

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
ООП Разработка программно-информационных систем
Отделение школы (НОЦ) Отделение информационных технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема работы
Проектирование и разработка кроссплатформенного игрового приложения на Unity для персональных компьютеров в жанре 2D платформер

УДК 004.42:004.382.7

Обучающийся

Группа	ФИО	Подпись	Дата
8К92	Лутошкин Степан Евгеньевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Е.С.	К.Т.Н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОИТ ИШИТР	Коровкин В.А.			

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов М.А.	Д.Э.Н.		

По разделу «Социальная ответственность»

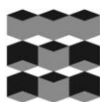
Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД	Мезенцева И.Л.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП/ОПОП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Е.С.	К.Т.Н.		

Результаты обучения

Код компетенции	Наименование компетенции
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
ОПК(У)-1	Способен применять естественнонаучные и общетеchnические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности
ОПК(У)-2	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК(У)-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью
ОПК(У)-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем
ОПК(У)-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов
ОПК(У)-7	Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой
ОПК(У)-8	Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий.
ПК(У)-1	Владение навыками разработки требований и проектирования программного обеспечения
ПК(У)-2	Владение навыками разработки документов и стратегии тестирования программного обеспечения
ПК(У)-3	Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения
ПК(У)-4	Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных
ПК(У)-5	Владение концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
ООП Разработка программно-информационных систем
Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП
_____ Чердынцев Е.С.
(подпись) (дата) (Ф.И.О.)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

Обучающийся:

Группа	ФИО
8K92	Лутошкин Степан Евгеньевич

Тема работы:

Проектирование и разработка кроссплатформенного игрового приложения на Unity для персональных компьютеров в жанре 2D платформер	
Утверждена приказом заведующего кафедрой – руководителем отделения на правах кафедры ОИТ ИШИТР	Приказ № 102-29/с от 12.04.2023 г.

Срок сдачи обучающимся выполненной работы:	10.06.2023
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом проектирования в работе является кроссплатформенное игровое приложение на Unity для персональных компьютеров в жанре 2D платформер
Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Аналитический обзор рынка 2D платформеров для персональных компьютеров и средств их разработки; 2. Проектирование и реализация персонажа пользователя и его компонентов; 3. Проектирование и разработка пяти игровых уровней; 4. Проектирование и реализация элементов пользовательского интерфейса игрового приложения; 5. Разработка системы обработки пользовательского ввода;

	6. Разработка системы сохранения и загрузки игрового процесса; 7. Разработка игровых систем управления миссиями, квестами и событиями; 8. Проектирование и реализация миниигры на примере головоломки с лучами света; 9. Разработка неигрового персонажа «Паук»
Перечень графического материала	1. Презентация в формате *.pptx; 2. Диаграммы классов, состояний, вариантов использования; 3. Рисунки, демонстрирующие результаты работы; 4. Диаграмма Ганта.
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов Магеррам Али оглы
Социальная ответственность	Мезенцева Ирина Леонидовна

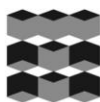
Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	04.03.2023
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Е.С.	к.т.н.		04.03.2023

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
8K92	Лутошкин Степан Евгеньевич		04.03.2023



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
ООП Разработка программно-информационных систем
Отделение школы (НОЦ) Отделение информационных технологий
Период выполнения весенний семестр 2022/2023 учебного года

Форма представления работы:

бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Обучающийся:

Группа	ФИО
8K92	Лутошкин Степан Евгеньевич

Тема работы:

Проектирование и разработка кроссплатформенного игрового приложения на Unity для персональных компьютеров в жанре 2D платформер

Срок сдачи обучающимся выполненной работы:	10.06.2023
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
16.03.2023	Анализ предметной области	25
17.04.2023	Проектирование программной системы	25
20.05.2023	Разработка программной системы	20
22.05.2023	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
22.05.2023	Социальная ответственность	15

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Е.С.	к.т.н.		03.03.2023

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Е.С.	к.т.н.		04.03.2023

Обучающийся

Группа	ФИО	Подпись	Дата
8K92	Лутошкин Степан Евгеньевич		04.03.2023

РЕФЕРАТ

Выпускная квалификационная работа содержит 88 страниц, 39 рисунков, 23 таблицы, 39 источников литературы.

Ключевые слова: видеоигра, проектирование игровых систем, разработка игр, геймдизайн, 2D платформер, Unity.

Объектом исследования является разработка кроссплатформенных игровых приложений на Unity для персональных компьютеров под управлением операционных систем Windows, GNU/Linux, OS X.

Предметом исследования являются архитектурные подходы и методы проектирования игровых приложений на Unity для персональных компьютеров.

Цель работы – проектирование и реализация кроссплатформенного игрового приложения на Unity в жанре 2D-платформер для персональных компьютеров под управлением операционных систем Windows, GNU/Linux и OS X.

В процессе выполнения работы использовались среда разработки JetBrains Rider, язык программирования C#, система контроля версий Plastic SCM, виртуальная доска для совместной работы Miro, пространство для командной работы Confluence.

Степень внедрения: частичная, текущее состояние TRL5 (англ. technology readiness level) по методике определения уровня готовности технологии.

Область применения – цифровые развлечения, индустрия видеоигр.

В результате работы были спроектированы и разработаны игровые механики для 5 игровых уровней, персонаж игрока и NPC, а также игровые системы для обработки ввода пользователя, сохранения/загрузки его прогресса и для управления миссиями, квестами и событиями. Также была реализована минигра на примере головоломки с лучами света.

В первой главе представлен аналитический обзор современных тенденций и технологий в разработке игровых приложений. Также в ней

описаны основные инструменты и платформы для разработки и сравнительный анализ аналогов.

Вторая глава описывает процесс проектирования и программной реализации компьютерной игры в жанре 2D-платформер.

Третья глава представляет собой выполненное задание по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение», в котором отображены потенциал, планирование и эффективность решения.

Четвертая глава представляет собой выполненное задание по разделу «Социальная ответственность», в котором рассмотрены организационно правовые, производственные и экологические аспекты безопасности, а также безопасность в чрезвычайных ситуациях.

СОДЕРЖАНИЕ

Введение.....	11
Список терминов, сокращений и условных обозначений	13
1 Аналитический обзор современных тенденций и технологий в разработке игровых приложений.....	14
1.1 Индустрия компьютерных игр	14
1.2 Виды компьютерных игр и их особенности	14
1.3 Анализ проектов в жанре «2D-платформер»	17
1.4 Основные инструменты и платформы для разработки игровых приложений.....	20
1.5 Вывод по главе	21
2 Проектирование и реализация игрового приложения.....	22
2.1 Общее видение системы.....	22
2.2 Геймдизайнерская документация на примере концепт-документа	23
2.3 Варианты использования	24
2.4 Игровой персонаж.....	27
2.4.1 Передвижение и взаимодействие с интерактивными объектами	29
2.4.2 Смена форм игрового персонажа	29
2.4.3 Прочие компоненты игрового персонажа	31
2.5 Основные внутриигровые системы.....	32
2.5.1 Система обработки ввода пользователя	32
2.5.2 Система сохранений	34
2.5.3 Система событий.....	37
2.5.4 Система миссий.....	37
2.5.5 Система квестов	38
2.6 Пользовательский интерфейс	40
2.7 Игровые уровни.....	43
2.8 Интерактивные объекты.....	46

2.8.1 Триггер	46
2.8.2 Головоломки с лучами света	47
2.9 NPC паук	49
Задание для раздела «финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	54
3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	55
Введение.....	55
3.1 Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения	55
3.1.1 Потенциальные потребители	55
3.1.2 Анализ конкурентных технических решений.....	56
3.1.3 Технология QuaD	57
3.1.4 SWOT-анализ.....	58
3.2 Планирование работ по научно-техническому исследованию	59
3.2.1 Структура работ в рамках научного исследования.....	59
3.2.2 Определение трудоемкости выполнения работ	60
3.2.3 Разработка графика проведения научного исследования.....	60
3.3 Бюджет научно-технического исследования	63
3.3.1 Расчет материальных затрат	63
3.3.2 Расчет амортизации специального оборудования	64
3.3.3 Основная заработная плата исполнителей темы	64
3.3.4 Дополнительная заработная плата	66
3.3.5 Отчисления во внебюджетные фонды.....	66
3.3.6 Накладные расходы	67
3.3.7 Формирование бюджета затрат научно–исследовательского проекта.....	67
3.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования .	68

3.5 Вывод по главе	70
Задание для раздела «социальная ответственность».....	71
4. Социальная ответственность	73
Введение.....	73
4.1 Правовые и организационные вопросы обеспечения безопасности	73
4.1.1 Специальные правовые нормы трудового законодательства	73
4.1.2 Основные эргономические требования к правильному расположению и компоновке рабочей зоны	74
4.2 Производственная безопасность	75
4.2.1 Монотонность труда.....	75
4.2.2 Повышенный уровень шума	76
4.2.3 Отсутствие или недостаток необходимого искусственного освещения.....	77
4.2.4 Нагрузка на зрительный аппарат.....	78
4.2.5 Отклонение показателей микроклимата.....	78
4.2.6 Длительность сосредоточенного наблюдения	80
4.2.7 Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий	80
4.3 Экологическая безопасность	81
4.4 Безопасность в чрезвычайных ситуациях	82
4.5 Вывод по главе	83
Заключение	84
Список источников	85

ВВЕДЕНИЕ

Разработка компьютерных игр в современном мире является популярным направлением разработки программного обеспечения. За 2022 год в одном из популярных магазинов игр Steam вышло более 12 000 новых игр. Согласно статистике SteamDB число релизов из года в год растет на протяжении 10 лет [1]. График, отражающий статистику за этот период изображен на рисунке 1.

Steam Game Releases by Year

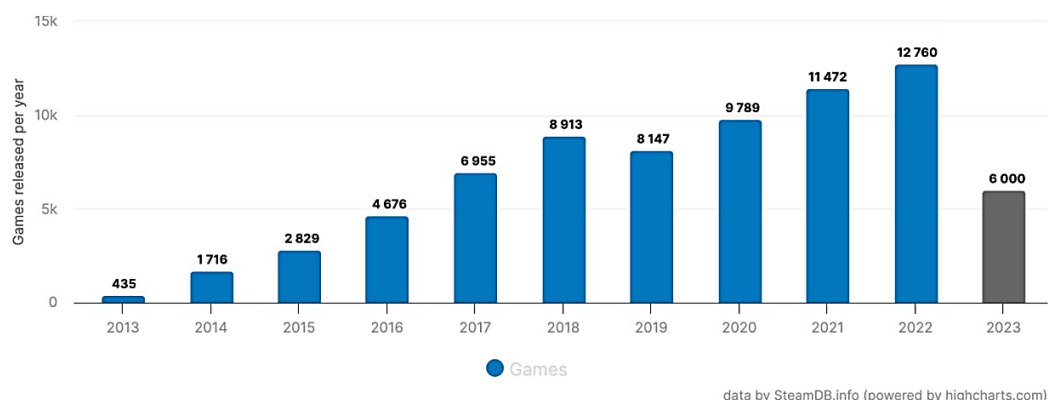


Рисунок 1 – Количество игровых релизов в год в Steam

На текущий момент существует множество игр различных жанров и масштабов. Так, например, по Кутлалиеву существует 7 основных жанровых групп и более 30 поджанров [2].

Конечная цель разработки компьютерных игр – создание уникального игрового опыта, который позволит игрокам окунуться в увлекательный виртуальный мир и получить максимальное удовольствие от игры. Современные технологии позволяют создавать игры, которые выходят за рамки обычных игровых жанров и устанавливают новые стандарты в индустрии компьютерных игр.

В рамках данной работы будут рассмотрены современные тенденции и технологии в разработке игровых приложений, а также основные инструменты и платформы для их создания. Особое внимание будет уделено популярной платформе Unity, которая является одним из основных инструментов для

разработки кроссплатформенных игр и позволяет создавать игры различных жанров, в том числе в жанре 2D-платформер.

В работе будут рассмотрены основные этапы разработки кроссплатформенного игрового приложения: от проектирования до реализации. Будут рассмотрены ключевые функции и особенности, которые необходимо учитывать при разработке игры в жанре 2D-платформер, а также возможности, которые предоставляет Unity для создания кроссплатформенных приложений.

Таким образом, целью данной работы является проектирование и реализация кроссплатформенного игрового приложения в жанре 2D-платформер.

Для достижения поставленной цели были сформированы следующие основные задачи:

- провести анализ вышедших игр для персональных компьютеров под управлением Windows, Linux и OS X в жанре 2D платформер, отметить их сильные стороны и отличительные особенности;
- выдвинуть функциональные требования и спроектировать основные игровые механики и пользовательский интерфейс;
- разработать концепт-документ игры;
- провести анализ программных средств, используемых при разработке игровых приложений;
- реализовать игровое приложение.

Результатом данной работы будет полноценное игровое приложение, которое сможет работать на персональных компьютерах под управлением Windows, Linux и OS X и соответствовать требованиям жанра 2D-платформер. Приложение будет включать 5 спроектированных и реализованных по линейному принципу игровых уровней, персонажа игрока, пользовательский интерфейс, а также вспомогательные игровые системы.

СПИСОК ТЕРМИНОВ, СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

Геймплей (англ. *gameplay*) – это основной игровой процесс, механика игры, которая включает в себя правила, задачи, управление, сценарий, взаимодействие персонажей и игрока.

Кроссплатформенность (англ. *cross-platform compatibility*) – это способность программного обеспечения работать на различных операционных системах и платформах.

Игровой ассет (англ. *game asset*) – это элемент, который используется в игровом процессе, например, текстура, модель персонажа, аудио или видео файл.

Игровой движок (англ. *game engine*) – это программное обеспечение, которое обеспечивает основу для разработки игр, включая графический движок, физический движок, звуковой движок, сценарные языки и т. д.

Игровая сцена (англ. *game scene*) – это среда, в которой происходит игровой процесс, включая персонажей, объекты, звуки, текстуры и другие элементы, которые могут взаимодействовать между собой и с игроком.

Префаб (англ. *prefab*) – это готовый компонент, который может быть использован в игровой сцене, например, готовая модель персонажа, которую можно повторно использовать в разных сценах игры.

Коллизия (англ. *collision*) – столкновение объектов в игре.

Дизайн уровней (англ. *level design*) – создание игровых уровней с учетом игровых механик и целей игры.

1 Аналитический обзор современных тенденций и технологий в разработке игровых приложений

1.1 Индустрия компьютерных игр

Индустрия компьютерных игр – это одно из наиболее быстроразвивающихся направлений современной развлекательной индустрии. На 2022 год объем рынка компьютерных игр превысил 38 млрд. долларов (что составляет 21% от всего рынка видеоигр), а рост, в сравнении с предыдущим годом, несмотря на послабления ковидных ограничений, составил около 1,8% [3-5]. С развитием технологий игровая индустрия стала еще более динамичной и инновационной, игры стали более реалистичными, интерактивными и доступными. На сегодняшний день компьютерные игры пользуются огромной популярностью у миллионов людей по всему миру.

Разработка игровых приложений является сложным процессом, который включает в себя множество аспектов, таких как дизайн, программирование, аудио и видео эффекты, тестирование и маркетинг. Раннее в основном крупные компании занимались созданием и публикацией игр, в то время как сейчас помимо них небольшие компании и даже одиночные разработчики имеют множество возможностей и инструментов для реализации своих идей.

1.2 Виды компьютерных игр и их особенности

Игровая индустрия находится в постоянном развитии, и это неизменно влияет на жанры и виды игр. В настоящее время существует множество жанров и видов компьютерных игр, каждый из которых имеет свои особенности и пользуется своей популярностью среди игроков.

В настоящее время можно выделить несколько критериев классификаций игр [2, 6]. Первый из них – платформа, на которой они выпущены: некоторые игры могут быть доступны лишь на одной платформе, в то время как другие могут являться кроссплатформенными. Второй – классификация игр по количеству игроков: игры могут быть разработаны как для одного игрока, так и для многопользовательской игры. Самым крупным

делением является классификация по жанру, который определяет конечную цель игры.

Один из самых популярных жанров – это шутеры от первого лица. В этом жанре игрок управляет персонажем, которому нужно сражаться с врагами, используя различное оружие и тактику. Шутеры от первого лица характеризуются быстрым темпом игры, множеством действий и возможностей, а также реалистичной графикой. Одним из представителей жанра является игра Call of Duty: Modern Warfare 2 [7]. Скриншот из игры представлен на рисунке 2.



Рисунок 2 – Скриншот из игры «Call of Duty: Modern Warfare 2»

Другим жанром являются стратегии, где игрок управляет армией, государством или группой персонажей, планирует различные действия и совершает их – например, нападение на чужую территорию или распределение ресурсов собственного государства. В этом жанре важна не только быстрота реакции, но и умение планировать и принимать взвешенные решения. К игре этого жанра относится Sid Meier's Civilization V [8], изображенная на рисунке 3.



Рисунок 3 – Скриншот из игры «Sid Meier's Civilization V»

Еще одним жанром являются платформеры. Платформеры представляют собой игры, где игрок управляет персонажем, который перемещается по 2D-пространству (или по 3D-пространству), преодолевая различные препятствия, сражаясь с врагами и собирая бонусы. Основные особенности игр этого жанра включают в себя разнообразие уровней, использование способностей персонажа, наличие боссов и прочих элементов, которые делают игру более интересной и захватывающей. Один из ярких представителей жанра – Hollow Knight [9]. На рисунке 4 представлен скриншот из игры.

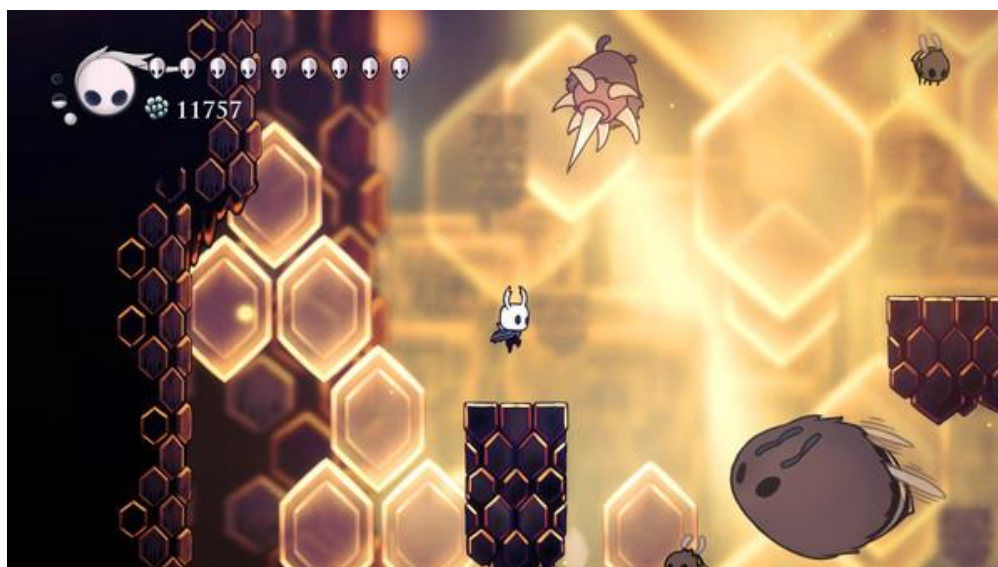


Рисунок 4 – Скриншот из игры «Hollow Knight»

Поскольку каждый жанр имеет свои особенности и привлекает свою аудиторию, разработчикам игр важно учитывать эти особенности при разработке.

1.3 Анализ проектов в жанре «2D-платформер»

Поскольку жанр 2D-платформер является довольно популярным жанром в игровой индустрии (согласно статистике SteamDB за последние 2 года доля вышедших в Steam игр в жанре 2D платформер составляет более 11% от общего числа релизов [1]), особенно среди инди-разработчиков, то, чтобы разработать продукт, отличающийся от большинства других игр в схожем жанре и привлечь внимание потенциальных пользователей, необходимо провести сравнительный анализ популярных представителей выбранного жанра и выделить их сильные и слабые стороны, а также изучить отличительные особенности. Результаты представлены в таблице 1.

Hollow Knight – платформер про королевство жуков, который выделяется своим потрясающим визуальным стилем, хорошо продуманным миром и атмосферой, сложными боссами и интересными механиками [9]. На рисунке 5 представлен скриншот из игры.



Рисунок 5 – Скриншот из игры «Hollow Knight»

Celeste – игра, которая получила много наград и признания за свой сюжет, музыку и геймплей. Она сочетает в себе элементы платформера со

сложным геймплеем и пазла, предлагает игроку разнообразные вызовы и задачи [10]. На рисунке 6 представлен скриншот из игры.



Рисунок 6 – Скриншот из игры «Celeste»

Ori and the Blind Forest – это красочный и захватывающий платформер, в котором игрок управляет маленьким существом по имени Ори, стремящимся спасти свой лес от опустошения. Игра привлекает своей красивой графикой и музыкой, создающими захватывающую атмосферу. Кроме того, в игре есть элементы RPG, например, система улучшений навыков, что добавляет глубины и интереса к прохождению игры [11]. На рисунке 7 представлен скриншот из игры.



Рисунок 7 – Скриншот из игры «Ori and the Blind Forest»

Super Meat Boy – это 2D-платформер, известный своей сложностью и чрезвычайно быстрым геймплеем. Игра имеет уникальный стиль визуального

оформления, которое сочетает в себе кровавый и ужасный характер с юмором и милыми персонажами [12]. На рисунке 8 представлен скриншот из игры.



Рисунок 8 – Скриншот из игры «Super Meat Boy»

Сравнение особенностей игр представлено в таблице 1.

Таблица 1 – сравнение особенностей игр в жанре 2D – платформер

	Hollow Knight	Celeste	Ori and the Blind Forest	Super Meat Boy
Сложность игры	Средне	Сложно	Средне	Сложно
Геймплей	Платформер, боевик, метроидвания	Платформер, головоломки	Платформер, головоломки, метроидвания	Платформер, экшн
Визуальный стиль	темный, готический мир с мистическими существами	пиксельная графика, милая и яркая атмосфера	Красивое сочетание 2D и 3D графики, проработанный мир, множество существ	пиксельная графика, мрачная и забавная атмосфера
Продолжительность, час	20-30	10-15	10-15	8-10
Движок	Unity	Unity	Unity	собственный
Платформы	Windows, Mac OS, GNU/Linux	Windows, Mac OS, GNU/Linux	Windows	Windows, Mac OS, GNU/Linux

В результате проведенного анализа можно сделать следующие выводы:

- большинство игр имеют уровень сложности средне-сложный, что повышает порог входа для новых игроков. Поэтому рекомендуется реализовать подробную систему обучения пользователей, чтобы сделать игру более доступной;

- для жанра платформеры характерен яркий визуальный стиль, поэтому стоит уделить ему большое внимание при разработке игры;
- в качестве движка чаще всего используется Unity;
- в качестве платформ чаще выступают Windows, Mac OS, GNU/Linux.
- рекомендуемое время прохождения игры составляет около 10 часов.

1.4 Основные инструменты и платформы для разработки игровых приложений.

Современная индустрия компьютерных игр предлагает широкий выбор инструментов и платформ для разработки и продвижения игровых приложений. Инструменты для разработки игр предоставляют различные возможности и функции, которые могут быть использованы для создания игр на разных платформах и в различных жанрах. Выбор инструмента зависит от потребностей и уровня опыта разработчика, а также от требований конкретного проекта и его бюджета. Использование современных инструментов и платформ для разработки игр позволяет ускорить и упростить этот процесс, а также создавать игры более высокого качества в рамках доступного бюджета.

Одним из популярных инструментов для разработки игр является Unreal Engine 4. Он предоставляет множество возможностей для создания игр и поддерживает множество платформ. Unreal Engine 4, в отличие от многих других игровых движков, использует продвинутую графику, что позволяет создавать реалистичные и детализированные игры [13].

Другим популярным инструментом является Unity. Помимо поддержки кроссплатформенности и большого количества готовых решений и плагинов, движок также предоставляет множество возможностей для работы с 2D графикой, среди которых есть инструменты для работы с 2D анимациями, спрайтами, светом, шейдерами и многим другим. Также среди особенностей стоит отметить поддержку языка программирования C# [14-15].

Существуют также более специализированные инструменты для разработки игр. Например, Construct и GameMaker Studio предназначены для создания 2D игр без необходимости написания кода. С помощью этих инструментов можно создавать игры с простыми механиками и меньшим бюджетом [16-17].

Кроме инструментов для разработки игр, есть также платформы для продажи и распространения игр. Наиболее популярной из них является Steam, которая предоставляет доступ к большому количеству игр и позволяет разработчикам продавать свои игры на этой платформе. Есть также другие платформы, такие как GOG и Epic Games Store, которые также предоставляют возможность продажи игр [18-20].

Использование Unity в качестве основного инструмента при разработке игры в жанре 2D платформер является целесообразным решением благодаря кроссплатформенной поддержке, мощным инструментам для работы с 2D графикой, а также поддержке языка программирования C#.

1.5 Вывод по главе

Рынок видеоигр представляет собой перспективное направление для развития бизнеса, именно поэтому многие разработчики ориентируются на создание новых и уникальных игр. Один из самых популярных жанров в игровой индустрии – это 2D платформеры, которые привлекают своим интересным геймплеем и увлекательными заданиями.

Игровой движок Unity отлично подходит работы над игрой в жанре 2D платформер. Его целесообразно использовать при разработке, поскольку он предоставляет не только удобный интерфейс, но и широкий выбор готовых ресурсов (в том числе много возможностей для работы с 2D графикой) и обширное сообщество разработчиков.

2 Проектирование и реализация игрового приложения

2.1 Общее видение системы

Система представляет из себя кроссплатформенное игровое приложение на Unity для персональных компьютеров в жанре 2D платформер. Игра является однопользовательской, все данные хранятся на клиентской части.

Slime's Castle - игра, где Вы - комок слизи, оказавшийся в таинственной башне безумного мага при незнакомых обстоятельствах. Исследуйте, прилипайте, похищайте уникальные способности встретившихся неприятелей, пожирая их! Решайте сложные, но увлекательные головоломки, чтобы проникать в недоступные области, а затем снова прилипайте - все, чтобы вспомнить историю своего возникновения и найти способ выбраться из этого жуткого места.

Основной механикой игры является переключение между формами главного персонажа для использования их особенностей в конкретных игровых ситуациях. Каждая форма включает в себя уникальный текстуру, а также особенности передвижения по игровому уровню и некоторые способности, которые может применять игрок.

Цель игры – выбраться из башни безумного мага и узнать настоящую историю возникновения персонажа игрока.

Геймплей включает в себя поэтапное прохождение уровней с помощью перемещения персонажа игрока в 2D пространстве, использование основной механики, а также взаимодействие с интерактивными объектами для решения головоломок и перехода на следующий уровень.

Приложение должно быть спроектировано для использования на персональных компьютерах под управлением Windows, GNU/Linux и OS X.

Общие характеристики приложения приведены в таблице 2.

Таблица 2 – Характеристика игрового приложения

Жанр	2D платформер с элементами приключения
Сеттинг	Фэнтези
Ключевые механики	Смена формы главного персонажа для продвижения по уровню, решение головоломок
Длительность	1-2 часа
Платформы	Windows, GNU/Linux, OS X или совместимые
Игровой движок	Unity

2.2 Геймдизайнерская документация на примере концепт-документа

При разработке игровых приложений используются различные виды документации. Некоторые виды используемой документации:

- питч – отрывок, представляющий из себя короткое и емкое описание проекта. Как правило, используется при презентации для привлечения спонсирования проекта;
- десятистраничник – документ из десяти страниц, является шаблоном по книге Скота Роджерса. На каждой странице вкратце описывается та или иная сторона проекта в максимально доступном и интересном потенциальной аудитории виде;
- концепт-документ – это документ, включающий в себя детальное описание разрабатываемых игровых механик, игровых локаций, пользовательского интерфейса и других элементов игры. Используется в индустрии видеоигр для организации работы разработчиков.

Проектирование приложения началось с создания концепт-документа, описывающего начальную проработку всех основных аспектов игры и подробно раскрывающего игровые механики.

Концепт-документ включал в себя следующие разделы:

- описание сеттинга игры;
- описание основных внутриигровых систем;

- список и описание игровых механик;
- описание игровых уровней;
- интерактивные объекты и неигровые персонажи;
- элементы пользовательского интерфейса;
- список игровых локаций;
- прочее.

Для ведения и поддержки документации использовалось пространство для командной работы Confluence от компании Atlassian [21].

2.3 Варианты использования

Игровое приложение можно разделить на три блока:

- 1) главное меню;
- 2) игровая сессия;
- 3) меню паузы.

При запуске приложения пользователь первым делом попадает в главное меню. Здесь он может начать игровую сессию, настроить параметры игры или же закрыть приложение. На рисунке 9 представлены варианты использования в главном меню.

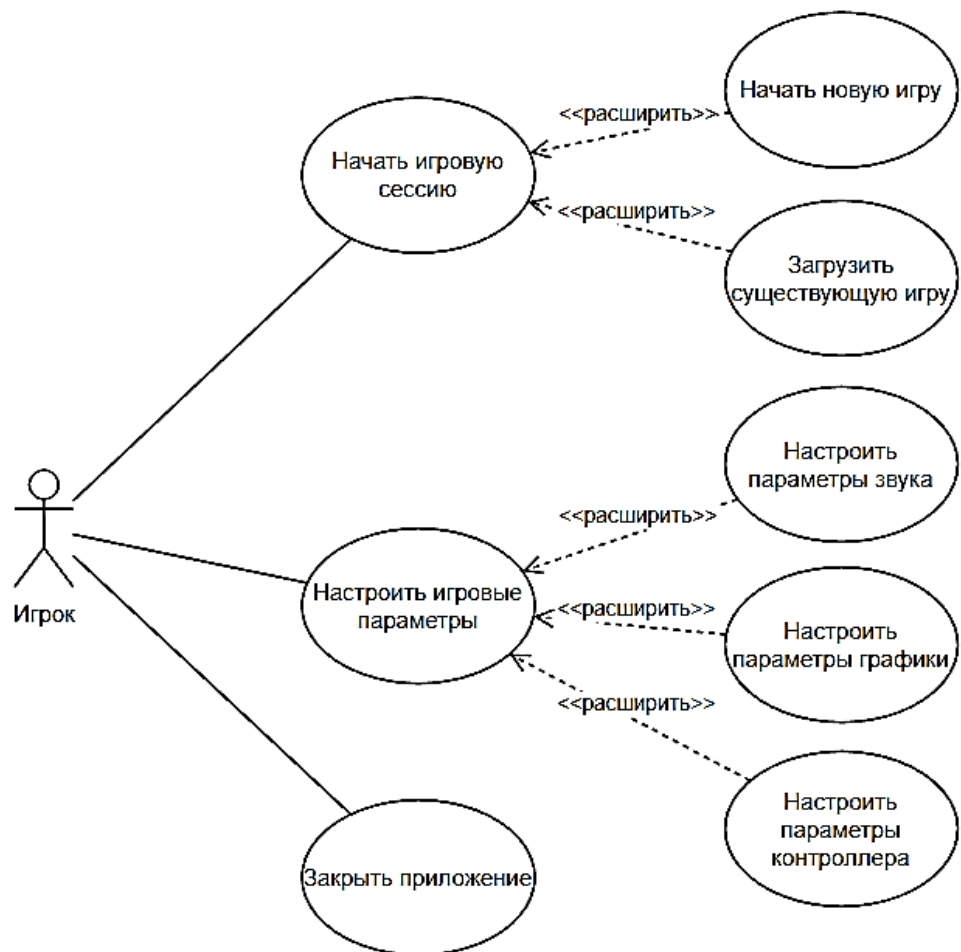


Рисунок 9 – Варианты использования в главном меню

При начале игровой сессии игрок начинает взаимодействовать с игровой сценой. При этом пользователь может перемещать игрового персонажа по сцене, совершать прыжок, изменять форму игрового персонажа, использовать способности и взаимодействовать с интерактивными объектами, а также открыть меню паузы. Все варианты использования на игровой сцене представлены на рисунке 10.

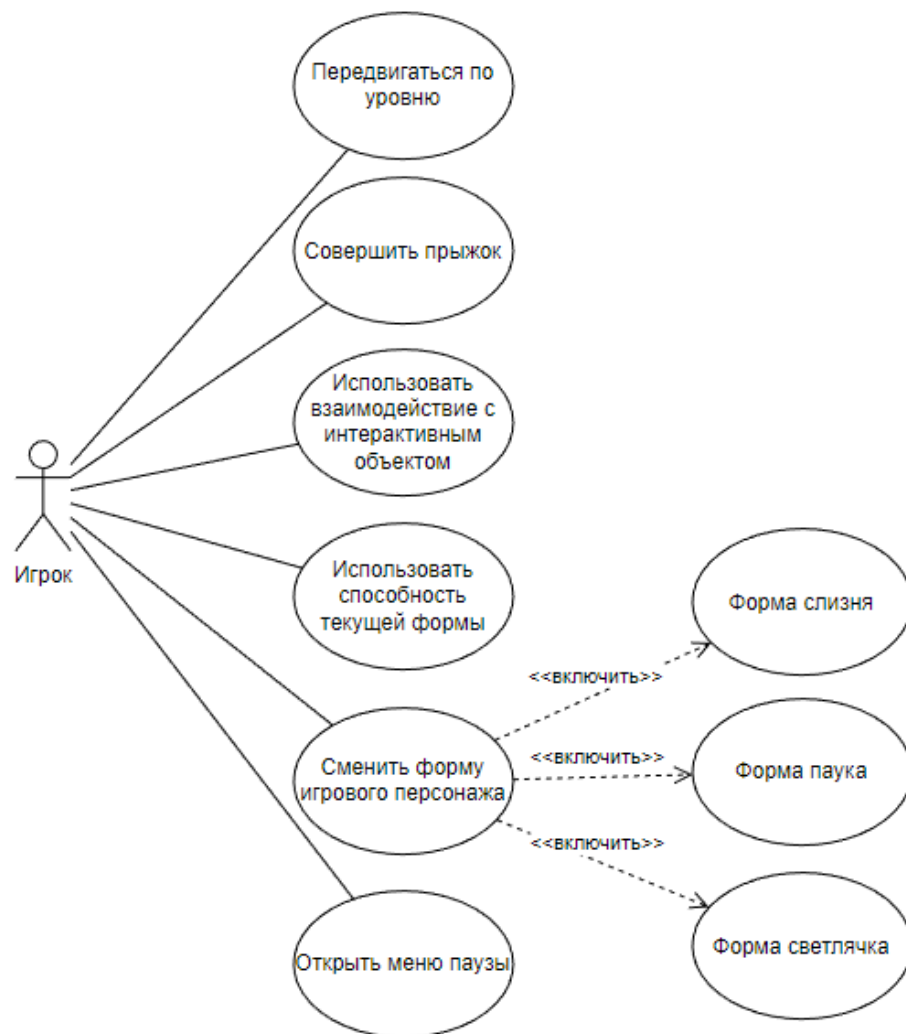


Рисунок 10 – Варианты использования на игровой сцене

Одним из вариантов использования на игровой сцене является открытие меню паузы. Открыв меню паузы, пользователь может настроить игровые параметры, сохранить или загрузить игровой прогресс или же продолжить игровую сессию, тем самым закрыв меню паузы. Варианты использования в меню паузы представлены на рисунке 11.

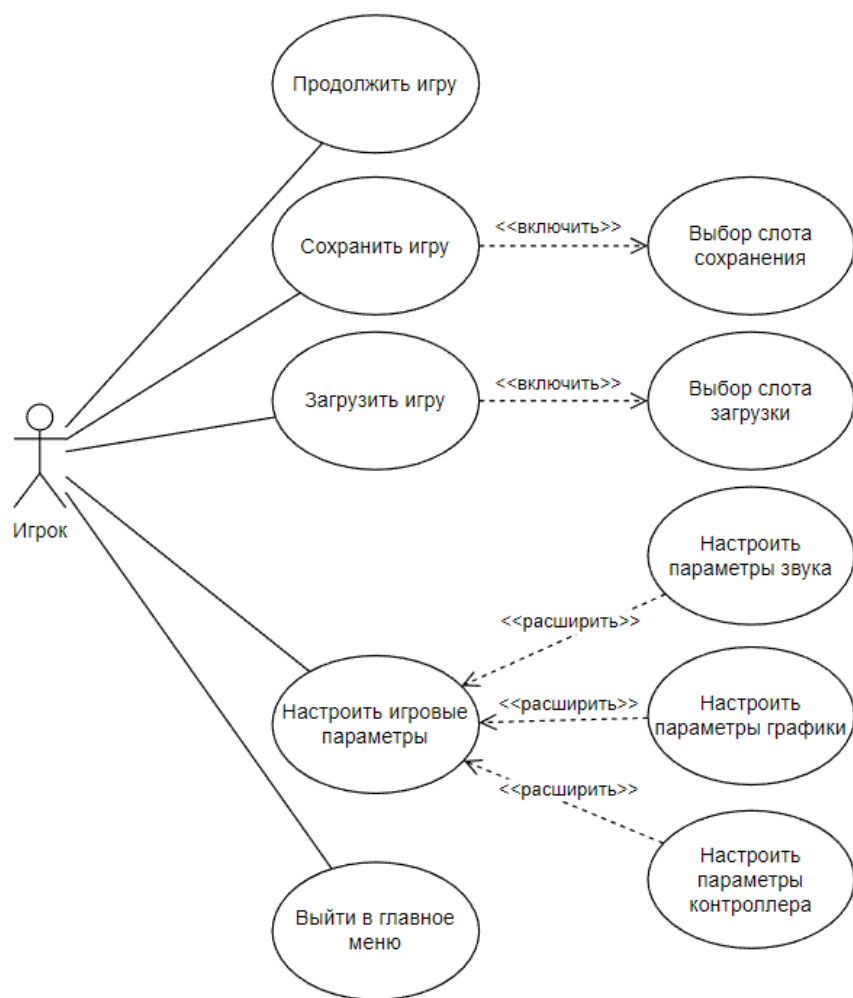


Рисунок 21 – Варианты использования в меню паузы

2.4 Игровой персонаж

Игровой персонаж является одним из основных составляющих игрового приложения. На игровой сцене в редакторе Unity он представляет из себя префаб с большим количеством разнообразных компонентов. Концептуальная диаграмма классов игрового персонажа представлена на рисунке 12.

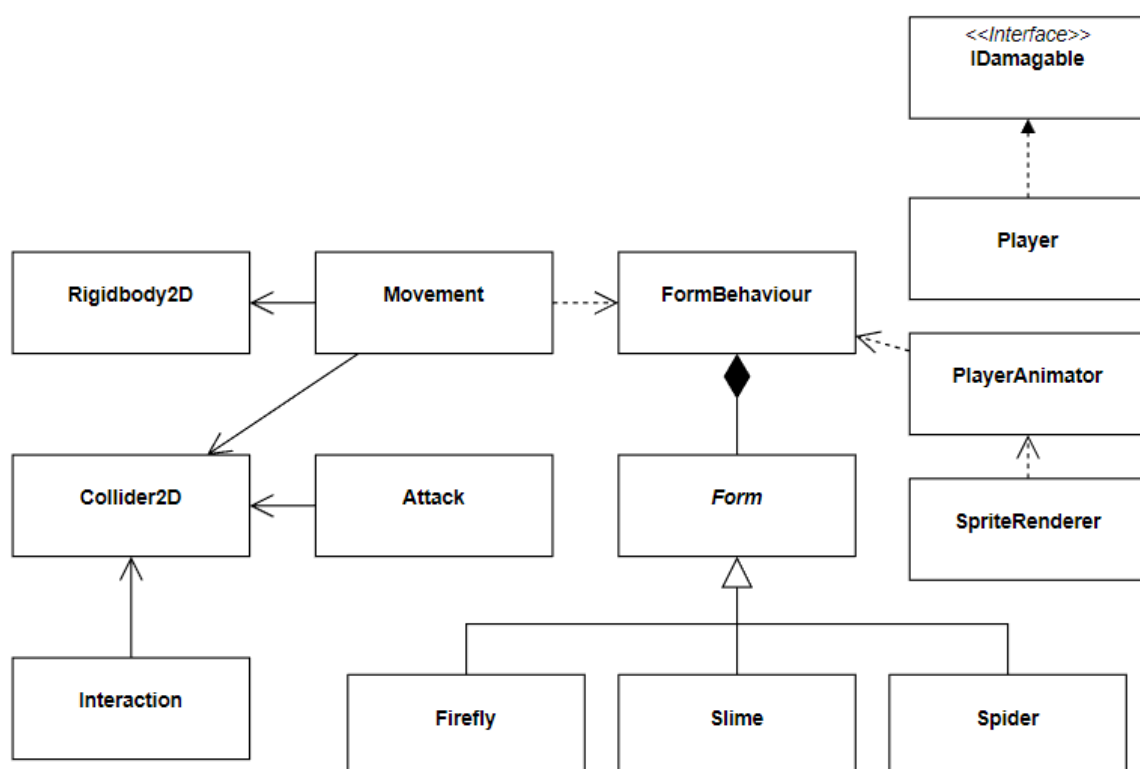


Рисунок 12 – Концептуальная диаграмма классов игрового персонажа

Классы «Movement», «Attack», «Interaction» отвечают за передвижение персонажа и взаимодействие его с игровым миром.

«FormBehavior», «Form», «Firefly», «Slime», «Spider» ответственны за механику смены форм, реализуя паттерн «стратегия».

«PlayerAnimator» выполняет функцию смены текстуры и проигрывания анимации персонажа игрока.

Класс «Player», реализуя интерфейс «IDamagable», отвечает за систему получения урона персонажем игрока, а также помимо этого хранит в себе некоторые параметры, например, уровень здоровья и количество собранных монет.

Таким образом, все компоненты персонажа можно разделить на несколько категорий:

- 1) передвижение и взаимодействие с интерактивными объектами;
- 2) смена форм игрового персонажа;
- 3) прочее.

2.4.1 Передвижение и взаимодействие с интерактивными объектами

Для взаимодействия с виртуальным миром пользователь использует игрового персонажа. Возможные действия пользователя, независимые от формы игрового персонажа представлены в таблице 3.

Таблица 3 – основные способности игрового персонажа

Способность	Описание
Горизонтальное передвижение	Равномерное перемещение игрового персонажа в горизонтальном направлении с определенной скоростью.
Прыжок	Равноускоренное перемещение игрового персонажа в вертикальном направлении под действием прикладываемого импульса и гравитации.
Взаимодействие с интерактивными объектами	При нахождении в поле действия интерактивного объекта игровой персонаж может с ним взаимодействовать. Характер взаимодействия полностью зависит от типа интерактивного объекта. Таким образом, реализация взаимодействия полностью делегируется ему.

Как видно из диаграммы классов, компоненты передвижения и взаимодействия используют компонент «Collider2D» для отслеживания коллизий. Кроме того, компонент передвижения дополнительно использует компонент «Rigidbody2D», поскольку система передвижения игрового приложения спроектирована на основе физики Unity [22].

2.4.2 Смена форм игрового персонажа

Основной механикой игры является переключение между формами главного персонажа для использования их особенностей в конкретных игровых ситуациях. Каждая форма включает в себя уникальную текстуру, а также особенности передвижения по игровому уровню и некоторые способности, которые может применять игрок.

Таким образом, реализация особенностей передвижения персонажа, а также применение им способностей, были делегированы на «текущую» форму. Способности каждой формы описаны в таблице 4.

Таблица 4 – способности форм игрового персонажа

Форма слизня
Дэш – резкий рывок в сторону, позволяющий быстро перемещаться в горизонтальном направлении
Простой удар – способность, позволяющая наносить урон противникам, попадающим в радиус действия
Удар со сплешем – способность, позволяющая делать рывок вертикально вниз и наносить урон противникам в области при соприкосновении с твердой поверхностью
Форма паука
Передвижение по паутине – специальным объектам происходит свободно, во всех направлениях плоскости экрана пользователя, игнорируя гравитацию
Форма светлячка
Двойной прыжок – позволяет совершать второй прыжок, аналогичный обычному в других формах
Планирование – способность, позволяющая снижать скорость падения до приземления на твердую поверхность
Вспышка – способность, позволяющая кратковременно освещать темные участки в определенном радиусе от персонажа и наносить урон противникам, реагирующим на свет

При проектировании стоило учесть возможность расширения персонажа игрока новыми способностями и/или формами. Исходя из этого, было решено применить паттерн «стратегия», выделив каждую форму в отдельный класс, а переключать формы с помощью специального класса «FormBehaviour».

Выбор форм пользователем в игре производится с помощью взаимодействия «ChooseForm», а также экрана, изображенного на рисунке 13.

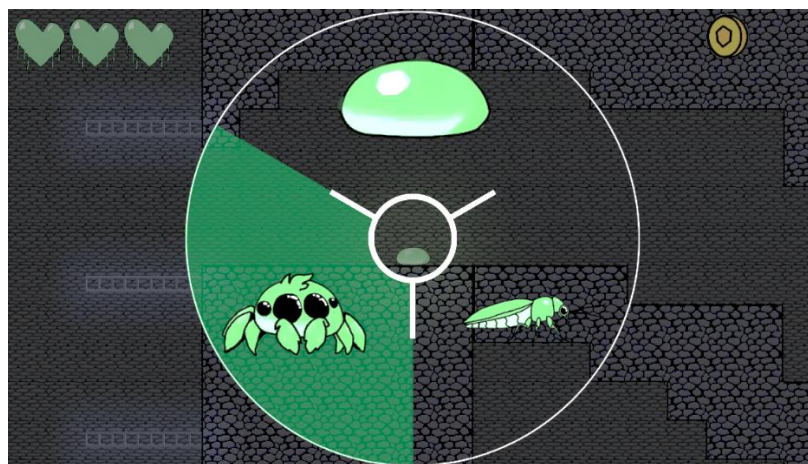


Рисунок 13 – Окно выбора форм в игре

Для выбора желаемой формы пользователю необходимо удерживать клавишу, выбранную для взаимодействия, до открытия окна выбора форм, затем передвинуть курсор мыши на желаемую форму в открывшемся окне и отпустить кнопку. Процесс изменения формы представлен на диаграмме DFD на рисунке 14.

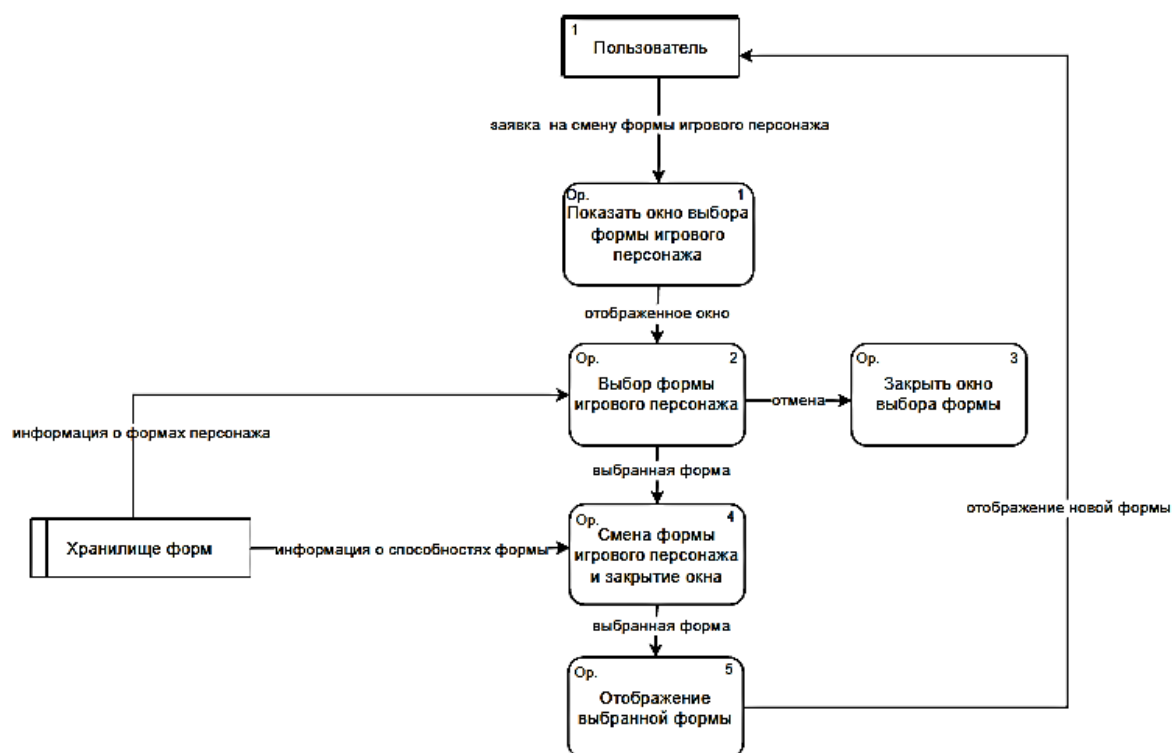


Рисунок 14 – Описание потоков данных функции «Смена формы игрового персонажа»

2.4.3 Прочие компоненты игрового персонажа

К прочим компонентам игрового персонажа можно отнести скрипт «Player», который отвечает за базовые параметры и методы персонажа, такие как, например, изменение уровня здоровья. Для этого класс реализует интерфейс «IDamagable».

Также в эту категорию можно отнести компонент, отвечающий за изменение спрайта персонажа в зависимости от выбранной формы. Для отображения спрайта используется базовый компонент Unity «SpriteRenderer» [22].

2.5 Основные внутриигровые системы

2.5.1 Система обработки ввода пользователя

Система предназначена для обеспечения взаимодействия пользователя с игровым персонажем с помощью контроллеров (клавиатура/мышь/геймпад).

В процессе анализа были выделены следующие функциональные требования:

- система должна отслеживать нажатия клавиш пользователем;
- система должна предоставлять возможность доступа к информации о пользовательском вводе из сторонних классов;
- система должна давать возможность пользователю изменять используемые клавиши;
- система должна отображать пользователю текущие отслеживаемые клавиши.

При проектировании системы было решено использовать встроенный в Unity класс «Input System» в качестве основы обработки нажатий [22]. Систему обработки ввода пользователя можно разбить на несколько компонентов:

- компонент «player input», который хранит в себе карты со всеми возможными взаимодействиями;
- скрипт для работы непосредственно с другими внутриигровыми системами;
- скрипт для переназначения клавиш.

На рисунке 15 изображен компонент «player input». Все взаимодействия разбиты на 3 карты:

- 1) «UI_default» – для взаимодействия с UI элементами от Unity;
- 2) «Slime» – для управления игровым персонажем;
- 3) «UI» – для работы с пользовательским интерфейсом.

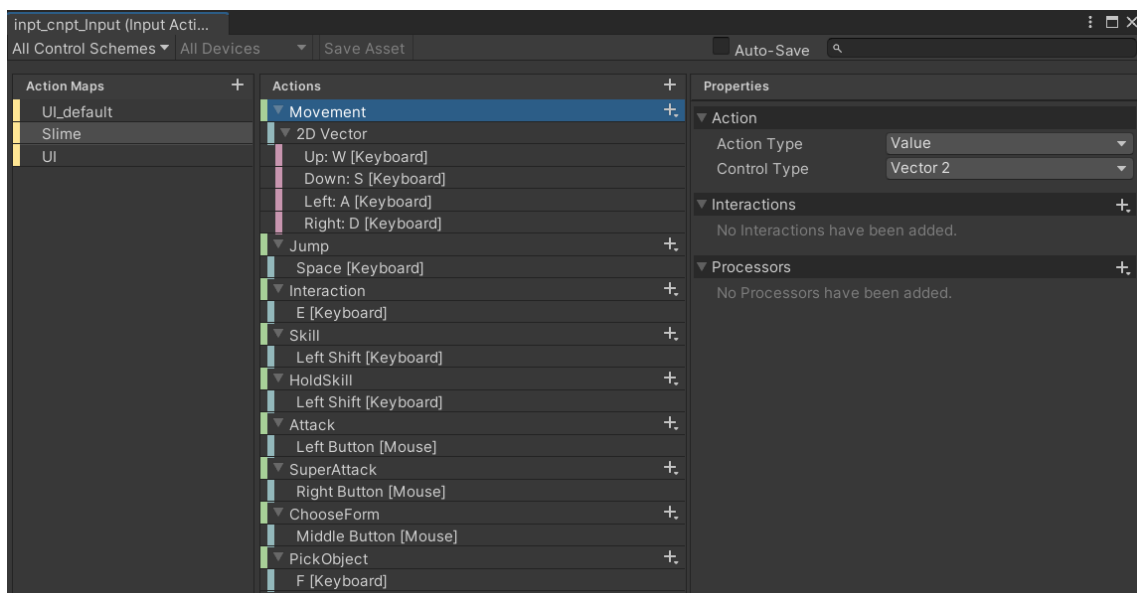


Рисунок 15 – Компонент «player input»

Система также включает в себя UI элемент для отображения в настройках контроллера выбранных на текущий момент клавиш, изображенный на рисунке 16.

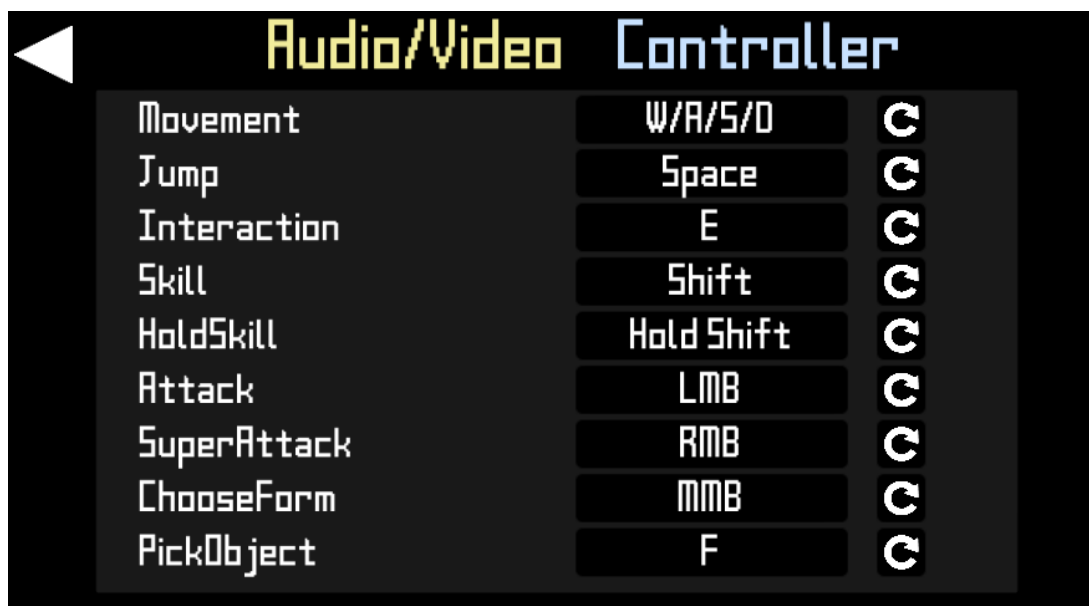


Рисунок 16 – UI элемент с настройками контроллера

При нажатии пользователем на кнопку напротив нужного взаимодействия, открывается окно переназначения отслеживаемой клавиши, изображенное на рисунке 17. Следующая нажатая пользователем клавиша будет выставлена вместо текущей для данного взаимодействия. При нажатии клавиши «Esc» процесс переназначения отменится. При нажатии на кнопку со стрелочкой настройки взаимодействия будут сброшены до стандартных.



Рисунок 17 – Окно переназначения клавиши

2.5.2 Система сохранений

Система ответственна за загрузку и сохранение игрового прогресса пользователя.

В процессе анализа были выделены следующие функциональные требования:

- система должна сохранять данные о прогрессе пользователя, а также состояние интерактивных объектов на сценах;
- система должна сохранять данные в файлы игры;
- система должна извлекать из файлов игры данные и сообщать их соответствующим системам.

На игровой сцене в Unity система представлена префабом, в состав которого входит скрипт, отвечающий за работу с данными. Файлы в формате JSON подходят для хранения параметров системы и данных о прогрессе, поэтому данный формат был использован в работе.

Система включает в себя блок сохранения состояний интерактивных объектов на игровых сценах – все изменения уровня, сделанные игроком, сохраняются в файлы игры точно также, как и данные персонажа. Это позволяет исключить большую часть ошибок при проектировании уровней.

Функция сохранения игрового прогресса показана на диаграмме DFD на рисунке 18.

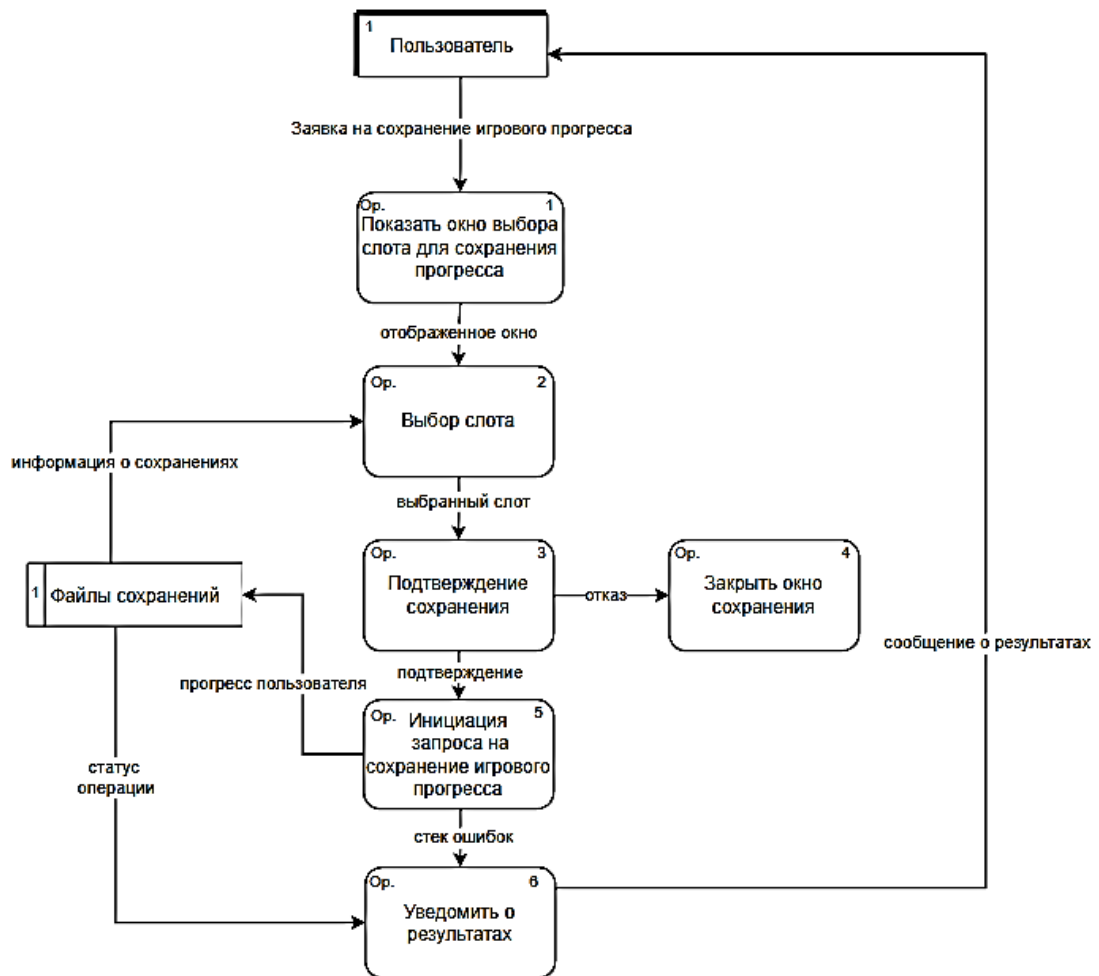


Рисунок 18 – Описание потоков данных функции «Сохранение игрового прогресса»

Пользователь взаимодействует с системой через UI – в меню паузы находятся кнопки «Save» и «Load». После нажатия на кнопку «Save» открывается окно с выбором слота, изображенное на рисунке 19. Если на какой-либо слот ранее записывались данные о прогрессе игрока, то под ним будет краткая информация о прогрессе – суммарное время игры в часах и время последнего сохранения.



Рисунок 19 – Окно выбора слота при сохранении игрового прогресса

При выборе пользователем слота (при нажатии на изображение дискеты) открывается окно с подтверждением действия. После подтверждения следует инициализация запроса на сохранение игрового прогресса пользователя в файлы игры и сообщение об успешности выполнения операции.

Чтобы загрузить игровой прогресс, пользователю необходимо нажать в меню паузы кнопку «Load». Окно выбора слота при загрузке прогресса представлено на рисунке 20. На слотах присутствуют три вида кнопок:

- 1) с изображением стрелки – продолжение игры;
- 2) с изображением знака «+» – начать новую игру;
- 3) с изображением корзины – удалить сохранение.



Рисунок 20 – Окно выбора слота при загрузке игрового прогресса

2.5.3 Система событий

Система предназначена для работы с событиями для уменьшения связности кода.

В процессе анализа были выделены следующие функциональные требования:

- система должна хранить существующие события;
- система должна предоставлять возможность доступа к информации о существующих событиях из сторонних классов;
- система должна единовременно иметь единственный экземпляр на игровых сценах.

В качестве способа поддержки функции обратного вызова используются «UnityEvents» [25].

Система реализует паттерн «одиночка», который позволяет удовлетворить одному из требований. Пример реализации представлен на листинге 1.

Листинг 1 – Пример реализации паттерна «одиночка»

```
public static EventSystem instance;
private void Awake()
{
    if (instance == null)
    {
        instance = this;
    }
    else if (instance != this)
    {
        Destroy(gameObject);
    }

    DontDestroyOnLoad(gameObject);
}
```

2.5.4 Система миссий

Система предназначена для работы с игровыми миссиями (игровые события, в которых хранится текущий прогресс игрока). Возвращает статус миссии при запросе от другого класса.

В процессе анализа были выделены следующие функциональные требования:

- система должна хранить прогресс по миссиям;
- система должна собирать прогресс по запросу и предоставлять внешнему классу.

На сцене в редакторе Unity система из себя префаб, в состав которого входит скрипт, управляющий инициализацией, изменением состояния игровых миссий, а также предоставляющий нужную информацию внешним классам по запросу.

Возможные статусы миссий представлены в перечислении «MissionStatus»:

- «NotAssigned» – для новых миссий;
- «InProgress» – миссия в процессе выполнения;
- «Achieved» – цель миссии достигнута;
- «Completed» – миссия «сдана»;
- «Failed» – миссия провалена.

2.5.5 Система квестов

Цель квестовой игровой системы состоит в управлении задачами игрока, а именно получение, отслеживание прогресса и завершение.

В процессе анализа были выделены следующие функциональные требования:

- система должна хранить свои параметры и все квесты игрока в файлах игры;
- система должна хранить список активных квестов;
- система должна назначать, инициализировать и отслеживать запрошенный квест;
- система должна позволять создавать и редактировать квесты вне редактора Unity;
- система должна уведомлять систему миссий о статусе квестов.

В Unity на игровой сцене система представлена префабом, в состав которого входит скрипт, управляющий извлечением данных из файлов игры и отвечающий за назначение заданий.

Система квестов представляет из себя группу связанных между собой классов «QuestSystem», «Quest» и «Goal». Их взаимосвязь отражена на диаграмме классов, представленной на рисунке 21.

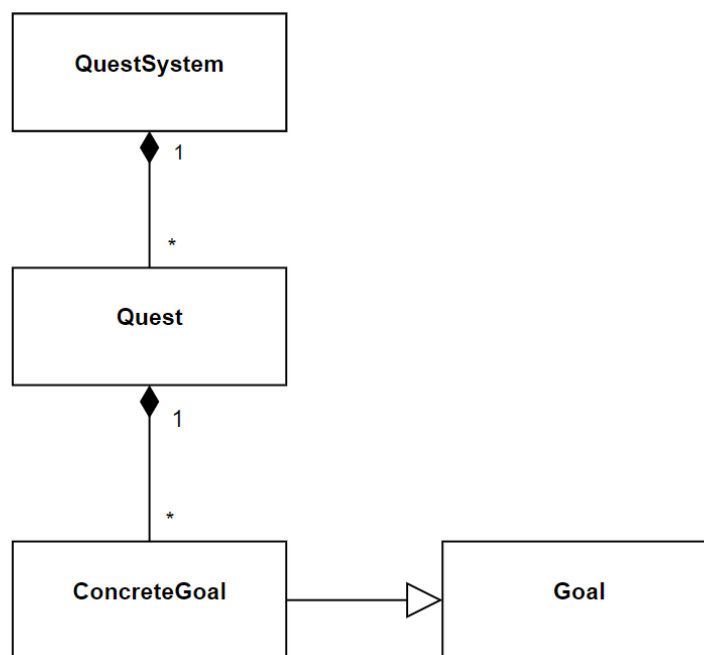


Рисунок 21 – Диаграмма классов квестовой системы

Система квестов хранит активные квесты в качестве компонентов. Каждый квест может включать в себя несколько задач. Задачи делятся на два типа, каждый из которых является классом наследником от Goal:

- 1) «KillGoal» – убийство монстров;
- 2) «GatheringGoal» – сбор предметов.

Квест считается выполненным, когда все задачи, включенные в него, завершены. Удаление квеста из активных и обновление миссий происходит автоматически.

В качестве формата файлов для хранения квестов был выбран JSON. Такой выбор позволяет редактировать квесты в большинстве текстовых редакторов без необходимости использования Unity, что упрощает разработку. JSON файлы также отлично подходят для хранения параметров системы.

Таким образом, система квестов была спроектирована так, что работающему с ней разработчику достаточно было лишь заполнить шаблон в формате JSON для создания нового квеста.

2.6 Пользовательский интерфейс

Весь пользовательский интерфейс в игре можно разделить на две категории. К первой относятся все меню, ко второй – все экраны, появляющиеся непосредственно во время игровой сессии.

Проектировка и моделирование пользовательского интерфейса осуществлялись с помощью платформы Miro [23].

В общем виде модель была построена с помощью карты wireframe'ов, изображенной на рисунке 22. Данный прототип включал также некоторые экраны, относящиеся к игровой сессии. В прототипе были настроены переходы между экранами по нажатию соответствующих кнопок, что позволяло оценить пользовательский опыт использования интерфейса.

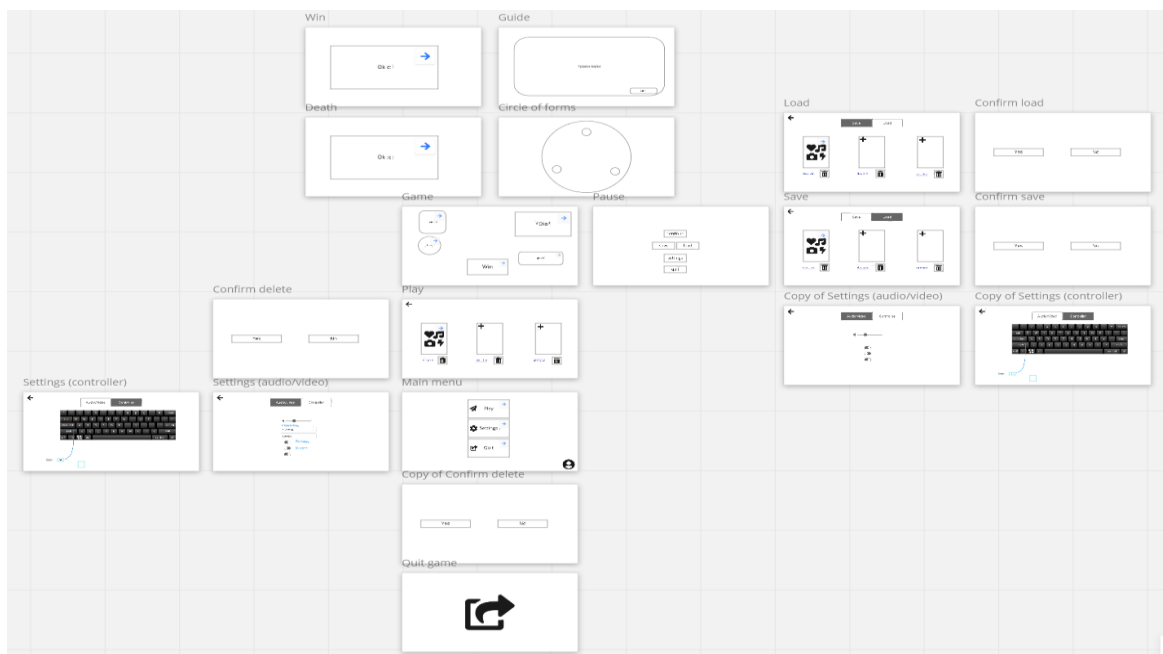


Рисунок 22 – Карта wireframe'ов

К элементам интерфейса, появляющимся во время игровой сессии, можно отнести отображение количества здоровья, собранных монет, а также колесо выбора форм. Экран представлен на рисунке 23.

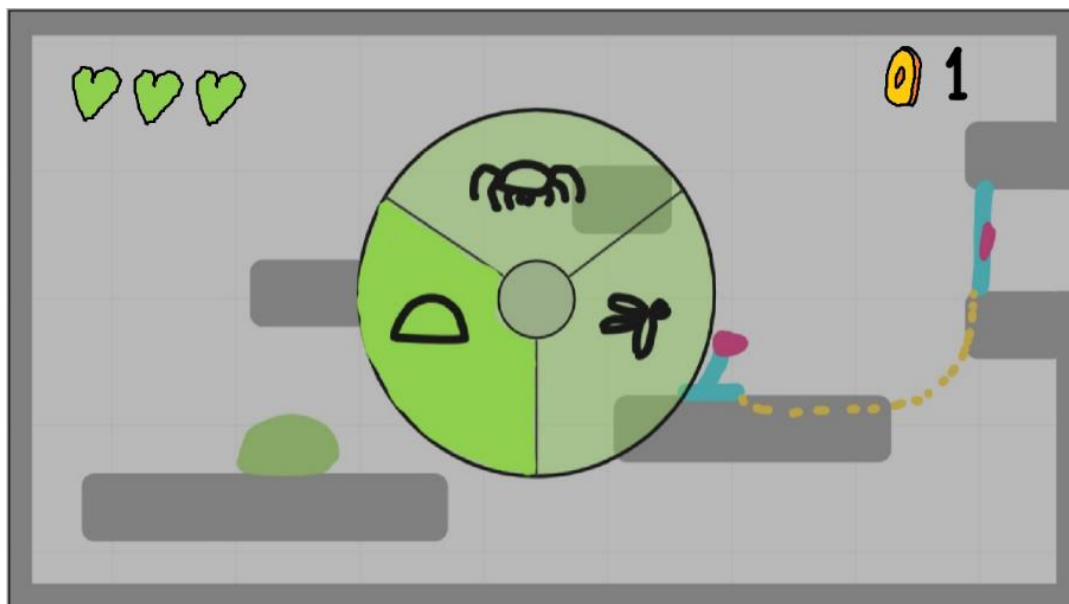


Рисунок 23 – Экран выбора форм

На основе спроектированной карты wireframe'ов были реализованы UI элементы в приложении Unity.

При запуске приложения пользователь попадает в главное меню, в котором находятся кнопки для начала игровой сессии, перехода в меню настроек и выхода из игры.

При нажатии на кнопку «Play» высвечивается окно с выбором варианта начала игровой сессии: пользователю предоставляется информация о текущем игровом прогрессе (при наличии), а также возможность начать новую игру или продолжить текущую. Интерфейс идентичен тому, что появляется при загрузке игрового прогресса.

После выбора варианта пользователь попадает на экран игры, изображенный на рисунке 24, где видит непосредственно игровой уровень и своего персонажа. Здесь у него есть возможность сменить форму персонажа (в таком случае вызывается окно выбора формы), а также поставить игру на паузу.

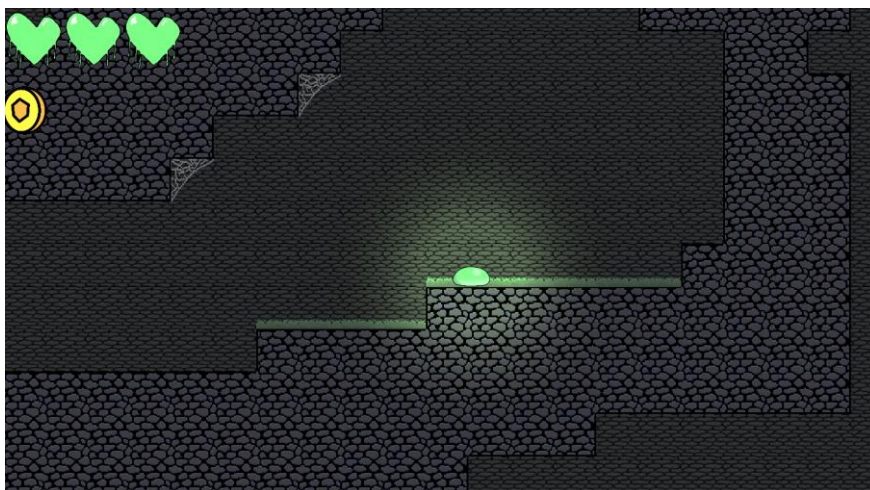


Рисунок 24 – Экран игры

В окне паузы, изображенном на рисунке 25, присутствуют кнопки для продолжения игровой сессии (снятия игры с паузы), сохранения/загрузки игрового прогресса, перехода в настройки и перехода в главное меню.

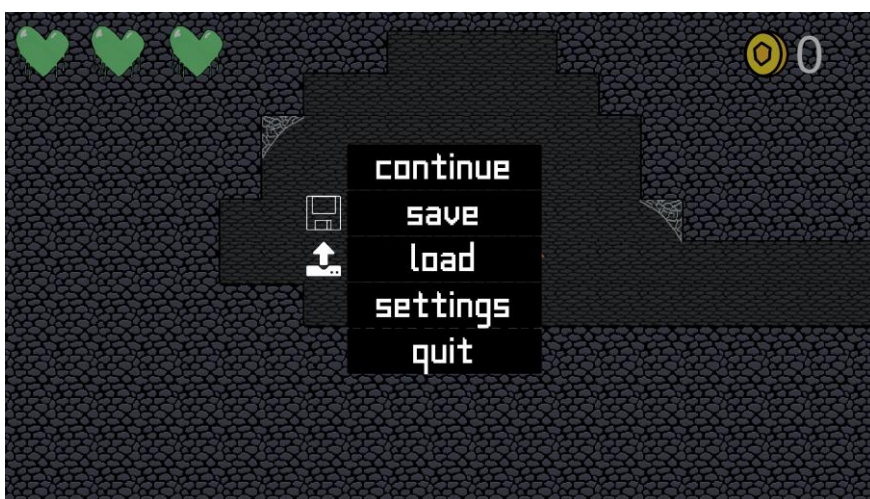


Рисунок 25 – Меню паузы

В раздел изменения параметров можно перейти из главного меню или меню паузы. Данный раздел делится на изменение видео/аудио настроек и настроек контроллера, и состоит из двух основных соответствующих экранов.

Экран настроек видео/аудио, изображенное на рисунке 26, содержит в себе кнопку для возврата в главное меню, кнопку для перехода в настройки контроллера, ползунок громкости музыки, флажок для включения/отключения полноэкранного режима, а также выпадающий список для изменения разрешения экрана.



Рисунок 26 – Экран настроек видео/аудио

Настройки контроллера включают в себя два экрана – непосредственно экран с текущими взаимодействиями и клавишами, а также экран переназначения выбранной клавиши. Данные экраны описаны в пункте 2.5.1 «Система обработки ввода пользователя».

2.7 Игровые уровни

Разработка уровней для игры разбилась на несколько этапов:

- создание прототипов;
- формирование метрик;
- реализация спроектированных уровней в редакторе Unity.

При визуальной разработке игровых уровней использовалась система тайлов Unity [25].

Проектирование началось с создания общих очертаний и идеи каждого уровня с помощью платформы Miro. Пример схемы уровня представлен на рисунке 27. Каждая часть имеет свой номер и/или название для удобства коммуникации.

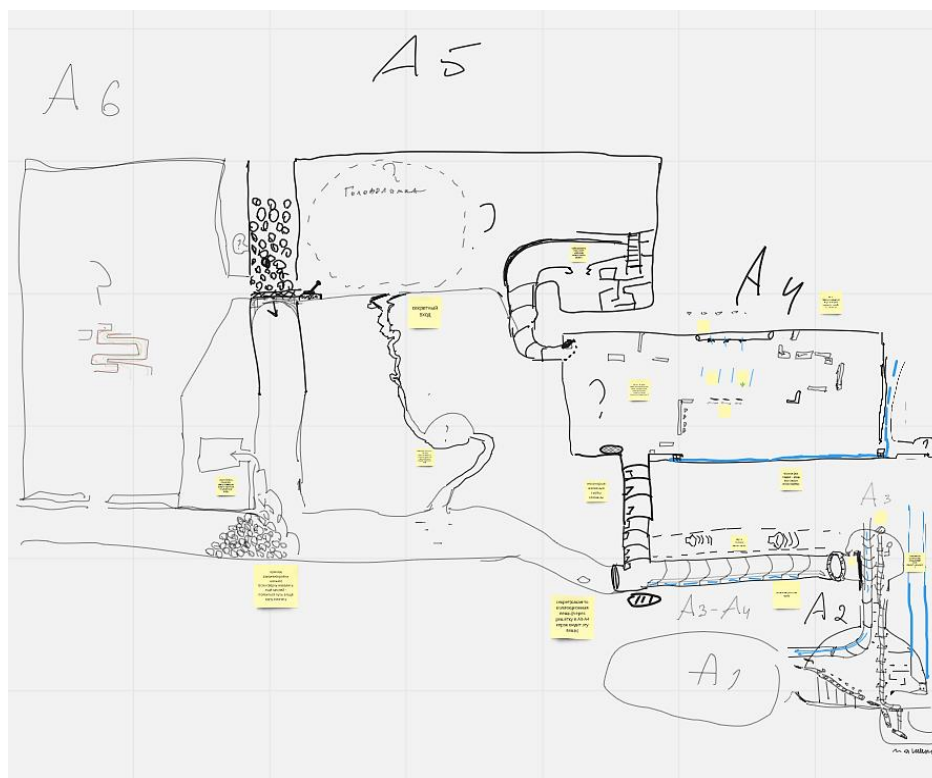


Рисунок 27 – Игровой уровень на начальном этапе проектирования

После выделения общих границ и идей уровня следовала детализация.

Пример детализации части с номером «А4» представлен на рисунке 28.

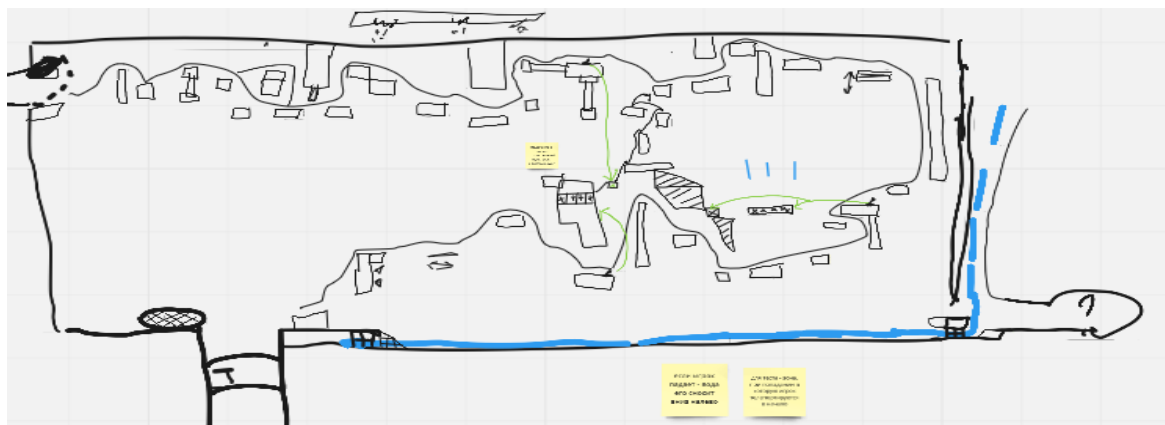


Рисунок 28 – Часть уровня с номером «А4»

Общая схема всех спроектированных на начальном этапе уровней представлена на рисунке 29.

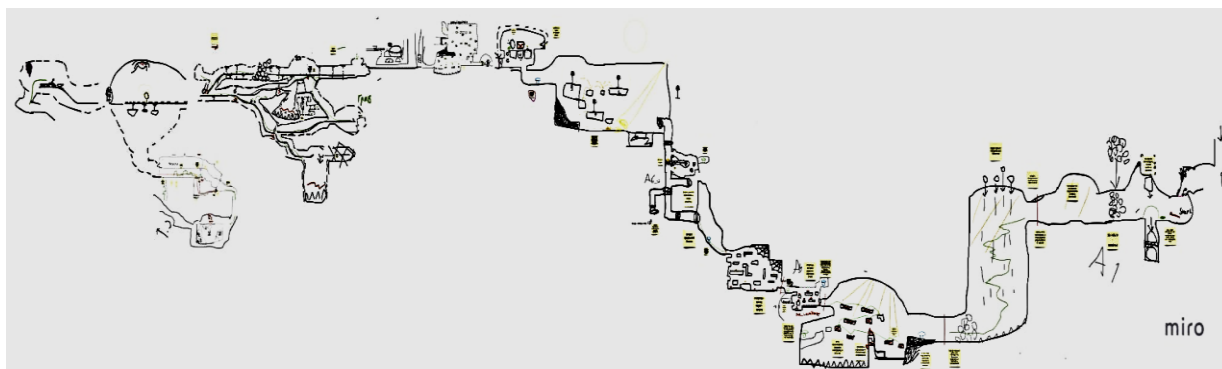


Рисунок 29 – Схема уровней

Для того, чтобы реализовать спроектированные уровни в редакторе Unity, было необходимо определиться с метриками. Размер игрового персонажа было решено принять в 1,0 блок в ширину и 0,5 в высоту. Высота и длина прыжка, траектория планирования светлячка и другие метрики представлены на рисунке 30.

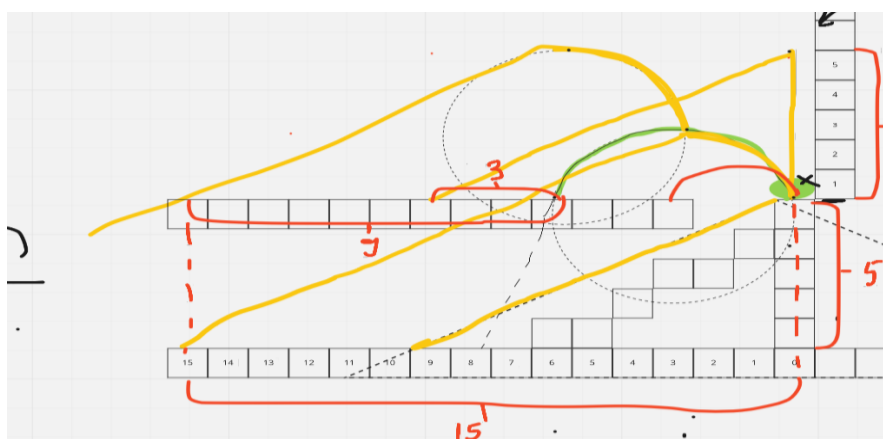


Рисунок 30 – Определение метрик персонажа

Перед реализацией уровней непосредственно в редакторе Unity, они были спроектированы с учетом определенных метрик для определения потенциально проблемных мест и ошибок при проектировании. Пример части «А4» с учетом метрик представлен на рисунке 31.

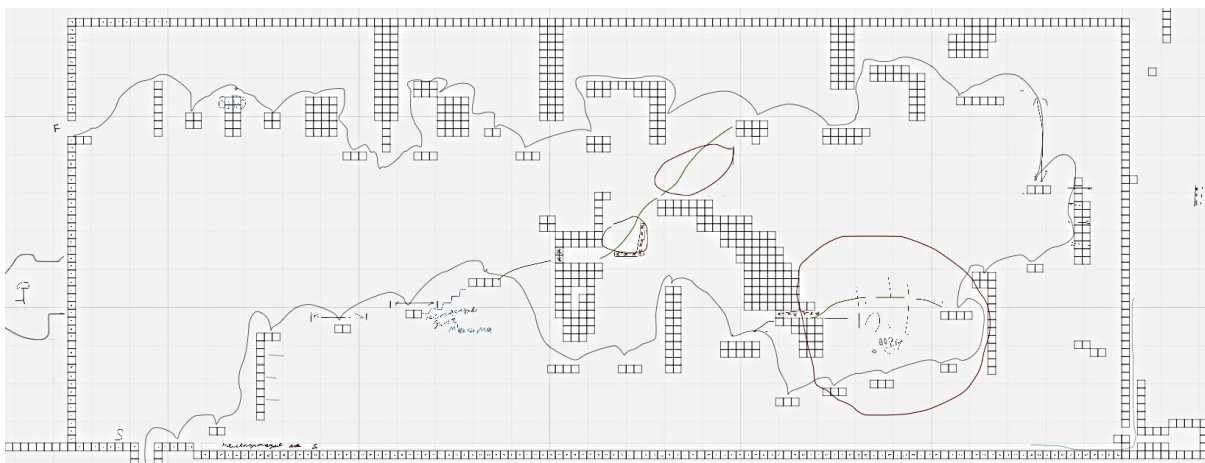


Рисунок 31 – Модель части «А4» с учетом метрик

Финальным этапом стала реализация спроектированных уровней в редакторе Unity. Пример реализации части «А4» в редакторе изображен на рисунке 32.

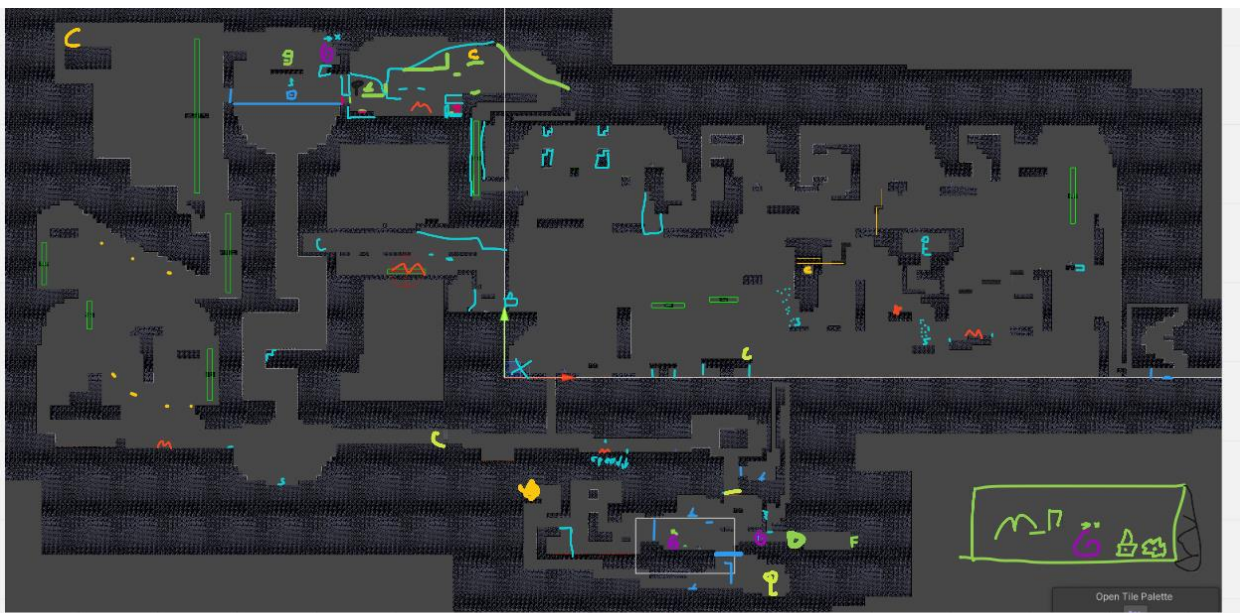


Рисунок 32 – Собранный в редакторе Unity уровень

2.8 Интерактивные объекты

2.8.1 Триггер

Игровой объект, предназначенный для передачи информации о коллизии с игроком выбранной системе или объекту.

В процессе анализа были выделены следующие функциональные требования:

- объект должен передавать информацию выбранной системе или объекту;
- объект должен удаляться с игровой сцены после передачи информации;

На игровой сцене в редакторе Unity представляет из себя префаб, включающий в себя в качестве компонентов коллайдер и скрипт. Триггер позволяет:

- запускать диалоги (отображаемые с помощью специального окна фрагменты текста) с помощью менеджера диалогов;
- изменять прогресс по миссиям с помощью системы миссий;
- изменять привязанный интерактивный объект.

Пример использования триггера на игровой сцене представлен на рисунке 33, здесь при коллизии с игроком запускается диалог.

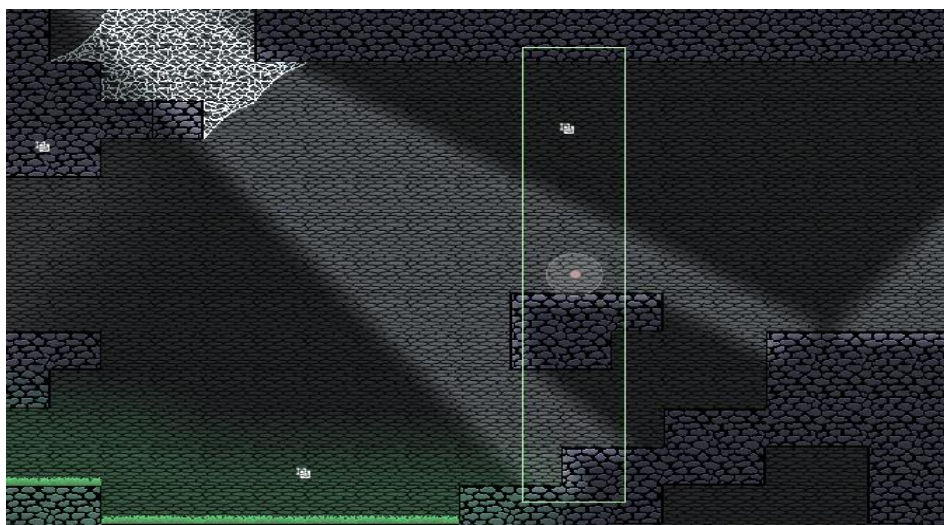


Рисунок 33 – Расположенный на игровой сцене триггер

2.8.2 Головоломки с лучами света

Головоломки с лучами света представляют из себя особые игровые ситуации, в которых игроку необходимо выполнить последовательность некоторых действий для прохождения дальше по уровню. Игровая логика световых головоломок связана с изменением углов падения/отражения лучей света. Для этого игроку необходимо вращать некоторые из приведенных ниже интерактивных объектов с помощью взаимодействия с нажимными

плитами/рычагами для того, чтобы направить луч света от источника к приемнику.

Перечень интерактивных объектов:

- источники – излучают луч света в определенном направлении;
- зеркала – отражают луч света на определенный градус согласно законам физики (подобно реальным зеркалам);
- линзы – фокусируют луч света при нахождении игрока. Работают подобно источникам, за исключением того, что для излучения света игрок должен находиться рядом с линзой. Излучают свет в противоположную игроку сторону;
- приёмники – активируется, если на них попадает луч света. Способны активировать связанные интерактивные объекты (например, открывать двери);
- концентраторы – направляют принятый луч света в определенном направлении (подобно преломлению).

Пример головоломки представлен на рисунке 34. Два луча света выходят из источников и попадают на зеркало, которое отражает их. Нажимные плиты, расположенные справа, вращают зеркало по часовой стрелке и против соответственно.

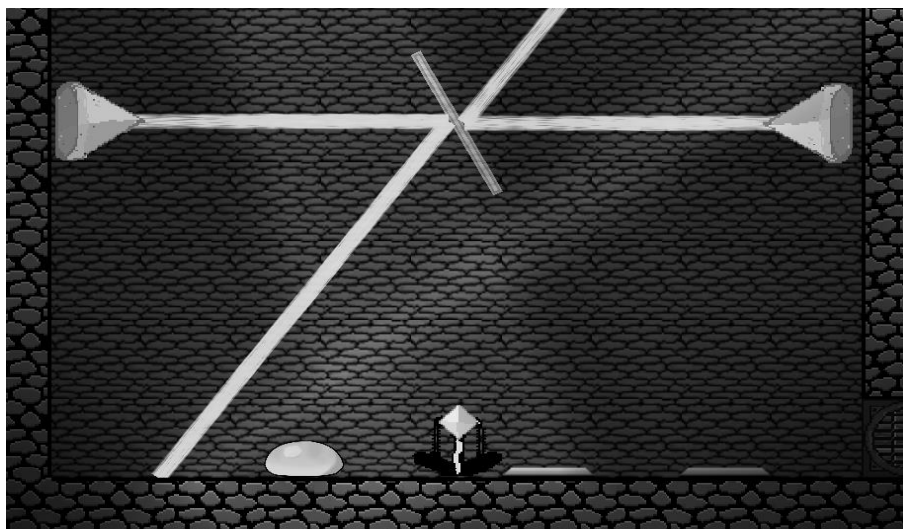


Рисунок 34 – Пример головоломки с лучами света

На рисунке 35 представлена решенная головоломка – один из лучей попадает на приемник и активирует его.

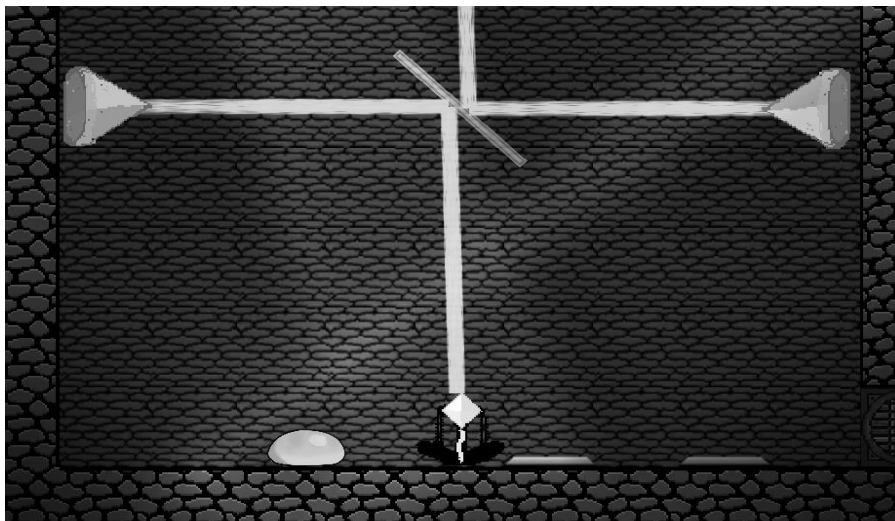


Рисунок 35 – Решенная головоломка с лучами света

2.9 NPC паук

Неигровой персонаж, который представляет из себя противника. Представлен на рисунке 36 в состоянии ожидания начала стадии 1.



Рисунок 36 – Босс паук в состоянии ожидания начала стадии 1

Имеет 3 стадии боя, в каждой из которых реализует паттерн «состояние». При проектировании были созданы диаграммы состояний, отражающие поведение персонажа.

Первая стадия представляет из себя битву в темной замкнутой комнате, где паук хаотично передвигается от точки к точке и, используя четыре вида атак (сбоку лапой, снизу лапой, укус сверху, стрельба паутиной издалека), пытается нанести урон игроку. Задача игрока – при помощи «вспышки» светлячка оглушать паука, а затем с помощью формы слизня наносить ему урон.

Возможные состояния паука приведены на рисунке 37. Как видно из диаграммы, атака «стрельба паутиной» является отдельным состоянием, в котором персонаж находится, пока игрок использует форму светлячка. Это связано с тем, что светлячок имеет возможность оглушить паука и в целом имеет большую интенсивность света, чем другие две формы. Таким образом, игрок вынужден использовать другую форму игрового персонажа, чтобы подпустить паука поближе. Тогда он переходит в состояние атаки, где случайным образом выбирается одна из трех оставшихся атак. После выбора начинается анимация подготовки к атаке, во время которой у игрока есть возможность перевести паука в состояние «оглушение» вспышкой светлячка, а затем в состояние «реакция на удар» ударом слизня. Если при этом уровень здоровья паука опустится ниже 0, начнется стадия 2. Если же игрок даст ему завершить анимацию – паук перейдет в состояние атаки, во время которой он наносит сам удар и является неуязвимым и невосприимчивым к любым действиям игрока.

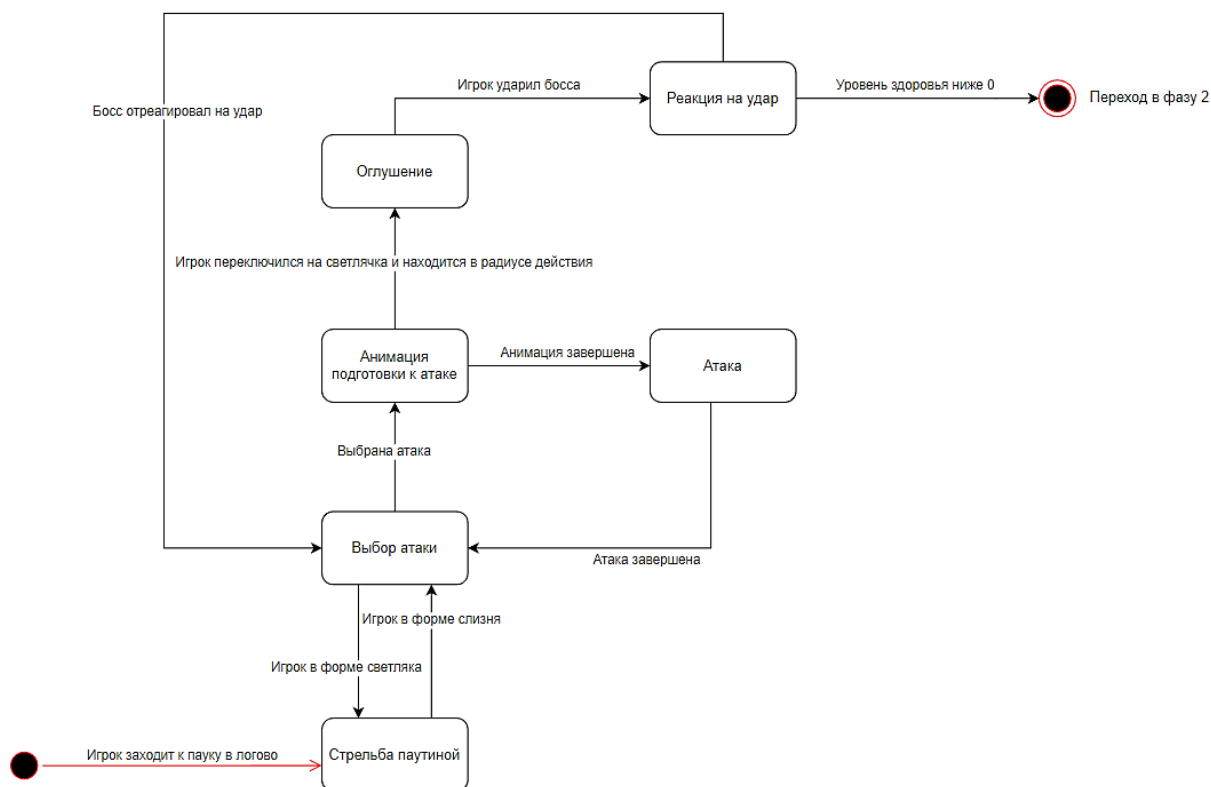


Рисунок 37 – Диаграмма состояний босса паука в стадии 1

Вторая стадия представляет из себя погоню паука за игроком в коридорном уровне с различными препятствиями. Задача игрока – добежать до конца коридора, не получив урон.

Состояния паука приведены на рисунке 38. Стадия начинается сразу после конца первой стадии, паук переходит в состояние погони за игроком, где движется по коридору вслед за ним. Если он приближается к игроку слишком близко, то переходит в состояние атаки и убивает игрока, уровень загружается с последнего сохранения. Также у паука есть состояние ожидания игрока, в которое он переходит примерно на середине пути, когда игрок забегает в безопасную зону. Когда игрок покидает эту зону, паук возвращается к состоянию погони. Если игрок дойдет до конца коридора, произойдет переход в стадию 3.

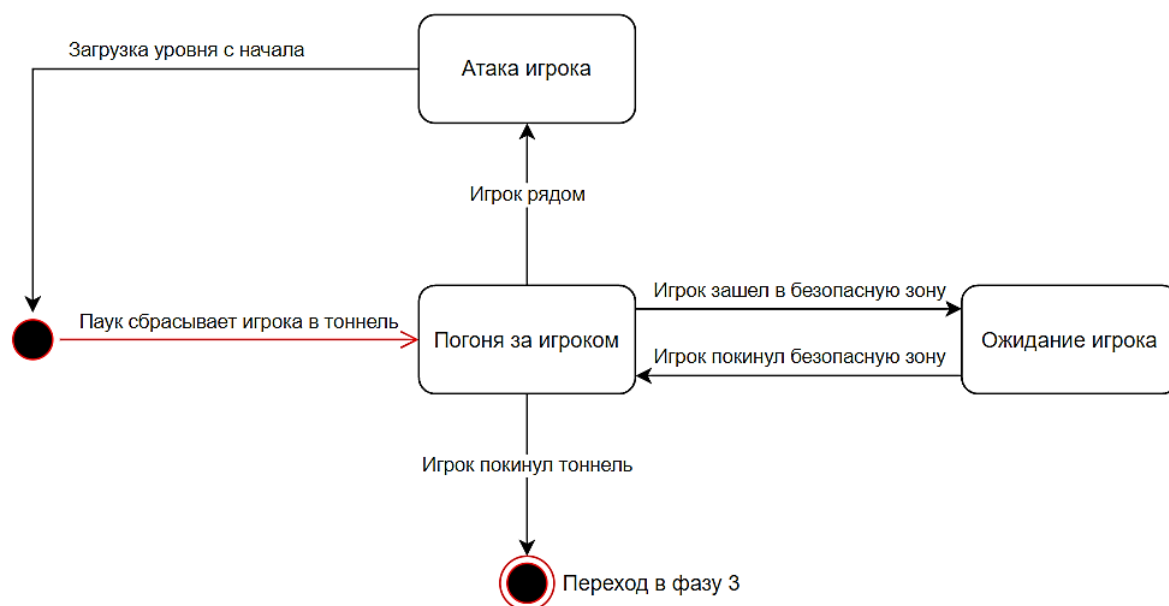


Рисунок 38 – Диаграмма состояний босса паука в стадии 2

Третья стадия представляет из себя погоню паука за игроком по уровню, состоящему из комнат, расположенных по типу лабиринта. Задача игрока – заманить паука и нанести урон с помощью интерактивных объектов – ламп и рычагов.

Состояния паука приведены на рисунке 39. Стадия начинается с состояния погони за игроком подобно состоянию в стадии 2. Также подобно второй стадии работает и атака паука, однако здесь она не убивает игрока с первого раза, а лишь наносит определенное количество урона. Подобно первой стадии работает и оглушение, однако теперь, чтобы перевести паука в это состояние, необходимо использовать лампы и рычаги. У паука также присутствуют состояния реакции на удар, анимация снятия оглушения. В состоянии особой атаки он переходит после снятия оглушения. В этом состоянии паук совершает привычные атаки лапой перед собой, однако их радиус, урон и скорость больше, чем у стандартных. Если во время состояния реакции на удар его уровень здоровья опустится ниже 0, паук перейдет в состояние смерти, после чего проиграет финальную анимацию и удалится из игрового мира.

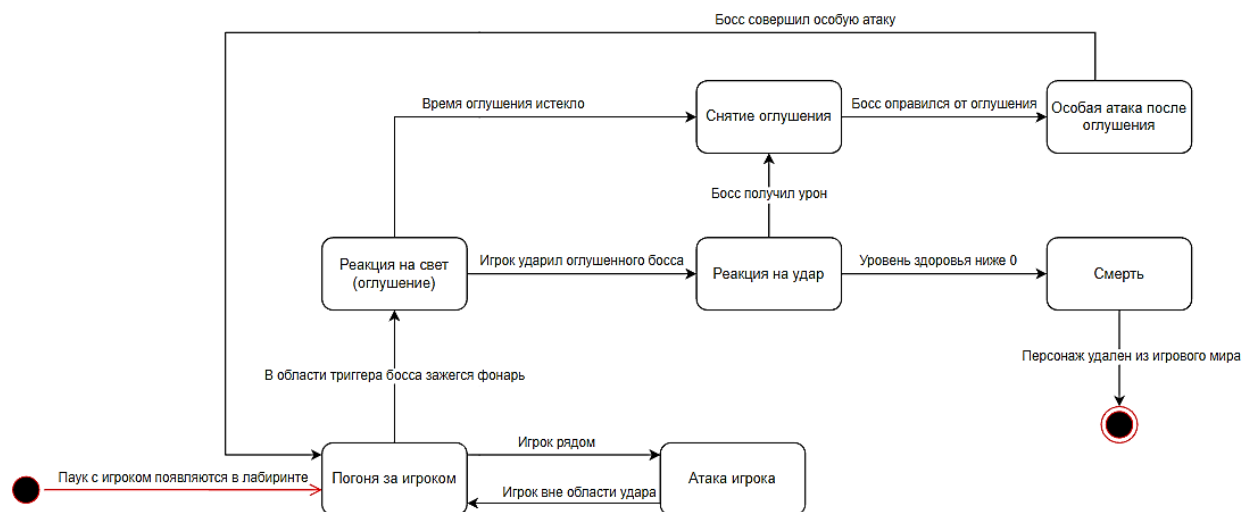


Рисунок 39 – Диаграмма состояний босса паука в стадии 3

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Обучающемуся:

Группа	ФИО
8K92	Лутошкин Степан Евгеньевич

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Оклад руководителя – 40000 руб. Оклад разработчика – 20000 руб.
2. Нормы и нормативы расходования ресурсов	Премияльный коэффициент 30%; Доплаты и надбавки руководителя 40%; Доплаты и надбавки разработчика 20%; Дополнительной заработной платы 15%; Накладные расходы 16%; Районный коэффициент 1,3.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Тариф отчислений во внебюджетные фонды 7,6%

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернативности проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Определение потенциального потребителя результатов исследования. SWOT-анализ разработанной стратегии.
2. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения

Перечень графического материала:

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. График проведения и бюджет НТИ
4. Оценка ресурсной, финансовой и экономической эффективности НТИ

Дата выдачи задания для раздела по линейному графику	01.03.2023 г.
--	---------------

Задание выдал консультант по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов Магеррам Али оглы	д.э.н.		

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
8K92	Лутошкин Степан Евгеньевич		

3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Введение

Выпускная квалификационная работа представляет собой проектирование и разработку кроссплатформенного игрового приложения на Unity для персональных компьютеров в жанре 2D платформер. Отличительной особенностью продукта является уникальная игровая механика смены форм игрового персонажа для преодоления препятствий с помощью особенностей каждой формы. Область применения приложения – цифровые развлечения.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проведение анализа продукта для установки его экономической ценности на рынке, оценка эффективности разработки, определение рисков и стратегий их устранения, формирование состава работ и бюджета проекта.

Для достижения обозначенной цели необходимо решить следующие задачи:

- оценить коммерческий потенциал и перспективность разработки;
- провести оценку научно-технического уровня исследования и оценку рисков;
- составить план работ по реализации проекта;
- рассчитать бюджет проекта.

3.1 Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения

3.1.1 Потенциальные потребители

Поскольку в рамках данной работы рассматривается разработка кроссплатформенного игрового приложения в жанре 2D платформер для персональных компьютеров, потенциальными потребителями продукта являются фанаты компьютерных игр, в частности игр схожего жанра.

3.1.2 Анализ конкурентных технических решений

Существует большое количество видеоигр в жанре 2D платформер, но все они отличаются друг от друга, каждая игра имеет свою уникальную игровую механику или другую отличительную особенность.

Поскольку рынок видеоигр находится в постоянном движении и каждый день появляются все новые продукты и решения, для успешных продаж приложение должно быть конкурентоспособным. Это означает, что нужно здраво оценивать свои сильные и слабые стороны и принимать решения в зависимости от них. Рассмотрим ближайших конкурентов разрабатываемого игрового приложения.

Hollow Knight – платформер-метроидвания про королевство жуков, который выделяется своим потрясающим визуальным стилем, хорошо продуманным миром и атмосферой, сложными боссами и интересными механиками.

Ori and the Blind Forest – это красочный и захватывающий платформер, в котором игрок управляет маленьким существом по имени Ори, стремящимся спасти свой лес от опустошения. Игра привлекает своей красивой графикой и музыкой, создающими захватывающую атмосферу. Кроме того, в игре есть элементы RPG, такие как система улучшений навыков, что добавляет глубины и интереса к прохождению игры.

Для оценки конкурентоспособности разрабатываемой системы была составлена карта сравнения конкурентных технических решений, представленная в таблице 5. Индексом «ф» обозначена собственная разработка, индексом «к1» – игра Hollow Knight, индексом «к2» – Ori and the Blind Forest.

Таблица 5 – Сравнительный анализ конкурентов

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Бф	Бк1	Бк2	Кф	Кк1	Кк2
Потребность в ресурсах памяти	0,15	5	5	3	0,75	0,75	0,45
Качество сюжета	0,25	4	5	5	1	1,25	1,25
Время прохождения игры	0,15	3	5	3	0,45	0,75	0,45
Удобство эксплуатации	0,2	4	5	5	1	1	1
Стоимость продукта	0,1	5	3	3	0,5	0,3	0,3
Срок выхода на рынок	0,05	4	4	4	0,2	0,2	0,2
Объем трудозатрат	0,1	5	3	3	0,5	0,3	0,3
Итого	1				4,4	4,55	3,95

Анализ конкурентоспособности показал, что продукт опережает технические и экономические показатели эффективности. Основными преимущественными факторами являются: стоимость продукта, потребность в ресурсах памяти, а также объем трудозатрат.

3.1.3 Технология QuaD

Для оценки перспективности разработки в соответствие с методологией QuaD была составлена оценочная карта, представленная в таблице 6.

Таблица 6 – Оценочная карта перспективности разработки

Критерии оценки	Вес критерия	Балл	Максимальный балл	Относительное значение	Средневзвешенное значение
Потребность в ресурсах памяти	0,15	90	100	0,9	0,135
Качество сюжета	0,25	70	100	0,7	0,175
Время прохождения игры	0,15	70	100	0,7	0,105
Удобство эксплуатации	0,2	85	100	0,85	0,17
Стоимость продукта	0,1	95	100	0,95	0,095
Срок выхода на рынок	0,05	70	100	0,7	0,035
Объем трудозатрат	0,1	90	100	0,9	0,09
Итого	1			5,7	0,805

Анализ, произведенный по технологии QuaD, показал, что разработка программного продукта перспективна, так как итоговое показательное значение попадает в диапазон от 80 до 100.

3.1.4 SWOT-анализ

Для того чтобы структурировано описать выбранный вариант решения поставленной цели, а именно разработку собственных программных модулей, применим один из методов стратегического планирования – SWOT-анализ. Анализ позволяет дать качественную оценку текущей ситуации, а также показывает, насколько достижима реализация имеющихся возможностей при наличии внешних угроз. Он представляет из себя выявление сильных и слабых сторон решения, возможностей, а также угроз, завершающий – определение соответствий утверждений, определенных на предыдущем этапе, между собой. Результаты выполненной в ходе данного этапа работы приведены в таблице 7.

Таблица 7 – SWOT анализ

	Сильные стороны: С1. Интуитивно понятный пользовательский интерфейс С2. Простота в эксплуатации С3. Уникальная игровая механика смены форм игрового персонажа	Слабые стороны: СЛ1. Временные затраты на разработку СЛ2. Повторное создание уже существующих решений СЛ3. Отсутствие рекламной компании
Возможности: В1. Популярность жанра игры В2. Использование актуальных и популярных инструментов разработки	Сопоставление сильных сторон и возможностей говорит о том, что благодаря упрощённому интерфейсу и уникальной игровой механике у проекта есть возможность привлечь первичных клиентов, также есть шансы на дальнейшее увеличение их числа благодаря популярности жанра.	Оценивая возможности и слабые стороны проекта будет уместным сделать вывод, что система может быть не разработана вовремя.
Угрозы внешней среды: У1. Отсутствие спроса на продукт У2. Вероятность создания аналогичного продукта крупной компанией	Сильные стороны и угрозы показывают, что, несмотря на полезность разработки, существует вероятность спада интереса к данному продукту. На данный момент это лишь потенциально возможная ситуация.	Слабые стороны и угрозы сигнализируют о том, что система может не иметь спрос у пользователей на длинном промежутке времени.

По результатам проведенного SWOT-анализа, можно сделать вывод, что несмотря на имеющуюся угрозу и наличие слабых сторон решения, открывающиеся перед командой возможности, достижимые за счет сильных сторон, достаточны для продолжения работ по проекту.

3.2 Планирование работ по научно-техническому исследованию

Грамотное планирование работ является основой успешного завершения проекта в указанных при согласовании временных рамках. Данный раздел посвящен определению структуры научно-исследовательских работ и их планированию.

3.2.1 Структура работ в рамках научного исследования

Перечень этапов работы и распределение исполнителей представлен в таблице 8.

Таблица 8 – Перечень работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Выбор направления исследований	1	Составление и утверждение темы бакалаврской работы	Руководитель, разработчик
Выявление требований	2	Анализ предметной области и аналогов	Руководитель, Разработчик
	3	Постановка требований к программному обеспечению	Руководитель, Разработчик
	4	Подбор и изучение материалов по теме	Разработчик
	5	Календарное планирование работ	Руководитель, Разработчик
Проектирование ПО	6	Выбор средств и инструментов проектирования и разработки	Разработчик
	7	Проектирование игрового приложения и его компонентов	Разработчик
Разработка программного обеспечения	8	Разработка игрового приложения	Разработчик
Тестирование	9	Модульное тестирование	Разработчик
	10	Интеграционное тестирование	Разработчик
Оформление отчета по ВКР	11	Оформление ВКР	Руководитель, Разработчик

3.2.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, который зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ожі}$ используется следующая формула:

$$t_{ожі} = \frac{3t_{\min i} + 2t_{\max i}}{5}, \quad (3)$$

где $t_{ожі}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{\min i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

$t_{\max i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ по нескольким исполнителями.

$$T_{pi} = \frac{t_{ожі}}{Ч_i}, \quad (4)$$

где T_{pi} – продолжительность одной работы, раб.дн.;

$t_{ожі}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.;

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

3.2.3 Разработка графика проведения научного исследования

Наиболее удобным и наглядным представлением проведения научных работ является построение ленточного графика в форме диаграммы Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для удобства построение графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (5)$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})}, \quad (6)$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

Расчет коэффициента календарности:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})} = \frac{365}{365 - 118} = 1,48 \quad (7)$$

Таблица 9 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ						Длительность работ в рабочих днях	Длительность работ в календарных днях
	Тmin, чел-дни		Тmax, чел-дни		Тож, чел-дни			
	Рук.	Разр.	Рук.	Разр.	Рук.	Разр.		
Составление и утверждение темы бакалаврской работы	1	5	5	10	2,6	7	7	10,3
Анализ предметной области и аналогов	3	7	7	15	4,6	10,2	11	16,3
Постановка требований к программному обеспечению	3	7	7	15	4,6	10,2	11	16,3
Подбор и изучение	0	20	0	35	0	26	26	38,4

материалов по теме								
Календарное планирование работ	3	5	3	5	3	5	5	7,4

Продолжение таблицы 9

Название работы	Трудоёмкость работ						Длительнос ть работ в рабочих днях	Длительнос ть работ в календарны х днях
	Тmin, чел- дни		Тmax, чел- дни		Тож, чел- дни			
	Рук.	Разр.	Рук.	Разр.	Рук.	Разр.		
Выбор средств и инструментов проектировани я и разработки	0	5	0	5	0	5	5	7,4
Проектирован ие игрового приложения и его компонентов	0	20	0	35	0	26	26	38,4
Разработка игрового приложения	0	40	0	70	0	52	52	76,8
Модульное тестирование	0	5	0	7	0	5,8	6	8,9
Интеграционн ое тестирование	0	7	0	14	0	9,8	10	14,8
Оформление ВКР	15	25	25	35	19	29	29	42,9
Итого	25	146	47	246	33,8	186	188	277,9

Для визуализации продолжительности этапов разработки была построена диаграмма Ганта, представленная в Таблице 10.

Таблица 10 – Диаграмма Ганта

Название работы	Исполнители	Продолжительность выполнения работ, по номеру месяца									
		1	2	3	4	5	6	7	8	9	10
Составление и утверждение темы бакалаврской работы	Руководитель	■									
	Разработчик	■									
Анализ предметной области и аналогов	Руководитель	■									
	Разработчик	■									
Постановка требований к программному обеспечению	Руководитель		■								
	Разработчик		■								

3.3.2 Расчет амортизации специального оборудования

В специальное оборудование входят оборудование для рабочего места и ПО, необходимое для реализации программного обеспечения. Затраты на специальное оборудование приведены в таблице 11.

Таблица 11 – Затраты на специальное оборудование

Наименование оборудования	Количество единиц оборудования	Цена за 1 ед. оборудования	Затраты, руб.
Персональный компьютер	2	80000	160000
Монитор	2	10000	20000
Компьютерная мышь	2	2000	4000
Клавиатура	2	5000	10000
Итого			194000

Общие затраты на оборудование составили 194 000 руб. Срок полезного использования офисных машин составляет от 2 до 3 лет. Для вычисления амортизации данный срок можно принять за 3 года.

Норма амортизации вычисляется по следующей формуле:

$$A_n = \frac{100\%}{3} = 33,33\% \quad (8)$$

Годовые амортизационные вычисления составляют:

$$A_g = S * \frac{A_n}{100\%} = 194000 * 0,33 = 64020 \text{ рублей} \quad (9)$$

Ежемесячные амортизационные вычисления составляют:

$$A_m = \frac{A_g}{12} = \frac{64020}{12} = 5335 \text{ рублей} \quad (10)$$

3.3.3 Основная заработная плата исполнителей темы

Основная зарплата – это та часть зарплата, которая выплачивается работнику гарантированно, не зависимо от результатов труда. В данном разделе рассчитывается заработная плата программиста и руководителя, помимо этого необходимо рассчитать расходы по заработной плате, определяемые трудоемкостью проекта и действующей системой оклада.

Основная заработная плата работника рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_p \quad (11)$$

где $Z_{\text{осн}}$ – основная заработная плата одного работника;

T_p – продолжительность работ, выполняемых работником, раб. дн.;

$Z_{дн}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{дн} = \frac{Z_m \cdot M}{F_d} \quad (12)$$

где Z_m – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года:

при отпуске в 24 раб. дня $M = 11,2$ месяца, 5–дневная неделя;

при отпуске в 48 раб. дней $M = 10,4$ месяца, 6–дневная неделя;

F_d – действительный годовой фонд рабочего времени научно–технического персонала, раб. дн.

Расчёт баланса рабочего времени приведён в таблице 12.

Таблица 12 – Баланс рабочего времени (для 6-дневной недели)

Показатель рабочего времени	Руководитель	Разработчик
Календарные дни	365	365
Нерабочие дни (праздники/выходные)	118	118
Потери рабочего времени (отпуск/невыходы по болезни)	48	72
Действительный годовой фонд рабочего времени	199	175

Месячный должностной оклад работника (руководителя):

$$Z_m = Z_{тс} \cdot (1 + k_{пр} + k_d) \cdot k_p \quad (13)$$

где $Z_{тс}$ – заработная плата по тарифной ставке, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3;

k_d – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5;

k_p – районный коэффициент, равный 1,3 (для Томска).

Для предприятий, не относящихся к бюджетной сфере, тарифная заработная плата (оклад) рассчитывается по тарифной сетке, принятой на данном предприятии. Расчет основной заработной платы представлен в таблице 13.

Таблица 13 – Расчет основной заработной платы

Исполнитель	$Z_{тс}$, руб.	$k_{пр}$	k_d	k_p	Z_m , руб.	F_d	M , мес	$Z_{дн}$, руб.	T_p , раб. дн.	$Z_{осн}$, руб.
Руководитель	40000	0,	0,4	1,3	88400	199	5	2221	41	91065
Разработчик	20000	0,3	0,2	1,3	39000	175	8	1783	184	328046

Общий размер основной заработной платы составил 419111 рублей.

3.3.4 Дополнительная заработная плата

Дополнительная заработная плата учитывает величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы рассчитывается по формуле:

$$З_{\text{доп}} = k_{\text{доп}} \cdot З_{\text{осн}}, \quad (14)$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы, принятый на стадии проектирования за 0,15.

Результат расчёта дополнительной заработной платы работников приведён в таблице 14.

Таблица 14 – Расчет дополнительной заработной платы

Исполнитель	З _{осн} , руб.	k _{доп}	З _{доп} , руб.
Руководитель	91065	0,15	13660
Разработчик	328046	0,15	49207

Общий размер дополнительной заработной платы составил 62867 рублей.

3.3.5 Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Расчет произведен в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ. Так как предстоящий проект является частью сферы информационных технологий, проводим дальнейший расчет с учетом письма ФНС России от 01.03.2022 N БС-4-11/2441:

- 6% на обязательное пенсионное страхование;
- 1,5% на обязательное социальное страхование;

– 0,1% на обязательное медицинское страхование.

Таким образом общий тариф составляет 7,6%. В таблице 15 представлен результат расчёта отчислений во внебюджетные фонды.

Таблица 15 – Расчет отчислений во внебюджетные фонды

Исполнитель	З _{осн} , руб.	З _{доп} , руб.	к _{внеб}	З _{отч} , руб.
Руководитель	91065	13660	0,076	7959,09
Разработчик	328046	49207	0,076	28671,22

3.3.6 Накладные расходы

Для учета дополнительных затрат, не относящихся к основному производству, используются накладные расходы. Величина данной статьи расходов определяется по формуле:

$$З_{\text{накл}} = (\sum \text{статей}) \cdot k_{\text{нр}} \quad (15)$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%. В таблице 16 представлен результат расчёта накладных расходов.

Таблица 16 – Расчет накладных расходов

Исполнитель	З _{осн} , руб.	З _{доп} , руб.	З _{отч} , руб.	Амортизация, руб.	к _{нр}	З _{отч} , руб.
Руководитель	91065	13660	7959,09	2667,5	0,16	18456,26
Разработчик	328046	49207	28671,22	2667,5	0,16	65374,68

3.3.7 Формирование бюджета затрат научно–исследовательского проекта

Рассчитанные ранее величины затрат научно-исследовательской работы формируют основу его бюджета. Сумма всех этих величин является нижней границей реальных затрат, необходимых для проведения работ. В таблице 17 представлен бюджет проекта.

Таблица 17 – Расчет бюджета затрат проекта

Наименование статьи	Сумма, руб.	Примечание
Материальные затраты	0	Пункт 4.3.1
Амортизация специального оборудования	64020	Пункт 4.3.2
Затраты на основную заработную плату	419111	Пункт 4.3.3
Затраты на дополнительную заработную плату	62867	Пункт 4.3.4
Отчисления во внебюджетные фонды	36631	Пункт 4.3.5
Накладные расходы	83831	Пункт 4.3.6
Общий бюджет	669460	

3.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности. Для этого учтем два варианта разработки текстур персонажей и окружения – с использованием уже готовых моделей (обозначение в таблицах 18-20 – И2) и создание новых моделей своими силами (обозначение в таблицах 18-20 – И1).

Интегральный показатель финансовой эффективности научного исследования определяется как:

$$I_{\text{фин.р}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}} \quad (16)$$

где $I_{\text{фин.р}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта.

Таблица 18 – Интегральные финансовые показатели

И1	И2
1	0,91

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum_{i=1}^n a_i \times b_i \quad (17)$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i^a, b_i^p – бальная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Для расчета показателя используются данные, представленные в таблице 19.

Таблица 19 – Сравнительная оценка характеристик вариантов исполнения

Критерии \ Объект исследования	Весовой коэффициент параметра	Исп.1	Исп.2
Простота реализации	0,15	3	5
Удобство в эксплуатации	0,15	5	4
Скорость работы	0,25	5	5
Простота применения	0,1	5	5
Масштабируемость	0,35	5	3
Итого	1	4,7	4,15

Интегральный показатель эффективности вариантов исполнения разработки ($I_{испi}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп1} = \frac{I_{р-исп1}}{I_{фин.р}^{исп1}} = \frac{4,70}{1} = 4,70; \quad (18)$$

$$I_{исп2} = \frac{I_{р-исп2}}{I_{фин.р}^{исп2}} = \frac{4,15}{0,91} = 4,56; \quad (19)$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта ($\mathcal{E}_{ср}$):

$$\mathcal{E}_{ср} = \frac{I_{исп2}}{I_{исп1}} \quad (20)$$

Таблица 20 – Сравнительная эффективность разработки

Показатели	Исп.1	Исп.2
Интегральный финансовый показатель разработки	1	0,91
Интегральный показатель ресурсоэффективности разработки	4,70	4,15
Интегральный показатель эффективности	4,70	4,56
Сравнительная эффективность вариантов исполнения	1	0,971

Сравнив значения интегральных показателей эффективности, можно сделать вывод, что реализация технологии в первом исполнении является более эффективным вариантом решения задачи, поставленной в данной работе с позиции финансовой и ресурсной эффективности. Стоит отметить, что с точки зрения финансового показателя разработки данное решение

проигрывает альтернативному, однако, его эффективность компенсируется более высокими качествами и возможностями, что отражается в более высоком показателе ресурсоэффективности.

3.5 Вывод по главе

В разделе исследовательской работы, посвященной финансовому менеджменту, ресурсоэффективности и ресурсосбережению, была дана оценка коммерческого потенциала разработки, спланирован график работ, сформирован бюджет затрат и определена эффективность исследования. Также проведен SWOT-анализ, составлены выводы на основе возможностей, угроз, сильных и слабых сторон проекта.

Кроме того, выполнено планирование научно-исследовательских работ по проекту. При планировании графика работ был составлен список задач, для каждой из которых определены исполнители и продолжительность. График работ визуализирован в виде диаграммы Ганта. Общая длительность проведения работ по проекту ориентировочно составляет 188 рабочих дней. С учётом продолжительности работ сформирован бюджет затрат научного исследования, размер которого 669460 руб.

Сравнение интегральных показателей эффективности вариантов исполнения показало, что наиболее выгодным с точки зрения ресурсоэффективности является 1 вариант исполнения, который и был реализован.

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Обучающемуся:

Группа 8K92		ФИО Лутошкин Степан Евгеньевич	
Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Проектирование и разработка кроссплатформенного игрового приложения на Unity для персональных компьютеров в жанре 2D платформер

Исходные данные к разделу «Социальная ответственность»:

Введение

- Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения.
- Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации

Объект исследования: игровое приложение на Unity для персональных компьютеров

Область применения: цифровые развлечения

Рабочая зона: офис

Размеры помещения: 6*3 м.

Количество и наименование оборудования рабочей зоны: 2, персональный компьютер, настольная лампа

Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне: проектирование, разработка и тестирование игрового приложения

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения

- специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;
- организационные мероприятия при компоновке рабочей зоны.

Трудовой кодекс РФ;
СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания
ТК РФ Статья 91. Понятие рабочего времени. Нормальная продолжительность рабочего времени;
ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя.

2. Производственная безопасность при разработке проектного решения:

- Анализ выявленных вредных и опасных производственных факторов

Вредные факторы:

1. Отсутствие или недостаток необходимого искусственного освещения;
2. Повышенный уровень шума;
3. Нагрузка на зрительный аппарат;
4. Отклонение показателей микроклимата;
5. Монотонность труда, вызывающая монотонию;
6. Длительность сосредоточенного наблюдения.

Опасные факторы:

Факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает рабочий.

Требуемые средства коллективной защиты от выявленных факторов: системы естественного освещения, приборы искусственного освещения, изоляционные средства, предохранительные устройства.

3. Экологическая безопасность при разработке проектного решения

Воздействие на селитебную зону не выявлено
Воздействие на литосферу не выявлено

	Воздействие на гидросферу при неверном способе утилизации рабочей техники Воздействие на атмосферу при неверном способе утилизации рабочей техники
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	Возможные ЧС: – Пожар; – Землетрясение; – Пандемия. Наиболее типичная ЧС: – Пожар.

Дата выдачи задания для раздела по линейному графику	01.03.2023 г.
---	----------------------

Задание выдал консультант по разделу «Социальная ответственность»:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
8K92	Лутошкин Степан Евгеньевич		

4. Социальная ответственность

Введение

Выпускная квалификационная работа представляет собой проектирование и разработку кроссплатформенного игрового приложения на Unity для персональных компьютеров в жанре 2D платформер. Область применения приложения – цифровые развлечения.

Данное мобильное приложение разрабатывалось в офисе. Помещение размерами 6*3м. с рабочим местом, оборудованным персональным компьютером и настольной лампой.

В данном разделе проведен анализ вредных и опасных факторов труда, определен комплекс мер организационного, правового, технического и режимного характера, который должен способствовать снижению возможности возникновения негативных последствий работы разработчика.

4.1 Правовые и организационные вопросы обеспечения безопасности

4.1.1 Специальные правовые нормы трудового законодательства

Законодательный акт "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 19.12.2022) [24] служит основным инструментом регулирования трудовых отношений между работником и работодателем. Он обеспечивает решение вопросов, включая организацию труда, управление рабочим процессом, заработную плату, трудовые споры и многое другое. Важнейшие к обязательному соблюдению пункты данного закона представлены ниже:

- нормальная продолжительность рабочего времени не может превышать 40 часов в неделю;
- работодатель обязан обеспечивать безопасность и условия труда, соответствующие государственным нормативным требованиям охраны труда;

- список обязанностей, режим работы и размер заработной платы должны быть зафиксированы в трудовом договоре;
- в течение рабочего дня работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут;
- всем работникам предоставляются выходные дни (еженедельный непрерывный отдых).

4.1.2 Основные эргономические требования к правильному расположению и компоновке рабочей зоны

Рабочее место должно быть организовано с учетом требований ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования» [25].

При организации работы с персональным компьютером, согласно указанному выше требованию, должны быть соблюдены следующие условия:

- конструкция рабочей мебели должна обеспечивать возможность индивидуальной регулировки соответственно росту пользователя и создавать удобную позу для работы;
- высота рабочего стола с клавиатурой должна составлять 680 - 800 мм над уровнем стола;
- экран монитора должен находиться от глаз пользователя на расстоянии 600-700 мм.;
- при работе двумя руками органы управления размещают с таким расчетом, чтобы не было перекрещивания рук.

При выполнении выпускной квалификационной работы правовых и организационных нарушений по указанным требованиям не было выявлено, рабочее место было оборудовано согласно всем нормам и правилам.

4.2 Производственная безопасность

В данном подразделе приведен анализ вредных и опасных факторов, которые воздействуют на разработчиков программного обеспечения на своих рабочих местах.

Согласно ГОСТ 12.0.003–2015 [26], на работника в течение рабочего дня могут воздействовать вредные и опасные факторы, каждый из которых влияет на производительность, работоспособность и физическое состояние.

Таблица 21 – Возможные опасные и вредные производственные факторы на рабочем месте за персональным компьютером

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Монотонность труда	Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 19.12.2022)
Повышенный уровень шума	ГОСТ 12.1.003–2014 ССБТ. Шум. Общие требования безопасности
Отсутствие или недостаток необходимого искусственного освещения	СП 52.13330.2016 Естественное и искусственное освещение. СП 2.2.3670–20 Санитарно-эпидемиологические требования к условиям труда СанПиН 1.2.3685–21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания
Нагрузка на зрительный аппарат	СанПиН 1.2.3685–21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания
Отклонение показателей микроклимата	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»
Длительность сосредоточенного наблюдения	Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 19.12.2022)
Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий	ГОСТ 12.1.019–2017 Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты

4.2.1 Монотонность труда

Работа программиста тесно связана с малоподвижной, однотипной деятельностью, что является показателем монотонного труда. Этот

негативный фактор играет важную роль в эмоциональном состоянии. При большом количестве монотонной работы, у сотрудника могут наблюдаться следующие негативные последствия:

- бессонница;
- депрессия;
- сонливость;
- повышенная утомляемость.

Для снижения уровня монотонности работы разработчика программного обеспечения следует принимать следующие меры:

- во время рабочего дня делать регулярные перерывы;
- вне рабочего времени стоит проводить время на свежем воздухе;
- стараться организовывать свое рабочее время таким образом, чтобы не заниматься одной задачей на протяжении длительного времени.

4.2.2 Повышенный уровень шума

Шум на рабочем месте программиста может исходить от вентиляторов, которые охлаждают помещение, а также от шума, доносящегося извне через окно. Этот шум представляет угрозу для здоровья человека и негативно влияет на рабочую обстановку. При длительном воздействии шума сотрудники могут испытывать раздражительность, головные боли, головокружение, ухудшение памяти, повышенную утомляемость, потерю аппетита и проблемы со сном. Такие нарушения организма могут привести к негативным изменениям в психическом состоянии человека, вплоть до развития стресса. Шум также снижает концентрацию внимания, нарушает физиологические функции и вызывает усталость из-за необходимости постоянной концентрации. В результате снижается работоспособность и производительность работника, а также ухудшается качество и безопасность его труда.

Уровень шума на рабочем месте программиста в течение номинального 8-часового рабочего дня не должен превышать нормативный эквивалентный уровень звука, установленный в СанПиН 1.2.3685-21 и равный 80 дБА [27-28].

Для уменьшения воздействий шума можно использовать отделку стен и потолка помещений, где расположены компьютеры, звукопоглощающими материалами, а также устанавливать оборудование, производящее минимальный шум.

4.2.3 Отсутствие или недостаток необходимого искусственного освещения

Недостаток освещения рабочей зоны или его отсутствие негативно сказываются на зрительной функции и влияют на психическое и эмоциональное состояние работников, вызывая усталость нервной системы.

Согласно СП 52.13330.2016, система общего освещения должна обеспечивать не менее 200 лк освещенности, с коэффициентом пульсации не превышающим 10% [29].

Согласно СП 2.2.3670–20 [30], компьютеры должны быть размещены таким образом, чтобы освещенность не превышала гигиенические нормативы, установленные в СанПиН 1.2.3685–21, представленные в таблице 22.

Таблица 22 – Требования к освещению на рабочих местах

Показатель	Норма
Освещенность поверхности рабочего стола	500 лк
Освещенность экрана монитора	Не более 200 лк
Коэффициент пульсации освещенности	Не более 5%

Для снижения влияния фактора недостаточной освещенности необходимо, чтобы уровень естественного освещения рабочего места и яркость дисплея персонального компьютера были приблизительно одинаковыми, так как яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение, что приводит к их быстрой утомляемости. При недостаточной освещенности помещения может помочь расширение

оконного проема помещения и установка качественных источников искусственного освещения.

4.2.4 Нагрузка на зрительный аппарат

Работа за компьютером в течение продолжительного времени оказывает значительное напряжение на глаза. В отличие от печатных материалов, где изображение представлено непрерывными линиями, на мониторе мы видим изображение, состоящее из отдельных точек, которые светятся и мерцают. В результате пользователь сталкивается с негативными последствиями, такими как ухудшение зрения, слезящиеся глаза, головная боль, усталость, двоение и искажение изображения.

Чтобы уменьшить вредное воздействие на зрительный аппарат, необходимо принять несколько мер. Во-первых, важно делать регулярные перерывы в работе за компьютером. Во время этих перерывов рекомендуется отвлечься от экрана и сфокусировать взгляд на дальних объектах, чтобы снять напряжение с глаз.

Кроме того, правильный выбор цветовой схемы, шрифтов и компоновки окон в используемых приложениях, а также правильная ориентация дисплея монитора также имеют большое значение для заботы о глазах. Необходимо выбирать цвета, которые не вызывают излишнего напряжения глаз, предпочтительно использовать тёмный фон и светлый текст. Шрифты должны быть достаточно крупными и хорошо читаемыми, чтобы избежать необходимости напрягать глаза для чтения.

Применение этих рекомендаций поможет снизить воздействие неблагоприятных факторов на зрение при работе за компьютером и поддерживать его в хорошем состоянии.

4.2.5 Отклонение показателей микроклимата

Микроклимат – это метеорологические условия внутренней среды, определяемые действующими на организм человека сочетаниями температуры, относительно влажности и скорости движения воздуха и др. Причиной отклонения показателей от установленных норм зачастую является

некорректная работа системы вентиляции офисного помещения, которая одновременно влияет и на температуру окружающего воздуха в помещении, на влажность, и на скорость его движения.

Данный фактор можно отнести к вредным, так как его воздействие на организм человека проявляется в виде быстрой утомляемости, потери работоспособности сотрудника.

Нормативные показатели микроклимата регламентируются СанПиН 1.2.3685-21. Санитарные нормы устанавливают оптимальные и допустимые значения показателей в рабочей зоне, что позволяет создавать благоприятные условия работы, соответствующие физиологическим потребностям организма человека.

Работа, выполняемая разработчиком игрового приложения, относится к категории Ia, так как она является мало подвижной и мало интенсивной, выполняется в положении сидя с минимальными физическими напряжениями. В таблице 23 представлены допустимые величины параметров микроклимата на рабочих местах для данной категории.

Таблица 23 – Допустимые величины параметров микроклимата

Период года	Температура воздуха, С°		Температура поверхностей, С°	Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
	диапазон ниже оптимальных величин	диапазон выше оптимальных величин			для диапазона температур воздуха ниже оптимальных величин, не более	для диапазона температур воздуха выше оптимальных величин, не более
Холодный	20,0-21,9	24,1-25,0	19,0-26,0	15-75	0,1	0,1
Теплый	21,0-22,9	25,1-28,0	20,0-29,0	15-75	0,1	0,2

Для поддержания микроклимата в помещении необходимо придерживаться следующих правил:

- оборудовать рабочее помещение кондиционерами, увлажнителями воздуха и приборами измерения температура;
- регулировать микроклимат помещения с помощью установленных приборов и/или с помощью окон;
- регулярно проводить влажную уборку помещения;
- рационально размещать рабочие места.

4.2.6 Длительность сосредоточенного наблюдения

Длительное сосредоточенное наблюдение, особенно при работе за компьютером, может представлять собой вредный производственный фактор, который способствует развитию различных заболеваний и проблем со здоровьем. При этом у сотрудника могут наблюдаться усталости и боли в глазах, синдром «сухого глаза», ухудшение зрения, развитие близорукости.

Также при длительной работе за компьютером могут возникать проблемы сердечно-сосудистой системы, мускулоскелетные или проблем с возникновением лишнего веса.

При работе важно учесть эти факторы и предпринять меры для снижения риска. К таким мерам могут относиться регулярные перерывы для отдыха и движения, упражнения для глаз, а также правильная организация рабочего места и рабочих процессов.

4.2.7 Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий

Работа программиста происходит в непосредственной близости от электрических сетей и приборов, поэтому работник должен с осторожностью обращаться с электропроводкой и компьютером, а также помнить об опасности поражения электрическим током. Величина травмы зависит от силы тока, проходящего через тело, и напряжения при соприкосновении. Электрический ток, проходящий через тело человека, может вызвать остановку сердца или привести к летальному исходу.

Факторами, повышающими вероятность удара током, являются:

- повышенная влажность (более 75%);
- высокая температура воздуха и поверхностей (более 35 °С);
- наличие токопроводящей пыли;
- неверная проектировка рабочего места;
- отсутствие защитных конструкций для проводов;
- наличие посторонних предметов на электроприборах.

Требования к защите от поражения электрическим током определены в ГОСТ 12.1.019–2017 [31]. Согласно правилам устройства электроустановок, рабочее помещение программиста не относится к помещениям повышенной опасности [32].

Безопасность сотрудников должна обеспечиваться выполнением следующих мер защиты [33]:

- соблюдение соответствующих расстояний до токоведущих частей или закрытие их ограждениями;
- применение блокировки аппаратов и ограждающих устройств для предотвращения ошибочного доступа к токоведущим частям;
- применение предупреждающих сигнализаций, надписей и плакатов;
- применение устройств для снижения напряженности электрических и магнитных полей до допустимых значений;
- использование индивидуальных электрозащитных средств.

4.3 Экологическая безопасность

При разработке игрового приложения в основном использовался только персональный компьютер и труд программиста. Персональные компьютеры не несут значимый вред окружающей среде в целом, однако, он есть.

Комплектующие для компьютера токсичны, и их неправильная утилизация может нанести серьезный ущерб экологии. Сжигание деталей ПЭВМ является источником загрязнения атмосферы планеты.

Для минимизирования воздействия на окружающую среду, необходимо соблюдать специальные процедуры утилизации компьютерной и офисной техники. Переработка отходов электронной промышленности регулируется ГОСТ 30775–2001 [34].

Для работы персонального компьютера требуется электроэнергия, увеличение потребления которой может привести к созданию дополнительных тепловых электростанций, что может оказать негативное воздействие на гидросферу и атмосферу за счёт выбросов токсичных веществ, а также заборов и сбросов воды разной температуры [35].

4.4 Безопасность в чрезвычайных ситуациях

Разработка игрового приложения проходила в офисе. Наиболее вероятная чрезвычайная ситуация, которая может возникнуть в таких условиях – пожар.

Наиболее частыми причинами возникновения пожара можно назвать короткое замыкание, перегрузку сетей, с последующим нагревом токоведущих частей и неисправность оборудования.

Согласно ГОСТ 12.1.004–91, при работе с компьютером необходимо соблюдать следующие нормы пожарной безопасности [36]:

- запрещается подключать к сети количество потребителей, превышающих допустимую нагрузку;
- работа за компьютером должна проходить только при исправном оборудовании и кабелях;
- должна происходить регулярная проверка технического состояния оборудования, особенно кабелей;
- необходимо поддерживать чистоту в рабочей зоне;
- разрешается курить только в специально отведенных местах.

Рабочее помещение, где работает программист, относится к категории В согласно стандарту пожарной безопасности. Это связано с наличием твёрдых горючих и трудногорючих материалов. По классификации пожаров,

установленной в Федеральном законе от 22 июля 2008 года № 123-ФЗ, возможный пожар в данном помещении относится к классу А в связи с наличием горючих твердых веществ, а также к классу Е из-за присутствия электроустановок под напряжением [37].

При появлении пожара, любой, увидевший пожар должен:

- незамедлительно заявить о происшествии в пожарную службу;
- принять меры по эвакуации людей согласно плану эвакуации;
- отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

4.5 Вывод по главе

В результате проведенной работы были изучены правовые и организационные аспекты обеспечения безопасности в рабочем окружении при разработке игрового приложения. В данной главе были определены потенциально опасные и вредные факторы, связанные с работой программиста, которые соответствуют установленным нормам.

Согласно правилам устройства электроустановок, рабочие места программистов относятся к зонам без повышенной опасности от поражения электрическим током. Рабочее помещение оборудовано с учетом требований по электро- и пожарной безопасности. Работа программистов относится к категории тяжести труда Ia и требует 1 группы по электробезопасности.

Относительно пожарной безопасности, рабочее помещение программиста классифицируется как пожароопасное категории В, а возможный пожар может быть классифицирован как класс А или Е.

С учетом негативного воздействия на окружающую среду, рабочее помещение инженера-программиста относится к объектам IV категории.

Таким образом, нарушений в организации рабочего процесса при выполнении проекта выявлено не было, и все необходимые требования и нормы безопасности были соблюдены.

ЗАКЛЮЧЕНИЕ

В результате работы было спроектировано и реализовано игровое приложение в жанре 2D платформер, которое включало игровые механики для 5 игровых уровней, персонажа игрока и пользовательский интерфейс, системы для обработки ввода пользователя, сохранения/загрузки его прогресса и для управления миссиями, квестами и событиями. Также были реализованы миниигра на примере головоломки с лучами света и NPC «Паук».

На начальном этапе были определены задачи для реализации игры и проведен аналитический обзор рынка 2D платформеров.

На следующем этапе был составлен концепт-документ, который включал выявленные функциональные требования к системе. В процессе проектирования были построены диаграммы вариантов использования и определены подходящие для реализации приложения инструменты.

На основе этапа проектирования были реализованы все компоненты системы. Стоит отметить их гибкую архитектуру, которая позволяет добавлять новые формы и способности игроку, игровые задания и миссии.

Игровое приложение разрабатывалась в команде, что дало дополнительный опыт работы в команде при разработке. Личным вкладом автора работы являются все вышеописанные механики и системы. В работе использовалась среда разработки JetBrains Rider и система управления версиями Plastic SCM [38-39].

На этапе выполнения задания по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» была дана оценка коммерческого потенциала разработки, спланирован график работ, сформирован бюджет затрат и определена эффективность разработки.

На этапе выполнения задания по разделу «Социальная ответственность» было определено, что нарушений в организации рабочего процесса при разработке игрового приложения выявлено не было, и все необходимые требования и нормы безопасности были соблюдены.

СПИСОК ИСТОЧНИКОВ

- 1) SteamDB: база данных Steam: официальный сайт игры. – URL: <https://steamdb.info>. – Текст: электронный.
- 2) Wikipedia: классификация компьютерных игр: сайт. – URL: https://ru.wikipedia.org/wiki/Классификация_компьютерных_игр. – Текст: электронный.
- 3) DTF: Тренды и тенденции игровой индустрии в 2023 году: сайт. – URL: <https://dtf.ru/u/249930-logrus-it/1603116-trendy-i-tendencii-igrovoy-industrii-v-2023-godu/>. – Текст: электронный.
- 4) Высшая школа бизнеса: игровая индустрия: геймдев: сайт. – URL: <https://hsbi.hse.ru/articles/igrovaya-industriya-geymdev/>. – Текст: электронный.
- 5) Tadviser: компьютерные и видеоигры (мировой рынок): сайт. – URL: [https://www.tadviser.ru/index.php/Статья:Компьютерные_и_видеоигры_\(мировой_рынок\)](https://www.tadviser.ru/index.php/Статья:Компьютерные_и_видеоигры_(мировой_рынок)). – Текст: электронный.
- 6) GameIsArt: классификация компьютерных игр: сайт. – URL: https://gamesisart.ru/game_class_all.html/. – Текст: электронный.
- 7) Call of Duty: Modern Warfare 2: официальный сайт игры. – URL: <https://www.callofduty.com/modernwarfare2/>. – Текст: электронный.
- 8) Sid Meier's Civilization V: официальный сайт игры. – URL: <https://civilization.com/civilization-5/>. – Текст: электронный.
- 9) Hollow Knight: официальный сайт игры. – URL: <https://www.hollowknight.com/>. – Текст: электронный.
- 10) Celeste: официальный сайт игры. – URL: <https://www.celestegame.com/>. – Текст: электронный.
- 11) Ori and the Blind Forest: официальный сайт игры. – URL: <https://www.oriandtheblindforest.com/blind-forest/>. – Текст: электронный.
- 12) Super Meat Boy: официальный сайт игры. – URL: <http://www.supermeatboy.com/>. – Текст: электронный.
- 13) Игровой движок Unreal Engine: сайт. – URL: <https://www.unrealengine.com/>. – Текст: электронный.

14) Unity: официальный сайт. – URL: <https://unity.com/>. – Текст: электронный.

15) Unity: форумы: сайт. – URL: <https://forum.unity.com/>. – Текст: электронный.

16) Игровой движок Construct: официальный сайт. – URL: <https://construct.net/>. – Текст: электронный.

17) Игровой движок GameMaker: официальный сайт. – URL: <https://gamemaker.io/>. – Текст: электронный.

18) Steam: официальный сайт. – URL: <https://store.steampowered.com/>. – Текст: электронный.

19) GOG: официальный сайт. – URL: <https://www.gog.com/>. – Текст: электронный.

20) Epic Games Store: официальный сайт. – URL: <https://www.epicgames.com/>. – Текст: электронный.

21) Confluence: пространство для командной работы: сайт. – URL: <https://www.atlassian.com/software/confluence>. – Текст: электронный.

22) Unity: официальная документация: сайт. – URL: <https://docs.unity3d.com/Manual/index.html>. – Текст: электронный.

23) Miro: платформа для совместной работы: сайт. – URL: <https://miro.com>. – Текст: электронный.

24) Российская федерация. Законы. Трудовой кодекс Российской Федерации: Федеральный закон № 197-ФЗ (ред. 01.03.2023): [принят Государственной думой 21 декабря 2001 года]. – Собрание законодательства РФ. – 07.01.2002. – №1 (ч.1).

25) ГОСТ 12.2.032-78. Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования: дата введения 1979-01-01. – URL: <https://docs.cntd.ru/document/1200003913>. – Текст: электронный.

26) ГОСТ 12.0.003-2015. Опасные и вредные производственные факторы. Классификация: дата введения 2017-03-01. – URL: <https://docs.cntd.ru/document/1200136071>. – Текст: электронный.

27) СанПин 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания: дата введения 2021-03-01. – URL: <https://docs.cntd.ru/document/573500115>.

28) ГОСТ 12.1.003-2014. Система стандартов безопасности труда. Шум. Общие требования безопасности: дата введения 2015-11-01. – URL: <https://docs.cntd.ru/document/1200118606>. – Текст: электронный.

29) СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*: дата введения 2017-05-08. – URL: <https://docs.cntd.ru/document/456054197>. – Текст: электронный.

30) СП 2.2.3670-20. Санитарно-эпидемиологические требования к условиям труда: дата введения 2021-01-01. – URL: <https://docs.cntd.ru/document/573230583>. – Текст: электронный.

31) ГОСТ 12.1.019-2017. Система стандартов безопасности труда. Электробезопасность. Общие требования и номенклатура видов защиты: дата введения 2019-01-01. – URL: <https://docs.cntd.ru/document/1200161238>. – Текст: электронный.

32) Правила устройства электроустановок. Глава 1.1 Общая часть: официальное издание: утверждены Министерством энергетики Российской Федерации от 08.07.02: введены в действие 01.01.03. – URL: <https://docs.cntd.ru/document/1200030216>. – Текст: электронный.

33) Правила по охране труда при эксплуатации электроустановок: официальное издание: утверждены Министерством труда и социальной защиты Российской Федерации от 15.12.20 с изменениями на 29.04.22: введены в действие 01.11.22. – URL: <https://docs.cntd.ru/document/573264184>. – Текст: электронный.

34) ГОСТ 30775-2001. Ресурсосбережение. Обращение с отходами. Классификация, идентификация и кодирование отходов. Основные положения: дата введения 2002-07-01. – URL: <https://docs.cntd.ru/document/1200028877>. – Текст: электронный.

35) Критерии отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II, III, IV категорий: официальное издание: утверждены постановлением Правительства Российской Федерации от 31.12.20: введены в действие 01.01.21. – URL: <https://docs.cntd.ru/document/573292854>. – Текст: электронный.

36) ГОСТ 12.1.004-91. Пожарная безопасность. Общие требования: дата введения 1992-07-01. – URL: <https://docs.cntd.ru/document/9051953>. – Текст: электронный.

37) Российская федерация. Законы. Технический регламент о требованиях пожарной безопасности: Федеральный закон № 123-ФЗ: [Принят Государственной Думой 4 июля 2008 года]. – URL: <https://docs.cntd.ru/document/902111644>. – Текст: электронный.

38) JetBrains Rider: среда разработки программного обеспечения: сайт. – URL: <https://www.jetbrains.com/rider>. – Текст: электронный.

39) Plastic SCM: система управления версиями: сайт. – URL: <https://www.plastic SCM.com>. – Текст: электронный.