы Б по суммарному числу считываемых или вводимых знаков за рабочую смену, но не более 40 000 знаков за смену;
- для группы В по суммарному времени непосредственной работы с ПК за рабочую смену, но не более 6 ч за смену.

Для предупреждения преждевременной утомляемости пользователей ПК рекомендуется организовывать рабочую смену путем чередования работ с использованием ПК и без него.

В случаях, когда характер работы требует постоянного взаимодействия с компьютером (набор текстов или ввод данных и т.п.) с напряжением внимания и сосредоточенности, при исключении возможности периодического переключения на другие виды трудовой деятельности, не связанные с ПЭВМ, предлагается осуществлять перерывы на 10-15 мин. через каждые 45-60 мин. работы. Продолжительность непрерывной работы без регламентированного перерыва не должна превышать 1 часа.

При работе с ПК в ночную смену (с 22:00 до 6:00) независимо от категории и вида трудовой деятельности продолжительность регламентированных перерывов желательно увеличивать на 30%.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

Заключение

Разработка игрового приложения является проектом, который требует знаний в области разработки мобильных приложений и геймдизайна. В процессе разработки необходимо использовать различные инструменты и методы для создания приложения. В ходе выполнения практики были освоены основные понятия и технологии разработки игр, изучены принципы работы сцен, компонентов, физики и скриптов, изучены основные инструменты разработки игр, изучены основные принципы разработки игр и основные инструменты.

					ПГУ1.090301.5.001 ПЗ	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

Список использованных источников

1. Джереми Гибсон Бонд. Unity и C#. Геймдев от идеи до реализации. СПб.: Питер, 2019.
2. Статистические данные о мобильных играх. [Электронный ресурс] / <https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-gaming-statistics>
3. Джозеф Хокинг. Unity и C#. Unity в действии. Мультиплатформенная разработка на C#. Второе международное издание. СПб.: Питер, 2019. – 352 с.
4. Дмитрий Денисов. Разработка игры на Unity с нуля до реализации, 2022.

					ПГУ1.090301.5.001 ПЗ	Лист
						58
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А

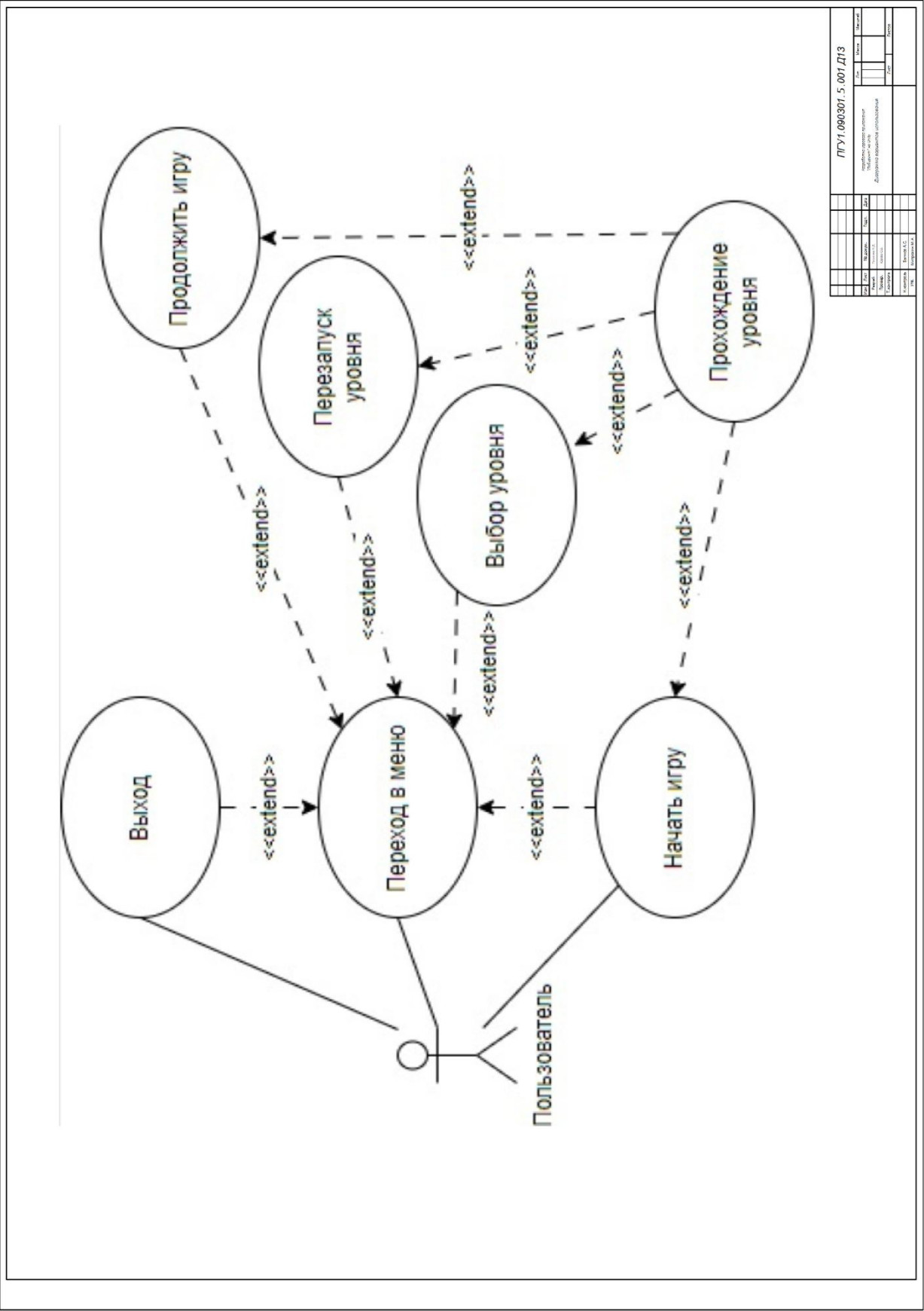
Разработка игрового приложения “Лабиринт” на *Unity*

(обязательное)

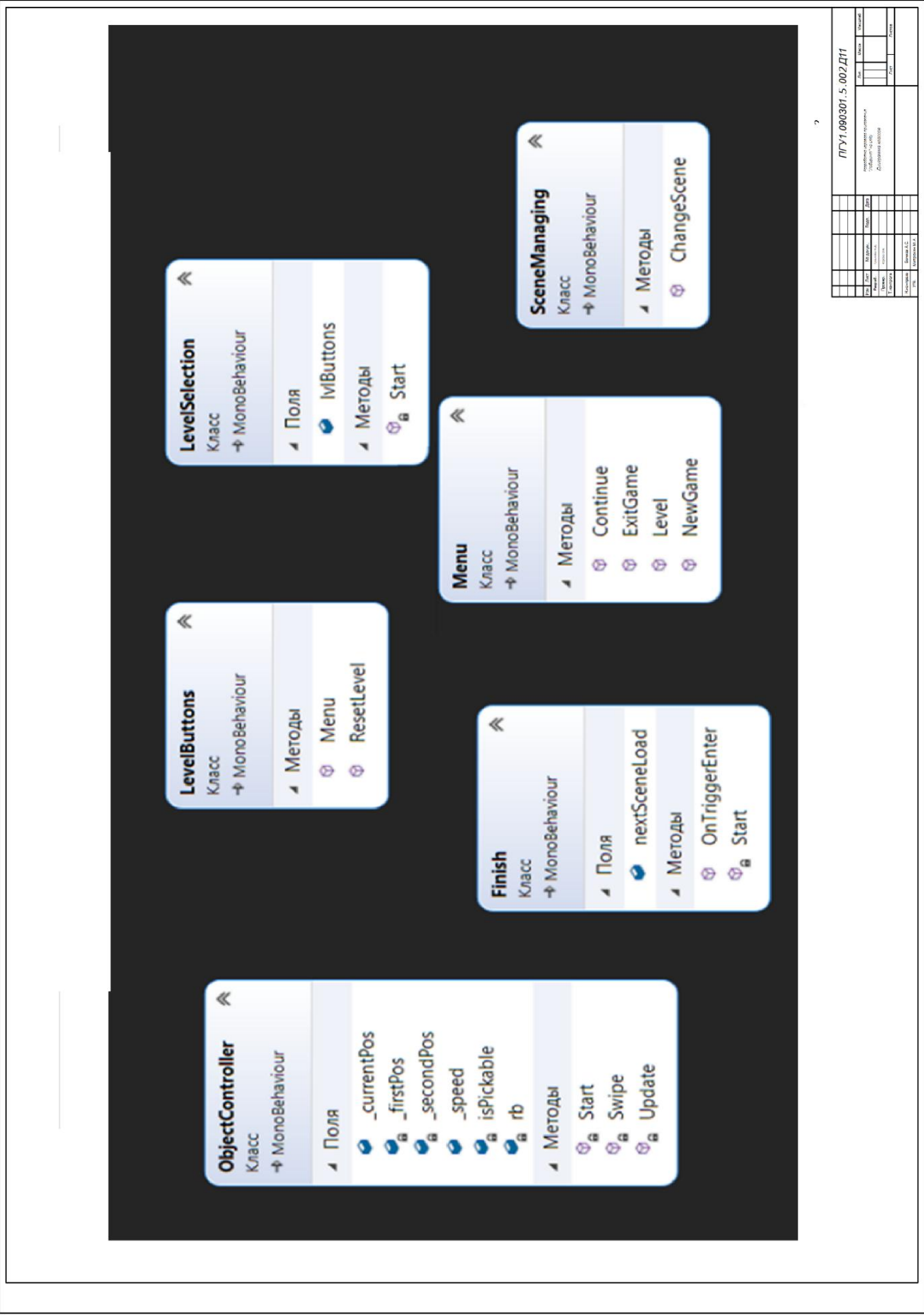
UML диаграммы

					ПГУ1.090301.5.001 ПЗ	Лист
						59
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А. 1 – UML – диаграмма прецедентов



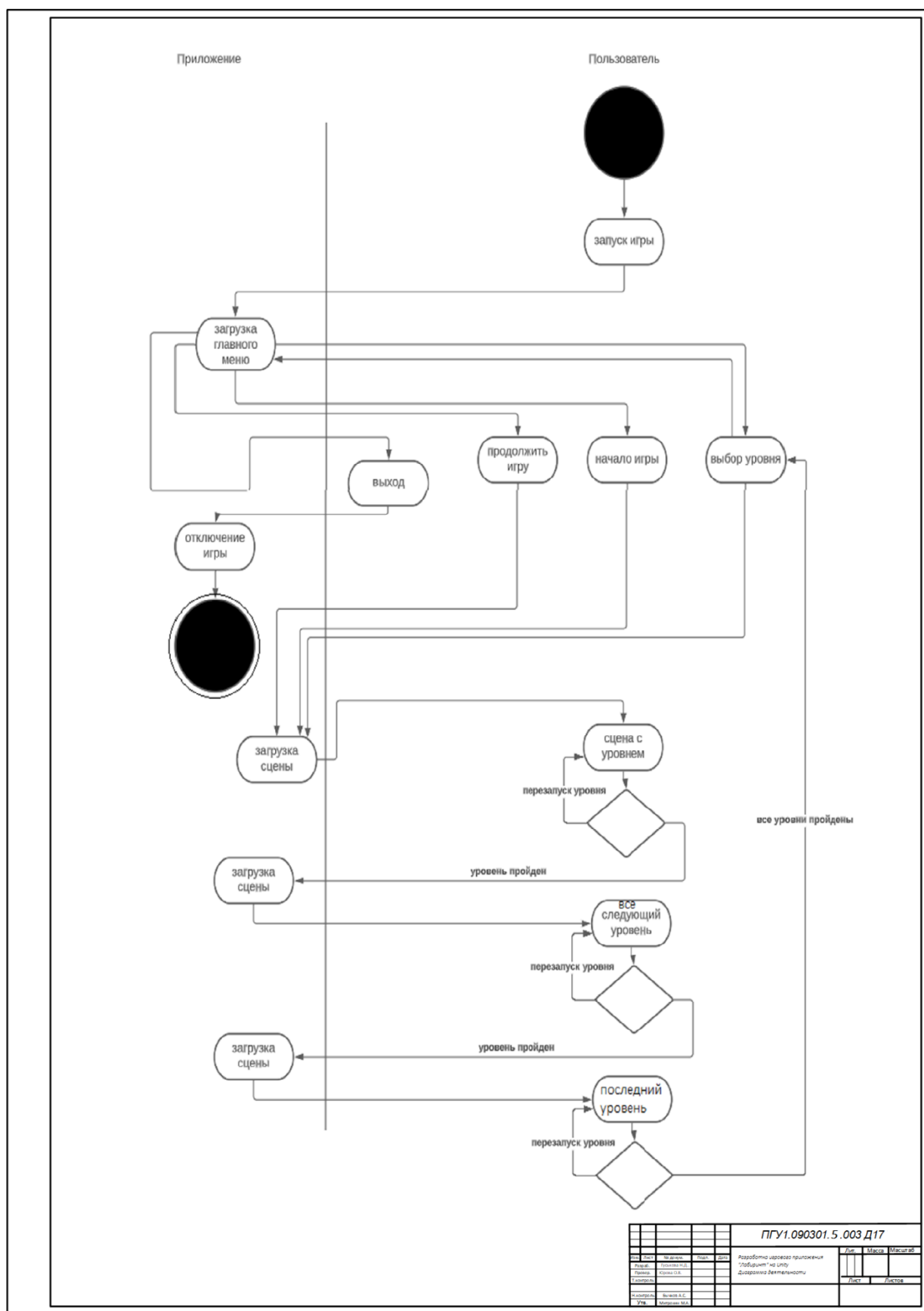
Приложение А.2 – UML – диаграмма классов



ПГУ1.090301.5.002Д11									
№	Стр.	Изм.	Дата	Исполн.	Провер.	Утверд.	Подпись	Дата	Подпись
1	1								
2	2								
3	3								
4	4								
5	5								
6	6								
7	7								
8	8								
9	9								
10	10								
11	11								
12	12								
13	13								
14	14								
15	15								
16	16								
17	17								
18	18								
19	19								
20	20								
21	21								
22	22								
23	23								
24	24								
25	25								
26	26								
27	27								
28	28								
29	29								
30	30								
31	31								
32	32								
33	33								
34	34								
35	35								
36	36								
37	37								
38	38								
39	39								
40	40								
41	41								
42	42								
43	43								
44	44								
45	45								
46	46								
47	47								
48	48								
49	49								
50	50								
51	51								
52	52								
53	53								
54	54								
55	55								
56	56								
57	57								
58	58								
59	59								
60	60								
61	61								
62	62								
63	63								
64	64								
65	65								
66	66								
67	67								
68	68								
69	69								
70	70								
71	71								
72	72								
73	73								
74	74								
75	75								
76	76								
77	77								
78	78								
79	79								
80	80								
81	81								
82	82								
83	83								
84	84								
85	85								
86	86								
87	87								
88	88								
89	89								
90	90								
91	91								
92	92								
93	93								
94	94								
95	95								
96	96								
97	97								
98	98								
99	99								
100	100								

Изм.	Лист	№ докум.	Подпись	Дата

Приложение А.3 –UML – диаграмма деятельности



Приложение Б

Разработка игрового приложения “Лабиринт” на *Unity*

(обязательное)

Листинг программы

					ПГУ1.090301.5.001 ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

Файл «ObjectController.cs»

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

public class ObjectController : MonoBehaviour
{
    private Rigidbody rb;
    private bool isPickable;
    private Vector2 _firstPos;
    private Vector2 _secondPos;
    public Vector2 _currentPos;
    public float _speed;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        rb.mass = 5;
    }

    void Update()
    {
        Swipe();
    }

    private void Swipe()
    {
        if (Input.GetMouseButtonDown(0))
        {
            RaycastHit hit;
            Ray ray =
Camera.main.ScreenPointToRay(Input.mousePosition);

            if (Physics.Raycast(ray, out hit))
            {
                if (hit.transform.name == gameObject.name)
                {
                    isPickable = true;
                    _firstPos = new Vector2(Input.mousePosition.x,
Input.mousePosition.y);
                }
            }

            if (Input.GetMouseButtonUp(0) && isPickable)
            {
                _secondPos = new Vector2(Input.mousePosition.x,
Input.mousePosition.y);

                _currentPos = new Vector2(
                    _secondPos.x - _firstPos.x,
```

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

```

        _secondPos.y - _firstPos.y
        );

        _currentPos.Normalize();

        if (_currentPos.y < 0 && _currentPos.x > -0.5f &&
        _currentPos.x < 0.5f)
        {
            // Back
            rb.velocity = Vector3.back * _speed;
        }
        else if (_currentPos.y > 0 && _currentPos.x > -0.5f &&
        _currentPos.x < 0.5f)
        {

            rb.velocity = Vector3.forward * _speed;
        }
        else if (_currentPos.x < 0 && _currentPos.y > -0.5f &&
        _currentPos.y < 0.5f)
        {

            rb.velocity = Vector3.left * _speed;
        }
        else if (_currentPos.x > 0 && _currentPos.y > -0.5f &&
        _currentPos.y < 0.5f)
        {

            rb.velocity = Vector3.right * _speed;
        }

        isPickable = false;
    }
}
}

```

Файл «*Finish.cs*»

```

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

public class Finish : MonoBehaviour
{
    public int nextSceneLoad;
    [SerializeField] GameObject WinPanel;
    void Start()
    {
        nextSceneLoad = SceneManager.GetActiveScene().buildIndex +
1;
        WinPanel.SetActive(false);
    }
}

```

					ПГУ1.090301.5.001 ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		

```

}

public void OnTriggerEnter(Collider other)
{
    if (other.gameObject.tag == "Player")
    {
        WinPanel.SetActive(true);
        if (SceneManager.GetActiveScene().buildIndex == 8)
        {
            WinPanel.SetActive(true);
        }
        else
        {
            if (nextSceneLoad > PlayerPrefs.GetInt("levelAt"))
            {
                PlayerPrefs.SetInt("levelAt", nextSceneLoad);
            }
        }
    }
}

public void NextLevel()
{
    SceneManager.LoadScene(nextSceneLoad);
}
}

```

Файл «*SceneManaging.cs*»

```

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

public class SceneManaging : MonoBehaviour
{
    public void ChangeScene(int sceneIndex)
    {
        SceneManager.LoadScene(sceneIndex);
    }
}

```

Файл «Menu.cs »

```

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

public class Menu : MonoBehaviour
{
    public void Level()
    {
        SceneManager.LoadScene("Levels");
    }
}

```

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		66

```

    }

    public void ExitGame()
    {
        Application.Quit();
    }

    public void Continue()
    {
        int nextSceneLoad = PlayerPrefs.GetInt("levelAt");
        SceneManager.LoadScene(nextSceneLoad);
    }

    public void NewGame()
    {
        PlayerPrefs.DeleteAll();
        SceneManager.LoadScene("Level1");
    }
}

```

Файл «*LevelSelection.cs*»

```

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

public class LevelSelection : MonoBehaviour
{
    public Button[] lvlButtons;
    void Start()
    {
        int levelAt = PlayerPrefs.GetInt("levelAt", 2);

        for (int i = 0; i < lvlButtons.Length; i++)
        {
            if (i + 2 > levelAt)
                lvlButtons[i].interactable = false;
        }
    }
}

```

Файл «*LevelButtons.cs*»

```

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

public class LevelButtons : MonoBehaviour
{

```

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67

```

public void ResetLevel()
{
SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
}
public void Menu()
{
SceneManager.LoadScene("Menu");
}
}

```

					ПГУ1.090301.5.001 ПЗ	Лист
						68
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение В

Разработка игрового приложения “Лабиринт” на *Unity*

(обязательное)

Презентация

					ПГУ1.090301.5.001 ПЗ	Лист
						69
Изм.	Лист	№ докум.	Подпись	Дата		

Тема работы:

«Разработка игрового приложения «Лабиринт Unity»

Выполнил: студент группы 19ВВВП1 Гуськова Н.И.
Руководитель: к.т.н., доцент каф. ВТ ПГУ И

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		70

Актуальность работы

мобильных устройствах — это одна из самых популярных кений. гь разработки игрового приложения на платформе рностью мобильных устройств и развитием мобильных смартфонов и планшетах уже не являются просто развл особом заработка

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		71

Цель и объект работы

цель – реализовать игровое приложение “Лабиринт” на Unity
изработки является игровое приложение , представляющие
перемещением объектов

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		72

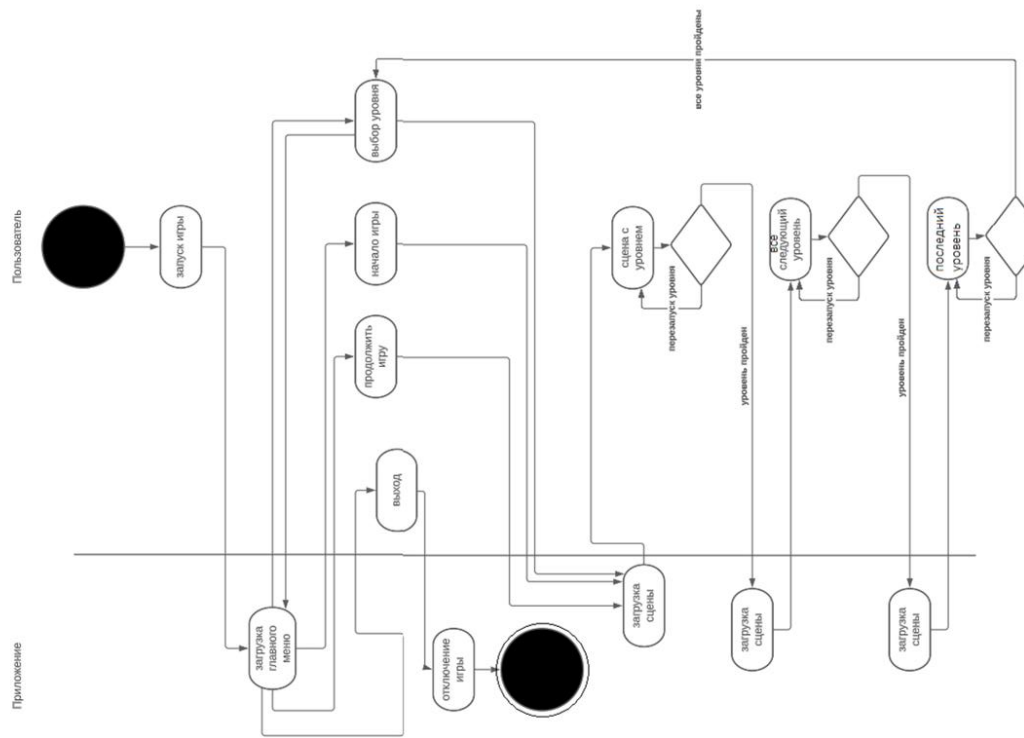
а задачи

кциями игры должны быть:
юсти лабиринта: расстановка блоков различна на каждом ур
юсти прохождения игроком уровня.
ние блоков: игрок должен иметь возможность перетаскивать (к
к выходу.
ализовать финиш игры, когда игрок пошёл уровень и может (и
нь заново : игрок в любой момент должен иметь возможность
/ровень
игрок должен иметь возможность начать игру, продолжить, в
и из игры
ню : игрок должен иметь возможность перейти в меню из игр
онятный графический интерфейс
сохранения прогресса: игрок должен иметь возможность сох
енных уровней в игре и продолжать играть с того же

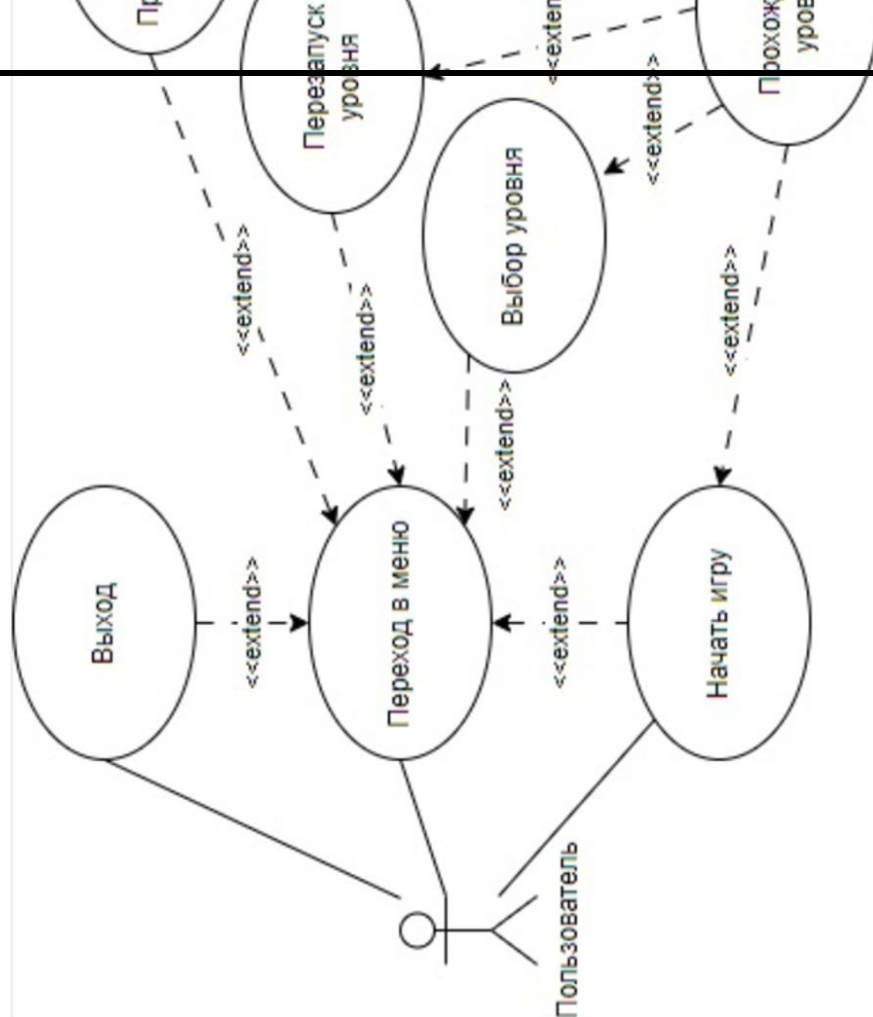
Аналоги



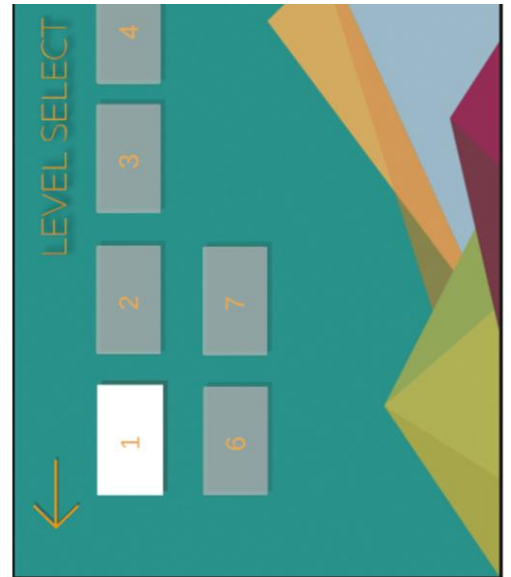
UML-диаграмма деятельности



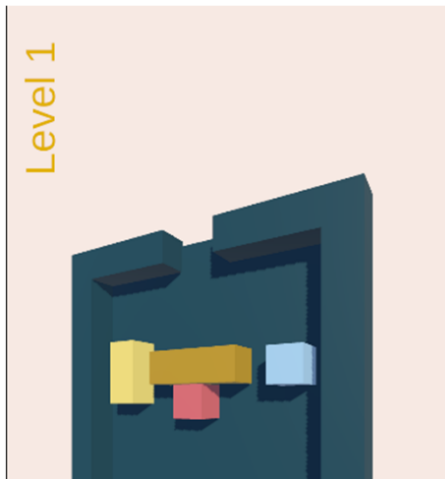
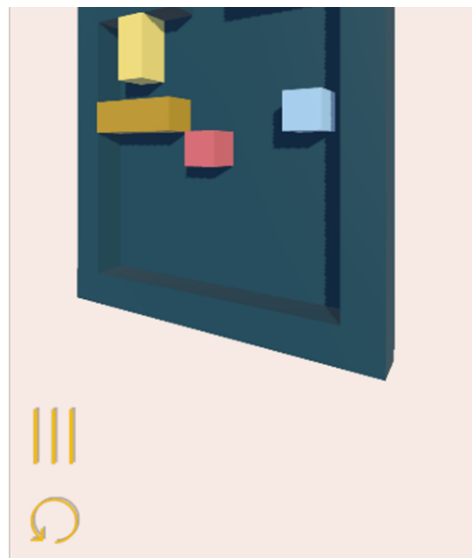
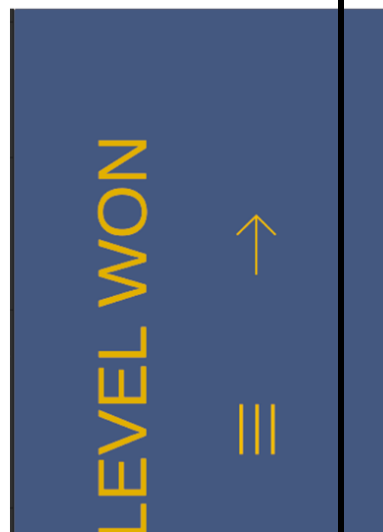
UML – диаграмма прецедентов



Тестирование



Тестирование



Заключение

Игрового приложения является проектом, который требует разработки мобильных приложений и геймдизайна. В процессе необходимо использовать различные инструменты и методологии. В ходе выполнения практики были освоены технологии разработки игр, изучены принципы работы с физикой и скриптами, изучены основные инструменты и методы основные принципы разработки игр и основные игровые механики.

					ПГУ1.090301.5.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		79

Спасибо за внимание!

					ПГУ1.090301.5.001 ПЗ	Лист
						80
Изм.	Лист	№ докум.	Подпись	Дата		

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

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

«Утверждаю»

Заведующий кафедрой

_____М.А. Митрохин

« 26 » _____ апреля _____ 20 23 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Гуськова Н.Д. гр 19ВВП1 факультета вычислительной техники.

2. Тема работы Образовательное приложение «Игровое приложение «Лабиринт» на Unity»

Тема утверждена приказом ПГУ № 458/о от "26" апреля 2023 г.

3. Руководитель бакалаврской работы к.т.н., доцент, каф. ВТ ПГУ Юрова О.В.

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

Реализовать игровое приложение “Лабиринт” на Unity, помогающее в развитии логики.
Реализовать лабиринт из нескольких уровней ,возможность сохранения и графический
интерфейс . Провести тестирование и отладку каждого элемента приложения.

5. Перечень подлежащих разработке вопросов

Требуется произвести анализ предметных областей: изучить существующие игры, выделить их достоинства и недостатки. Выбрать язык программирования, который подойдет для написания логики приложения. Определить архитектуру приложения и вспомогательные средства разработки.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Изучение предметной области	50	26.04.2023-30.04.2023	
2.	Анализ требований	10	01.05.2023-02.05.2023	
3.	Выбор технологий, применяемых при проектировании и разработке	50	03.05.2023-07.05.2023	
4.	Описание алгоритмов	40	08.05.2023-11.05.2023	
5.	Реализация игрового приложения	150	12.05.2023-29.05.2023	
6.	Оформление пояснительной записки	24	30.05.2023-31.05.2023	

Дата выдачи задания "26" апреля 2023 г.

Руководитель бакалаврской работы _____ Юрова О.В.
(подпись, дата)

Задание к исполнению принял студент _____ Гуськова Н.Д.
(подпись, дата)

Работу к защите допустить
Декан факультета _____ Фионова Л.Р.
(подпись, дата)