**1 Постановка задачи**

**1. Организационно-экономическая сущность задачи**

Наименование задачи: мобильная игра Roguelike-Arkanoid “RogArk”

Цель разработки: создание уникального и захватывающего игрового процесса, сочетающего в себе элементы случайной генерации, процедурно генерируемых уровней и управления мячом.

Назначение: Данная игра в может заинтересовать широкую аудиторию игроков, включая фанатов рогаликов, аркадных игр, ищущих новый игровой опыт, а также мобильных игроков.

Периодичность использования: в зависимости от пользователя, это может быть как каждый день, так и раз в неделю и т.п. Кроме того, периодичность использования может зависеть от количества свободного времени и вовлечённости в игровой процесс.

Источники и способы получения данных: интернет, различная документация, а также на примерах существующих аналогичных проектов в том же жанре.

Обзор существующих аналогичных ПП:

Для того, чтобы рассмотреть похожие ПП нужно выделить то, что мы взяли от жанра, а также дополнили. Рассмотрим следующие три игры: The Binding of Isaac, Slay The Spire, а также классический аркадный Arkanoid. В двух представителях жанра Roguelike (Slay The Spire, The Binding of Isaac) присут-ствует процедурная генерация, которая создаёт этаж с десятками комнат, которые пользователю необходимо пройти в течении забега. В Slay The Spire всего три этажа, в конце каждого босс, точно также и в The Binding of Isaac, но этажей там больше. В этих играх есть комната «магазин», в котором можно потратить заработанную валюту с противников/комнат. Но эти комнаты проходятся по-разному, в Slay The Spire у вас есть колода карт, которую в те-чении забега вы дополняете новыми карточками, с помощью которых вы ата-куете, защищаетесь, либо же получаете какой-то бонус. А вот в The Binding of Isaac вы можете управлять своим персонажем, передвигаться во всех 8 направлениях. По течению забега в обеих играх, вы можете получить предметы, которые будут как-то влиять на геймплей. Сутью жанра Roguelike явля-ется постепенное развитие навыка прохождения игры, а также открытия ка-ких-то заблокированных активностей. Теперь мы должны перейти к игре Arkanoid, там есть движущаяся в горизонтальном направлении платформа, а также шарик, который отбиваясь от платформы должен уничтожить «кирпичи», в этой игре присутствуют какие-то временные улучшения, который могут как-то повлиять на геймплей.

Геймплей:

Общее описание игры:

Конечная цель игры - пройти все уровни, собрать максимальное количество монеток и победить всех противников. Игрок управляет движущейся плат-формой, на которой находится шарик, который можно отбрасывать. Шарик может атаковать противников и собирать монетки и сердца. В каждом забеге генерируются 5 разных этажей с 20 уровнями на каждом из них. На каждом уровне есть несколько видов комнат: комната с противниками, которых нуж-но будет убить; комната с магазином, где можно купить здоровье или пред-меты для стрельбы, платформы или на противника; комната с элитным про-тивником. После прохождения забега, убийства босса и элитных противников игрок получает "души", валюту для метапрогрессии, которая позволяет от-крывать так называемые "классы", за которые можно будет играть: Пулемет-чик (много пуль, но меньше урона), Снайпер (меньше пуль, но выше точ-ность и урон), штурмовик (среднее кол-во пуль). В игре есть возможность собрать до 100 различных предметов, которые влияют на стрельбу, платфор-му или на противника. Например, предмет, который позволяет шарику стре-лять двумя выстрелами одновременно. Игра предоставляет возможность иг-рать за различные классы и выбирать режим игры в зависимости от уровня сложности, что делает ее интересной и разнообразной для игроков всех уров-ней.

Моя часть:

Магазин: на экране у игрока будет появляться некоторое количество предме-тов разной редкости на выбор, так же представлены их характеристики и сто-имость. Игрок сможет выбрать предмет и у него на экране появится сообще-ние о подтверждении покупки, после подтверждения предмет появится в ин-вентаре у игрока. Таких комнат будет от 1 до 3

Сокровищница: В комнате будет находиться только сундук или несколько, игрок может его открыть и получить от туда как что-то полезное(предметы, ресурсы), так и что-то, что может ему навредить(урон, враг). Но так же игрок может не открывать сундук, если посчитает это рискованным или неуместным т просто пропустить комнату. Комнат будет 1-4

Комната отдыха(лагерь): В комнате игроку будет предоставлен выбор между пополнением здоровья и получением бонуса(какого-либо улучше-ния/предмета или ресурсов) от 3 до 5 на этажах и обязательно перед боем с боссом.

В игре будут элементы жанра Arkanoid: Платформа, которая может двигаться по горизонтали и отталкивать от себя шарик. Платформа может изменяться в размерах – становиться короче, для уклонения от атак. Шарик будет отталки-ваться от платформы, так же у него будет возможность стрелять, нанося урон врагам.

В игре будут предметы, которые будут облегчать прохождение. Предметы будут повышать какие-либо характеристики шарика(урон, скорость, размер и др.), платформы(здоровье, скорость, размер), так же некоторые предметы бу-дут давать и другие бонусы, которые так или иначе помогают игро-ку(временное бессмертие, эффекты от атак др. )

Предметы:

Shield: предмет даёт игроку дополнительное броню, которая будет действо-вать как дополнительное здоровье и защищать от атак противников.

healing hearts: предмет будет восстанавливать очки здоровья игроку

Power ball: предмет будет увеличивать визуальный размер снарядов, а так же урон, наносимый врагам.

speed potion: предмет будет временно ускорять движения платформы

Double shot : предмет, который будет давать определенный шанс на выстрел сразу 2 снарядами, при этом тратиться будет только 1

Time-slowing amulet: предмет будет немного замедлять врагов и их атаки для того, чтобы игроку стало проще уклоняться и попадать во врагов

poisonous attacks: предмет будет наделять некоторые из снарядов игрока ядом. При попадании таких снарядов во врага, им на протяжении некоторого времени будет продолжать наноситься урон

Ricochet bullets: Предмет будет давать возможность снарядам игрока отталки-ваться от врага к врагу

Invincibility potion: предмет будет давать игроку неуязвимость на небольшое количество времени

hit potion: предмет на некоторое время делает снаряды игрока самонаводящи-мися

friendly sphere: создаёт сферу, которая будет летать около шарика и наносить урон противникам

long shoot: предмет увеличивает дальность поражения снарядами

power stone: предмет уменьшает скорострельность, но увеличивает урон

fire attacks: предмет будет наделять некоторые из снарядов игрока огнём. При попадании таких снарядов во врага, им на протяжении некоторого времени будет продолжать наноситься урон

220 volt: раз в определенное время враги, находящиеся рядом с шаром будут поражаться током(им будет наноситься небольшой урон)

Health life: предмет добавляет место для дополнительной единицы здоровья

LIFE??: имея этот предмет, после смерти игрок возрождается, но имеет толь-ко 1 единицу здоровья

power engineer: предмет увеличивает скорость движения снарядов

lightning: предмет увеличивает скорость движения платформы

reducing ring: предмет уменьшает размер платформы

Пикап: Healing Hearts, Gold

Активный: hit potion, invisibility potion, speed potion

Обычный: shield, power ball, friendly sphere, lightning

Редкий: 220 volt, fire attacks, poisounous attacks, health life

Эпик: Ricochet bullets , Time-slowing amulet, double shot, reducing ring

Босс: Life??

2. Функциональные требования

Описание перечня функций и задач, которые должен выполнять

будущий ПП:

Игрок:

1. Прохождение уровней

2. Сбор предметов и ресурсов

3. Сражение с противниками

4. Сбор информации о предметах и противниках

5. Отдых в лагере

6. Открытие сундуков

7. Совершение покупок в магазине

3. Описание процессов с входной, выходной и условно-постоянной ин-формацией

Выходная:

- Сообщение о смерти.

- Сообщение о получении награды (золото/души/предметы)

- Сообщение о нехватке какого-либо ресурса

- Сообщение о подтверждении покупки в магазине

- Сообщение о проходящем событии

- Сообщение о текущем уровне (при входе в комнату)

- Взаимодействие с лагерем (Выбор между отдыхом/бонусом)

- При пополнении здоровья/пуль, рядом с игроком отображается кол-во вос-становленного здоровья/пуль

- Диалоговое сообщение

Входная информация:

- Выбор варианта ответа в диалоге, событии

- Согласие/несогласие о подтверждении покупки/исцелении

- Выбор следующего уровня на карте

Условно-постоянная информация:

- В зависимости от предметов и типа противника рядом с ним будет отобра-жаться полоска здоровья, а также нанесённый урон

- Состояние здоровья/боеприпасов

- Количество ресурсов и предметов, а также информация о них

- Доступ к карте

- Настройки

- Информация о текущем уровне

4. Эксплуатационные требования:

Требования к применению: помогает обеспечить досуг игрокам, поднять настроение, обеспечить увлекательное занятие, возможно и на постоянной ос-нове.

Требования к реализации: Реализация игры в жанре roguelike с элементами арканоида требует увлекательной игровой механики, качественной графики и звука, удобного интерфейса, надежных языков и фреймворков программиро-вания, хорошо структурированного кода и документации, а также обширного тестирование. Это все будет реализовано при помощи C# и игрового движка Unity.

Требования к надежности: В проекте будет обеспечена стабильность, произ-водительность, совместимость, безопасность, обработка ошибок, тестирова-ние, обслуживание и техподдержка.

Требования к интерфейсу:

Главное меню включает в себя вкладки: Играть, Настройки, Справка, Выход из игры, Средства связи.

В самой игре экран будет разделён на 2 части: В верхней части игрового экрана должна отображаться информация о текущем уровне, количестве жизней, предметах, карта, а также настройки. В нижней части экрана должно быть размещено игровое поле, на котором игрок будет двигаться и сражаться с противниками с помощью шарика.

Системные требования:

Минимальные:

Операционная система: Android 10.0 или более поздняя версия

Процессор: 64-битный процессор ARMv8-A или более поздний

Оперативная память: 2 ГБ и более

Рекомендуемые:

Операционная система: Android 10.0 или более поздняя версия

Процессор: 64-битный процессор ARMv8-A с тактовой частотой 1,8 ГГц или более быстрый

Оперативная память: 3 ГБ и более

**2. Проектирование задачи**

Разработанные UML-диаграммы представлены в приложении А.

Для проекта разработаны UX и UI макеты, они представлены в приложении Б

**3.Руководство программиста**

**3.1 Общие сведения о программном продукте**

Программный продукт, разработанный под названием "RogArk", представляет собой инновационную мобильную игру, специально создан-ную для развлечения и отдыха пользователей. Основной целью разработ-чиков является выпуск уникального продукта, способного заявить о себе на рынке мобильных игр.

"RogArk" адресован широкому кругу пользователей, готовых по-грузиться в увлекательный мир данного проекта. Однако основную целе-вую аудиторию составляют люди, увлеченные мобильными играми, кото-рые ищут новые и захватывающие впечатления.

Минимальные системные требования для комфортного использова-ния включают в себя следующее:

- Операционная система Android 10.0.

- Минимум 2 гигабайта оперативной памяти.

- Процессор mediatek helio g37.

- Архитектура ARM 64.

Программа "RogArk" обладает широким спектром возможностей, предоставляя пользователям захватывающие игровые сценарии и разно-образные функциональные особенности. Игра призвана стать идеальным источником развлечения и отдыха, поглощая внимание пользователей сво-им уникальным геймплеем и высококачественной графикой.

Периодичность использования "RogArk" зависит от индивидуаль-ных предпочтений пользователей, предоставляя возможность погружения в игровой мир как по мере наличия свободного времени. Программный продукт идеально функционирует в мобильной среде, обеспечивая плав-ный геймплей и отзывчивость.

Дополнительные рекомендации включают в себя обеспечение доста-точного объема внешней памяти для хранения данных игры, а также ис-пользование устройств с соответствующими мониторами и периферийным оборудованием для оптимального визуального и управленческого опыта.

**3.2 Организация данных**

Определив атрибуты для каждой сущности и установив связи между ними, получили следующую схему данных (рисунок 1).

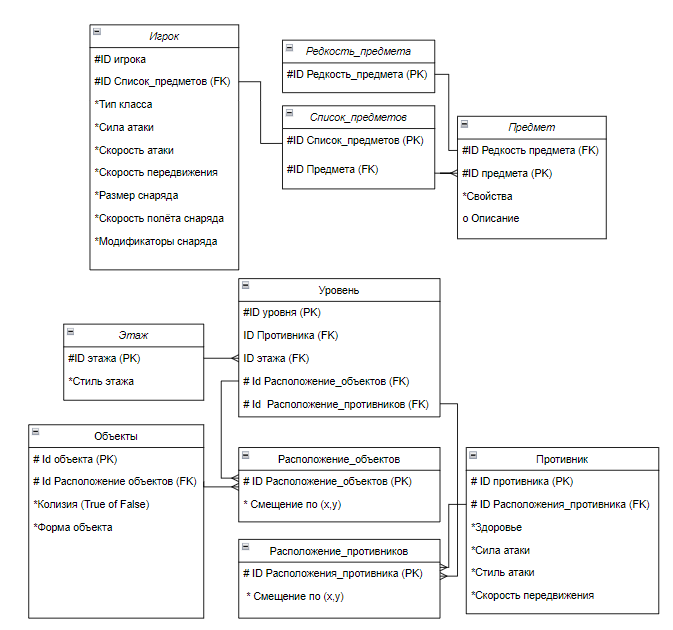


Рисунок 1 - Модель данных проекта

**3.3 Структура программы**

Cтруктура программы, а именно модули и их описание представленны в таблице 1

|  |  |  |
| --- | --- | --- |
| Таблица 1 – Структура программы | | |
| № | Название модуля | Описание |
| 1 | 2 | 3 |
| 1 | Ball.cs | Скрипт для мяча |
| 2 | Bullet.cs | Cкрипт для пуль |
| 3 | CollectableController.cs | Cкрипт для собираемых предметов |
| 4 | BaseEnemy.cs | Общий скрипт для противников |
| 5 | ChildMoving.cs | Скрипт, наследующий движение |
| 6 | FirstEnemy.cs | Скрипт для 1 вида соперника |
| 7 | SecondEnemy.cs | Скрипт для 2 вида соперника |
| Продолжение таблицы 1 | | |
| 1 | 2 | 3 |
| 8 | SmileController.cs | Скрипт для изменения лица противников при попадании |
| 9 | Spawn.cs | Скрипт появления противников |
| 10 | Platform.cs | Скрипт платформы |
| 11 | GameController.cs | Общий скрипт организации логики игры |
| 12 | gameplay | Сцена с игровым уровнем |
| 13 | MainMenu | Сцена с меню приложения |

Все файлы спрайтов, префабов, звуков и другие файлы UI находятся в разных папках по структуре и в папке “PNG”

**3.4 Структура и описание процедур и функций пользователя**

Все процедуры, используемые в программе находятся в таблице 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица 2 – Процедуры и функции | | | | |
| № | Имя процедуры | В каком модуле находится | За каким элементом управления закреплена | Назначение |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Awake | GameController, collectableController, BaseEnemy | - | Выполняет действия перед полным запуском игры |
| 2 | FixedUpdate | GameController, BaseEnemy, Spawn | - | Выполняет действия каждый кадр |
| 3 | DamagePlayer(damage) | GameController | - | Обеспечивает нанесение урона противнику |
| Продолжение таблицы 2 | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 4 | HealPlayer(HealPoints) | GameController | - | Производит исцеление игрока |
| 5 | addStats(int healthAmount = 0, float BallSpeedAmount = 0, float PlatformSpeedAmount = 0, float bulletSize = 0, int score = 0) | GameController | ball | Добавляет определенные статы шарика |
| 6 | ChangeScoreAmount(int amount) | GameController | Text | Обновление счета |
| 7 | ChangePlatformSpeed(int amount) | GameController | Platform | Изменение скорости платформы |
| 8 | KillPlayer() | GameController | - | Действия после смерти |
| 9 | Resume() | GameController | - | Обновление сцены игры |
| 10 | Start() | Ball, platform, ChildMoving, SmileController | - | Производит действия при запуске игры |
| 11 | OnTriggerEnter2D(Collider2D collision) | Ball, bullet, collectableController, platform, BaseEnemy | Ball, bullet, collectables, platform, enemy | Обрабатывает столкновения |
| 12 | respawnCooldown() | ball | ball | Восстановление шарика после падения |
| 13 | Update() | Ball, bullet, ChildMoving, SmileController | - | Выполняет действия каждый кадр |
| 14 | waitForBall() | bullet | bullet | Ожидает появления мячика на экране |
| 15 | play() | collectableController | - | Запуск звука сбора предметов |
| 16 | MovementLogic() | Platform, BaseEnemy | Platform, enemy | Создает логику движения объектов |
| Продолжение таблицы 2 | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 17 | Damage(int amount) | BaseEnemy | enemy | В ней прописана логика нанесения урона |
| 18 | Attacking() | BaseEnemy, SecondEnemy | enemy | Обработка логики выстрелов врага |
| 19 | MovementDelay() | BaseEnemy | enemy | Задержка перед движением врага |
| 20 | fireDelay() | BaseEnemy | enemy | Задержка перед выстрелом врага |
| 21 | dmgDelay() | BaseEnemy | enemy | После получения урона противник невосприимчив на время |
| 22 | WaitingForContinue() | BaseEnemy | enemy | Ожидание продолжения после падения шарика |
| 23 | Spawning(int Col) | Spawn | enemy | Обеспечивает появление противников |

**3.5 Описание используемых компонентов**

Все используемые компоненты и х описание находятся в таблице 3

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 3 – Использованные компоненты | | | |
| № | Компонент | На какой форме расположен | Назначение |
| 1 | 2 | 3 | 4 |
| 1 | Text | MainMenu, gameplay | Показ текста |
| 2 | Button | MainMenu, gameplay | Кнопки |
| 3 | Audio Sourse | MainMenu, gameplay | Компонент для воспроизведения звука |
| Продолжение таблицы 3 | | | |
| 1 | 2 | 3 | 4 |
| 4 | Line render | MainMenu, gameplay | Для “хвоста” у шарика |
| 5 | Main Camera | MainMenu, gameplay | Камера для отображения сцены на экране |
| 6 | GameObjects | MainMenu, gameplay | Все остальные игровые объекты |
| 7 | Canvas | MainMenu, gameplay | Для отображения всех UI элементов |
| 8 | Event System | MainMenu, gameplay | Для обработки всех событий на сценах |

**3.6 Спецификация программы**

Все файлы, используемые в приложении описаны в таблице 4

|  |  |  |
| --- | --- | --- |
| Таблица 4 – Файлы используемые в приложении | | |
| № | Имя файла | Назначение |
| 1 | RogArk.apk | Исполняемый файл проекта, используется для запуска программы на выполнение |
| 2 | \*.png | Спрайты и UI интерфейс |
| 3 | \*.cs | Скрипты |
| 4 | \*prefabs | Префабы |
| 5 | \*.mp4 | Звуки |
| 6 | \*Physicsmaterial2D | Настройки физики для объекта |
| 7 | MainMenu | Сцена меню |
| 8 | gameplay | Сцена уровня |

**4. Тестирование**

**4.1 Тесты на использование**

Все тесты проекта представлены в приложении В.

**4.2 Отчёт о результатах тестирования**

При разработке данного проекта многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения реализации было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов в автономном режиме. Все элементы были проверены, и было установлено, что все они работают правильно и выполняют поставленные задачи.

**5. Руководство пользователя**

**5.1 Запуск программы**

Инсталяция происходит через .apk файл, его нужно запустить на своём устройстве и после всех подтверждений он начнёт установку приложения на ваше устройство. После установки ярлык приложения появится у вас ан экране.

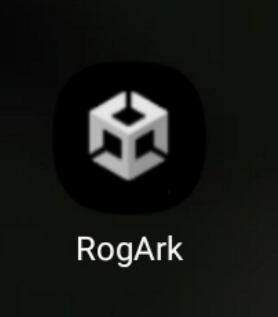
Запустить программу можно нажатием на ярлык на вашем устройстве (рисунок 2)

Рисунок 2 - Ярлык приложения

**5.2 Инструкция по работе с программой**

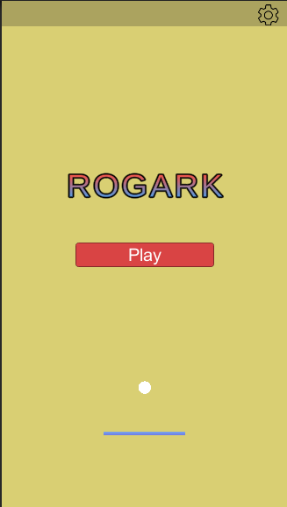
После загрузки открывается главное меню приложения (рисунок 3)

Рисунок 3 - Главное меню

При нажатии на кнопку настроек в правом верхнем углу, откроется меню настроек(рисунок 4, рисунок 5)

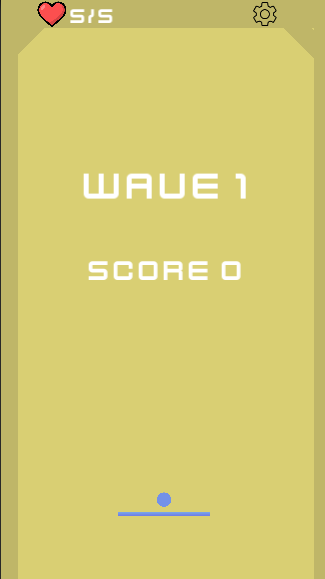
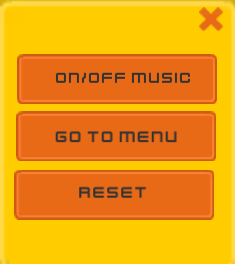
После нажатия на кнопку «Play» игрок переходит на игровой уровень (рисунок 6)

Рисунок 4 - Настройки в главном меню

Рисунок 6 - Игровой уровень

Рисунок 5 - Настройки в игровой сцене

Тут отображается здоровье, кнопка настроек и счёт. Так же в начале каждой волны пишется её номер.

После запуска на сцене появляются противники(рисунок 7, рисунок 8)

Рисунок 7 - Противник №1

Рисунок 8 - Противник №2

Чтобы начать игру, нужно оттянуть шарик и отпустить его в любую сторону. Чтобы убить противника, нужно попасть по нему шариком 4 раза, после 3 попадания противник становится неактивным и его нужно просто добить.

С каждой волной противники становятся сильнее и их становится больше

Так же из них могут выпасть предметы: сердечко(для исцеления), увеличение скорости платформы, увеличение скорости шарика и монетка(добавление к счёту)

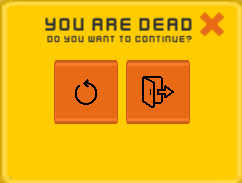
У игрока всего 5 жизней, при падении шарика и при попадании противника в платформу, отнимается 1 жизнь. Когда все жизни закончились появляется окно(рисунок 9), где можно выйти в меню или продолжить. Так же уровень можно перезапускать в настройках и там же есть возврат в меню и выключение звука.

Рисунок 9 - Меню после смерти

Завершить работу с приложением можно с помощью кнопки в настройках главного меню(рисунок 4), а так же просто закрыв приложение.

**Приложение А**

Структура проекта

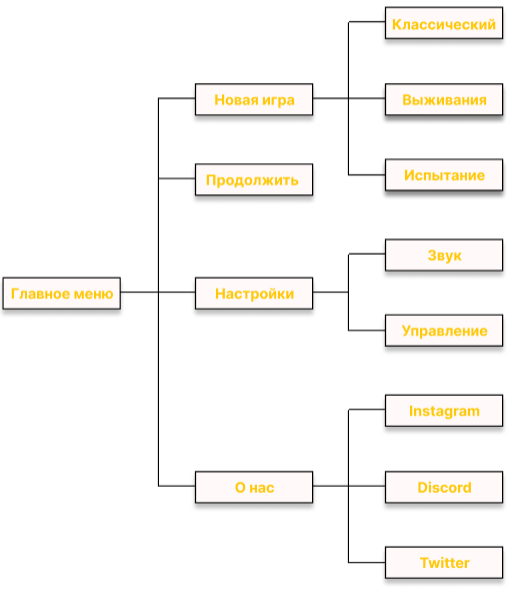


Рисунок А.1 – Структура сайта

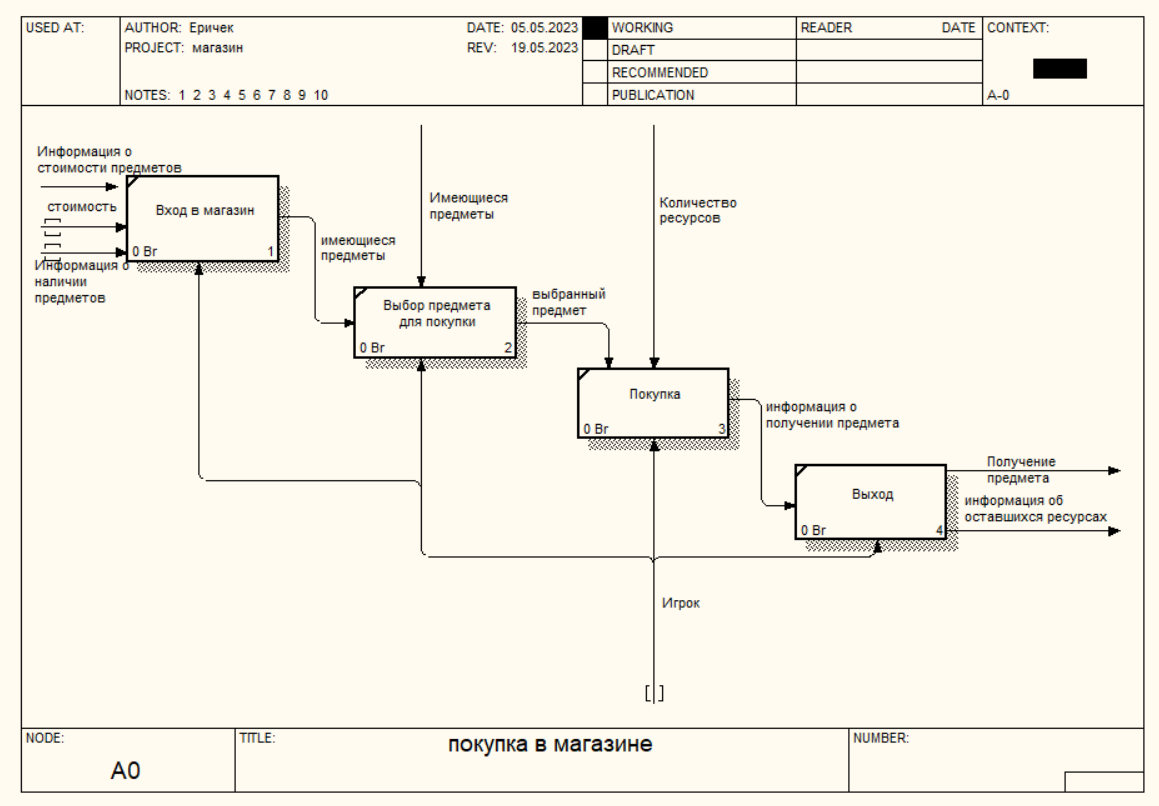


Рисунок А.2 – Функциональная модель

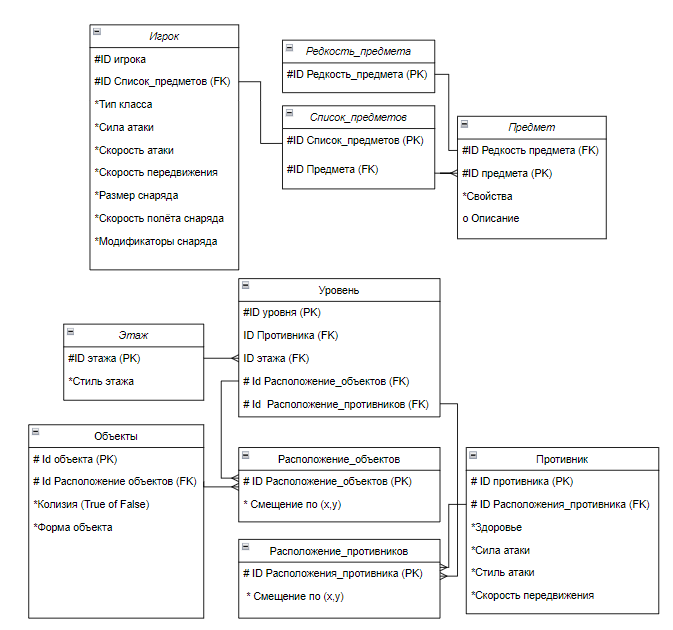


Рисунок А.3 – Модель данных

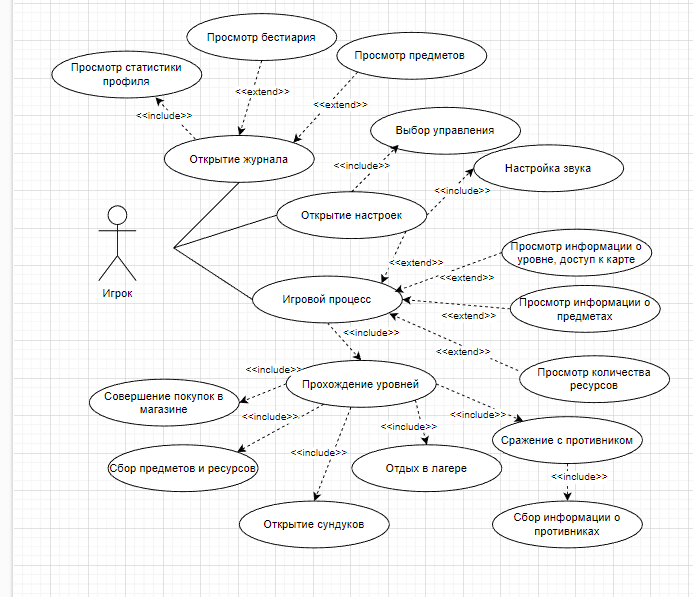
****

Рисунок А.4 – Диаграмма вариантов использования

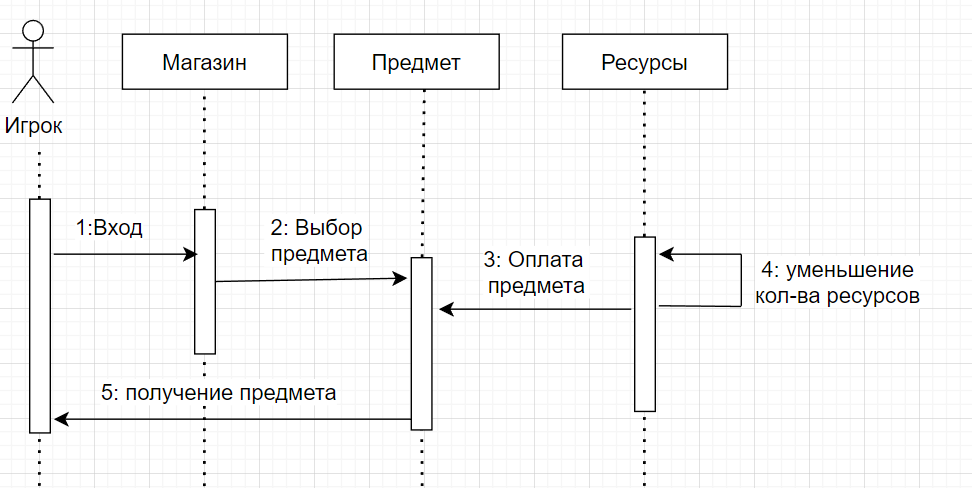


Рисунок А.5 – Диаграмма последовательности

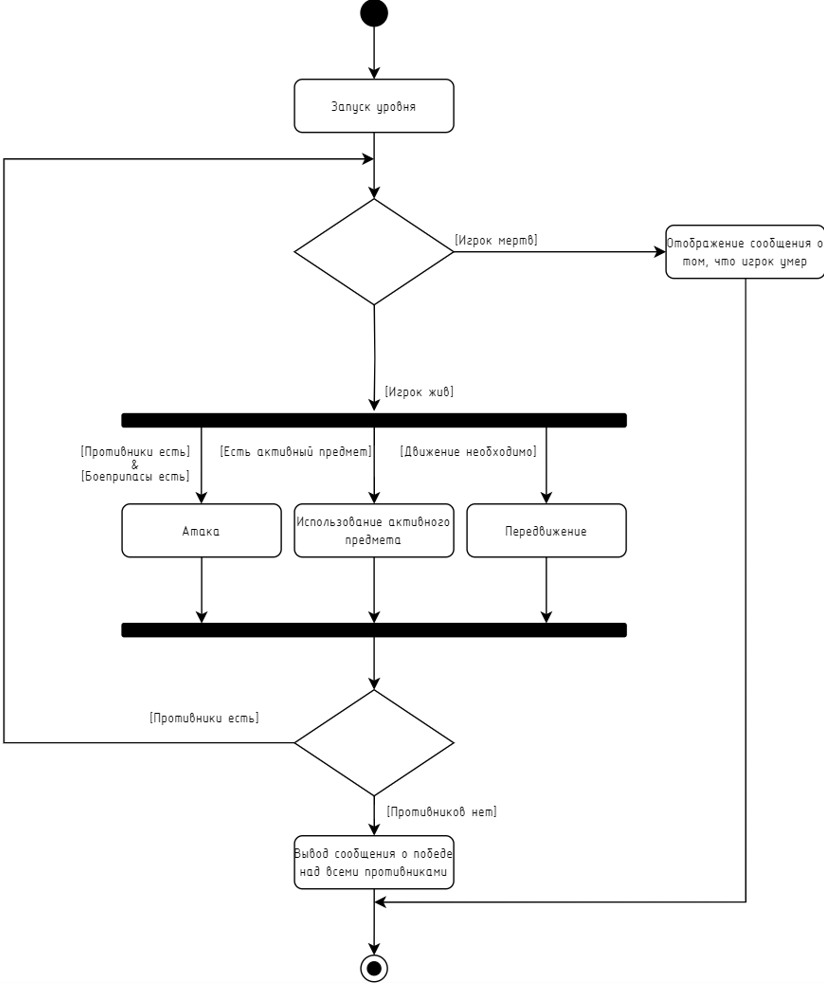


Рисунок А.6 – Диаграмма деятельности

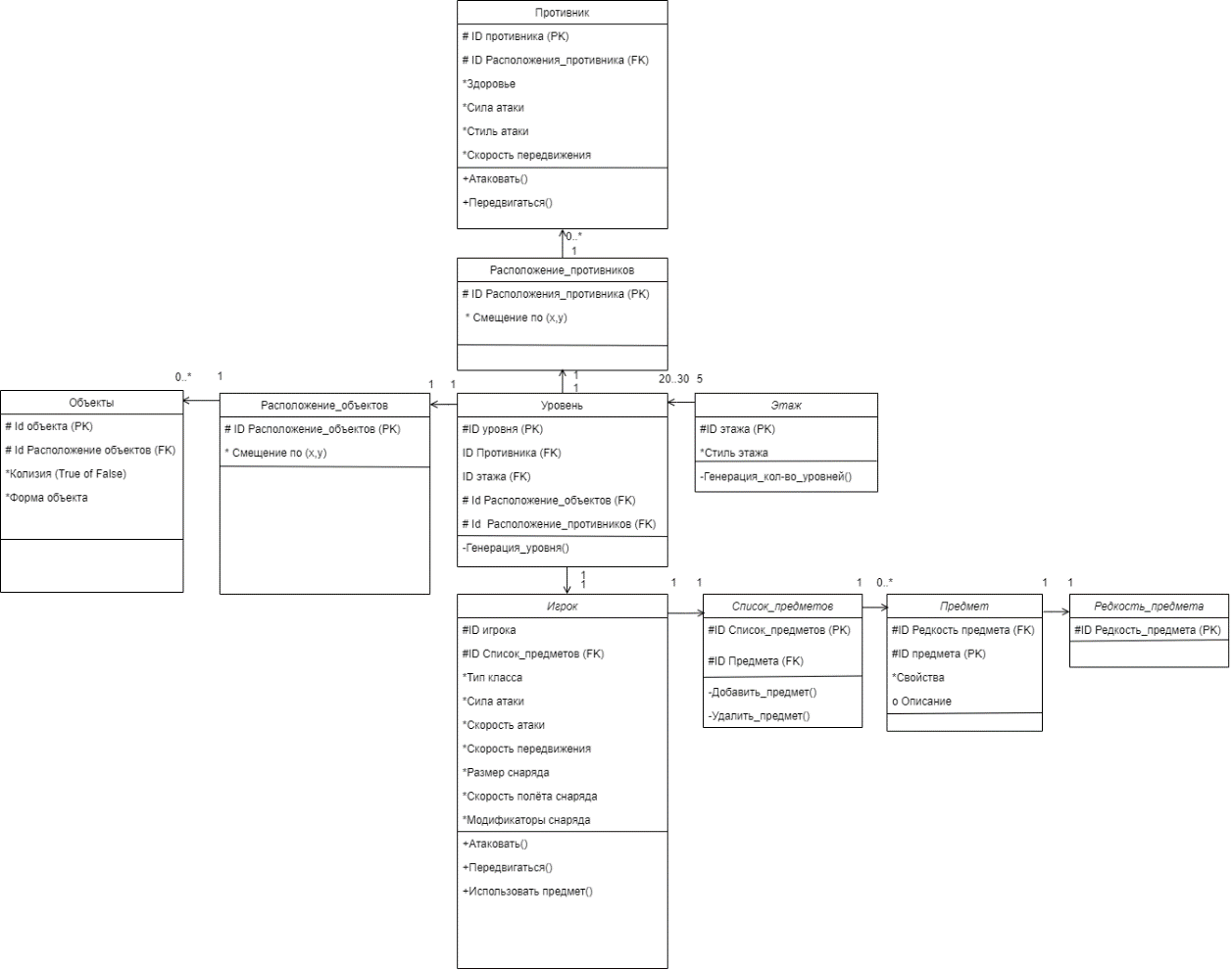


Рисунок А.7 – Диаграмма классов

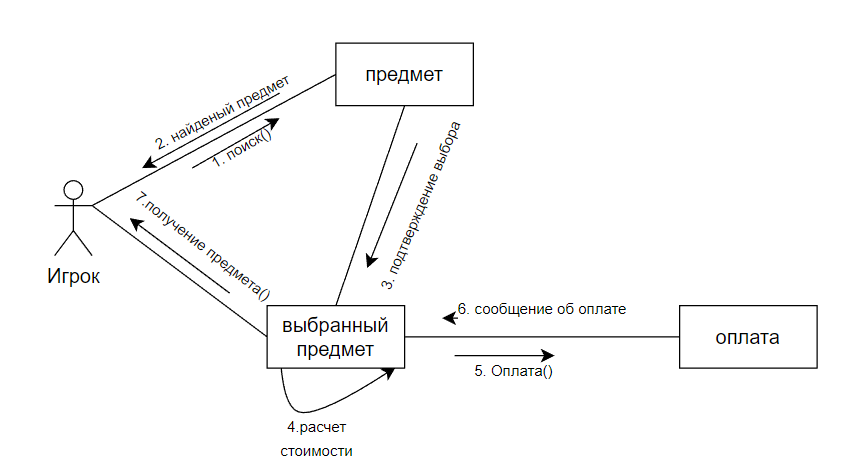


Рисунок А.8 – Диаграмма объектов

**Приложение Б**

UX, UI дизайн



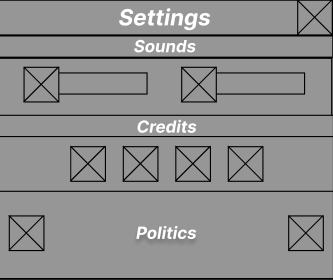
Рисунок Б.1 - UX главной формы

Рисунок Б.3- UX настроек

Рисунок Б.2 - UX - игрового уровня

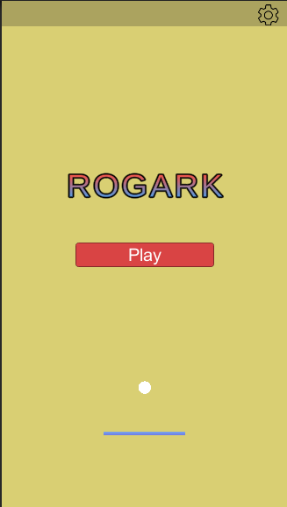
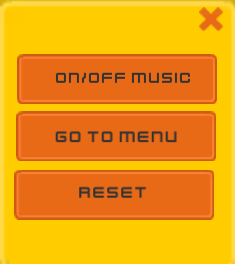
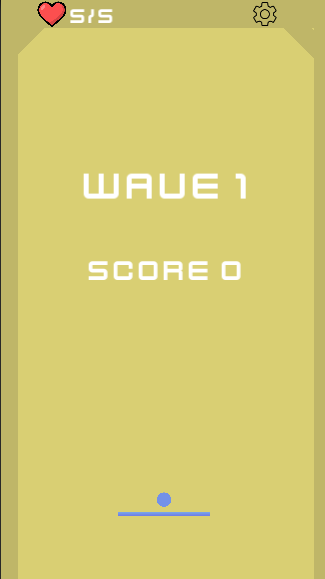


Рисунок Б.7 - UI игрового уровня

Рисунок Б.5 - UI настроек игрового уровня

Рисунок Б.6 - UI настроек

Рисунок Б.4 - UI меню

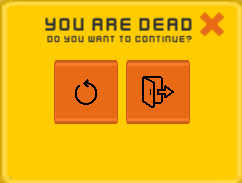


Рисунок 7 - UI меню после смерти

**Приложение В**

Тестирование

Рисунок В.1 - Тестирование

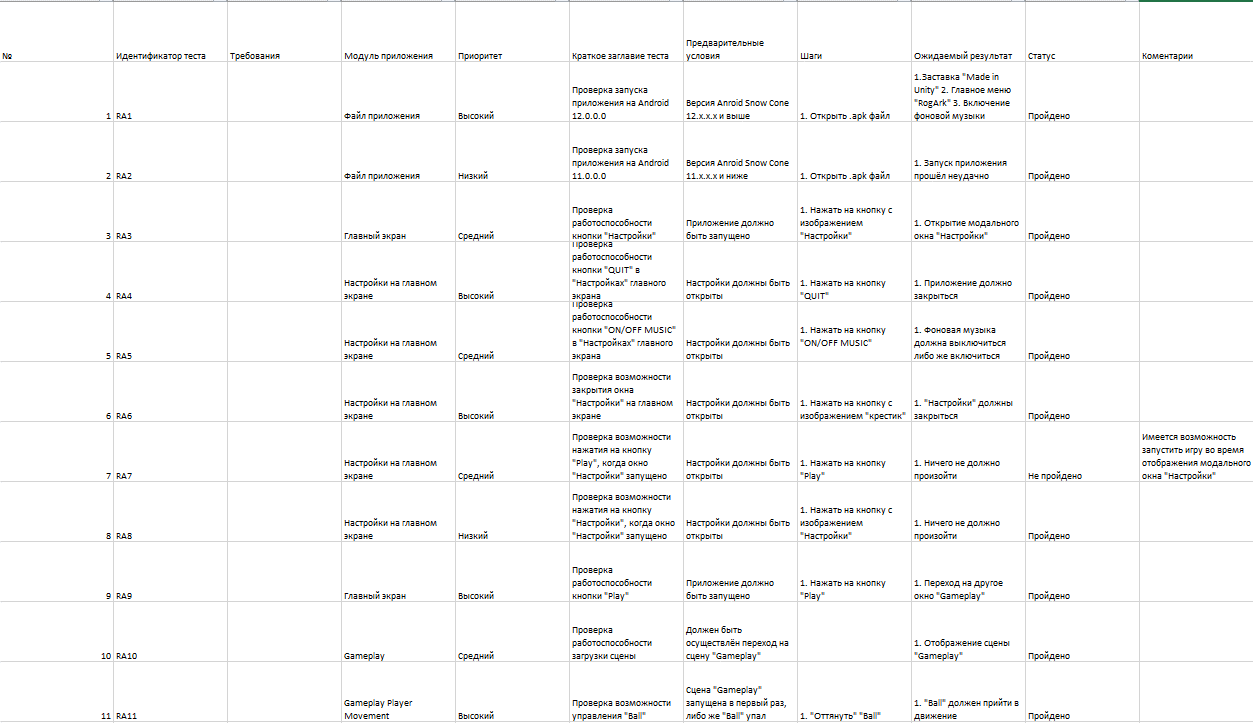
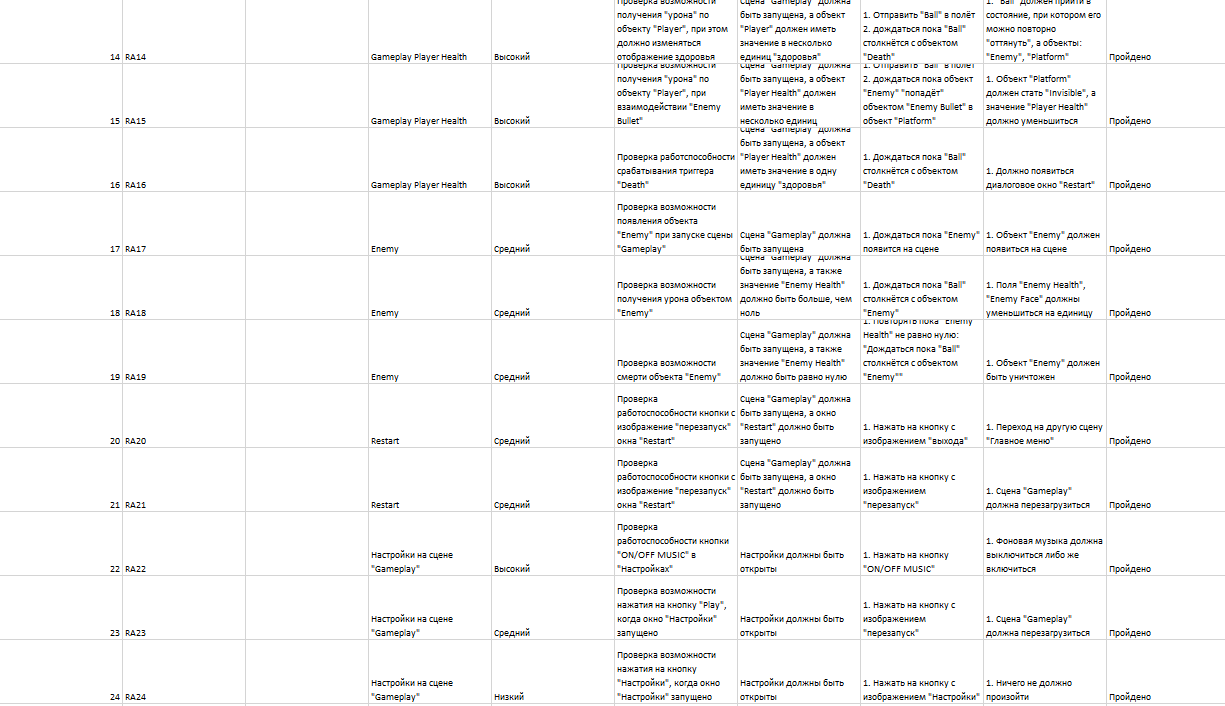


Рисунок 8 - Тестирование