

## Содержание

Введение .....	4
1 Сбор, анализ и формирование требований к программному продукту	5
1.1 Сбор требований.....	5
1.2 Анализ и формирование требований .....	6
2 Проектирование и разработка архитектуры программного продукта..	7
2.1 Построение диаграммы связей .....	7
2.2 Разработка сценария использования .....	9
2.3 Архитектура программного продукта .....	13
2.4 Прототипирование и дизайн программного продукта .....	15
3 Разработка программного продукта .....	18
3.1 Инструментальные и программные средства разработки .....	18
3.2 Календарный план разработки.....	19
4 Тестирование программного продукта .....	21
4.1 Выбор метода обеспечения качества.....	21
4.2 Тестирование программного продукта.....	21
5 Разработка документации на программный продукт .....	28
Заключение .....	29
Список литературы .....	30
Приложение А .....	31
Приложение Б.....	38
Приложение В .....	42
Приложение Г .....	43

					РКСИ.К22.09.02.03.МДК.03.01.000.00ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Разработка игрового приложения «Weaken» Пояснительная записка	Лит.	Лист	Листов
Разраб.		Иванов А.С.						
Провер.		Петрова М.Е.					3	44
Реценз.						ПОКС-44		
Н. Контр.								
Утверд.								



## Введение

Целью курсового проекта является разработка игрового приложения «Weaken».

Обучаясь в Ростовском-на-Дону колледже связи и информатики, я заинтересовался разработкой дизайна/мобильных игровых приложений на различных игровых движках. В качестве графического редактора/игрового движка я выбрал Aseprite/Unity, так как интерфейс интуитивно понятен и функционал прост в освоении. Проект предназначен для развлечения, отдыха и интересного времяпрепровождения.

В приоритете была разработка основного функционала и дизайна, а основной целью является реализация идей в виде игры.

					РКСИ.К22.09.02.03.МДК.03.01.000.00ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

# 1 Сбор, анализ и формирование требований к программному продукту

## 1.1 Сбор требований

На этапе сбора требований необходимо составить вопросы для заказчика и сформировать их в таблицу-опросник.

Таблица 1 – Интервью с заказчиком

Вопрос	Ответ
Назначение и цели программного продукта (ПП).	
Кто вы и чем занимаетесь ваш бизнес?	Разработка игровых приложений
Есть ли у компании действующий сайт?	Нет
На какую целевую аудиторию будет направлено игровое приложение?	Подростки, геймеры
Какие цвета Вы бы хотели видеть в игровых локациях? (нужное подчеркните)	<u>Темно-голубой</u> <u>Зелёный</u> <u>Оранжевый</u> <u>Белый</u> <u>Черно-серый</u> <u>Синий</u> <u>Чёрный</u>
Какая информация должна быть размещена на главной странице?	Новая игра, продолжить игру, настройки, выход
Какими возможностями будут обладать игроки?	Взаимодействия с игровым/не игровыми персонажами, взаимодействие с игровым окружением
Каковы сроки этого проекта?	До xx.06.2023
Планируется ли развитие этого проекта в будущем?	Нет
Есть ли системные требования к оборудованию?	Конкретных системных требований к оборудованию нет.
Кто из сотрудников будет иметь доступ к админ-панели?	Геймдизайнер, Программист
Какие данные должны редактироваться в админ панели?	Игровые механики, дизайн

Нужен ли сотрудникам разный уровень доступа?	Нет
Будет ли защищаться игровое приложение лицензионным соглашением?	Нет
Какими методами будет продвигаться игровое приложение в сети интернет?	С помощью игровых платформ

## 1.2 Анализ и формирование требований

Целевой аудиторией продукта являются люди покупательного возраста, интересующиеся играми и игровой индустрией.

Целями создания проекта является:

1. Привлечение, заинтересованных в данной сфере развлечений, пользователей;
2. Разработка отличительных игровых особенностей, таких как наличие внутриигровых персонажей с возможностью взаимодействия с ними:
  - Противостояние главного персонажа с не игровыми, враждебными персонажами;
  - Общение с мирными персонажами;
  - Взаимодействие с объектами окружающей среды;
  - Возможность управления движениями игрового персонажа с помощью клавиатуры;
  - Возможность использовать функцию атаки игрового персонажа с помощью компьютерной мыши;
  - Разнообразие дизайна игровых уровней и не игровых персонажей (NPC);
  - Возможность взаимодействия с не игровыми персонажами (NPC);
  - Возможность взаимодействия с объектами окружающей среды;
  - Разнообразие набора атакующих действий не игровых персонажей, которые являются боссами локаций: у каждого будет от двух до трех видов атак, которые придется избегать по-разному;
  - Наличие ключевых враждебных не игровых персонажей (NPC): каждый из них является боссом локаций, и после победы над ним даётся возможность пройти дальше.

## 2 Проектирование и разработка архитектуры программного продукта

### 2.1 Построение диаграммы связей

Диаграмма связей программного продукта представляет собой графическую схему взаимодействия объектов проектируемого ПП. Схема взаимодействия программных модулей показывает структуру программы, её составные части (модули), отражает связь между ними. Схема отражает не только подчиненность модулей, но и порядок их вызова или функционирования программы. Схема дополняется расшифровкой функций, выполняемых модулями.

Графическое изображение схемы взаимодействия модулей игрового приложения «Weaken» представлено на рисунках 1-3.

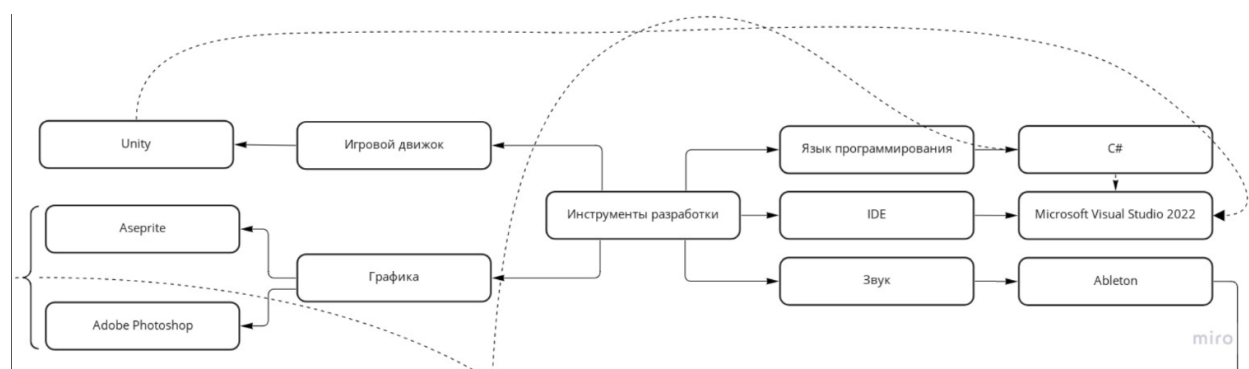


Рисунок 1 – Верхняя часть схемы Mind Map

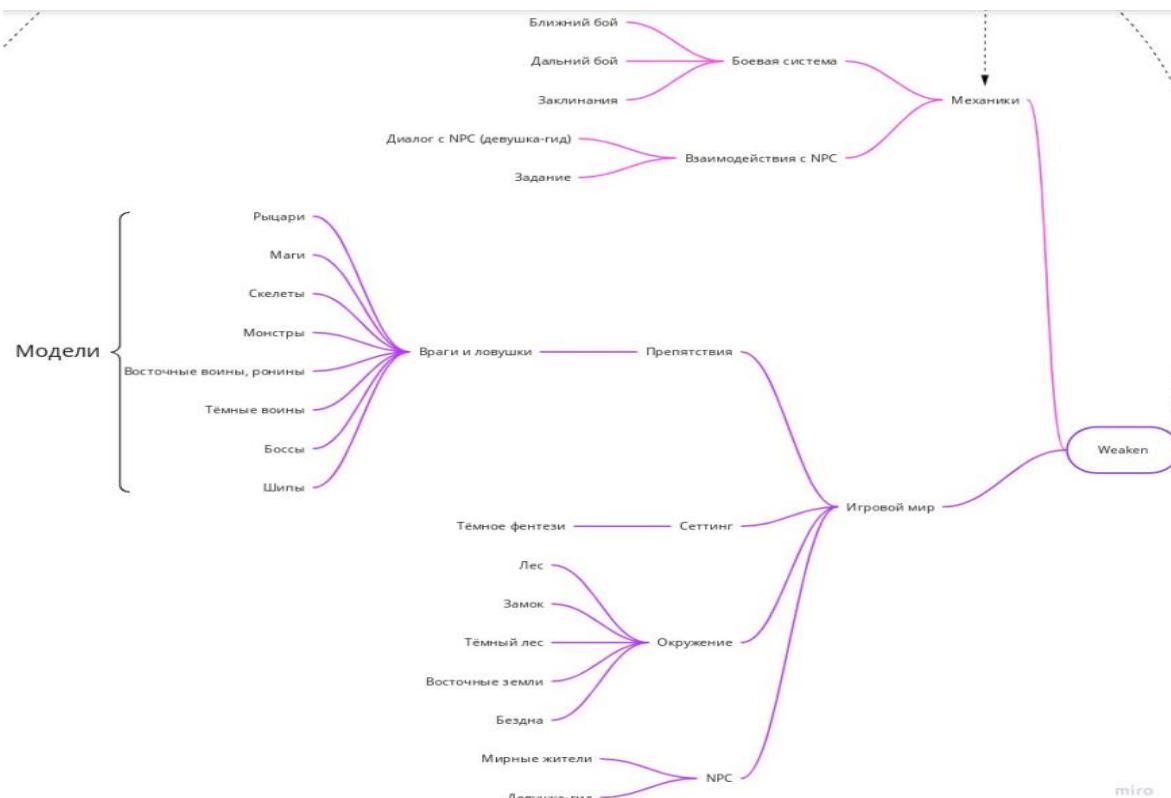


Рисунок 2 – Левая часть схемы Mind Map

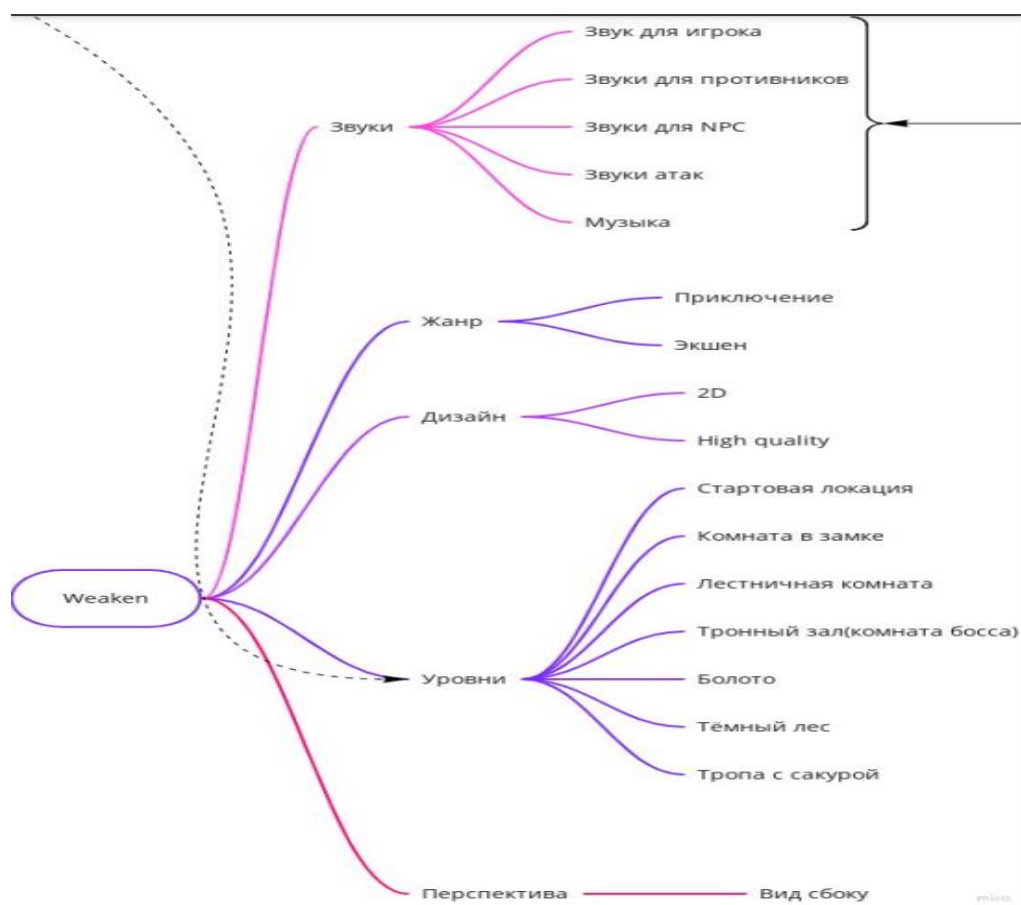


Рисунок 3 – Правая часть схемы Mind Map

## 2.2 Разработка сценария использования

Сценарий использования, проектируемого ПП можно разработать с помощью диаграмм вариантов использования (Use-case) и деятельности UML. Диаграмма Use-case разработана с целью определения функционала игрового приложения «Weaken» для его пользователей, и представлена на рисунках 4-5.

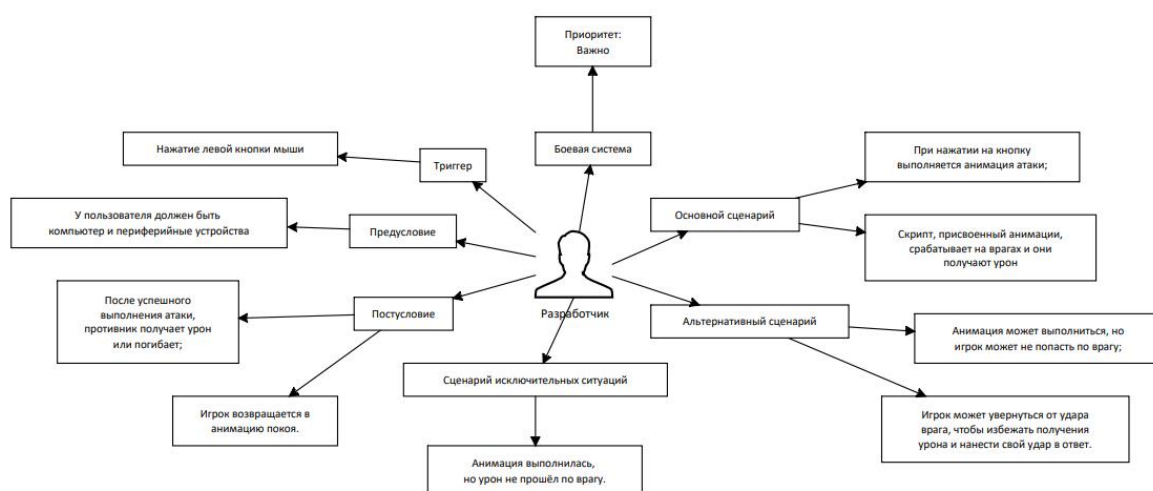


Рисунок 4 - Диаграмма вариантов использования со стороны разработчика, проектируемого ПП



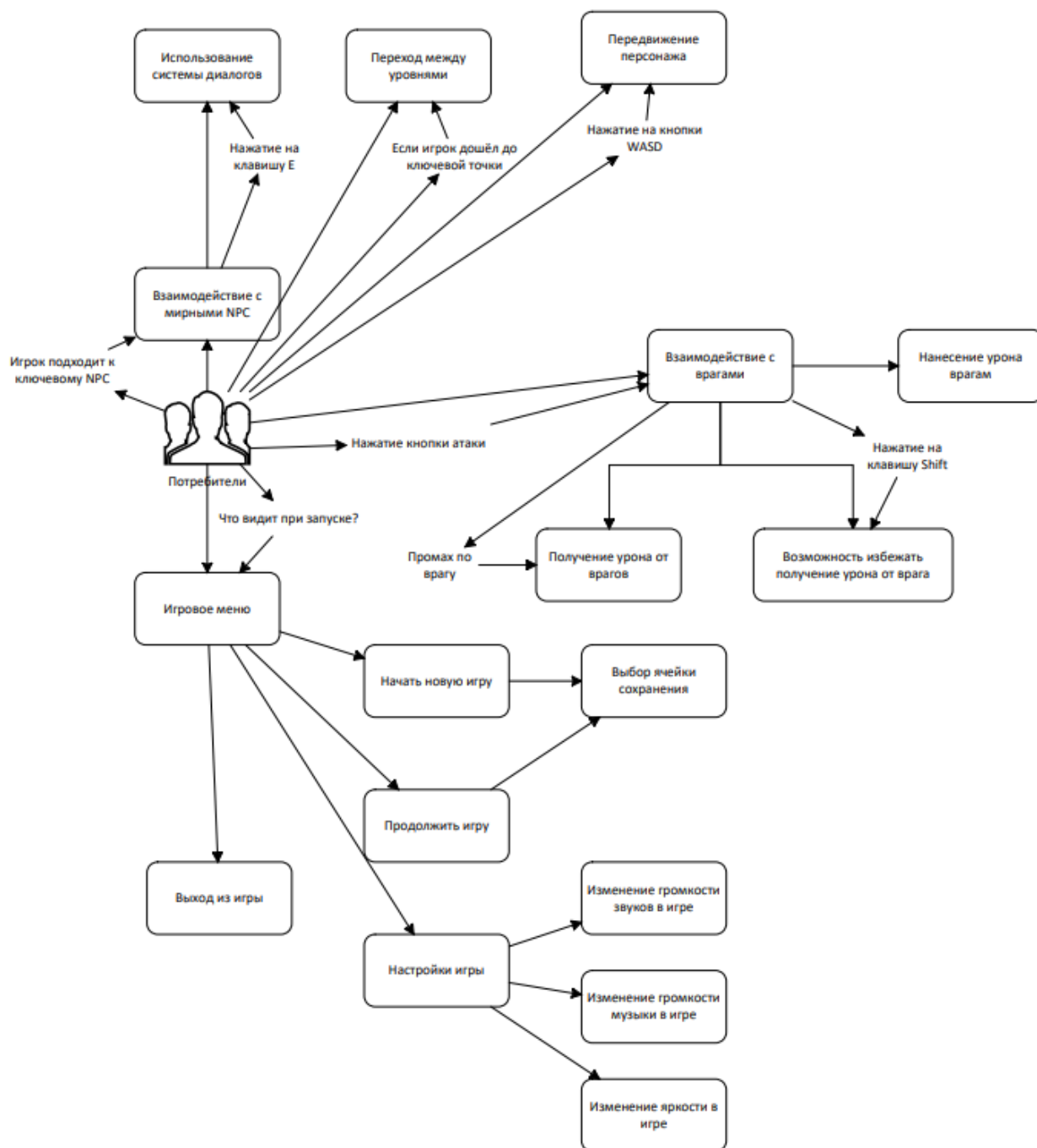


Рисунок 5 - Диаграмма вариантов использования со стороны пользователя

Диаграмма деятельности (активности) UML позволяет более детально визуализировать конкретный вариант использования. Эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой. Диаграмма деятельности для варианта использования (функции) «Редактирование справочника видеодисков» представлена на рисунке 6.

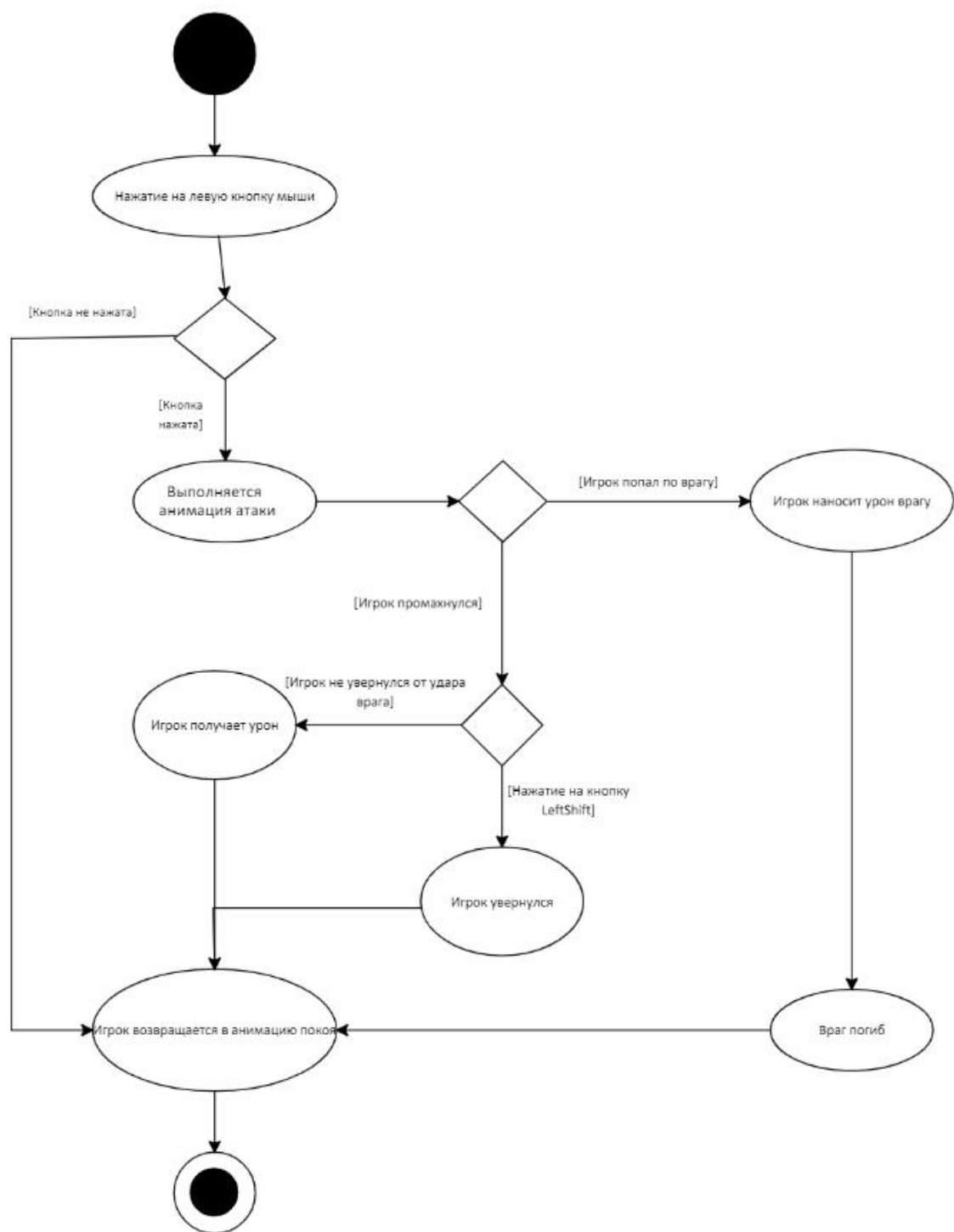


Рисунок 6 - Диаграмма деятельности для функций боевой системы

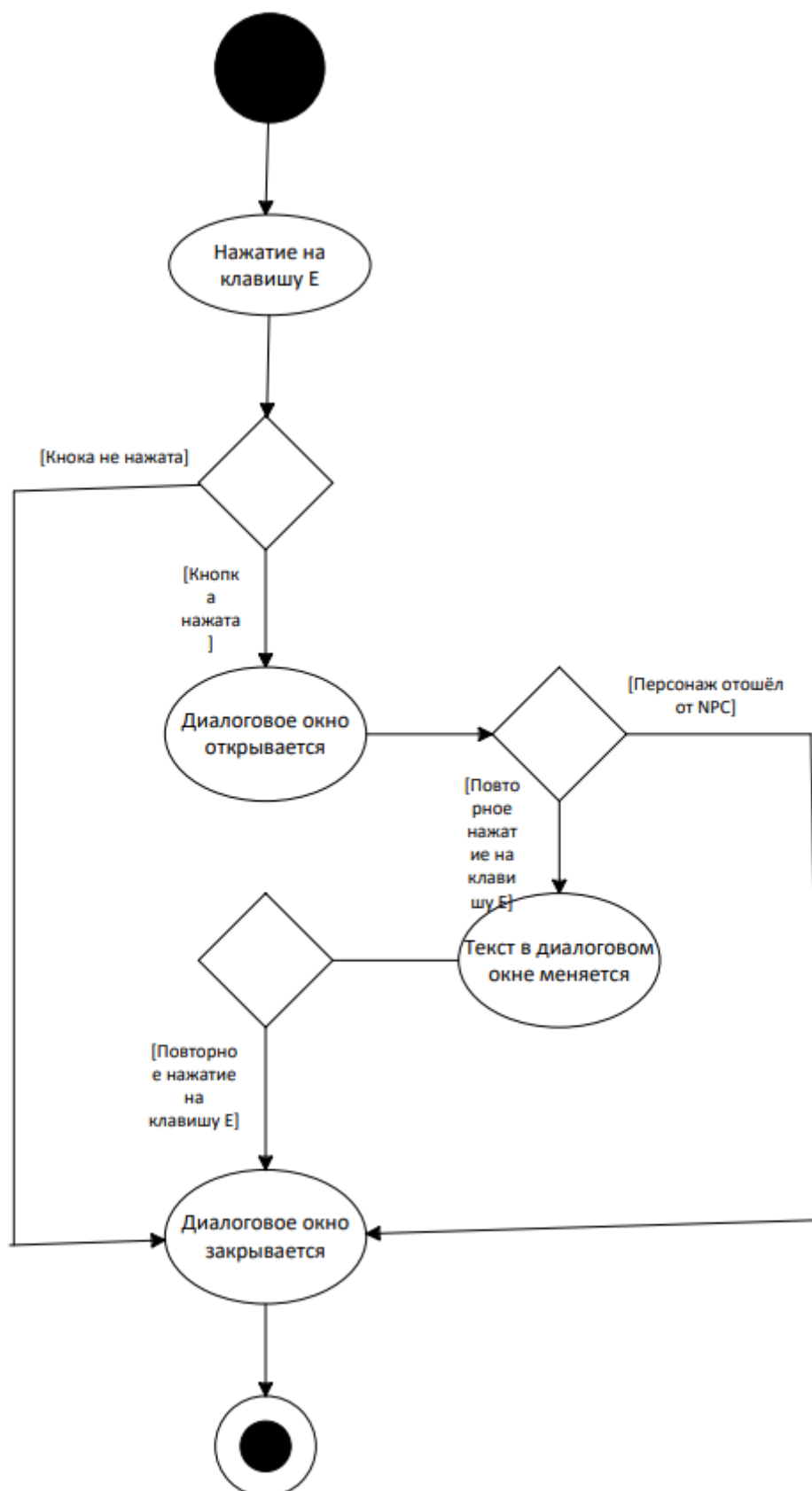


Рисунок 7 – Диаграмма деятельности для функций диалоговой системы

## 2.3 Архитектура программного продукта

При проектировании игрового приложения «Weaken» были использованы такие программы как Unity на языке программирования С# и графический редактор Aseprite, лаунчер будет устанавливаться на твердотельный накопитель в компьютере пользователя.

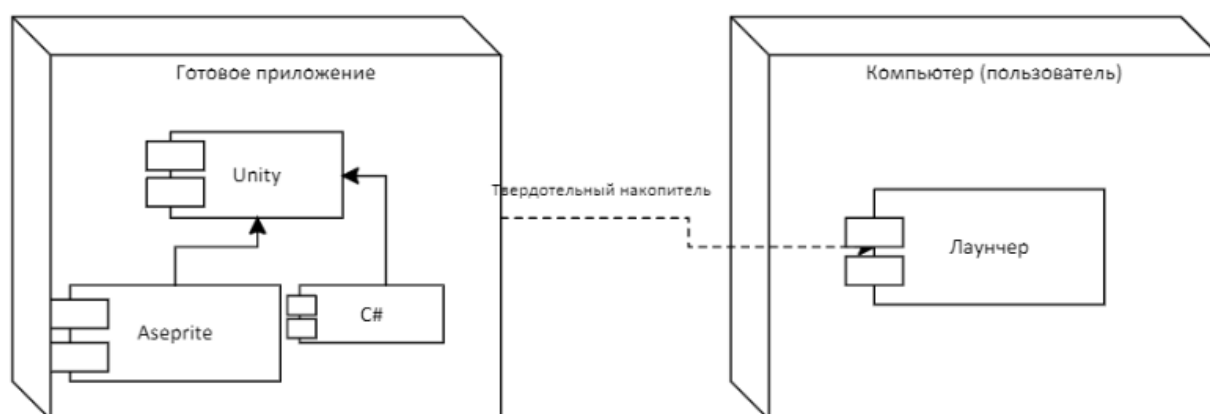


Рисунок 8 – Архитектура ПП

Почти всё современное программирование построено на принципах ООП и для проектирования внутренней структуры объектно-ориентированного приложения используют диаграмму классов (class diagram), предназначенную для представления программы в виде классов и связей между ними. Диаграмма классов игрового приложения «Weaken» представлена на рисунке 9.

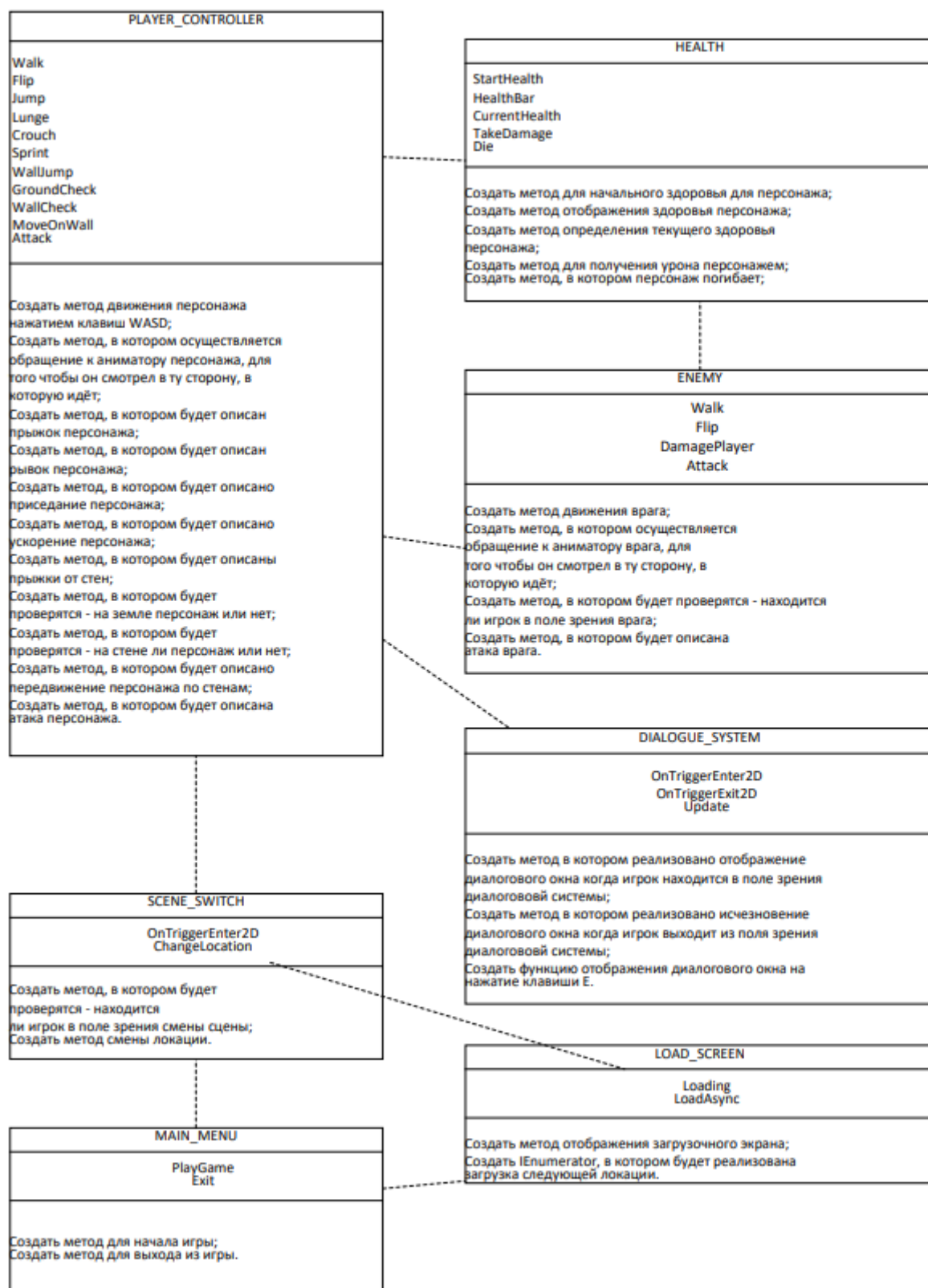


Рисунок 9 – Диаграмма классов проектируемого ПП

2.4 Прототипирование и дизайн программного продукта

Прототип — это набросок продукта, в котором заключены его внешний вид, логика работы и основная функциональность. Для презентации прототипа показывают, к чему и куда приводит взаимодействие будущего пользователя с элементами интерфейса.

Прототипы главного и внутриигрового меню игры «Weaken», разработанные с помощью сервиса Figma, представлены ниже на рисунках 10-16.



Рисунок 10 – окно главного меню после запуска игры

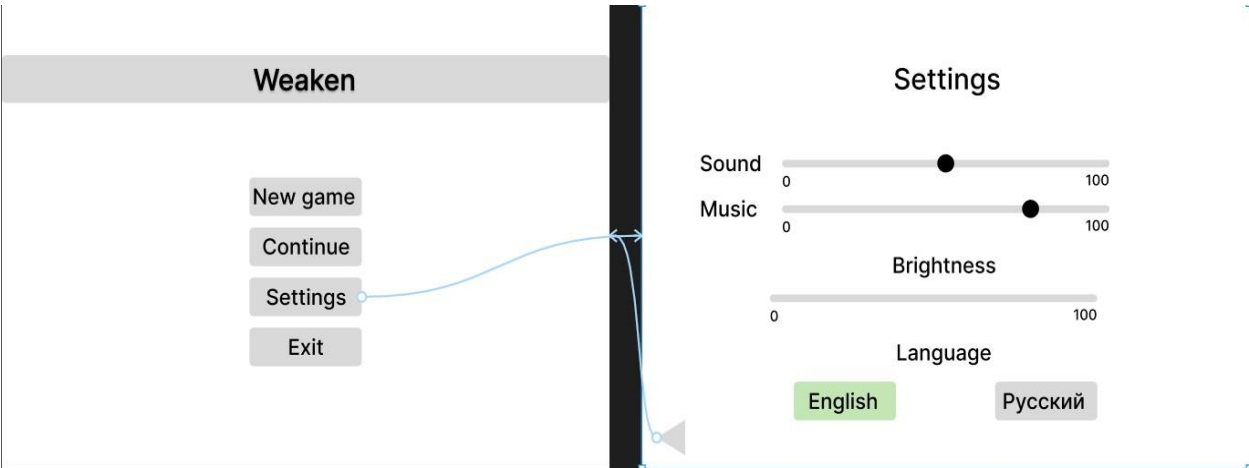


Рисунок 11 – переход из окна главного меню в окно «Settings» и обратно

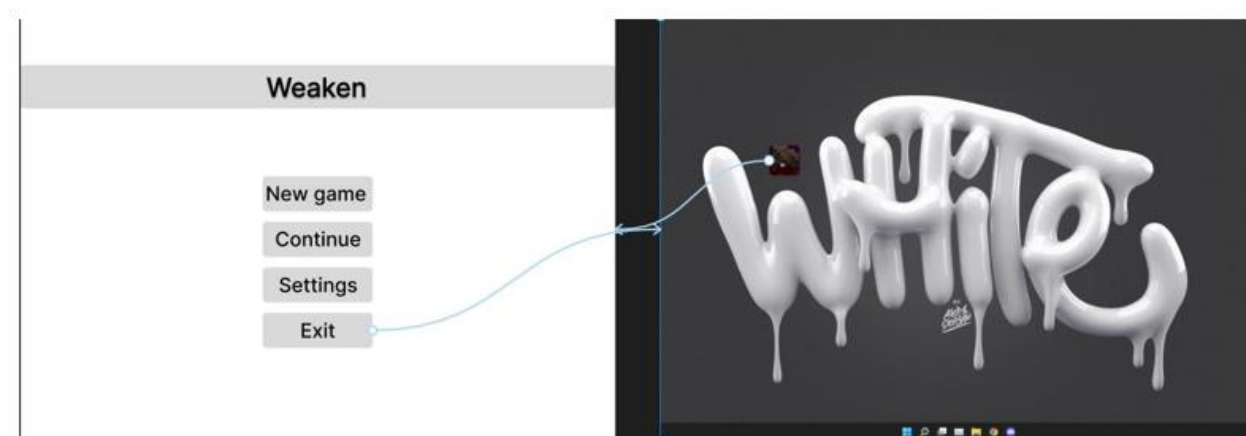


Рисунок 12 – выход из игры на рабочий стол пользователя через кнопку «Exit» и запуск игры через ярлык

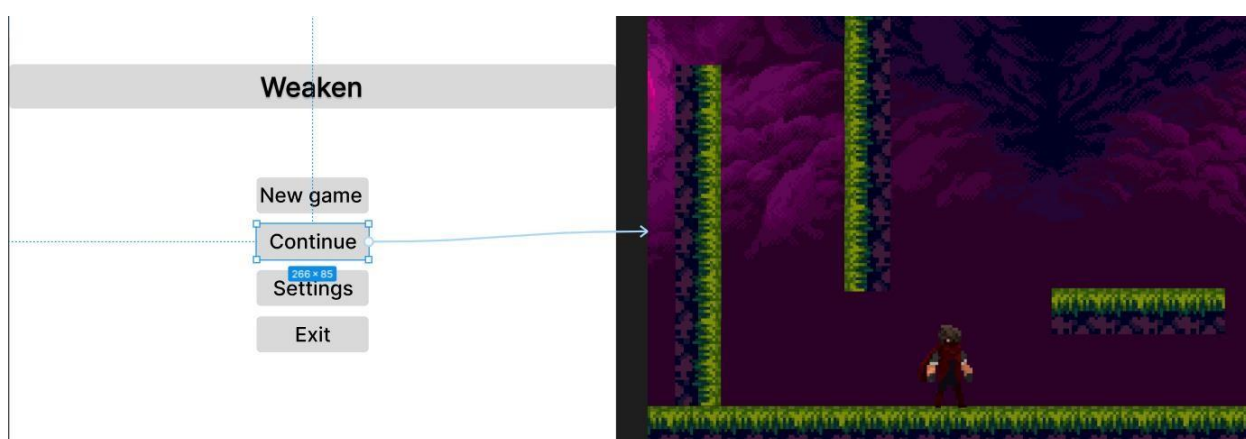


Рисунок 13 – продолжение игры с того момента, когда игрок запускал её в последний раз через кнопку «Continue»

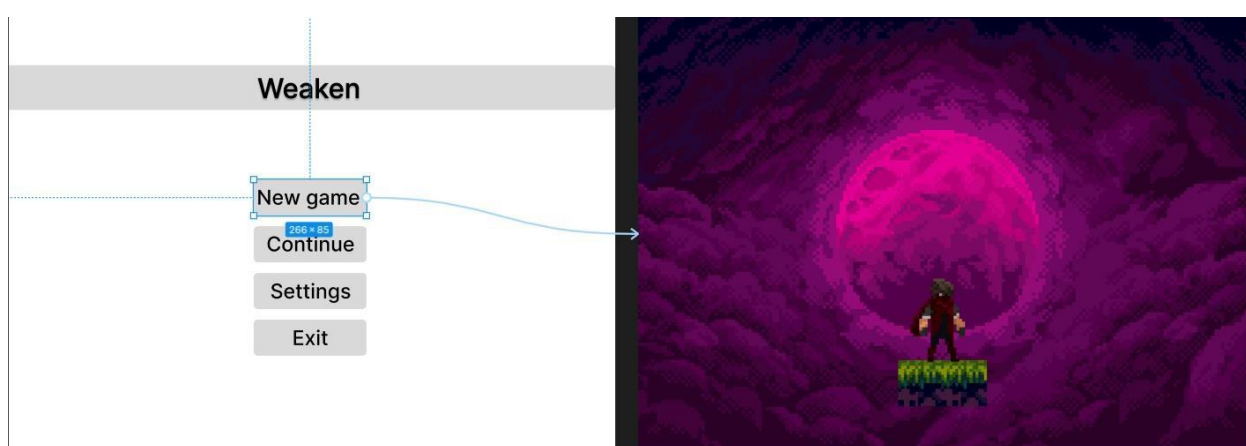


Рисунок 14 – начало новой игры через кнопку «New game»



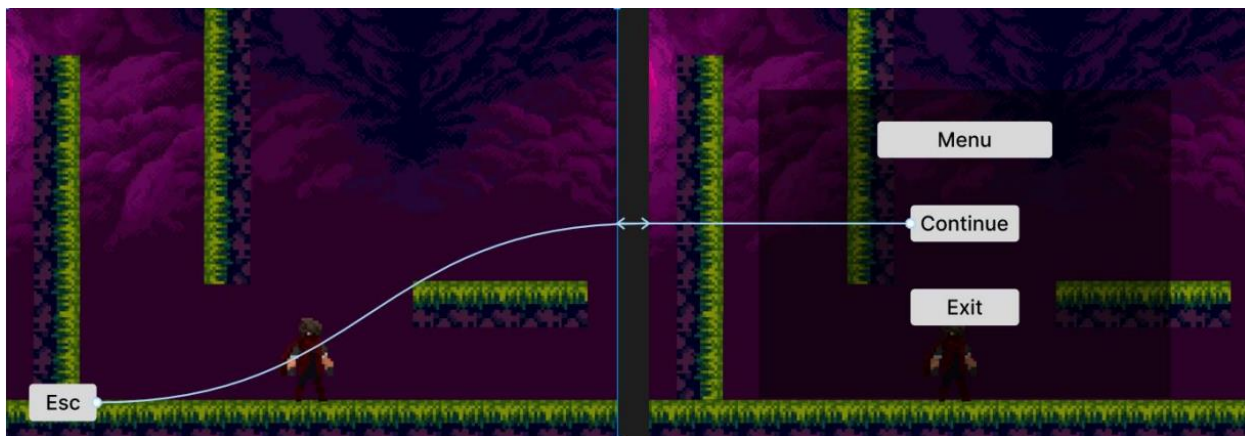


Рисунок 15 – вызов внутриигрового меню через клавишу «Esc» на клавиатуре в самой игре и возвращение в игру из внутриигрового меню через кнопку «Continue»

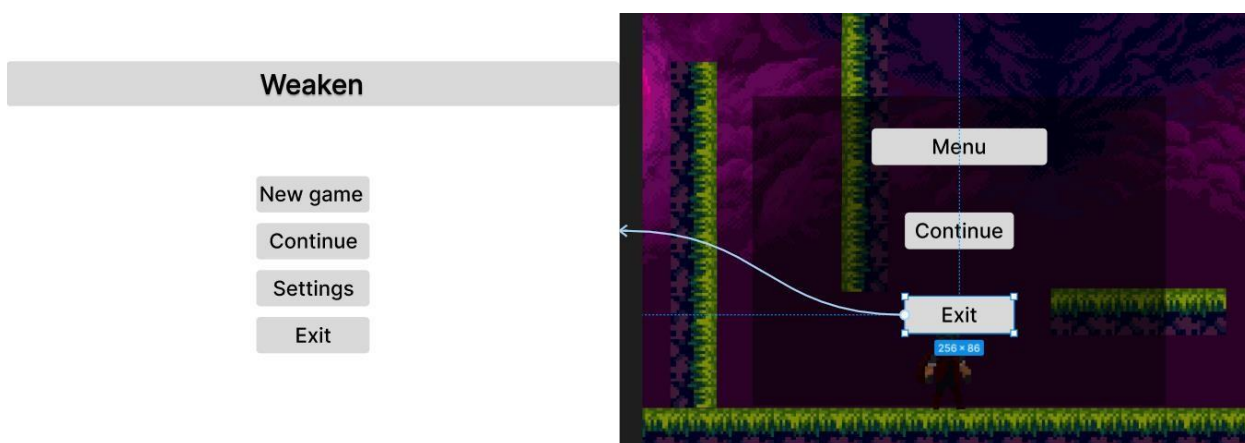


Рисунок 16 – выход из игры в главное меню из внутриигрового меню через кнопку «Exit»



### 3 Разработка программного продукта

#### 3.1 Инструментальные и программные средства разработки

Для разработки программного продукта были использованы следующие инструментальные и программные средства:

Создание игрового приложений, их поддержка и развитие осуществляется с помощью специализированного ПО.

Программно-инструментальные средства — это программные продукты, предназначенные для разработки программного обеспечения. К ним относят системы программирования, которые включают систему команд процессора и периферийных устройств, трансляторы с различных языков программирования.

Для разработки проекта были использованы следующие программные средства разработки:

Unity — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие.

GitHub - сервис онлайн-хостинга репозитория, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git и даже больше. Также GitHub может похвастаться контролем доступа, багтрекингом, управлением задачами и вики для каждого проекта.

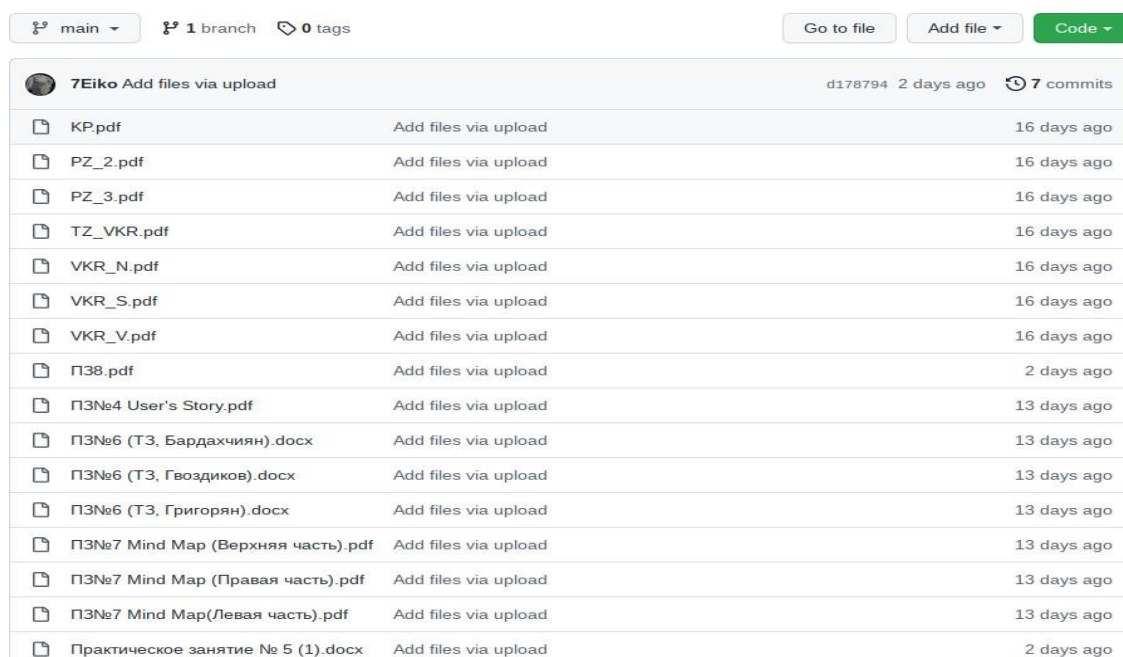


Рисунок 17 — репозиторий GitHub

Visual Studio 2022 - продукт компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Miro - платформа для совместной работы распределенных команд, разработанная в России и вышедшая на международный рынок.

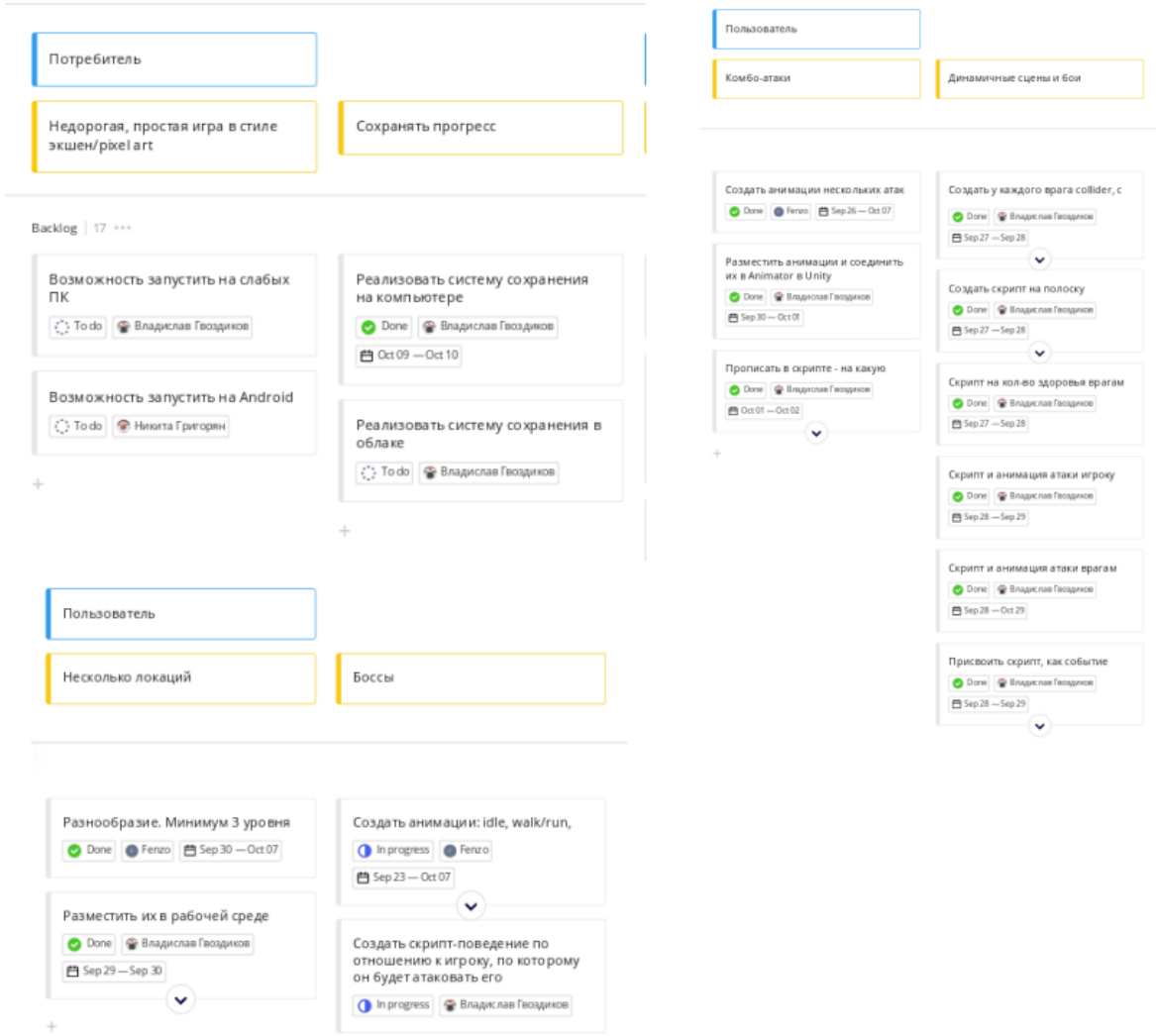


Рисунок 18 — доска Miro

3.2 Календарный план разработки

План разработки удобно представлять с помощью диаграммы Ганта. Диаграмма Ганта — это визуальный способ отображения

запланированных задач.

Горизонтальные графики широко используются для планирования проектов любых размеров в разных отраслях и сферах. Это удобный способ показать, какая работа планируется к выполнению в определенный день и время.

Календарный план разработки игрового приложения в виде диаграммы Ганта представлен на рисунке 19.

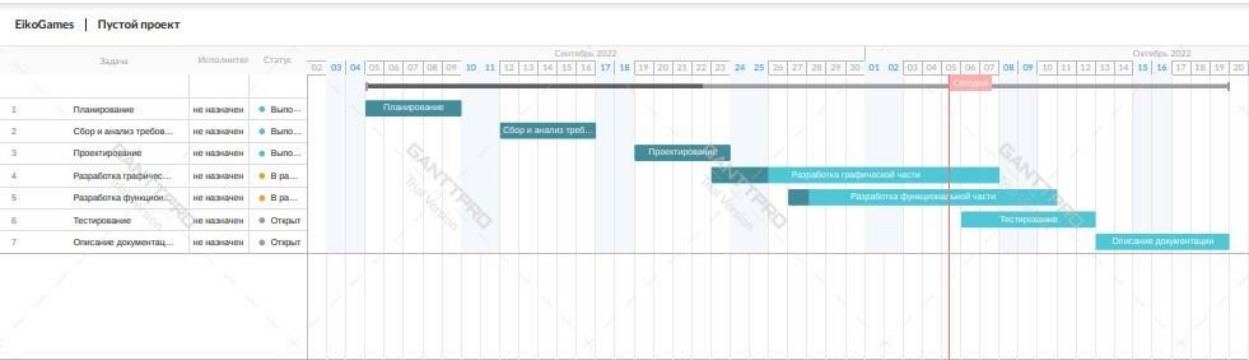


Рисунок 19 – Календарный план работы

## 4 Тестирование программного продукта

### 4.1 Выбор метода обеспечения качества

Тестирование программного обеспечения - это оценка разрабатываемого программного обеспечения/продукта, чтобы проверить его возможности, способности и соответствие ожидаемым результатам. Существуют различные типы методов, используемые в области тестирования и обеспечения качества.

Тестирование – процесс исследования и контроль качества, который состоит из планирования, проектирования, собственно проверки и анализа ее результатов.

Таблица 2 – План тестирования

Тест кейс, №	Название	Описание
1	Тест функционала главного героя	Тестирование перемещения, атаки и взаимодействия с окружающим миром
2	Тест системы здоровья	Тестирование шкалы здоровья во время получения урона
3	Тест функционала врагов	Тестирование поведения врагов по отношению к главному герою
4	Тест диалоговой системы	Тестирование диалоговой системы во время взаимодействия главного героя с NPC
5	Тест главного меню	Тестирование переходов между игрой, главным меню и рабочим столом пользователя
6	Тест боевой системы	Тестирование нанесения урона

### 4.2 Тестирование программного продукта

Для проведения тестирования программного продукта был составлен план проведения тестирования, который представлен в таблице 3.

Таблица 3 – Тест-кейс 1

Название	Описание
Идентификатор	1
Название	Тест функционала главного героя
Приоритет	Высокий
Описание	Тестирование функционала игрового персонажа
Предусловие	Предусловие отсутствует

Шаги тестирования	<ol style="list-style-type: none"> <li>1) Запустите приложение “Weaken” на устройстве</li> <li>2) Нажмите на кнопку “New game” в главном меню</li> <li>3) Проверьте анимацию покоя, когда персонаж стоит на месте</li> <li>4) Нажмите на клавиши WASD, чтобы персонаж начал двигаться</li> <li>5) Нажмите на клавишу Space, чтобы персонаж прыгнул</li> <li>6) Нажмите на клавишу LeftCtrl, чтобы персонаж присел</li> <li>7) Нажмите на клавишу C, чтобы персонаж осуществил пережат</li> <li>8) Нажмите на клавишу LeftShift для ускорения</li> <li>9) Проверьте анимацию покоя, когда персонаж находится на стене, прыгнув на неё</li> <li>10) Нажмите на клавишу Space, находясь на стене, для прыжка в противоположную сторону</li> <li>11) Нажмите на клавиши WS, чтобы персонаж начал взбираться вверх или вниз</li> <li>12) Нажмите на клавишу W, находясь на стене, чтобы персонаж добрался до края стены и повис на ней</li> <li>13) Находясь на краю стены, нажмите на клавишу W, чтобы забраться на стену</li> </ol>
Ожидаемый результат	<ol style="list-style-type: none"> <li>1) После запуска приложения появилось главное меню игры;</li> <li>2) После нажатия на кнопку “New game” появился игровой мир;</li> <li>3) У персонажа проигрывается анимация покоя, когда пользователь ничего не нажимает;</li> <li>4) У персонажа проигрывается анимация ходьбы;</li> <li>5) У персонажа проигрывается анимация прыжка, перепрыгивает препятствия;</li> <li>6) У персонажа проигрывается анимация приседания, проходит узкие пространства;</li> <li>7) У персонажа проигрывается анимация пережата, быстро преодолевает узкие пространства и уверачивается от ударов врагов;</li> <li>8) У персонажа проигрывается анимация ускорения, увеличивается скорость передвижения;</li> <li>9) У персонажа проигрывается анимация прыжка от стены, прыгает по стенам;</li> <li>10) У персонажа проигрывается анимация покоя при нахождении на стене;</li> <li>11) Проигрывается анимация, персонаж карабкается по стене;</li> <li>12) Проигрывается анимация, персонаж цепляется за горизонтальную поверхность;</li> <li>13) Проигрывается анимация, персонаж взобрался на горизонтальную поверхность.</li> </ol>
Постусловие	Постусловие отсутствует

Фактический результат	<ol style="list-style-type: none"> <li>1) После запуска приложения появилось главное меню игры;</li> <li>2) После нажатия на кнопку “New game” появился игровой мир;</li> <li>3) У персонажа проигрывается анимация покоя, когда пользователь ничего не нажимает;</li> <li>4) У персонажа проигрывается анимация ходьбы;</li> <li>5) У персонажа проигрывается анимация прыжка, перепрыгивает препятствия;</li> <li>6) У персонажа проигрывается анимация приседания, проходит узкие пространства;</li> <li>7) У персонажа проигрывается анимация переката, быстро преодолевает узкие пространства и уворачивается от ударов врагов;</li> <li>8) У персонажа проигрывается анимация ускорения, увеличивается скорость передвижения;</li> <li>9) У персонажа проигрывается анимация прыжка от стены, прыгает по стенам;</li> <li>10) У персонажа проигрывается анимация покоя при нахождении на стене;</li> <li>11) Проигрывается анимация, персонаж карабкается по стене;</li> <li>12) Проигрывается анимация, персонаж цепляется за горизонтальную поверхность;</li> <li>13) Проигрывается анимация, персонаж взобрался на горизонтальную поверхность.</li> </ol>
Статус	Пройден успешно

Таблица 4 – Тест-кейс 2

Название	Описание
Идентификатор	2
Название	Тест системы здоровья
Приоритет	Высокий
Описание	Тестирование здоровья главного героя и врагов
Предусловие	Предусловие отсутствует
Шаги тестирования	<ol style="list-style-type: none"> <li>1) Запустите приложение “Weaken” на устройстве</li> <li>2) Нажмите на кнопку “New game” в главном меню</li> <li>3) Проверьте наличие шкалы здоровья в левом верхнем углу</li> <li>4) Столкнитесь с врагом, и ждите до тех пор, пока враг вас не атакует, и шкала здоровья не уменьшится</li> <li>5) Нажмите на ЛКМ, чтобы атаковать врага</li> <li>6) Проверьте анимацию при получении урона от врагов</li> <li>7) Проверьте анимацию при нанесении урона по врагам</li> <li>8) Проверьте анимацию смерти у главного героя</li> <li>9) Проверьте анимацию смерти у врагов.</li> </ol>

Ожидаемый результат	1) После запуска приложения появилось главное меню игры; 2) После нажатия на кнопку “New game” появился игровой мир; 3) В левом верхнем углу есть шкала здоровья; 4) При получении урона от врагов здоровье у главного героя уменьшается; 5) При нанесении урона врагам их здоровье уменьшается; 6) При получении урона, у главного героя проигрывается анимация; 7) При нанесении урона, у врагов проигрывается анимация; 8) При обнулении здоровья, у главного героя проигрывается анимация смерти; 9) При обнулении здоровья, у врагов проигрывается анимация смерти.
Постусловие	Постусловие отсутствует
Фактический результат	1) После запуска приложения появилось главное меню игры; 2) После нажатия на кнопку “New game” появился игровой мир; 3) В левом верхнем углу есть шкала здоровья; 4) При получении урона от врагов здоровье у главного героя уменьшается; 5) При нанесении урона врагам их здоровье уменьшается; 6) При получении урона, у главного героя проигрывается анимация; 7) При нанесении урона, у врагов проигрывается анимация; 8) При обнулении здоровья, у главного героя проигрывается анимация смерти; 9) При обнулении здоровья, у врагов проигрывается анимация смерти.
Статус	Пройден успешно

Таблица 5 – Тест-кейс 3

Название	Описание
Идентификатор	3
Название	Тест функционала врагов
Приоритет	Высокий
Описание	Тестирование функционала игрового персонажа
Предусловие	Предусловие отсутствует
Шаги тестирования	1) Запустите приложение “Weaken” на устройстве; 2) Нажмите на кнопку “New game” в главном меню; 3) Столкнитесь с врагом в игре 4) Проверьте анимацию покоя, когда враг стоит на месте; 5) Проверьте передвижение врага, при нахождении главного героя в его поле зрения; 6) Нажмите на клавиши WASD, чтобы персонаж начал двигаться и удостоверьтесь в том, что враг преследует персонажа
Ожидаемый результат	1) После запуска приложения появилось главное меню игры; 2) После нажатия на кнопку “New game” появился игровой мир; 3) Главный герой встретил врага 4) У врага проигрывается анимация покоя 5) Враг замечает главного героя 6) Враг преследует главного героя
Постусловие	Постусловие отсутствует
Фактический результат	1) После запуска приложения появилось главное меню игры; 2) После нажатия на кнопку “New game” появился игровой мир;



	3) Главный герой встретил врага; 4) У врага проигрывается анимация покоя; 5) Враг замечает главного героя; 6) Враг преследует главного героя.
Статус	Пройден успешно

Таблица 6 – Тест-кейс 4

Название	Описание
Идентификатор	4
Название	Тест диалоговой системы
Приоритет	Высокий
Описание	Тестирование диалоговой системы между главным героем и NPC
Предусловие	Предусловие отсутствует
Шаги тестирования	1) Запустите приложение “Weaken” на устройстве; 2) Нажмите на кнопку “New game” в главном меню; 3) Встретить NPC с которым можно взаимодействовать; 4) Нажать на клавишу E, чтобы начать диалог с NPC; 5) Проверьте наличие диалогового окна; 6) Проверьте наличие текста в диалоговом окне; 7) Нажмите на клавишу E, чтобы переключить диалоговое окно на следующее; 8) Проверьте смену текста в диалоговом окне; 9) Нажимайте на клавишу E до тех пор, пока диалоговое окно не закроется.
Ожидаемый результат	1) После запуска приложения появилось главное меню игры; 2) После нажатия на кнопку “New game” появился игровой мир; 3) Главный герой встретил NPC с которым можно взаимодействовать; 4) При нажатии на клавишу E начинается диалог с NPC; 5) Появляется диалоговое окно; 6) Появляется текст в диалоговом окне; 7) При нажатии клавиши E диалоговое окно меняется; 8) В диалоговом окне меняется текст; 9) При нажатии клавиши E диалоговое окно закрывается.
Постусловие	Постусловие отсутствует
Фактический результат	1) После запуска приложения появилось главное меню игры; 2) После нажатия на кнопку “New game” появился игровой мир; 3) Главный герой встретил NPC с которым можно взаимодействовать; 4) При нажатии на клавишу E начинается диалог с NPC; 5) Появляется диалоговое окно; 6) Появляется текст в диалоговом окне; 7) При нажатии клавиши E диалоговое окно меняется; 8) В диалоговом окне меняется текст; 9) При нажатии клавиши E диалоговое окно закрывается.
Статус	Пройден успешно

Таблица 7 – Тест-кейс 5

Название	Описание
Идентификатор	5



Название	Тест главного меню
Приоритет	Высокий
Описание	Тестирование главного меню игры
Предусловие	Предусловие отсутствует
Шаги тестирования	<ol style="list-style-type: none"> <li>1) Запустите приложение “Weaken” на устройстве</li> <li>2) Нажмите на кнопку “New game” в главном меню</li> <li>3) Нажмите на клавишу ESC в игре, чтобы вызвать внутриигровое меню</li> <li>4) В всплывающем меню нажать на кнопку “Exit” и перейти в главное меню</li> <li>5) Нажмите на кнопку “Continue” в главном меню</li> <li>6) Проверить, сохранился ли прогресс игры с прошлого запуска</li> <li>7) В игре вызвать внутриигровое меню, нажав на клавишу ESC</li> <li>8) В всплывающем меню нажать на кнопку “Exit” и перейти в главное меню</li> <li>9) Нажмите на кнопку “Settings” в главном меню</li> <li>10) Проверьте наличие настроек;</li> <li>11) В окне настроек нажать на кнопку “Back” и перейти в главное меню</li> <li>12) Нажмите на кнопку “Exit” и выйти на рабочий стол.</li> </ol>
Ожидаемый результат	<ol style="list-style-type: none"> <li>1) После запуска приложения появилось главное меню игры;</li> <li>2) После нажатия на кнопку “New game” появился игровой мир;</li> <li>3) При нажатии на клавишу ESC появляется внутриигровое меню;</li> <li>4) В всплывающем меню после нажатия на кнопку “Exit” осуществляется переход в главное меню;</li> <li>5) После нажатия на кнопку “Continue” появился игровой мир;</li> <li>6) Прогресс игры сохранился;</li> <li>7) При нажатии на клавишу ESC появляется внутриигровое меню;</li> <li>8) В всплывающем меню после нажатия на кнопку “Exit” осуществляется переход в главное меню;</li> <li>9) После нажатии на кнопку “Settings” осуществляется переход в меню настроек;</li> <li>10) Все элементы настроек на месте;</li> <li>11) В меню настроек после нажатия на кнопку “Back” осуществляется переход в главное меню;</li> <li>12) После нажатия кнопки “Exit” осуществляется переход на рабочий стол.</li> </ol>
Постусловие	Постусловие отсутствует
Фактический результат	<ol style="list-style-type: none"> <li>1) После запуска приложения появилось главное меню игры;</li> <li>2) После нажатия на кнопку “New game” появился игровой мир;</li> <li>3) При нажатии на клавишу ESC появляется внутриигровое меню;</li> <li>4) В всплывающем меню после нажатия на кнопку “Exit” осуществляется переход в главное меню;</li> <li>5) После нажатия на кнопку “Continue” появился игровой мир;</li> <li>6) Прогресс игры сохранился;</li> <li>7) При нажатии на клавишу ESC появляется внутриигровое меню;</li> <li>8) В всплывающем меню после нажатия на кнопку “Exit” осуществляется переход в главное меню;</li> <li>9) После нажатии на кнопку “Settings” осуществляется переход в меню настроек;</li> <li>10) Все элементы настроек на месте;</li> </ol>

	11) В меню настроек после нажатия на кнопку “Back” осуществляется переход в главное меню; 12) После нажатия кнопки “Exit” осуществляется переход на рабочий стол.
Статус	Пройден успешно

Таблица 8 – Тест-кейс 6

Название	Описание
Идентификатор	6
Название	Тест боевой системы
Приоритет	Высокий
Описание	Тестирование боевой системы главного героя и врагов
Предусловие	Предусловие отсутствует
Шаги тестирования	<ol style="list-style-type: none"> <li>1) Запустить приложение “Weaken” на устройстве</li> <li>2) Нажмите на кнопку “New game” в главном меню</li> <li>3) Нажмите на клавиши WASD для того, чтобы начать двигаться и встретить врага</li> <li>4) Нажмите на ЛКМ и проверить анимацию атаки главного героя</li> <li>5) Проверьте анимацию при нанесении урона по врагам во время атаки главного героя</li> <li>6) Проверьте анимацию атаки врагов</li> <li>7) Проверьте анимацию при нанесении урона по главному герою во время атаки врагов</li> </ol>
Ожидаемый результат	<ol style="list-style-type: none"> <li>1) После запуска приложения появилось главное меню игры;</li> <li>2) После нажатия на кнопку “New game” появился игровой мир;</li> <li>3) Главный герой встретил врага;</li> <li>4) После нажатия на ЛКМ проигрывается анимация атаки главного героя;</li> <li>5) Срабатывает анимация атаки главного героя и враг получает урон;</li> <li>6) Проигрывается анимация атаки врагов;</li> <li>7) Срабатывает анимация атаки врагов и главный герой получает урон.</li> </ol>
Постусловие	Постусловие отсутствует
Фактический результат	<ol style="list-style-type: none"> <li>1) После запуска приложения появилось главное меню игры;</li> <li>2) После нажатия на кнопку “New game” появился игровой мир;</li> <li>3) Главный герой встретил врага;</li> <li>4) После нажатия на ЛКМ проигрывается анимация атаки главного героя;</li> <li>5) Срабатывает анимация атаки главного героя и враг получает урон;</li> <li>6) Проигрывается анимация атаки врагов;</li> <li>7) Срабатывает анимация атаки врагов и главный герой получает урон.</li> </ol>
Статус	Пройден успешно

## 5 Разработка документации на программный продукт

К видам документации, разрабатываемой на разных этапах жизненного цикла программного продукта, относятся:

- 1) технические требования;
- 2) технические спецификации;
- 3) сведения о выпуске;
- 4) руководства (например, по установке ПП, пользователя, администратора, программиста, по технической поддержке и т.д.);
- 5) описание проекта;
- 6) планы;
- 7) задания исполнителям (задание, распределённое между конкретными людьми или группами, участвующими в реализации проекта);
- 8) отчёты о ходе работ - создаются менеджерами для контролирующих органов;
- 9) протоколы встреч и обсуждений;
- 10) отчёты о результатах активности;
- 11) журналы.

По своему назначению и ориентации на определенные задачи и группы пользователей, документацию ПП можно разбить на две группы:

- 1) программные документы (технологические документы управления разработкой ПП) - документы, которые предназначены, прежде всего, для самих разработчиков и их начальства;
- 2) эксплуатационные программные документы (документы, входящие в состав программных систем) - документы, предназначенные для конечных пользователей или же обслуживающего персонала, позволяющие им осваивать и квалифицированно применять эти системы для решения конкретных функциональных задач.

В процессе создания игрового приложения «Weaken» было разработано приложение А, в котором находится программный код приложения, в приложении Б был разработан «Документ о концепции и границах проекта», в приложении В находится эксплуатационный программный документ «Руководство пользователя программным продуктом», а в приложении Г предоставлена графическая часть игрового приложения «Weaken».

## Заключение

В данной курсовой работе мы познакомились с технологиями разработки ПО и получили навыки работы с различными инструментами, таких как: Aspose, Miro, а также диаграммами Ганта.

Было разработано игровое приложение «Weaken». Игра была реализована в установленные сроки и полностью соответствует потребностям пользователей.

Так же стоит отметить, что по итогу курсовой работы были проверены и закреплены знания в области GameDev, а также в движке Unity и в самом языке C#/а также в графическом редакторе Aseprite.

Планируется добавить в игру элементы функционала: смена экипировки, смена оружия, инвентарь, расходные материалы.

					РКСИ.К22.09.02.03.МДК.03.01.000.00ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

## Список литературы

1. Интернет-ресурс Aseprite - <https://www.aseprite.org>
2. Интернет-ресурс C# - [https://ru.wikipedia.org/wiki/C\\_Sharp](https://ru.wikipedia.org/wiki/C_Sharp)
3. Интернет-ресурс Unity - <https://unity.com/ru>
4. Интернет-ресурс Игровой движок Unity - [https://ru.wikipedia.org/wiki/Unity\\_\(игровой\\_движок\)](https://ru.wikipedia.org/wiki/Unity_(игровой_движок))
5. Интернет-ресурс GitHub - <https://github.com>

					РКСИ.К22.09.02.03.МДК.03.01.000.00ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

```

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

public class PlayerController : MonoBehaviour
{
    public Rigidbody2D rb;
    public Vector2 mVector;
    public float speed = 2f;

    public Animator anim;
    public SpriteRenderer sr;
    public bool faceRight = true;

    public float jumpForce = 7f;
    private int jumpCount = 0;
    private int maxJumpValue = 2;

    public bool onGround;
    public Transform GroundCheck;
    public float checkRadius;
    public LayerMask Ground;

    public bool onWall;
    public Transform WallCheckUp;
    public Transform WallCheckDown;
    public float WallcheckRadiusUp;
    public float WallcheckRadiusDown;
    public LayerMask Wall;

    public int lungeImpulse = 5000;
    private bool locklunge = false;

    public float GroundCheckRadius;
    public Transform TopCheck;
    private float TopCheckRadius;
    public LayerMask Roof;
    public Collider2D poseStand;
    public Collider2D poseCrouch;
    private bool jumpLock = false;

```

```

public float fastSpeed = 5f;
private float realSpeed;
private bool speedLock;

public float upDownSpeed = 4f;
public float slideSpeed = 0;
private float gravityDef;

private bool blockMoveX;
public float jumpWallTime = 0.3f;
private float timerJumpWall;
public Vector2 jumpAngle = new Vector2(3.5f, 10);

void Start()
{
    rb = GetComponent<Rigidbody2D>();
    anim = GetComponent<Animator>();
    sr = GetComponent<SpriteRenderer>();

    realSpeed = speed;

    GroundCheckRadius =
GroundCheck.GetComponent<CircleCollider2D>().radius;
    TopCheckRadius = TopCheck.GetComponent<CircleCollider2D>().radius;

    WallcheckRadiusUp =
WallCheckUp.GetComponent<CircleCollider2D>().radius;
    WallcheckRadiusDown =
WallCheckDown.GetComponent<CircleCollider2D>().radius;

    gravityDef = rb.gravityScale;
}

void Update()
{
    Walk();
    Reflect();
    Run();
    Jump();
    CheckGround();
    Lunge();
    CrouchCheck();
    CheckWall();
    MoveOnWall();
}

```

```

    WallJump();
}

void Walk()
{
    if (!blockMoveX)
    {
        mVector.x = Input.GetAxisRaw("Horizontal");
        rb.velocity = new Vector2(mVector.x * realSpeed, rb.velocity.y);
        anim.SetFloat("moveX", Mathf.Abs(mVector.x));
    }
}

void Reflect()
{
    if ((mVector.x > 0 && !faceRight) || (mVector.x < 0 && faceRight))
    {
        transform.localScale *= new Vector2(-1, 1);
        faceRight = !faceRight;
    }
}

void Jump()
{
    if (Input.GetKeyDown(KeyCode.Space) && !jumpLock)
    {
        if (onGround)
        {
            anim.StopPlayback();
            anim.Play("Jump");
            rb.AddForce(Vector2.up * jumpForce);
            rb.velocity = new Vector2(rb.velocity.x, 0);
        }
        else if (++jumpCount < maxJumpValue)
        {
            anim.StopPlayback();
            anim.Play("DoubleJump");
            rb.velocity = new Vector2(0, 6);
        }
    }

    if (onGround)
    {

```



```

        jumpCount = 0;
    }

    if (Input.GetKeyDown(KeyCode.S))
    {
        Physics2D.IgnoreLayerCollision(3, 7, true);
        Invoke("IgnoreLayerOf", 0.5f);
    }
}

void CheckGround()
{
    onGround = Physics2D.OverlapCircle(GroundCheck.position,
checkRadius, Ground);
    anim.SetBool("onGround", onGround);
}

void CheckWall()
{
    onWall = Physics2D.OverlapCircle(WallCheckUp.position,
WallcheckRadiusUp, Wall) &&
Physics2D.OverlapCircle(WallCheckDown.position, WallcheckRadiusDown,
Wall);
    anim.SetBool("onWall", onWall);
}

void IgnoreLayerOf()
{
    Physics2D.IgnoreLayerCollision(3, 7, false);
}

void Lunge()
{
    if (Input.GetKeyDown(KeyCode.LeftControl) && !locklunge && onGround)
    {
        locklunge = true;
        Invoke("LungeLock", 2f);
anim.StopPlayback();
        anim.Play("Lunge");

        rb.velocity = new Vector2(0, 0);

        if (!faceRight)
        {

```

```

        rb.AddForce(Vector2.left * lungeImpulse);
    }
    else
    {
        rb.AddForce(Vector2.right * lungeImpulse);
    }
}

void LungeLock()
{
    locklunge = false;
}

void CrouchCheck()
{
    if (Input.GetKey(KeyCode.C) && onGround)
    {
        speed = 3;
        anim.SetBool("crouch", true);
        poseStand.enabled = false;
        poseCrouch.enabled = true;
        jumpLock = true;
    }
    else if (!Physics2D.OverlapCircle(TopCheck.position, TopCheckRadius,
Roof))
    {
        speed = 4;
        anim.SetBool("crouch", false);
        poseStand.enabled = true;
        poseCrouch.enabled = false;
        jumpLock = false;
    }
}

void Run()
{
    if (Input.GetKey(KeyCode.LeftShift) && onGround)
    {
        anim.SetBool("run", true);
        realSpeed = fastSpeed;

        if (Input.GetKeyDown(KeyCode.Space)) { speedLock = true; }
    }
    else

```

```

    {
        anim.SetBool("run", false);
        if (!speedLock) { realSpeed = speed; }
        else if (speedLock && onGround) { speedLock = false; }
        else { realSpeed = fastSpeed; }
    }
}

void MoveOnWall()
{
    if (onWall && !onGround)
    {
        mVector.y = Input.GetAxisRaw("Vertical");
        anim.SetFloat("UpDown", mVector.y);

        if (!blockMoveX)
        {
            anim.StopPlayback();
            anim.Play("UpDown");

            if (mVector.y == 0)
            {
                rb.gravityScale = 0;
                rb.velocity = new Vector2(0, slideSpeed);
            }
        }

        if (mVector.y > 0)
        {
            rb.velocity = new Vector2(rb.velocity.x, mVector.y * upDownSpeed /
2);
        }
        else if (mVector.y < 0)
        {
            rb.velocity = new Vector2(rb.velocity.x, mVector.y * upDownSpeed);
        }
    }

    else if (!onGround && !onWall) { rb.gravityScale = gravityDef; }

}

void WallJump()
{

```

```

if (onWall && !onGround && Input.GetKeyDown(KeyCode.Space))
{
    blockMoveX = true;

    mVector.x = 0;

    anim.StopPlayback();
    anim.Play("JumpingOnWall");

    transform.localScale *= new Vector2(-1, 1);
    faceRight = !faceRight;

    rb.gravityScale = gravityDef;
    rb.velocity = new Vector2(0, 0);

    rb.velocity = new Vector2(transform.localScale.x * jumpAngle.x,
jumpAngle.y);
}

if (blockMoveX && (timerJumpWall += Time.deltaTime) >=
jumpWallTime)
{
    if (onWall || onGround || Input.GetAxisRaw("Horizontal") != 0)
    {
        blockMoveX = false;
        timerJumpWall = 0;
    }
}
}
}

```

## Документ о концепции и границах проекта

## 1. Бизнес-требования

## 1.1. Исходные данные

Игровая индустрия постоянно развивается и даёт возможность начинающим разработчикам проявить свой творческий взгляд и создать свой собственный мир и поделиться им с окружающими. Для большинства, игры стали одним из основных способов развлечений и отдыха, а для некоторых искусством и неотъемлемой частью своей жизни. В связи с этим было принято решение разработать игровое приложение под названием «Weaken».

## 1.2. Возможности бизнеса

На сегодняшний день почти каждый человек использует интернет-ресурсы в различных целях. В игровой индустрии существует множество интересных игр с различными игровыми механиками, и каждая отличается от остальных, говоря нам о том, что любая идея может быть реализована, независимо от того, насколько сильно придётся конкурировать с другими продуктами.

## 1.3. Бизнес-цели и критерии успеха

Целями создания проекта является:

- 1) Создание игрового мира, с которым может познакомиться каждый;
- 2) Разработка отличительных игровых особенностей, таких как наличие внутриигровых персонажей с возможностью взаимодействия с ними:
  - противостояние главного персонажа с не игровыми, враждебными персонажами;
  - общение с мирными персонажами;
  - взаимодействие с объектами окружающей среды;
- 3) Привлечение пользователей будет осуществляться через интернет, т.к. продукт будет размещаться на соответствующих платформах.
- 4) Достигнуть объёма продаж, доход с которых будет составлять 1000 рублей за 12 месяцев;
- 5) Добиться признания продукта среди аудитории игроков и, в целом, творческих людей.

					РКСИ.К22.09.02.03.МДК.03.01.000.00ПЗ	Лист
						38
Изм.	Лист	№ докум.	Подпись	Дата		

б) Если продукт будет набирать популярность и в него будут играть, то это будет являться успехом.

#### 1.4. Потребности клиентов или рынка

Одной из самых важных частей повседневной жизни является развлечение.

Игры — один из самых лучших видов развлечений. Аудитория, которая окружает данный вид, обладает потребностями. Например, каждая игра должна быть стабильной и оптимизированной, чтобы каждый потребитель, даже со слабыми комплектующими устройства, мог запустить продукт. Обычно, большинство потребителей в игровой индустрии, сначала оценивают внешний вид продукта, поэтому готовое приложение оснащено соответствующими графическими данными, такие как дизайн мира, игрока и персонажей. Также, интерфейс готового продукта должен быть интуитивно понятен и легко запоминающимся.

#### 1.5 Бизнес-риски

Важнейшим риском в разработке игрового приложения является то, что идея, вложенная в игру, сюжет, геймплей, внешний вид может не понравиться потребителям и они могут потребовать обратно свои средства, потраченные на продукт. Поэтому важно учитывать потребности клиентов или рынка и оценивать свои возможности, чтобы оправдать эти потребности.

### 2. Образ решения

#### 2.1. Положение об образе проекта

Продукт под названием «Weaken» является игровым приложением, которое предназначена для того, чтобы отвлечься от рутины жизни и познакомить свою аудиторию с новым и неизведанным миром. Продукт разрабатывается для потребителей, которые любят играть в свободное время и наслаждаться временем, проведенным за игрой. В отличие от большинства игр, созданных в таком же жанре, игровое приложение «Weaken» обладает наличием сюжета и разнообразием игрового процесса.

					РКСИ.К22.09.02.03.МДК.03.01.000.00ПЗ	Лист
						39
Изм.	Лист	№ докум.	Подпись	Дата		

## 2.2 Основные функции

Основные функции продукта:

- Возможность управления движениями игрового персонажа с помощью клавиатуры;
- Возможность использовать функцию атаки игрового персонажа с помощью компьютерной;
- Разнообразие дизайна игровых уровней и не игровых персонажей (NPC);
- Возможность взаимодействия с не игровыми персонажами (NPC);
- Возможность взаимодействия с объектами окружающей среды;
- Разнообразие набора атакующих действий не игровых персонажей;
- Наличие ключевых враждебных не игровых персонажей (NPC).

## 3. Масштабы и ограничения проекта

Система проекта обладает красивым и привлекательным внешним видом и не сложным, но фундаментальным функционалом, который понятен для каждого. Но, система ограничена в высокоинтеллектуальных вычислениях и не имеет высоких графических настроек, т.к. в этом нет необходимости.

### 3.1. Объем проекта

Проект будет содержать следующие игровые механики: возможность перемещения игрока в разные направления, в том числе прыжки, приседания и рывок;

Набор движений игрока, которые упрощают передвижение по миру, такие как ползание по стенам и прыжки по стенам;

Набор атакующих способностей с разнообразными анимациями и видами атак;

Наличие различных уровней, в которых будут находиться несколько локаций, каждая будет отличаться своей стилистикой и цветовой гаммой.

## 4. Бизнес-контекст

### 4.1. Профили заинтересованных лиц

В проекте присутствуют следующие ценности, которые могут привлечь покупателей:

					РКСИ.K22.09.02.03.МДК.03.01.000.00ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

- стабильная работоспособность приложения;
- возможность запускать приложения на устройствах со слабыми комплектующими;
- низкая стоимость продуктами;
- наличие интересных игровых механик;
- наличие не игровых персонажей;
- разнообразие игрового мира.

## 4.2 Приоритеты проекта

Для эффективной разработки приложения, основные силы были направлены на создание фундаментальных игровых механик и качественного дизайна игрока, персонажей и игрового мира.

## 4.2 Операционная среда

Проект разрабатывается для компьютеров на игровом движке Unity, установленном на компьютер с операционной системой Windows, а для написания кода используется язык программирования C# в среде Visual Studio 2022.

					РКСИ.К22.09.02.03.МДК.03.01.000.00ПЗ	Лист 41
Изм.	Лист	№ докум.	Подпись	Дата		



## Руководство пользователя

1. Для запуска игры нажмите на соответствующий ярлык на рабочем столе.
2. Чтобы начать новую игру, в главном меню нажмите на кнопку «New Game».
3. Для перемещения по локации нажимайте клавиши WASD.
4. Для прыжка нажмите клавишу Space.
5. Для ускорения нажмите клавиши LeftShift + AD.
6. Для приседания нажмите клавишу LeftCtrl.
7. Для переката нажмите клавишу C.
8. Для карабканья нажмите клавиши WS.
9. Для атаки нажмите ЛКМ.
10. Для взаимодействия с NPC нажмите клавишу E.

					РКСИ.К22.09.02.03.МДК.03.01.000.00ПЗ	Лист
						42
Изм.	Лист	№ докум.	Подпись	Дата		



Рисунок 20 – Состояние покоя главного героя



Рисунок 21 – Атака главного героя



Рисунок 22 – Ходьба главного героя



Рисунок 23 – Приседание главного героя



Рисунок 24 – Состояние покоя врага



Рисунок 25 – Атака врага



Рисунок 26 – Ходьба врага

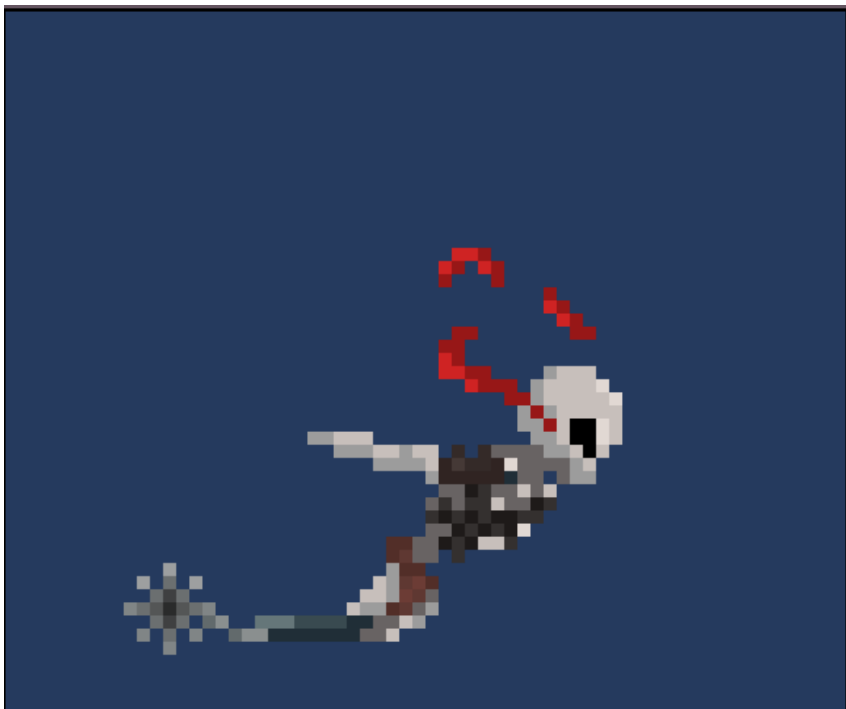


Рисунок 27 – Смерть врага