# 1.1 Постановка задачи

Наименование задачи: “last life”

Цель разработки:

Периодичность использования ПП:  при свободном времени, желании расслабиться и развлечься.

Источники и способы получения данных:

Обзор существующих аналогичных ПП: “рассмотрим на примере Portal” в данной игре пользователю постоянно прописывают диалоги-подсказки , игра от первого лица и взаимодействует через кнопки и рычаги.

# 1.1.2 Функциональные требования

Выбор уровня:

Tower defense - это поджанр стратегических видеоигр, цель которых заключается в защите территорий или владений игрока путем препятствования атакующим врагам, обычно достигаемых путем размещения защитных сооружений на или вдоль их пути атаки. Игра обычно имеет заданный путь, по которому перемещаются враги, часто в виде волн, и игрок должен размещать башни или другие защитные сооружения в стратегических точках вдоль пути, чтобы предотвратить достижение врагами своей цели.

1 где пользователь расставляет башни в определенных ячейках (башни -защитные сооружения с определенным радиусом действия, и скоростью атаки, башни не могут расставлять друг на друга, главная их цель уничтожать юнитов( юниты- это противники которые двигаются по определённому маршруту и имеют свою скорость передвижения, а так же количество hit point, главная цель дойти до конца), каждая башня отличается друг от друга, радиусом действия, скоростью стрельбы,  стоимостью, а также визуально). При убийстве юнитов даётся  определенное количество денег для постройки новых башен. Если юниты доходят до конца своего пути то выводиться на экран надпись lose с возможностью повторить уровень званого. Если же башни справляются со всеми юнитами то выводиться на экран надпись Win и переход к новому уровню или же пройти уровень повторно

**Описание перечня функций и задач, которые должен выполнять**

**будущий ПП:**

**Гость:**

1 передвигаться

2 прыгать и использовать спринт

3 Использовать меню на esc

4 взаимодействовать с обьектами

GamePlay:

Передвижение осуществляетяс на wasd и space, присутствует спринт и кол-во выносливости. При перемещении будет меняться текст-подсказка, взаимодействие осуществляется при помощи платформ, кнопок и рычагов.

Рисунок 1

**Описание процессов с входной, выходной и условно-постоянной Информацией**:

Так как у нас игра без регистрации, то как таковой входной информации не будет, выходной информацией будет являться:

1 вывод на экран текста с подсказками

2 при нажатии на wasd shift space передвигаться

3 при нажатии на esc появление меню паузы

4 выйти из игры при нажатии на esc и выйти

5 перейти на следующий уровень через портал

Эксплуатационные требования:

Требования к реализации:

Для реализации можно воспользоваться языками python и C#, для создания текстур blender. Для подключения P2P.

Требования к надежности: игра не нуждается в повышенным требования надёжности

Требования к интерфейсу: при запуске игры, будет выведен главный экран приложения, в которому будут переходы между уровней, настройки, справка, игра будет запущена в полноэкранном режиме. Сам интерфейс будет выполнен в Pixel ART still

Минимальные системные требования:

* windows 7-10
* свободное пространство на диске не менее 1- гб
* Минимальный объем оп не менее 2-гб
* Объём видео памяти 512-мб

# 1.2 Диаграмма вариантов использования

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования.

C диаграммой вариантов использования можно ознакомится в приложения А Диаграмма вариантов использования.

**2 Проектирование**

**2.1 Выбор стратегии разработки и модели жизненного цикла**

Таблица1- Выбор модели жизненного цикла на основе

характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? |  |  |  | Нет | Нет | Нет |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? |  |  |  |  | Да | Да |
| 4. | Нужно ли демонстрировать требования с целью их определения? |  |  | Да |  | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или системы? |  |  | Да |  | Да | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? |  |  |  | Да | Да | Да |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |

Вычисления: 2 за каскадную, 2 за V- образную, 4 за RAD, 4 за инкрементную, 7 за быстрого прототипирования и 7 за эволюционную.

Итог: На основе результатов заполнения табл. 3 подходящей является Эволюционная и быстрого прототипирования .

Таблица 2 **–** Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков  проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да |  |  |  | Да |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? |  |  |  | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? |  |  | Нет |  | Нет | Нет |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да |  | Да | Да | Да |

Вычисления: 4 за каскадную, 4 за V-образную, 3 за RAD, 4 за инкрементную, 5 за быстрого прототипирования и 6 за эволюционную.

Итог: На основе результатов заполнения табл. 4 подходящими являются каскадная, Эволюционная и прототипирования.

Таблица 3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? |  |  | Нет |  | Нет |  |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? |  |  |  | Да | Да | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет |  | Нет |  | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? |  |  |  |  | Да | Да |

Вычисления: 1 за каскадную, 1 за V-образную, 1 за RAD, 2 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: на основе результатов заполнения табл. 5 подходящей является модель эволюционная и протежированная.

Таблица 4 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? |  |  |  |  |  |  |
| 2. | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да |  |  |
|  |  |  |  |  |  |  |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 3. | Будет ли проект крупно- или среднемасштабным? |  |  |  | Да | Да | Да |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Да |  | Да |  | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? |  | Да |  | Да |  | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? |  | Да |  | Да |  | Да |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? |  |  |  | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? |  |  |  | Да | Да | Да |
| 8. | Является ли график сжатым? |  |  | Да | Да | Да | Да |
| 9. | Предполагается ли повторное использование компонентов? |  |  | Да | Да | Да | Да |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? |  |  |  |  | Да | Да |

Вычисления: 2 за каскадную, 3 за V-образную, 3 за RAD, 8 за инкрементную, 6 за быстрого прототипирования и 8 за эволюционную.

Итог: На основе результатов заполнения табл. 6 подходящей является инкрементная и эволюционная модели.

9 за каскадную ,10 за V-образную  ,11 за RAD , 18 за инкрементную ,

21 за быстрого прототипирования, 24 эвалюционная

Общий итог: в итоге заполнения табл. 3 – 6 наиболее подходящей является эволюционная

# 2.2. Проектирование системы главного меню

С диаграммой системы главного меню можно ознакомится в приложение (Б) Система главного меню.

# 2.2 Модель бизнес процесса

С диаграммой Модель бизнес процесса вы сможете ознакомится в приложение (В) Модель бизнес процесса.

# 2.3 Разработка UML-диаграмм

В ходе создания проекта требовалось создать две UML-диаграммы, что помогли бы понять определенные процессы самого интернет-ресурса. Этими диаграммами являются диаграмма последовательности и диаграмма деятельности.

Диаграмма последовательности (англ. sequence diagram) — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента (приложение Д Диаграмма последовательности).

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. (приложение Г Диаграмма деятельности).

Диаграмма модели данных (приложение Е Диаграмма модели данных).

Диаграмма классов (приложение Ё Диаграмма классов).

Диаграмма объектов (приложение Ж Диаграмма объектов).

Диаграмма компонентов (приложение З Диаграмма компонентов).

# 2.4 Разработка пользовательского интерфейса

Поставленной задачей на практику была реализация ux/ui интерфейсов. При

разработке интерфейсов были использованы следующие разрешения: 1920px, 1080px. При создании UX/UI интерфейсов были использованы модульные сетки с целью создания пропорционального, понятного интерфейса. Использовались преимущественно оттенки серого и черного цвета. Все доступно сразу. Таким образом был реализован понятный пользовательский интерфейс, созданы макеты под такое устройства как: компьютер.

С диаграммами UX/UI можно ознакомится в приложение И.

# 3 Реализация

# 3.1 Руководство программиста

# 3.1.1 Организация данных

В данном проекте мы используем движок Unity.встроенную базу данных в C#, ее будет достаточно для реализации проекта. В данной базе хранится всё информация о кнопках и их взаимодействия.

# 3.1.2 Структура и описание процедур и функций пользователя

Пользователь может выбирать инструменты для рисования, редактировать ширину и тип линий, может очистить при помощи ластика или очистки рабочей поверхности, может сохранить изображение на устройстве. Может выйти из приложения нажав на крестик.

Организационные данные:

Файл 1 munu.cs:

ChechIsOnAAtop– проверка нахождения чего-либо сверху.

void Update() – передвижение персонаже, движение камеры, движение обьектов.

Jump()-прыжок

Die() – смерть.

RaloadLevel() – перезапуск уровня.

IsGrounded() – на земле ли он.

Rotate()-Вращение обьекта.

AtractiveObjectAndParents()-активирует или дизкативирует обьекты.

OnCollisionEnter()-при соприкосновении.

HelpingThint()-удаляет обьект с карты.

Start()-присваивает множество переменных и значений по умолчанию при старте.

Resume-Продолжить после паузы.

Pause-Останавливает игру.

QuitGame() – выход.

OnCollisionExit() – при несоприкосновении.

# 4 Тестирование

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы. Так же в случае провала должно помочь в раннем устранении неполадок связанные с ошибками в коде или ошибках системы.

Отчёт о результатах тестирования предоставлен в таблице 5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название теста | Шаги | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Т1 | Передвижение на wasd | Нажать на клавиши wasd | Персонаж двигается | Персонаж двигается | Выполнено |
| Т2 | Вращать камерой | Подключить мышь и провести ею | Персонаж вращает головой | Персонаж вращает головой | Выполнено |
| Т3 | Пауза | Нажав на esc | Откроется меню паузы | Открылось меню паузы | Выполнено |
| Т4 | Продолжить | Нажать esc затем нажать продолжить | Игра продолжится | Игра продолжилась | Выполнено |
| Т5 | выйти | Нажать esc Нажать выйти | Игра закроется | Игра закрылать | Выполнено |
| Т6 | прыжок | Нажать кнопку space | Персонаж подпрыгнет | Персонаж подпрыгнул | Выполнено |
| Т7 | Спринт | Зажать w и shift | Персонаж будет двигаться быстрее и терять выносливость | Персонаж бежит быстрее и теряет выносливость | Выполнено |
| Т8 | подсказки | Пройтись вперёд | Текст будет меняться при приближении к обьектам | Текст меняется при приближении к обьектам | Выполнено |
| Т9 | рестарт | Спрыгнуть в бездну | Уровень начнётся заново | Уровень начался заново | Выполнено |
| Т10 | рычаг | Подойти к рычагу и коснуться его | Рычаг изменит своё положение и запустит действие | Рычаг изменил положение и активировал действие | Выполнено |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Т11 | кнопка | Коснуться кнопки | При соприкосновении запустит действие | При соприкосновении запустилось действие | Выполнено |
| Т12 | Переход на следующий уровень | Коснуться портала | Загрузится следующий уровень | Загрузился следующий уровень | Выполнено |

В ходе тестировании не было выявлено ошибок, игра работает корректно

# 5 Руководство пользователя

# 5.1 Общие сведения о программном продукте

Наименованием программного продукта является The last life. Данная игра была разработана с учетом потребностей пользователей в развлекательных приложениях. Она предназначена для того, чтобы обеспечить пользователям уникальный опыт игры, который позволит им наслаждаться игровым процессом в полной мере.

# 5.2 Запуск приложения

Для запуска приложения нажать на файл MyProject.exe

# 5.3 Инструкции по работе с приложением

После запуска игрового приложения сразу же начнётся обучение. В правом верхнем углу всегда будет виден текст, который будет меняться по мере продвижения игрока

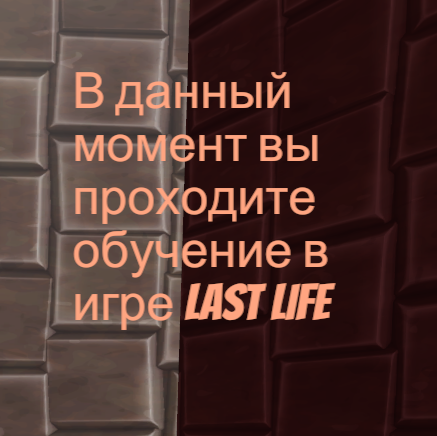


Рисунок 2 – Текст-подсказка

Передвижение осуществляется на клавиши wasd, space и LShift.

При использовании бега(LeftShift), вы будете быстрее передвигаться, а так же расходовать выносливость. О текущем количестве выносливости можно узнать, посмотрев на полосу внизу экрана.

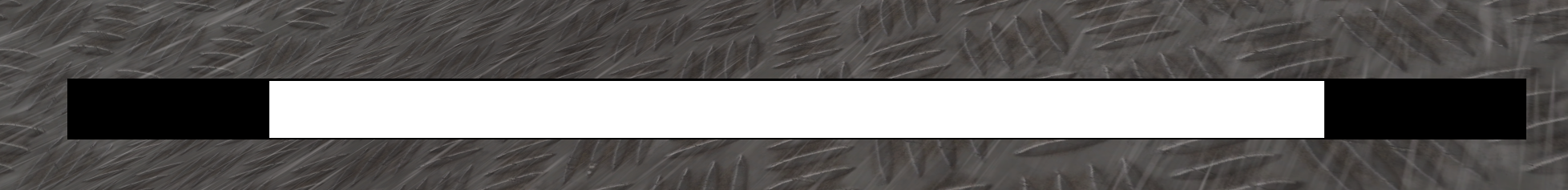


Рисунок 3 – Выносливость

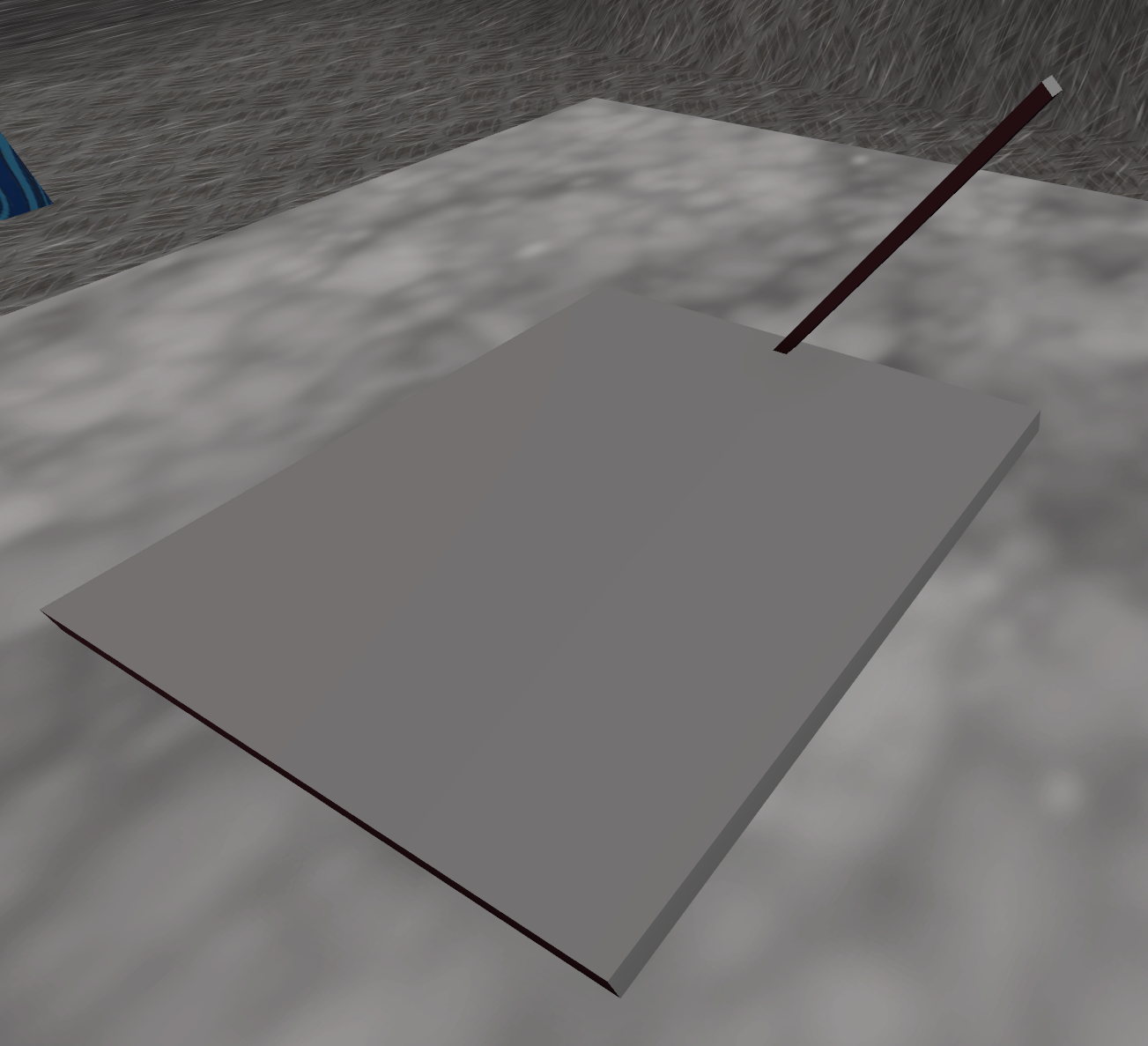
Взаимодействие с игровыми объектами осуществляется через рычаги и кнопки.  


Рисунок 4 – Рычаг

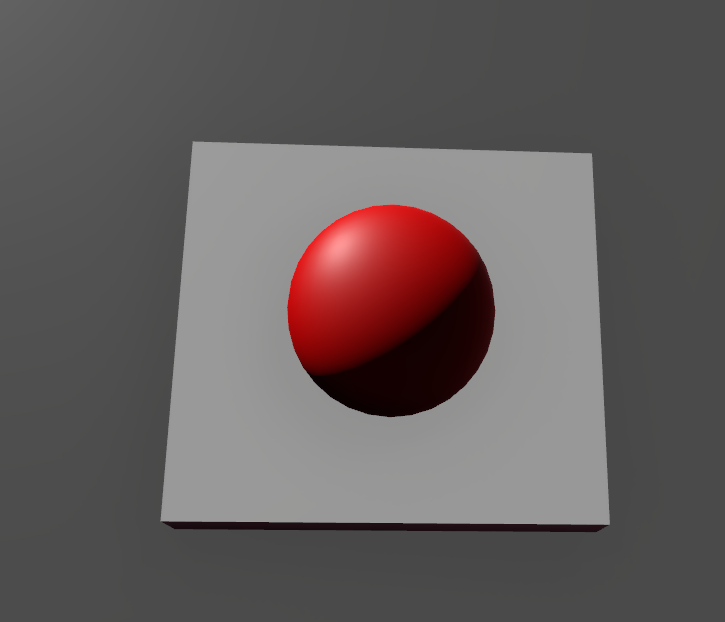


Рисунок 5 – Кнопка

# Для взаимодействия достаточно коснуться рычага либо кнопки.

При нажатии клавиши ecs, игра ставится на паузу и появляется игровая меню-пауза в выбором: Продолжить игру или выйти.



Рисунок 6 – Меню паузы

Для перехода на следующий уровень используются порталы. Для взаимодействия с ними необходимо его коснуться.

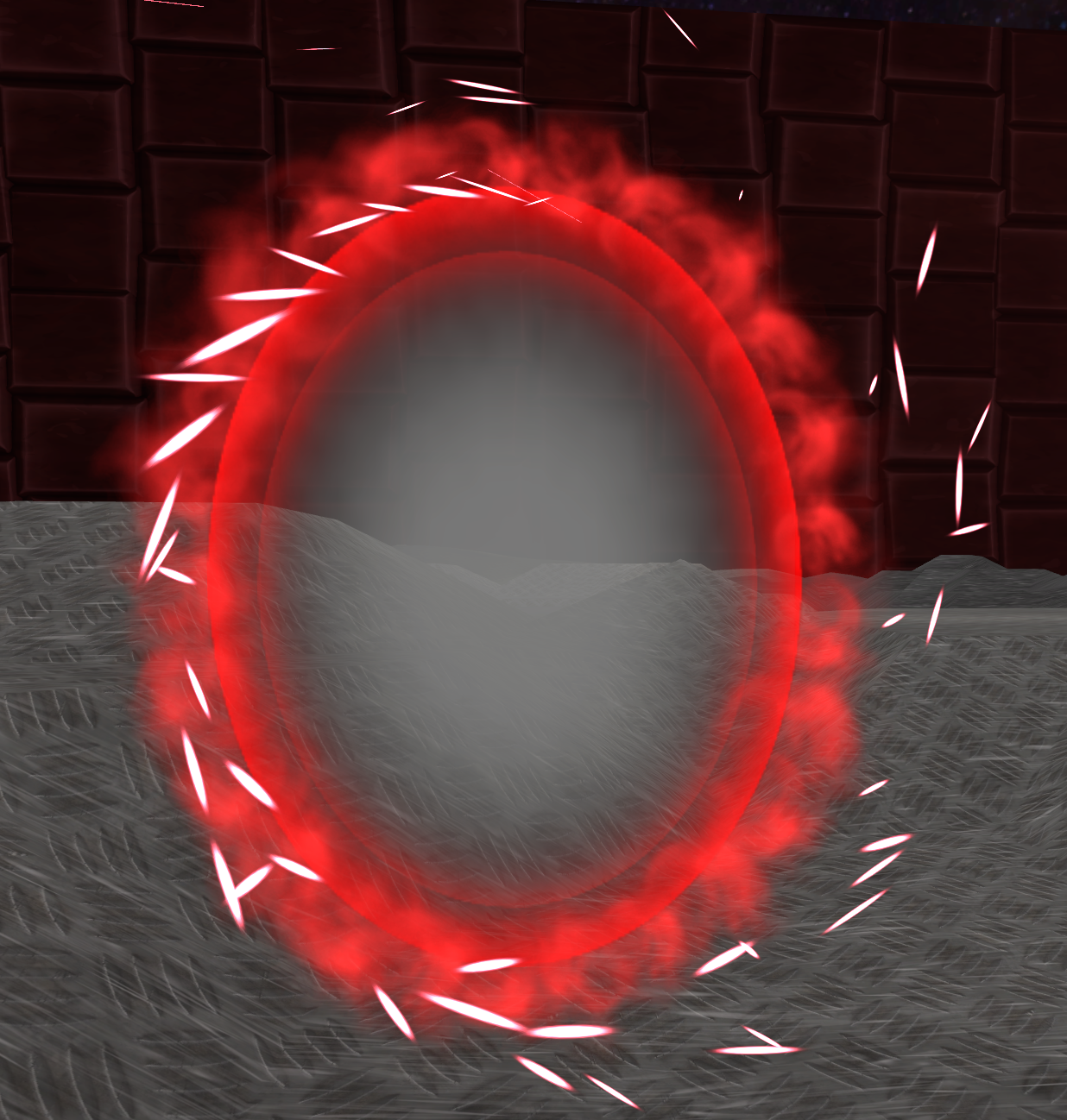


Рисунок 7 – Портал

При выпадении за карту уровень начинается заново.



Рисунок 8 – Пространство за картой

# 5.4 Завершение работы с приложением

# Заключение

Целью данного проекта являлась разработка игрового приложения The last life.

В ходе реализации поставленной задачи был укреплён пройденный изученный материал по дисциплине «Технология разработки программного обеспечения», а также получено много дополнительной информации о её возможностях.

Созданная программа предоставит материал, благодаря которому вы сможете изучить такой жанр игры как Tower Defe.

Следует также учесть, что в поставленной задачи был реализован простой интерфейс, который позволяет использовать приложение пользователю, не обладающим опытов в подобных приложениях.

После тщательного тестирования приложения были выявлены некоторые недоработки, которые были исправлены на стадии проектирования, и полностью исключены на стадии тестирования программы. В целом при реализации программы, были выполнены все условия, перечисленные в предыдущих разделах пояснительной записки. Таким образом, можно сказать, что программа была реализована вполне успешно.

# Приложение А

**Диаграмма вариантов использования**

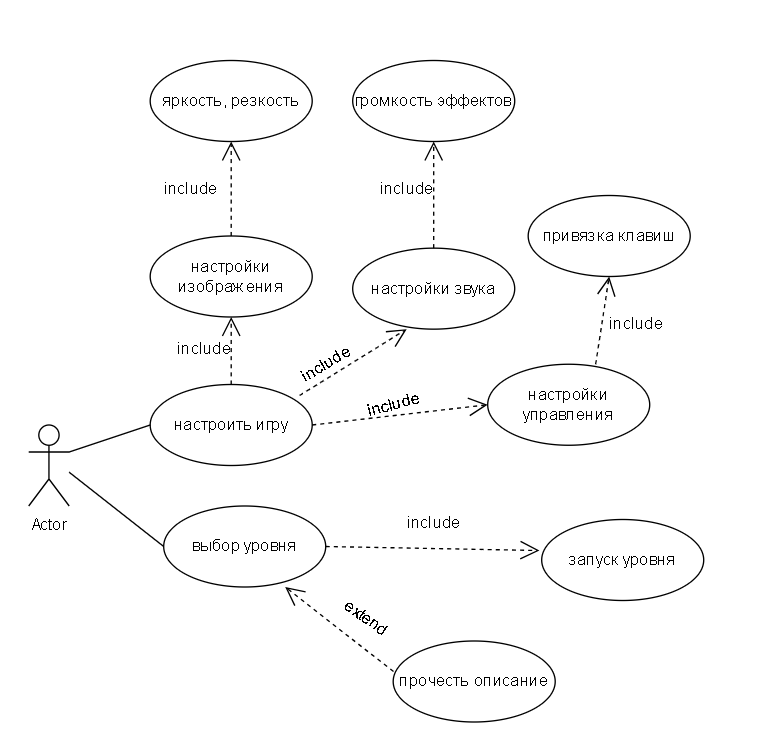


Рисунок 9 - Диаграмма вариантов использования

# Приложение Б

# Диаграмма системы главного меню

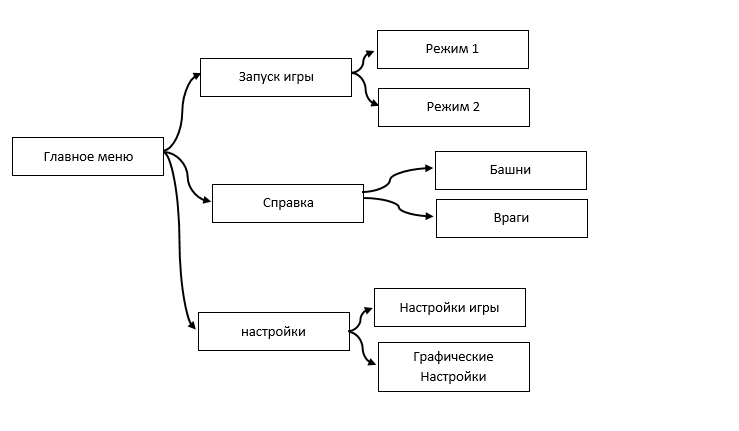


Рисунок 10 – Диаграмма системы главного меню

# Приложение В

**диаграммой Модель бизнес-процесса**

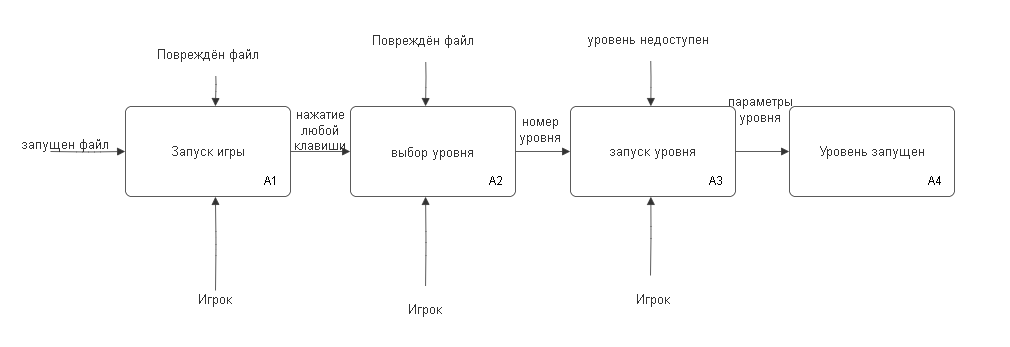


Рисунок 11 – Модель бизнес-процесса

# Приложение Г

**Диаграмма деятельности**



Рисунок 12 – Диаграмма деятельности часть 1

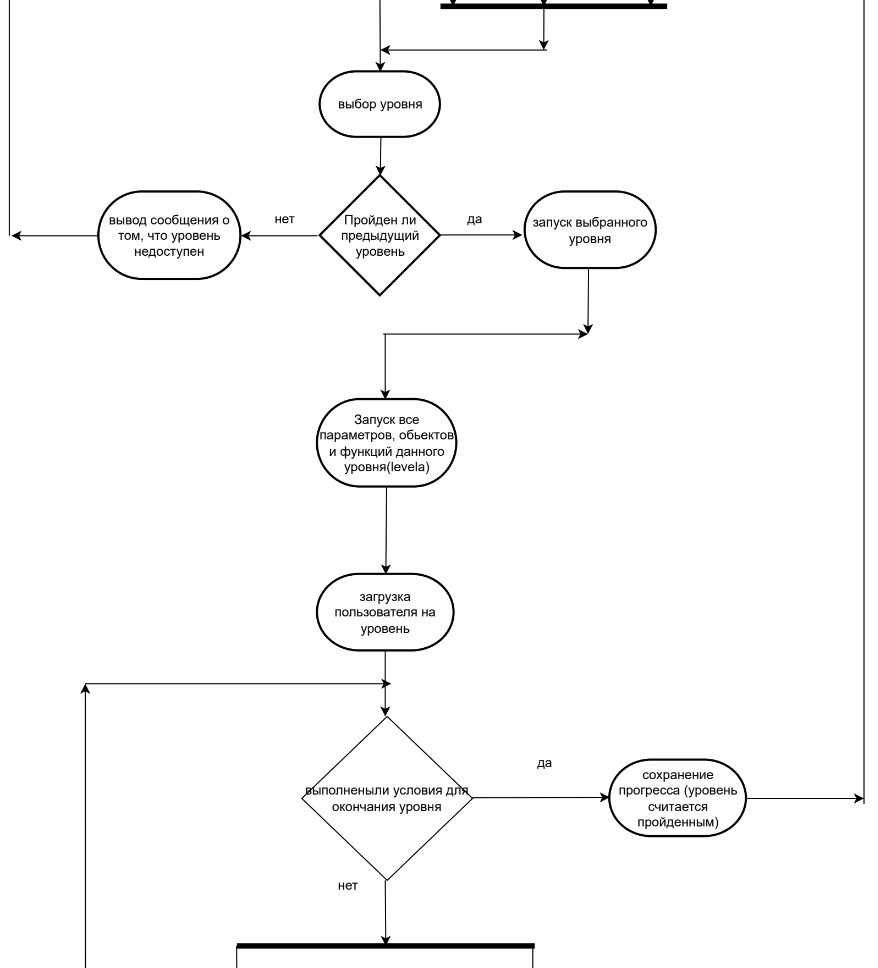


Рисунок 13 – Диаграмма деятельности часть 2

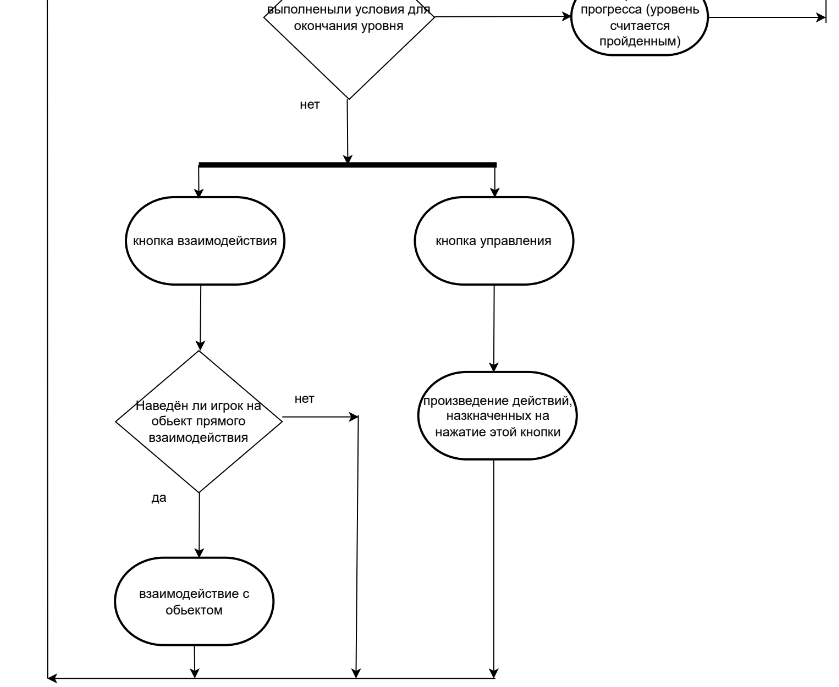


Рисунок 14 – Диаграмма деятельности часть 3

# Приложение Д

# Диаграмма последовательности

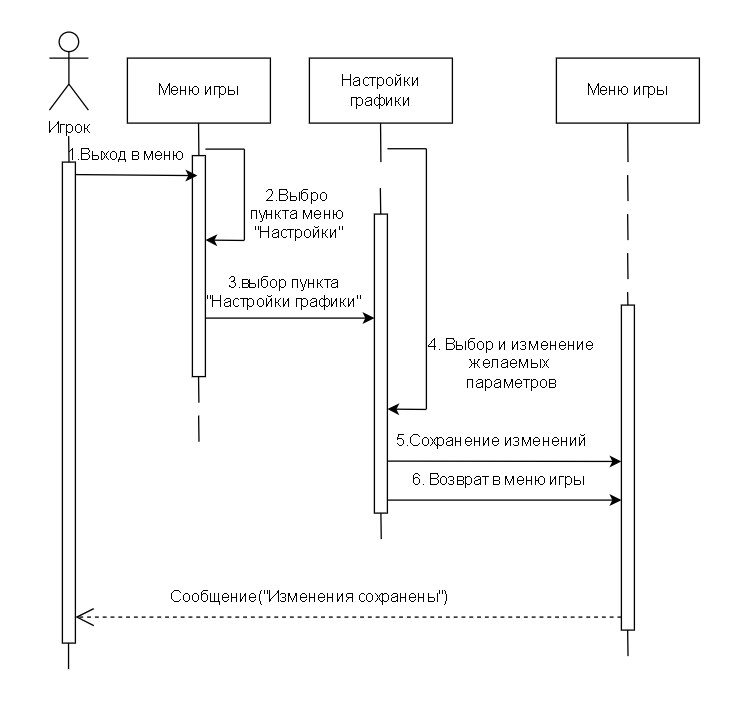
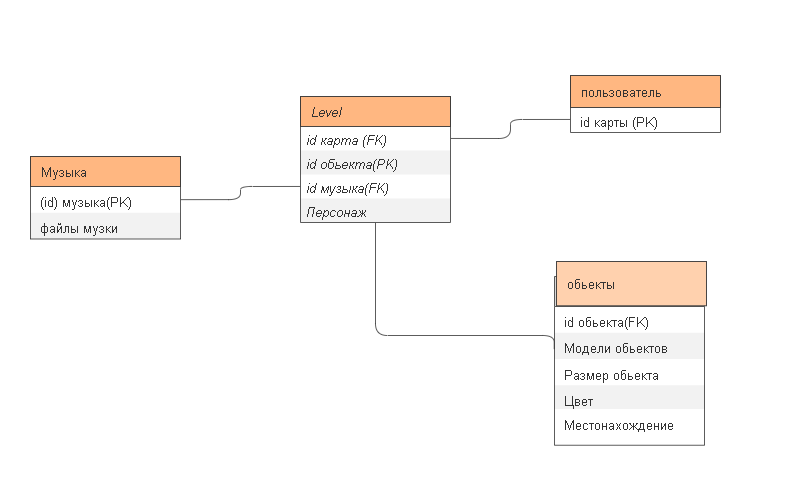


Рисунок 15 – Диаграмма последовательности

# Приложение Е

# Диаграмма модели данных



# Рисунок *16* – Диаграмма модели данных

# Приложение Ё

**Диаграмма классов**

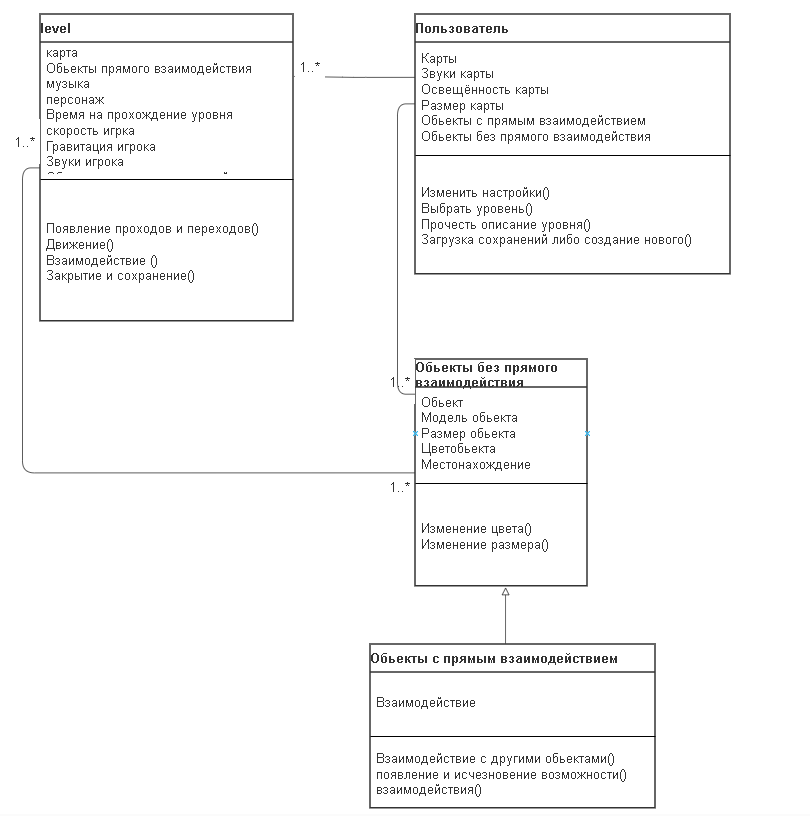


Рисунок 17 – Диаграмма классов

# Приложение Ж

**Диаграмма объектов**

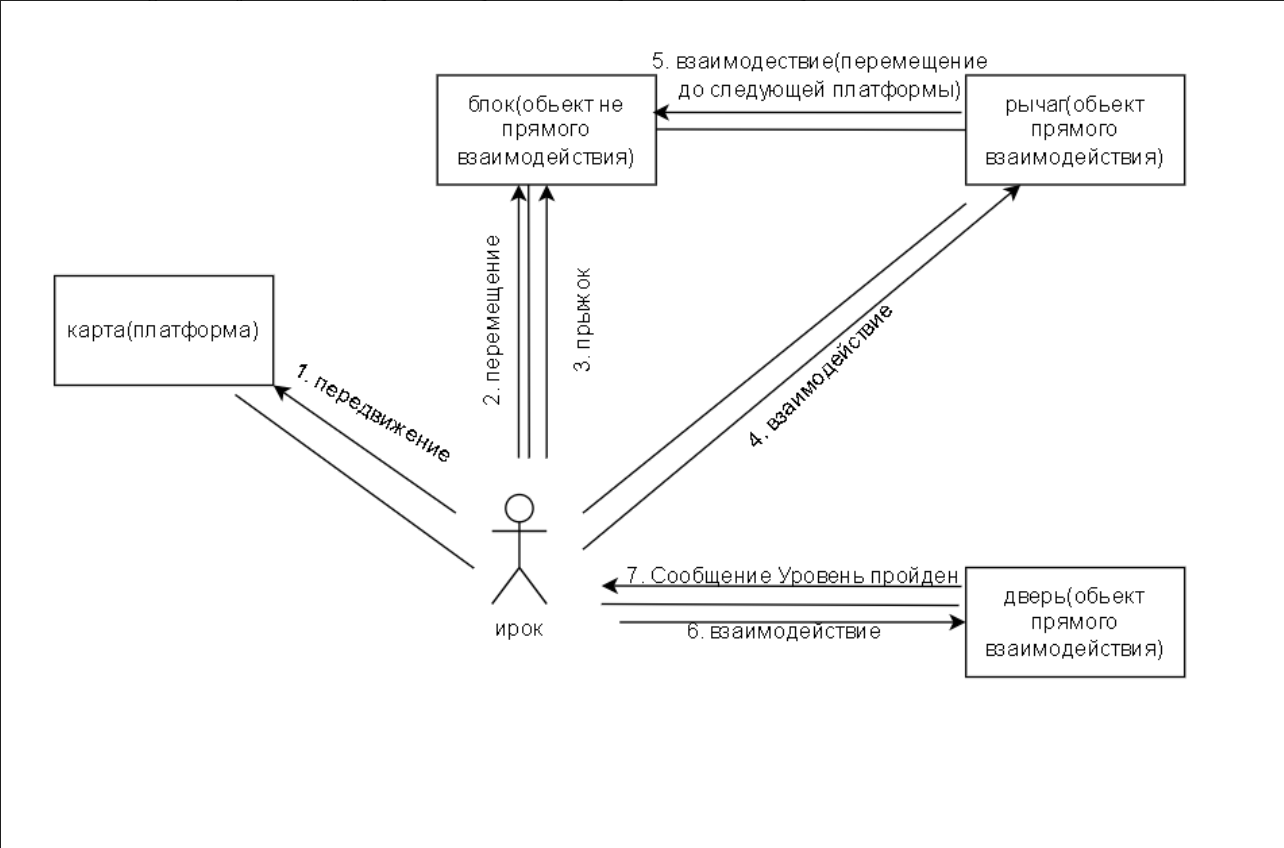


Рисунок 18 Диаграмма объектов

# Приложение З

**Диаграмма компонентов**

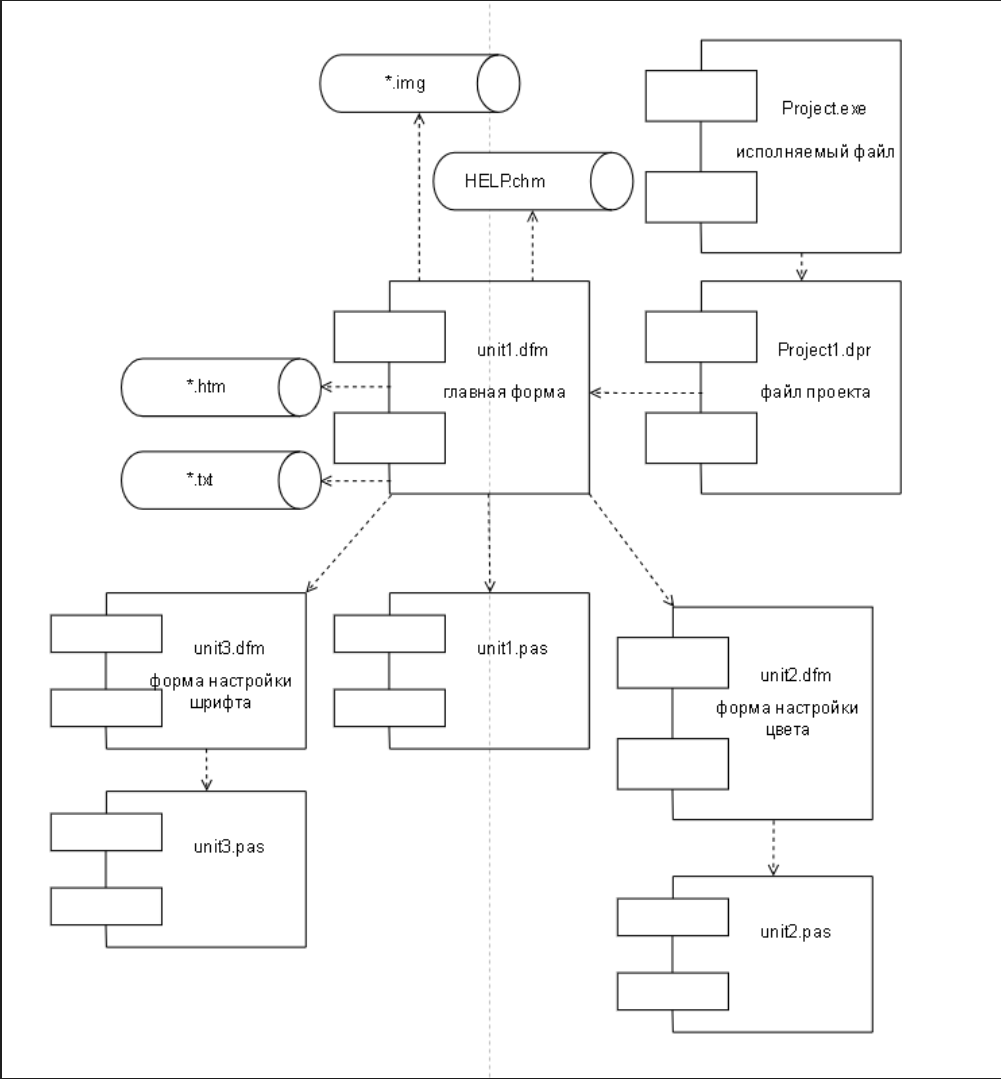
******

Рисунок 19 Диаграмма компонентов

# Приложение И

**Диаграммы UX/UI**

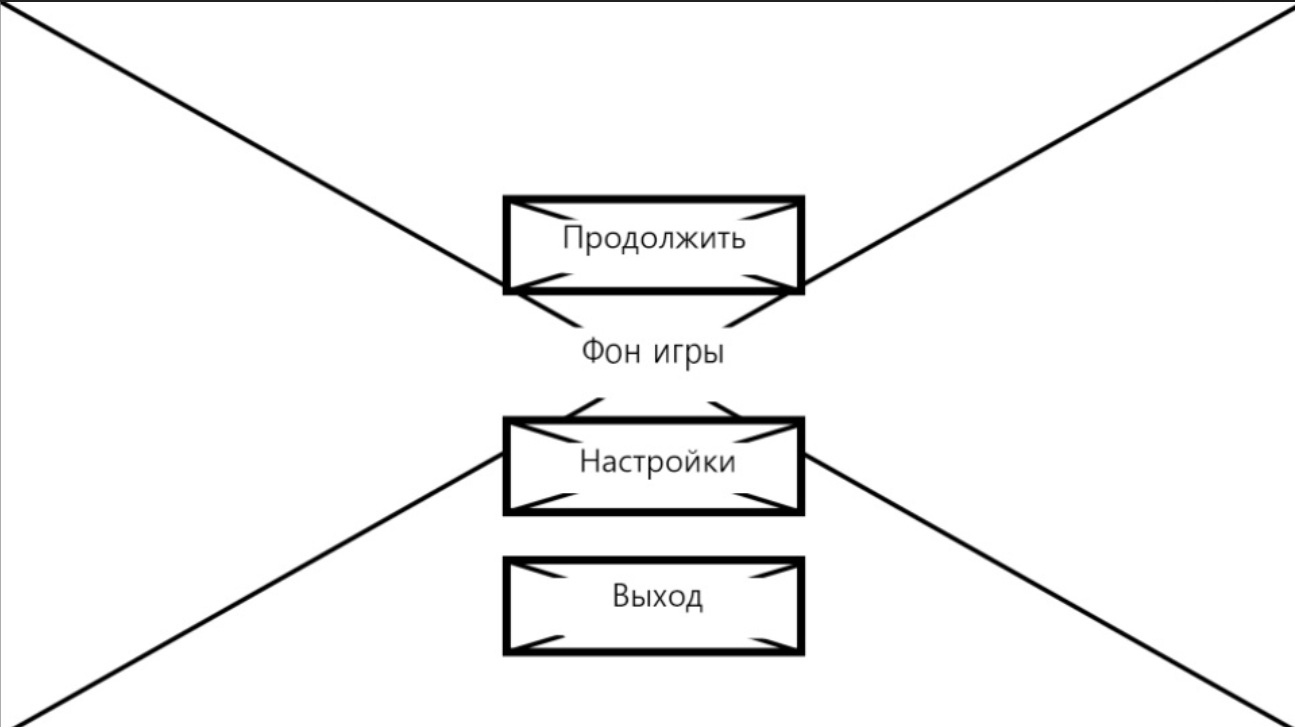


Рисунок 20 – UX для меню паузы

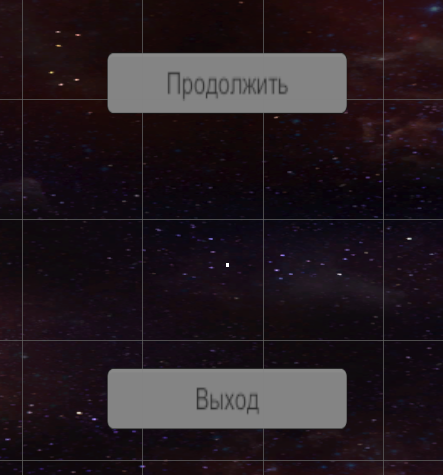


Рисунок 21 – UI меню паузы