

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

«___»_____ 2024 г.

**Разработка игры в жанре «Top-Down Shooter»
на платформе Unity**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2024.308-318.ВКР

Научный руководитель,
доцент кафедры СП, к.т.н.
_____ М.В. Сухов

Автор работы,
студент группы КЭ-401
_____ Р.В. Шайхулин

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
«___»_____ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
29.01.2024 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра
студенту группы КЭ-401
Шайхулину Роману Валерьевичу,
обучающемуся по направлению
02.03.02 «Фундаментальная информатика и информационные технологии»

1. Тема работы (утверждена приказом ректора от 22.04.2024 г. № 764-13/12
Разработка игры в жанре «Top-Down Shooter» на платформе Unity.

2. Срок сдачи студентом законченной работы: 03.06.2024 г.

3. Исходные данные к работе

3.1. UnityHub. [Электронный ресурс] URL: <https://unityhub.ru> (дата обращения: 19.01.2024 г.).

3.2. Unity Documentation. [Электронный ресурс] URL: <https://docs.unity.com> (дата обращения: 17.01.2024 г.).

3.3. Unity User Manual 2021.3 (LTS). [Электронный ресурс] URL: <https://docs.unity3d.com/Manual/Unity2D.html> (дата обращения: 19.01.2024 г.).

3.4. Wolf M. J. P., Perron B. The video game theory reader. – Routledge, 2013. – 164 p.

4. Перечень подлежащих разработке вопросов

4.1. Проанализировать предметную область и инструменты разработки на Unity.

- 4.2. Спроектировать концепцию игры.
- 4.3. Реализовать механики и контент игры.
- 4.4. Выполнить тестирование программного продукта.
- 5. Дата выдачи задания: 29.01.2024 г.**

Научный руководитель,
доцент кафедры СП, к.т.н.

М.В. Сухов

Задание принял к исполнению

Р.В. Шайхулин

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1. Описание предметной области	7
1.2. Анализ существующих решений для реализации проекта.....	10
2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	11
2.1. Концепция игры	11
2.2. Функциональные и нефункциональные требования.....	12
2.3. Диаграмма вариантов использования	13
3. АРХИТЕКТУРА СИСТЕМЫ.....	14
3.1. Общее описание архитектуры системы.....	14
4. РЕАЛИЗАЦИЯ СИСТЕМЫ	17
4.1. Реализация компонентов системы	17
4.2. Реализация пользовательского интерфейса	22
5. ТЕСТИРОВАНИЕ	26
ЗАКЛЮЧЕНИЕ	29
ЛИТЕРАТУРА.....	30
ПРИЛОЖЕНИЯ.....	32
Приложение А. Спецификация вариантов использования.....	32
Приложение Б. Скриншоты итоговой версии игры	37

ВВЕДЕНИЕ

Актуальность темы

Индустрия видеоигр сегодня пользуется большой популярностью как среди игроков, так и среди разработчиков.

Видеоигры помогают игрокам отдохнуть и развлечься, погружают их в увлекательные миры и сюжеты, снимают стресс, увлекают игровым процессом. Также благодаря интерактивности игры способны подарить эмоции, которые нигде больше человек получить не сможет.

Игры могут быть не только развлечением, но и способом развить ментальные навыки, такие как концентрация, стратегическое мышление и реакция.

Разработчикам же открыт простор для творчества. Создать собственную игру могут не только крупные компании с огромными ресурсами, но и небольшие студии, и даже один человек. Игроки тепло воспринимают работы разных направлений и с разными бюджетами.

В ходе развития индустрии видеоигр сформировались огромное количество жанров: от стратегий и ролевых игр до спортивных симуляторов и приключенческих игр.

Одним из первых жанров видеоигр стали top-down шутеры. Они имеют долгую историю, и для многих игроков они приносят ностальгические воспоминания и ощущение классического игрового опыта.

Механики таких игр зачастую просты, игровой процесс динамичен, поэтому они легки в освоении и не требуют глубокого изучения.

Постановка задачи

Целью выпускной квалификационной работы является разработка компьютерной игры в жанре «Top-Down Shooter» на платформе Unity. Для достижения поставленной цели необходимо решить следующие задачи.

1. Провести анализ предметной области.
2. Спроектировать компьютерную игру.
3. Реализовать компьютерную игру.

4. Провести тестирование компьютерной игры.

Структура и объем работы

Работа состоит из введения, пяти глав, заключения и списка литературы. Объем работы составляет 41 страницу, объем списка литературы – 15 источников.

В первой главе «Анализ предметной области» описываются особенности игрового жанра «Top-Down Shooter», ключевые свойства разрабатываемой игры и схожие проекты.

Вторая глава «Проектирование системы» посвящена описанию функциональных и нефункциональных требований, а также проектированию и описанию диаграммы вариантов использования.

В третьей главе «Архитектура системы» описываются общая архитектура системы, представлена диаграмма компонентов, и каждый компонент подробно описан.

В четвертой главе «Реализация системы» подробно описана реализация всех компонентов игры и пользовательского интерфейса.

В пятой главе «Тестирование» произведено тестирование разработанной компьютерной игры на основе функционального и юзабилити тестирования.

В приложении А содержится подробное описание вариантов использования, представленные во второй главе. В приложении Б содержатся скриншоты итоговой версии игры.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание предметной области

«Top-Down Shooter» – это 2D игра с видом сверху, где главная задача игрока – побеждать врагов. Основа таких игр – простота и динамичность. В них нет сложных механик, поэтому они легки в освоении. По большей части от игрока требуется перемещаться по игровой арене, избегать получения урона и наносить урон самому [1]. На этом и строится основной игровой процесс.

Такие игры не требуют глубокого изучения и умственных усилий, они требуют реактивной работы мозга. Игроку не нужно держать в голове огромное количество информации, а нужно грамотно реагировать на происходящее внутри игровой арены.

Игры в жанре «Top-Down Shooter» зачастую лишены глубокого сюжета, ведь он им просто не нужен [13].

Жанр зародился ещё в 1962 году [3, 12] с игры «SpaceWar». Далее жанр получил активное распространение, и были созданы многие знаковые для индустрии игры [2, 7].

1. «Tank» 1974 года – аркада для игровых автоматов на двух игроков, выпущенная компанией «Atari» (рисунок 1). Белый и чёрный танк сталкиваются в лабиринте из блоков и мин. Игрокам начисляются очки за уничтожения танка соперника. Побеждает игрок, набравший наибольшее количество очков [6].

2. «Galaxian» 1979 года (рисунок 2) – аркадная игра, выпущенная компанией «Namco». Смысл игры – отражать волны пришельцев, управляя космическим кораблем «Galaxip». С каждой новой волной сложность игры повышается [8].

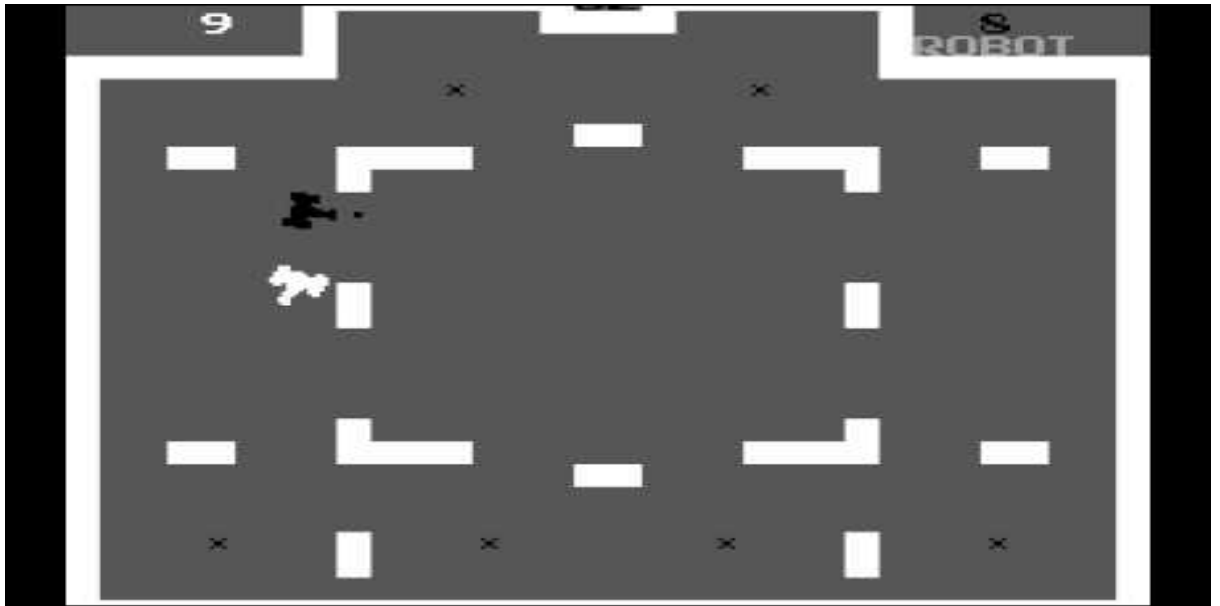


Рисунок 1 – Игра «Tank» 1974 года



Рисунок 2 – Игра «Galaxian» 1979 года

3. «Metal Gear» 1987 года (рисунок 3) – стелс-экшн, разработанный Хидео Кодзимой, про солдата сил специального назначения, которому поручено в одиночку проникнуть в крепость «Outer Heaven» и уничтожить находящийся в ней «Metal Gear» – двуногий танк, способный осуществлять ядерные удары из любой точки мира [4].

А также «Gauntlet», «Commando», «GTA» и многие другие.



Рисунок 3 – Игра «Metal Gear» 1987 года

Жанр пользуется популярностью и по сей день. Среди вышедших за последние годы top-down шутеров можно выделить несколько достойных представителей.

1. «The Ascent» 2021 года.
2. «Brotato» 2022 года.
3. «20 Minutes Till Dawn» 2022 года.
4. «Remedium: Sentinels» 2023 года.
5. «Tiny Rogues» 2022 года.
6. «Intravenous» 2021 года.

Причина, по которой игры в жанре «Top-Down Shooter» остаются актуальными до сих пор – простой и понятный, но в то же время затягивающий динамичный игровой процесс.

От игрока требуется проявлять тактическое мышление: грамотно позиционироваться на поле боя, выбирать цели для атаки и быстро реагировать на действия противника [14]. В зависимости от количества противни-

ков, их позиций и способностей игроку необходимо в реальном времени выбирать оптимальную тактику боя. Top-down шутеры крайне динамичные, и стечение игровых обстоятельств приводит к разнообразным ситуациям, в которых игроку предстоит быстро принимать тактические решения. Именно это формирует у игрока запоминающийся игровой опыт [5]. Часто top-down шутеры перекликаются с другими жанрами: «Bullethell», «Stealth-Action», «Roguelike», «Hack & Slash» и прочие [15].

1.2. Анализ существующих решений для реализации проекта

При создании игры использовалась платформа для разработки Unity [9–11]. Она содержит огромное количество технических средств создания компьютерных игр.

1. Встроенный физический движок Ageia PhysX.
2. Высоко оптимизированный графический «пайплайн».
3. Потокосное аудио и видео.
4. Поддержка трех языков скриптования: JavaScript, C#, Boo.

Игровой движок Unity2D полностью интегрирован в среду разработки Unity. Тесная интеграция позволяет прямо в редакторе получить всё то, на что будет способна выпущенная игра.

Также использовался пакет с оптимизированным алгоритмом поиска A* Pathfinding Project.

Все спрайты в игре были взяты из открытых источников: Unity Asset Store, itch.io.

Вывод по первой главе

В этой главе была проанализирована специфика top-down шутеров, выявлены общие для жанра свойства. Также были рассмотрены прародители жанра и современные его представители. Были проанализированы существующие решения для реализации проекта.

2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1. Концепция игры

Цель игры – отразить как можно больше волн врагов. Волна считается завершенной, когда все враги повержены. Тогда начинается новая волна, но врагов появляется на одного больше. Более того, каждую десятую волну у всех врагов увеличивается запас здоровья и урон от атаки. Также каждую пятую и десятую волну появляются боссы.

Для уничтожения врагов у игрока есть три типа оружия: пистолет, винтовка и дробовик. Они отличаются уроном, скорострельностью и дальностью стрельбы. У пистолета и винтовки после выстрела вылетает по одной пуле в прямом направлении, а у дробовика вылетает пять пуль по дуге.

Изначально игроку доступен только пистолет. Дробовик и винтовку можно купить, при этом доступ к пистолету остается бесплатным.

Винтовка стреляет быстрее и на более дальнюю дистанцию, дробовик стреляет медленнее и на менее дальнюю дистанцию, пистолет является усреднённым вариантом. При этом дробовик выпускает несколько пуль за один выстрел, из-за чего на ближней дистанции имеет наибольший урон.

Враги преследуют игрока по всей арене и обходят препятствия. За их убийство игроку начисляется внутриигровая валюта, которую он может потратить на улучшение характеристик или покупку оружия.

Игроку доступно четыре типа улучшений: увеличение здоровья, урона, скорострельности и регенерации здоровья.

На игровой арене в большом количестве расположены объекты, которые невозможно пройти и прострелить напрямую. Игра построена на том, чтобы метко стрелять во врагов, при этом передвигаться по игровой арене между объектами. Потеряв концентрацию и столкнувшись с таким объектом, игрок рискует быть зажатым, вследствие чего проиграть.

2.2. Функциональные и нефункциональные требования

Разрабатываемое приложение должно удовлетворять следующим функциональным требованиям.

1. Игра должна предоставлять возможность начинать и завершать игровую сессию.
2. Игра должна предоставлять возможность настраивать громкость звука, громкость эффектов и разрешение экрана.
3. Игра должна предоставлять информацию о лучшем игровом результате игрока.
4. Игра должна предоставлять возможность заработка внутриигровой валюты.
5. Игра должна предоставлять информацию о текущих характеристиках персонажа и стоимости их улучшения.
6. Игра должна предоставлять возможность игроку поставить паузу.
7. С каждой новой волной врагов должно становиться больше.
8. На определенных волнах должны появляться боссы.
9. В игре должны появляться временные улучшения – предметы, которые временно повышают скорость, восстанавливают здоровье или дают неуязвимость.

Разрабатываемое приложение должно удовлетворять следующим нефункциональным требованиям.

1. Игра должна быть разработана на платформе Unity.
2. В игре должно быть реализовано главное меню.
3. В игре должно присутствовать несколько видов врагов, в том числе боссов.
4. В игре должны быть реализованы разные виды оружия.
5. В игре должны быть анимации передвижения персонажа, врагов, выстрела из оружия, получения урона у врагов.
6. В игре должна быть реализована миникарта.

2.3. Диаграмма вариантов использования

На рисунке 4 представлена диаграмма вариантов использования игры в жанре «Top-Down Shooter».

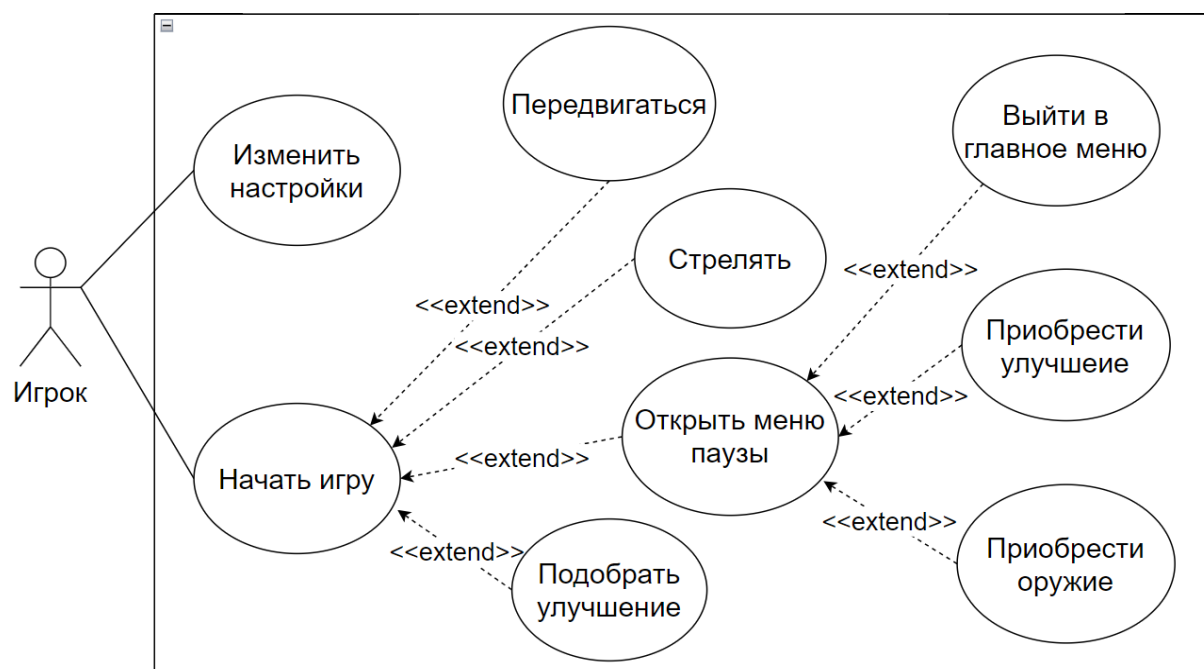


Рисунок 4 – Диаграмма вариантов использования

С системой взаимодействует только один основной актер – игрок.

Игрок – это человек, играющий в игру. При запуске системы первое, что увидит игрок – сцена главного меню, на которой расположены кнопки, отвечающие за начало игры и выхода из нее. В ходе игры игрок может перемещаться по игровой сцене, стрелять и подбирать улучшения. Помимо этого, игрок может изменить игровые настройки и вернуться в главное меню. Спецификации вариантов использования приведены в приложении А.

Вывод по второй главе

В данной главе были описаны функциональные и нефункциональные требования к системе, описан концепт игры на основании первой главы, а также определены и описаны варианты использования.

3. АРХИТЕКТУРА СИСТЕМЫ

3.1. Общее описание архитектуры системы

Игра состоит из двух сцен – главного меню и игровой локации.

Главное меню открывается при запуске игры. Из него игрок может начать игру и перейти на игровую локацию кнопкой «Play», зайти в меню настроек кнопкой «Options» и выйти из игры кнопкой «Quit».

В меню настроек игрок может поменять разрешение экрана, громкость музыки и звуков, а также переключиться с полноэкранного режима на оконный.

На игровой локации игрок получает управление персонажем. Игрок может открыть меню магазина клавишей «Esc». Там ему будут доступны кнопки для улучшения характеристик и покупки оружия. Из меню магазина игрок может выйти в главное меню.

На рисунке 5 представлена диаграмма компонентов игры.

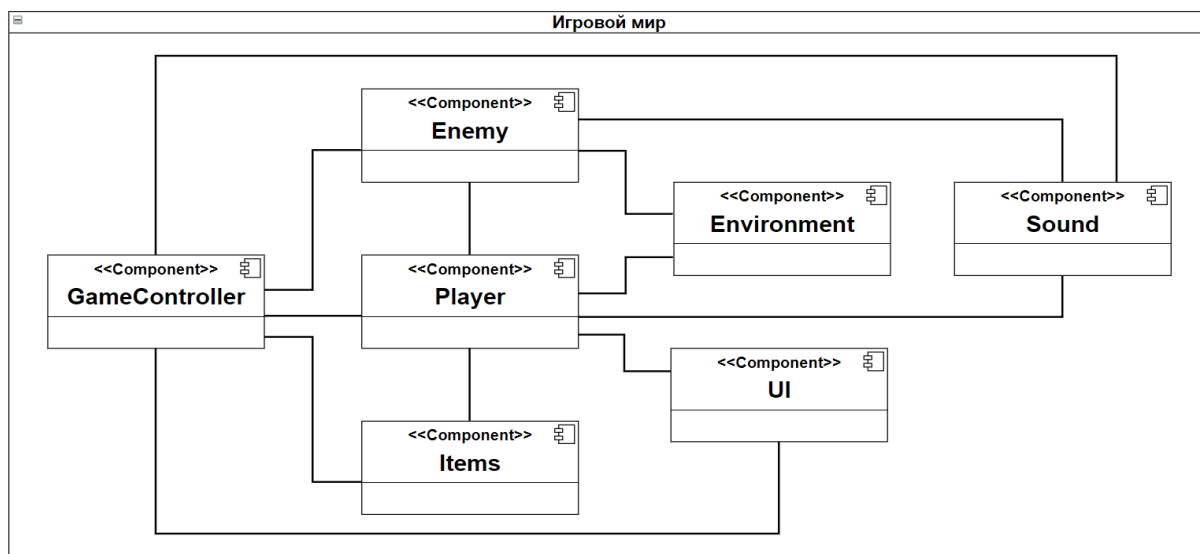


Рисунок 5 – Диаграмма компонентов

Игра состоит из следующих основных компонентов.

1. Player – компонент, предоставляющий игроку возможность управления игровым персонажем.
2. Enemy – компонент, отвечающий за противников.

3. `Items` – компонент, отвечающий за предметы временных улучшений: ускорение, неуязвимость и лечение.

4. `Environment` – компонент, отвечающий за объекты окружения: камни, бревна, траву и т.д.

5. `GameController` – компонент, отвечающий за настройки игрового мира.

6. `UI` – компонент, предоставляющий игроку интерфейс.

7. `Sound` – компонент, отвечающий за звуки в игре.

Разберем более подробно компоненты, составляющие программную систему.

Компонент `Player`

Основной задачей компонента является предоставление игроку возможности управления игровым персонажем. Для этого в компоненте представлена функция перемещения, прицеливания и стрельбы. Также компонент хранит информацию о скорости перемещения, здоровье персонажа, восстановлении здоровья, уроне оружия, скорострельности и дальности стрельбы.

Компонент `Enemy`

Компонент содержит искусственный интеллект, который заставляет врагов преследовать персонажа. Также содержит информацию о врагах: урон от атаки, награда за убийство, здоровье и скорость перемещения.

Компонент `Items`

Компонент содержит информацию о предметах, которые появляются на игровой локации: время неуязвимости, время увеличения скорости, величина увеличения скорости, величина восстановления здоровья.

Компонент `Environment`

Компонент содержит в себе все статичные объекты на игровой карте. Персонаж и враги не могут проходить через эти объекты, а также через них не проходят выстрелы.

Компонент GameController

Компонент содержит информацию о настройках игрового мира: стоимость улучшений, стоимость покупки оружия, начальное количество денег, значение, на которое увеличиваются характеристики при покупке улучшения, количество волн, места появления врагов и лучший игровой результат.

Компонент UI

Компонент предоставляет игроку интерфейс: полоска здоровья, количество денег, текущая волна, миникарта, перекрестье прицела. В меню магазина: стоимость приобретения улучшений и оружия, кнопки для приобретения улучшений и оружия, кнопка выхода. В случае смерти показывает соответствующее сообщение.

Компонент Sound

Компонент отвечает за все звуки в игре. Звуки выстрела, получения урона, смерти врага, появления боссов и саундтрек.

Вывод по третьей главе

Архитектура игры была спроектирована на основе требований к системе, диаграммы функциональных и нефункциональных требований. Был реализован компонент для предоставления игроку интерфейса взаимодействия с игрой. Возможности игрока были спроектированы на основе диаграммы вариантов использования. Также был выделен компонент для настройки игры, чтобы иметь возможность гибко изменять игровой баланс в сторону повышения или понижения сложности.

4. РЕАЛИЗАЦИЯ СИСТЕМЫ

4.1. Реализация компонентов системы

Player

Компонент предоставляет игроку возможность управлять игровым персонажем. Включает в себя три скрипта: `Player`, `Weapon` и `Health`.

В скрипте `Player` содержатся функции для перемещения `MovePlayer`, вращения `RotateBody` (листинг 1) и `RotateFeet`, добавления скорости `TakeSpeed` при подборе предмета временного улучшения и снижения скорости `RemoveSpeed` после того, как действие временного улучшения закончилось.

В скрипте `Weapon` содержатся функции для выстрела `Fire` и создания пули `SpawnBullet`.

В скрипте `Health` содержатся функции для получения урона `TakeDamage`, подбора аптечки `TakeHealth`, подбора брони `TakeArm`, дающей неуязвимость, а также для прекращения действия неуязвимости `RemoveArm` и пассивной регенерации здоровья `HpRegen`. Скрипт `Health` также включает в себя логику для управления здоровьем врагов.

Листинг 1 – Функция вращения модельки персонажа

```
private void RotateBody()
{
    Vector3 difference;
    float distance = Vector3.Distance(gameObject.transform.position,
    Camera.main.ScreenToWorldPoint(Input.mousePosition));

    if (distance > 11f)
    {
        difference = Cam-
era.main.ScreenToWorldPoint(Input.mousePosition) - muzzle.position;
    }
    else
    {
        difference = Cam-
era.main.ScreenToWorldPoint(Input.mousePosition) - transform.position;
    }

    difference.Normalize();

    float rotZ = Mathf.Atan2(difference.y, difference.x) *
    Mathf.Rad2Deg; // find the angle in degrees
    if (top.activeSelf)
        topTorso.transform.rotation = Quaternion.Euler(0f, 0f, rotZ);
    if (topShotgun.activeSelf)
```

```

        topShotgunTorso.transform.rotation = Quaternion.Euler(0f, 0f,
rotZ);
        if (topRifle.activeSelf)
            topRifleTorso.transform.rotation = Quaternion.Euler(0f, 0f,
rotZ);
    }

```

Enemy

Компонент отвечает за противников. Включает в себя три скрипта: `Enemy`, `Seeker` и `Health`. Компонент содержит в себе частичную реализацию искусственного интеллекта (листинг 2) для поиска и преследования игрока в скрипте `Seeker` и `Enemy`. Также в скрипте `Enemy` содержится функция для обработки столкновений с игроком и с другими врагами.

Листинг 2 – Одна из частей реализации поведения врагов

```

void FixedUpdate()
{
    Vector3 vectorToTarget = player.transform.position - trans-
form.position;
    float angle = Mathf.Atan2(vectorToTarget.y, vectorToTarget.x) *
Mathf.Rad2Deg - rotationModifier;
    Quaternion q = Quaternion.AngleAxis(angle, Vector3.forward);
    transform.rotation = Quaternion.Slerp(transform.rotation, q,
Time.deltaTime * speedRotation);

    if (path == null)
    {
        return;
    }
    if (currentWaypoint >= path.vectorPath.Count)
    {
        reachedEndOfPath = true;
        return;
    }
    else
    {
        reachedEndOfPath = false;
    }

    Vector2 direction = ((Vector2)path.vectorPath[currentWaypoint] -
rb.position).normalized;
    Vector2 force = direction * speed * Time.deltaTime;

    if (distance < nextWaypointDistance)
    {
        currentWaypoint++;
    }
}

```

Items

Компонент отвечает за предметы временных улучшений: ускорение, неуязвимость и лечение. Включает в себя скрипт `ItemSpawn`, в котором реализована функция для проверки наличия других улучшений на карте `CheckBoosters`, а также функция для создания на карте `SpawnBoost` (листинг 3).

Листинг 3 – Функция создания на карте предмета временного улучшения

```
void SpawnBoost()
{
    int boosterNumber = UnityEngine.Random.Range(1, 4);
    switch (boosterNumber)
    {
        case 1:
            boost = speed;
            break;
        case 2:
            boost = arm;
            break;
        case 3:
            boost = health;
            break;
        default:
            break;
    }

    Instantiate(boost, new Vector2(gameObject.transform.position.x, gameObject.transform.position.y), new Quaternion(0, 0, 0, 0));
}
```

Environment

Компонент отвечает за объекты окружения: камни, бревна, траву и т.д. (рисунок 6). Он реализован в виде статичных объектов на карте, которые помечены специальным тегом `Environment`. Персонаж, пули и враги не могут проходить через объекты с этим тегом.

GameController

Компонент отвечает за настройки игрового мира. Включает в себя два скрипта: `GameController` и `WaveManager`.



Рисунок 6 – Часть предметов окружения на игровой локации

GameController включает в себя функции для изменения данных в интерфейсе игрока и в меню магазина после убийства врага, приобретении улучшения или смены оружия. GameController отвечает за изменение характеристик персонажа после соответствующих приобретений в меню магазина. А также GameController отвечает за приведение игры в состояние паузы (листинг 4) и выход в главное меню.

WaveManager содержит функцию для генерации волн врагов SpawnWave и для появления отдельно взятых врагов SpawnEnemies. Функция SpawnEnemies вызывается из функции SpawnWave.

Листинг 4 – Функция переключения паузы

```
void PauseGame()
{
    pause = !pause;

    if (pause)
    {
        crossHair.SetActive(false);
        Time.timeScale = 0;
        player.GetComponent<Player>().disablePlayer = true;
        Cursor.visible = true;
    }
    else
    {
```

```

        crossHair.SetActive(true);
        Cursor.visible = false;
        Time.timeScale = 1;
        player.GetComponent<Player>().disablePlayer = false;
    }
}

```

UI

Компонент предоставляет игроку интерфейс на игровой локации, в меню магазина и после поражения. В игровой локации это шкала здоровья, счетчик волн, количество денег, подсказка для вызова меню магазина и миникарта. В меню магазина это кнопки для покупки улучшений характеристик персонажа и оружия (рисунок 7), а также подписи к этим кнопкам и стоимость покупки. В меню паузы доступна кнопка для выхода в главное меню. После поражения персонажа отображается панель, где показан текущий и лучший игровой результат, а также расположены кнопки выхода и рестарта.



Рисунок 7 – Кнопки в меню магазина

Sound

Компонент отвечает за звуки в игре. Включает в себя скрипт `AudioPlayer`, который содержит функцию для воспроизведения звуков и музыки `PlaySound` (листинг 5).

Листинг 5 – Функция воспроизведения звука

```
public void PlaySound (string _name)
{
    for (int i = 0; i < sounds.Length; i++)
    {
        if (sounds[i].name == _name)
        {
            sounds[i].Play();
            return;
        }
    }

    Debug.LogWarning("AudioManager: Sound not found in list, " +
        _name);
}
```

4.2. Реализация пользовательского интерфейса

Пользовательский интерфейс реализован двумя компонентами: UI, который включает в себя элементы интерфейса, и GameController, который отвечает за изменение данных в элементах интерфейса.

Интерфейс имеет четыре состояния: на игровой локации (рисунок 8), в меню магазина после поражения и в главном меню. Также в главном меню доступно меню настроек.

На игровом интерфейсе представлены следующие элементы:

- 1) счетчик волн;
- 2) подсказка, на какую кнопку открыть меню магазина;
- 3) количество денег;
- 4) иконка с текущим временным улучшением;
- 5) шкала здоровья;
- 6) перекрестье прицела;
- 7) миникарта, где отображаются все основные объекты.

В меню магазина игроку доступна информация о текущих характеристиках персонажа: максимальное здоровье (Max HP), восстановление здоровья в секунду (HP Regen), урон оружия (Damage), скорострельность оружия (Firerate).



Рисунок 8 – Игровой интерфейс

В меню магазина доступны кнопки для покупки оружия. Представлены три вида оружия: пистолет (Pistol), винтовка (Rifle) и дробовик (Shotgun). После покупки и смены оружия предыдущее оружие сохраняется и его можно снова экипировать из меню паузы. При этом его стоимость будет равна нулю.

У пистолета (Pistol) средние показатели урона и скорострельности, у дробовика (Shotgun) низкая скорострельность и высокий урон, у винтовки (Rifle) высокая скорострельность и низкий урон. При выборе оружия отображаются именно его характеристики. Также прирост урона и скорострельности у всех трех видов оружия разный.

В меню магазина доступны кнопки для улучшения вышеперечисленных характеристик. Стоимость улучшения показана на самих кнопках. После поражения появляется экран с оповещением (рисунок 9), текущем и лучшем игровом результате. На этом экране доступны кнопки для перезапуска уровня (Restart) и выхода в главное меню (Quit).



Рисунок 9 – Надпись после поражения

В главном меню (рисунок 10) представлены три кнопки:

- 1) кнопка `Play` запускает сцену с игровой локацией;
- 2) кнопка `Options` открывает меню настроек;
- 3) кнопка `Quit` закрывает игру.



Рисунок 10 – Интерфейс в главном меню

В меню настроек (рисунок 11) доступен слайдер для регулирования громкости музыки и эффектов. Также доступна возможность изменять раз-

решение экрана, настройки графики (низкие, средние и высокие) и режим экрана (полноэкранный и оконный). Кнопка для сохранения настроек (Save) и выхода из меню настроек (Exit).

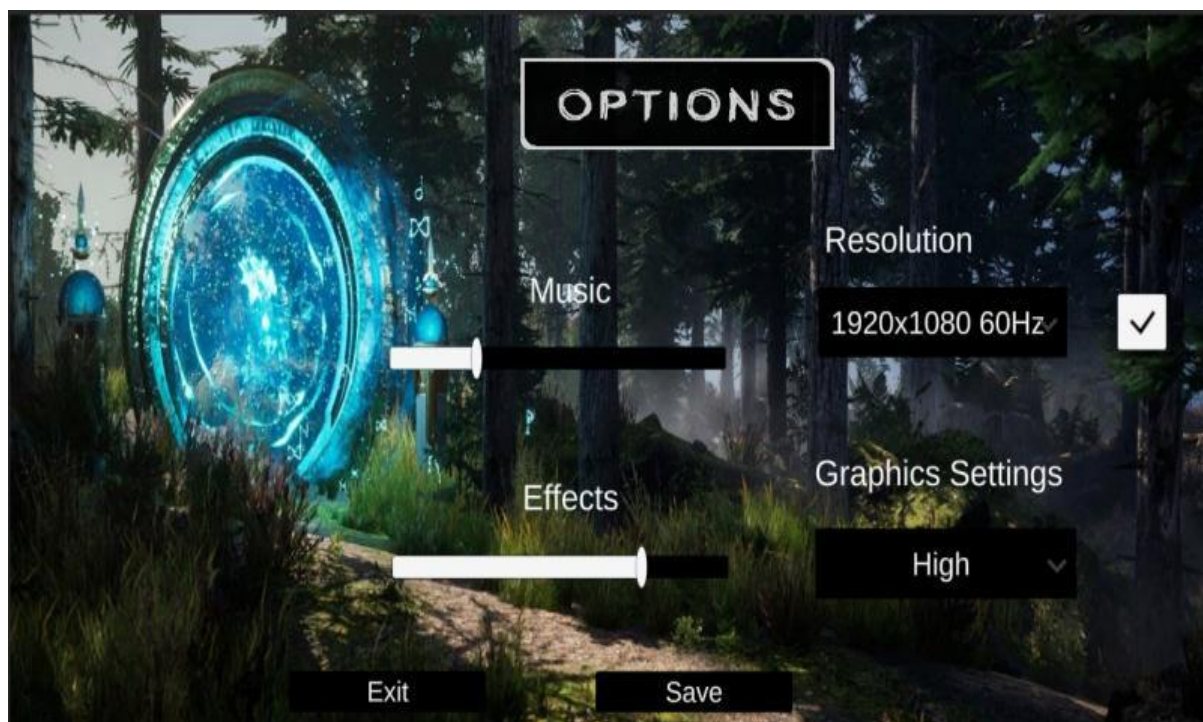


Рисунок 11 – Интерфейс в меню настроек

Вывод по четвертой главе

В соответствии с определенными требованиями и спроектированной архитектурой системы была реализована компьютерная игра в жанре «Top-Down Shooter». Была подробно описана реализация всех компонентов игры. В реализации каждого компонента представлены элементы исходного кода, для каждого состояния пользовательского интерфейса приведены скриншоты.

5. ТЕСТИРОВАНИЕ

Функциональное тестирование

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям. Набор тестов на функциональность представлен в таблице 1.

Таблица 1 – Тестирование системы

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
1.	Тестирование главного меню	1. Запустить игру нажатием кнопки Play. 2. Открыть меню настроек нажатием кнопки Options. 3. Выйти из игры нажатием кнопки Quit.	Корректная реакция на нажатие кнопок.	Да
2.	Тестирование меню настроек	1. Изменить настройки звука в меню настроек. 2. Изменить настройки экрана в меню настроек.	Настройки изменяются в соответствии с действиями пользователя.	Да
3.	Тестирование персонажа	1. Использовать клавиши wasd для перемещения персонажа. 2. Использовать ЛКМ для стрельбы. 3. Использовать мышку для вращения персонажа.	Персонаж корректно реагирует на действия игрока.	Да
4.	Тестирование врагов	1. Проверить, следуют ли враги за персонажем. 2. Проверить, обходят ли враги препятствия.	Враги корректно ведут себя на игровой локации.	Да
5.	Тестирование предметов временного улучшения	1. Проверить, появляются ли предметы временного улучшения на карте через заданный промежуток времени. 2. Проверить, не появляется ли на карте более 1 предмета улучшения. 3. Проверить, появляются ли предметы случайным образом.	Предметы временного улучшения появляются через заданный промежуток времени. Предметы появляются случайным образом, но не более одного за раз.	Да

Продолжение таблицы 1

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
6.	Тестирование интерфейса	<ol style="list-style-type: none"> 1. Проверить, реагирует ли шкала здоровье на получение урона персонажем. 2. Проверить, реагирует ли счетчик денег на убийство врага. 3. Проверить, реагирует ли миникарта на движение персонажа и врагов. 	Данные на интерфейсе отображаются корректно.	Да
7.	Тестирование меню магазина	<ol style="list-style-type: none"> 1. Приобрести улучшение максимального здоровья. 2. Приобрести улучшение регенерации здоровья. 3. Приобрести улучшение урона. 4. Приобрести улучшение скорострельности. 5. Приобрести винтовку. 6. Приобрести дробовик. 	Улучшения и покупка оружия происходят, если для этого достаточно денег. Если недостаточно, то ничего не происходит.	Да
8.	Тестирование меню после поражения	<ol style="list-style-type: none"> 1. Проверить, корректно ли отображается текущий игровой результат. 2. Проверить, корректно ли фиксируется и отображается лучший игровой результат. 3. Проверить работу кнопки перезапуска уровня. 4. Проверить кнопку выхода в главное меню. 	Данные отображаются корректно. Кнопка перезапуска перезапускает уровень. Кнопка выхода выводит в главное меню.	Да
9.	Тестирование взаимодействия персонажа и врагов	<ol style="list-style-type: none"> 1. Столкнуться персонажем с врагом. 2. Выстрелить во врага. 	Персонаж получает урон после столкновения с врагом. Враг получает урон при попадании по нему выстрелом.	Да
10.	Тестирование оружия	<ol style="list-style-type: none"> 1. Выстрелить из пистолета. 2. Сменить оружие на винтовку и выстрелить из винтовки. 3. Сменить оружие на дробовик выстрелить из дробовика. 	Оружие меняется корректно. Выстрелы соответствуют каждому типу оружия.	Да
11.	Тестирование появления врагов	Проверить, корректно ли появляются враги и боссы.	Враги и боссы появляются корректно.	Да

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
12.	Тестирование анимаций	1. Проверить анимации персонажа: перемещение, вращение, выстрел. 2. Проверить анимации врагов: перемещение, получение урона, гибель. 3. Проверить анимацию появления экрана после поражения.	Анимации воспроизводятся корректно.	Да
13.	Тестирование звуков	1. Проверить, воспроизводится ли музыка в главном меню. 2. Проверить, воспроизводится ли музыка на игровой локации. 3. Выстрелить из всех видов оружия. 4. Получить урон от врага. 5. Дождаться волны с боссом. 6. Убить врага.	Соответствующие звуки воспроизводятся в меню паузы, на игровой локации, при выстреле из оружия, при появлении боссов, при смерти врага.	Да

Юзабилити тестирование

Данное тестирование производилось при участии 8 одноклассников, в его ходе были выявлены и исправлены некоторые недочеты.

1. Кнопка выхода перенесена из левой части экрана в правую.
2. В меню настроек, если частота обновления экрана монитора была 144 Гц, строки накладывались друг на друга. Баг исправлен.
3. При смене разрешения экрана пропадал прицел. Баг исправлен.

В приложении Б представлены скриншоты итоговой версии игры после реализации и проведения тестирования.

Вывод по пятой главе

На основании функциональных требований было проведено функциональное тестирование, а также юзабилити тестирование. Все тесты были успешно пройдены – реализованная компьютерная игра полностью соответствует сформированным требованиям.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была разработана компьютерная игра в жанре «Top-Down Shooter» на платформе Unity. Достигнуты следующие основные результаты.

1. Проведен анализ предметной области. Были рассмотрены механики аналогичных проектов, для разрабатываемой игры были выбраны некоторые из них.

2. Выполнено проектирование игры. А именно, были выделены функциональные и нефункциональные требования, сформирована концепция игры и составлена диаграмма вариантов использования.

3. Спроектирована архитектура игры и описаны все компоненты приложения.

4. Реализована компьютерная игра. Реализация каждого компонента игры была подробно описана.

5. Проведено функциональное и юзабилити тестирование. Все функциональные тесты были успешно пройдены, а в результате юзабилити тестирования были выявлены недочеты, которые были исправлены.

В дальнейшем планируется монетизация игры. Для этого нужно провести аналитику, посчитать юнит-экономику и основные маркетинговые метрики. Это нужно сделать как для уже имеющейся десктопной версии игры, так и для мобильной, потому что рассматривается вариант портирования. После этого можно будет сделать вывод, на какой платформе игра принесёт больше денег: на ПК с продаж на специализированных сервисах, или на мобильных устройствах с рекламных интеграций.

ЛИТЕРАТУРА

1. Hetherington B. Time Warp and Prediction Analysis in a Top-Down Shooter Game. – Worcester Polytechnic Institute, 2020. – 47 p.
2. Marchand A., Hennig-Thurau T. Value creation in the video game industry: Industry economics, consumer benefits, and research opportunities. – Journal of interactive marketing, 2013. – 16 p.
3. Fish C. The History of Video Games. – USA: White Owl, 2021. – 120 p.
4. Konami. Metal Gear. [Электронный ресурс] URL: <https://www.konami.com/mg/archive/mg/index.html> (дата обращения: 11.02.2024 г.).
5. McGuire M., Jenkins O.C. Creating games: Mechanics, content, and technology. – CRC Press, 2008. – 500 p.
6. Museum of the Game. Tank. [Электронный ресурс] URL: https://www.arcademuseum.com/game_detail.php?game_id=12989 (дата обращения: 11.02.2024 г.).
7. Kent S.L. The Ultimate History of Video Games. – Crown, 2010. – 624 p.
8. Kohler C. Retro gaming hacks: Tips & tools for playing the classics. – O'Reilly Media, Inc., 2005. – 164 p.
9. Goldstone W. Unity game development essentials. – Packt Publishing Ltd, 2009. – 316 p.
10. Бонд Д.Г. Unity и C#. Геймдев от идеи до реализации. – СПб: Питер, 2019. – 928 с.
11. Хокинг Д. Unity в действии. Мультиплатформенная разработка на C#. – СПб: Питер, 2016. – 336 с.
12. Тристан Д. Играй! История видеоигр. – Москва: Белое яблоко, 2014. – 648 с.
13. Казакова Н.Ю. Классификация видео-и компьютерных игр как основополагающий фактор отбора формальных признаков в гейм-дизайне. – Научное обозрение. Серия 2: Гуманитарные науки, 2015. – 10 с.

14. Назаров Ю.В., Казакова Н.Ю. История возникновения гейм-дизайна как самостоятельной формы визуального искусства. Жанры видеоигр и основные этапы их разработки. – Дизайн и технологии, 2014. – 92 с.

15. Титоренко А.А. Метафизика видеоигр. – Наука телевидения, 2011. – 151 с.

ПРИЛОЖЕНИЯ

Приложение А. Спецификация вариантов использования

Спецификация вариантов использования (ВИ) системы приведена в таблицах 1–9.

Таблица 1 – Спецификация ВИ «Изменить настройки»

Прецедент: Изменить настройки
ID: 1
Краткое описание: Изменение громкости и разрешения экрана игры.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Отсутствуют
Основной поток: 1. Вариант использования начинается, когда игрок меняет любую из настроек игры в меню настроек. 2. Система меняет настройки в соответствии с теми, что установил игрок.
Постусловия: Настройки изменены.
Альтернативные потоки: Нет

Таблица 2 – Спецификация ВИ «Начать игру»

Прецедент: Начать игру
ID: 2
Краткое описание: Запустить сцену с игровой картой.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Нет
Основной поток: 1. Вариант использования начинается, когда игрок нажимает кнопку Play. 2. Система запускает сцену с игровой картой.
Постусловия: Запущена сцена с игровой картой.
Альтернативные потоки: Нет

Таблица 3 – Спецификация ВИ «Передвигаться»

Прецедент: Передвигаться
ID: 3
Краткое описание: Игрок передвигается по игровой карте.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Игра не на паузе.
Основной поток: 1. Вариант использования начинается, когда игрок нажимает кнопку W, A, S или D. 2. Система передвигает персонажа по игровой карте.
Постусловия: Нет
Альтернативные потоки: Нет

Таблица 4 – Спецификация ВИ «Стрелять»

Прецедент: Стрелять
ID: 4
Краткое описание: Игрок выстреливает из оружия, которое находится у него в руках.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Игра не на паузе.
Основной поток: 1. Вариант использования начинается, когда игрок нажимает кнопку выстрела. 2. Из оружия по направлению выстрела вылетает пуля или несколько пуль.
Постусловия: Пуля или несколько пуль летит по направлению выстрела.
Альтернативные потоки: Нет

Таблица 5 – Спецификация ВИ «Открыть меню паузы»

Прецедент: Открыть меню паузы
ID: 5
Краткое описание: Игра приостанавливается и открывается меню паузы.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Игра не на паузе.
Основной поток: 1. Вариант использования начинается, когда игрок нажимает кнопку Esc. 2. Система приостанавливает игру и открывает меню паузы.
Постусловия: Игра приостановлена и открыто меню паузы
Альтернативные потоки: Нет

Таблица 6 – Спецификация ВИ «Выйти в главное меню»

Прецедент: Выйти в главное меню
ID: 6
Краткое описание: Игрок переходит в сцену с главным меню.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Открыто меню паузы.
Основной поток: 1. Вариант использования начинается, когда игрок нажимает кнопку выхода в главное меню 2. Игрок подтверждает действие. 3. Система запускает сцены с главным меню.
Постусловия: Запущена сцена с главным меню.
Альтернативные потоки: I. Игрок не подтверждает выход в главное меню. 1. Меню паузы остается открытым.

Таблица 7 – Спецификация ВИ «Купить улучшение характеристики»

Прецедент: Купить улучшение характеристики
ID: 7
Краткое описание: Игрок покупает улучшение здоровья, регенерации, скорострельности или урона.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Открыто меню паузы и у игрока достаточно денег на покупку улучшения.
Основной поток: 1. Вариант использования начинается, когда игрок нажимает на кнопку покупки улучшения. 2. Система до конца игры увеличивает характеристику персонажа.
Постусловия: Улучшена характеристика.
Альтернативные потоки: Нет

Таблица 8 – Спецификация ВИ «Купить оружие»

Прецедент: Купить оружие
ID: 8
Краткое описание: Игрок покупает оружие и берет его в руки.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: 1. Открыто меню паузы. 2. У игрока достаточно валюты.
Основной поток: 1. Вариант использования начинается, когда игрок нажимает кнопку покупки оружия. 2. Система меняет оружие персонажа. 3. Система изменяет стоимость оружия на 0\$.
Постусловия: 1. У персонажа сменилось оружие. 2. Новая стоимость оружия – 0\$.
Альтернативные потоки: Нет

Таблица 9 – Спецификация ВИ «Подобрать улучшение»

Прецедент: Подобрать улучшение
ID: 9
Краткое описание: Игрок подбирает временное улучшение.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: На карте появилось улучшение.
Основной поток: 1. Вариант использования начинается, когда персонаж задевает модельку улучшения. 2. Система временно повышает характеристику персонажа и убирает улучшение с игровой карты. 3. Система отображает на интерфейсе игрока, что улучшение активно.
Постусловия: 1. На карте нет улучшений. 2. У пользователя временно повышена характеристика. 3. В интерфейсе игрока отображено активное временное улучшение.
Альтернативные потоки: Нет

Приложение Б. Скриншоты итоговой версии игры

На рисунках 1–10 представлены скриншоты итоговой версии игры.



Рисунок 1 – Главное меню игры

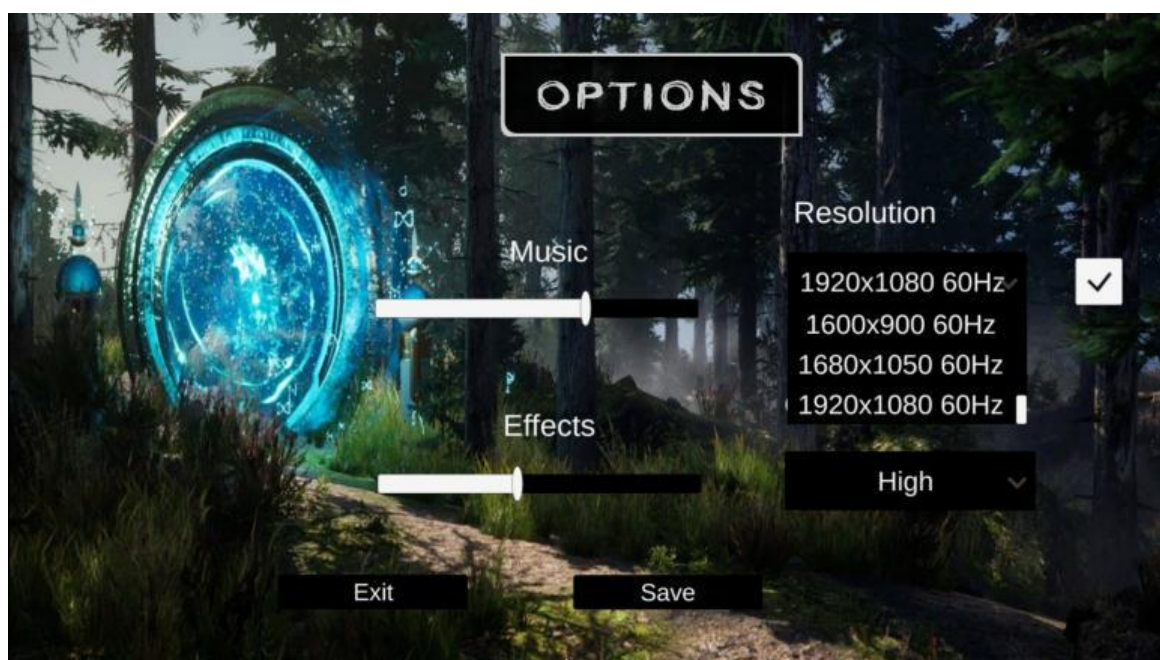


Рисунок 2 – Меню настроек



Рисунок 3 – Игровой процесс



Рисунок 4 – Броня, которая дает временную неуязвимость

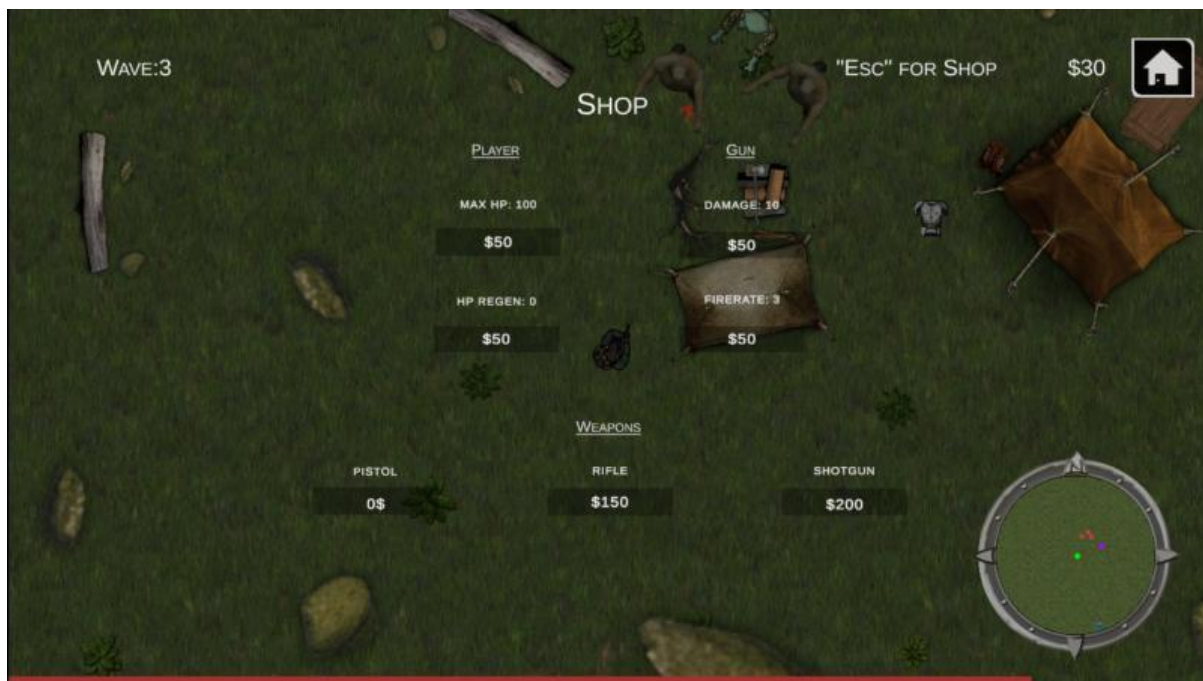


Рисунок 5 – Меню магазина



Рисунок 6 – Босс на 5 волне



Рисунок 7 – Босс на 10 волне



Рисунок 8 – Выстрел из дробовика



Рисунок 9 – Выстрелы из винтовки



Рисунок 10 – Меню после поражения