



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

НИУ ВШЭ-Пермь, УДК 09.03.04,
Факультет экономики, менеджмента
и бизнес-информатики, кафедра
информационных технологий в
бизнесе

РЕАЛИЗАЦИЯ МНОГОПОЛЬЗОВАТЕЛЬСКОЙ КОМПЬЮТЕРНОЙ ИГРЫ В ЖАНРЕ РПГ

Работу выполнил студент группы ПИ-18-2
3-го курса, факультета экономики,
менеджмента и бизнес-информатики
Чепокhov Е.С.

Научный руководитель:
преподаватель кафедры информационных
технологий
Лебедев В.В.

Пермь, 2021



ПЛАН ДОКЛАДА

1. [Проблема](#)
2. [Постановка цели работы и задач](#)
3. [Анализ предметной области](#)
4. [Обзор аналогов](#)
5. [Требования к разрабатываемой игре](#)
6. [Основание выбора средств разработки](#)
7. [Проектирование алгоритмов](#)
8. [Проектирование интерфейса](#)
9. [Результаты тестирования](#)
10. [Заключение](#)

ПРОБЛЕМА КУРСОВОЙ РАБОТЫ

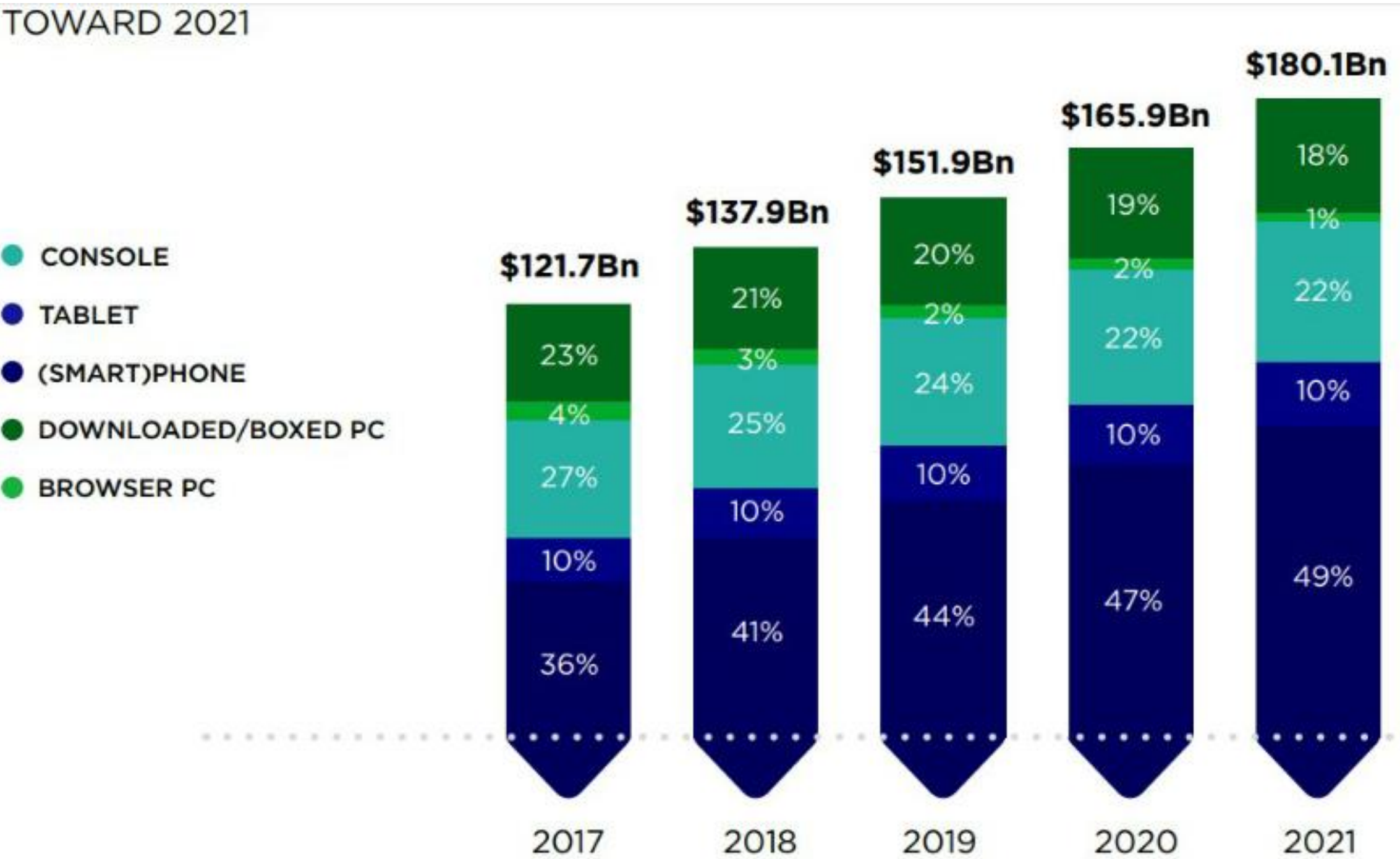


Рисунок 1 - Рост рыночной стоимости игровой индустрии

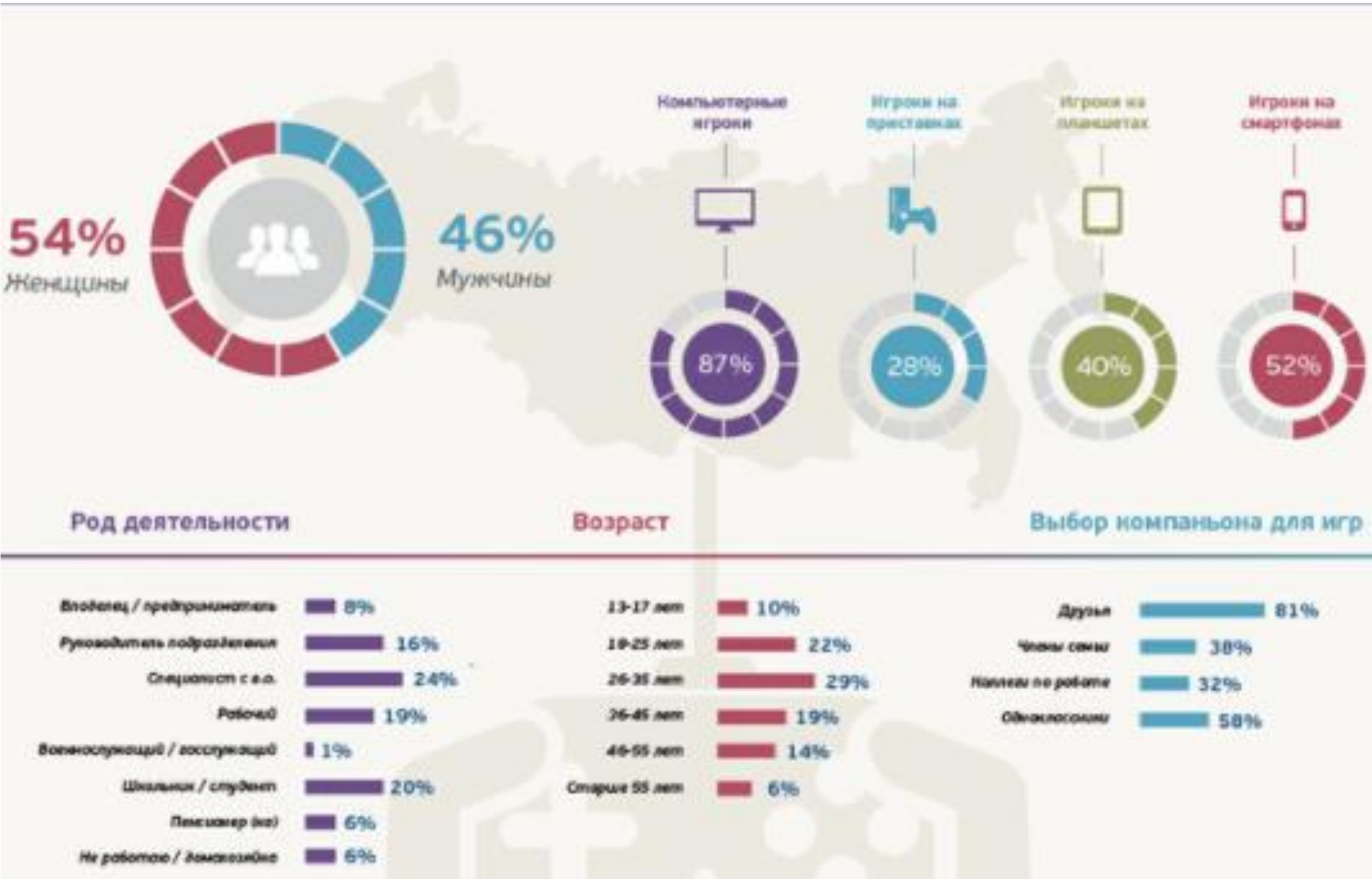


Рисунок 2 - Характеристика пользователей



ЦЕЛИ И ЗАДАЧИ

Предмет: разработка алгоритма многопользовательской игры.

Объект: компьютерные игры в жанре «RPG»

Цель: разработка компьютерной игры в жанре «RPG» с наличием многопользовательской игры.

Задачи:

1. Провести анализ действующих алгоритмов.
2. Проанализировать, сравнить и выбрать визуализатор для игры.
3. Изучить способы разработки многопользовательских игр.
4. Спроектировать алгоритм многопользовательской игры.
5. Спроектировать интерфейс для игры.
6. Провести опрос и проанализировать ответы по поводу интерфейса, внести изменения.
7. Реализовать код программы и предварительный интерфейс.
8. Реализовать проект с использованием алгоритма многопользовательской игры.
9. Написать пользовательскую документацию для программы.
10. Провести тестирование, для выявления недочетов и сбора критики тестировщиков.
11. Внести изменения на основании выявленных недочетов на этапе тестирования.

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

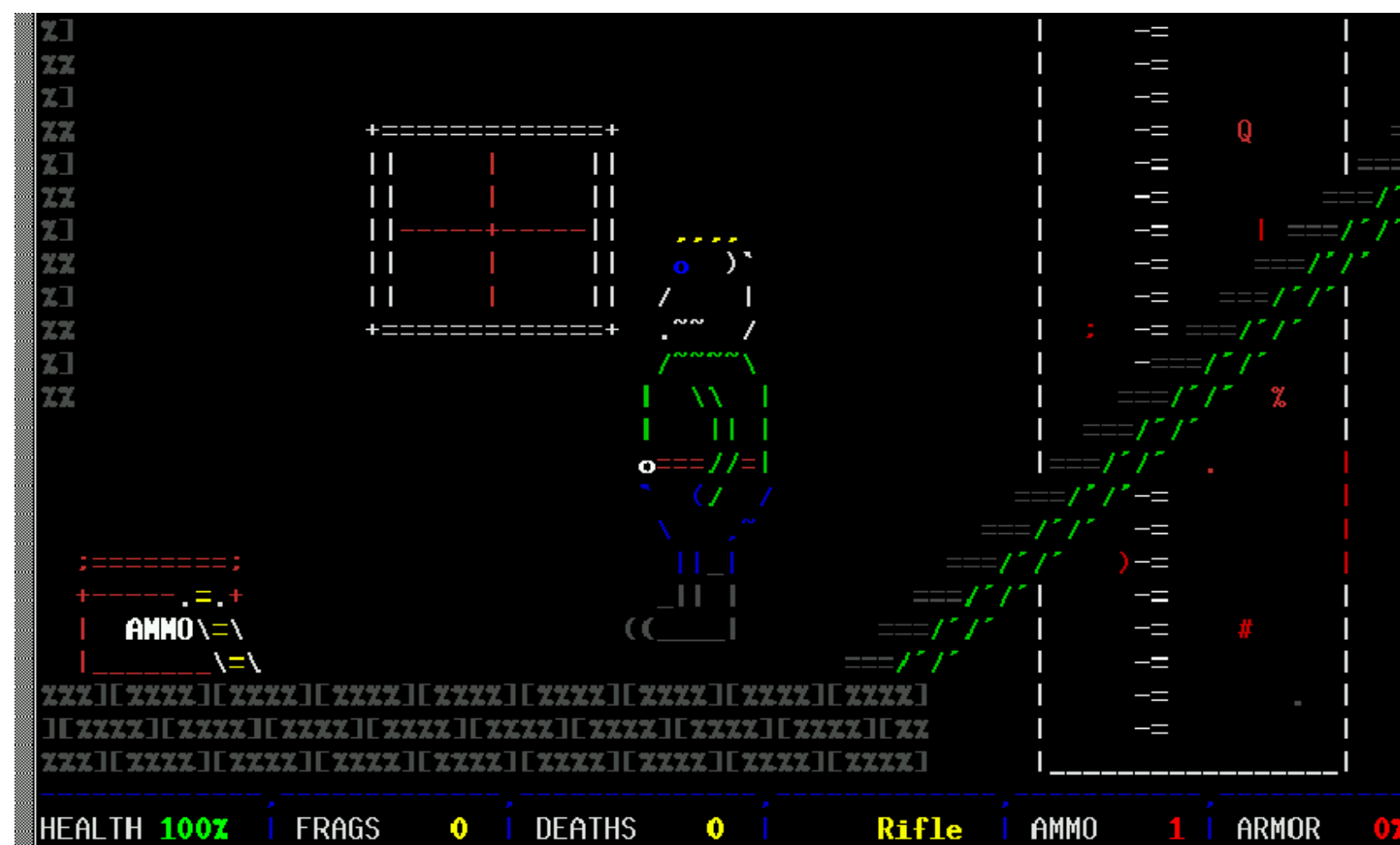


Рисунок 3 - Пример текстовой графики



Рисунок 4 - Пример двухмерной графики

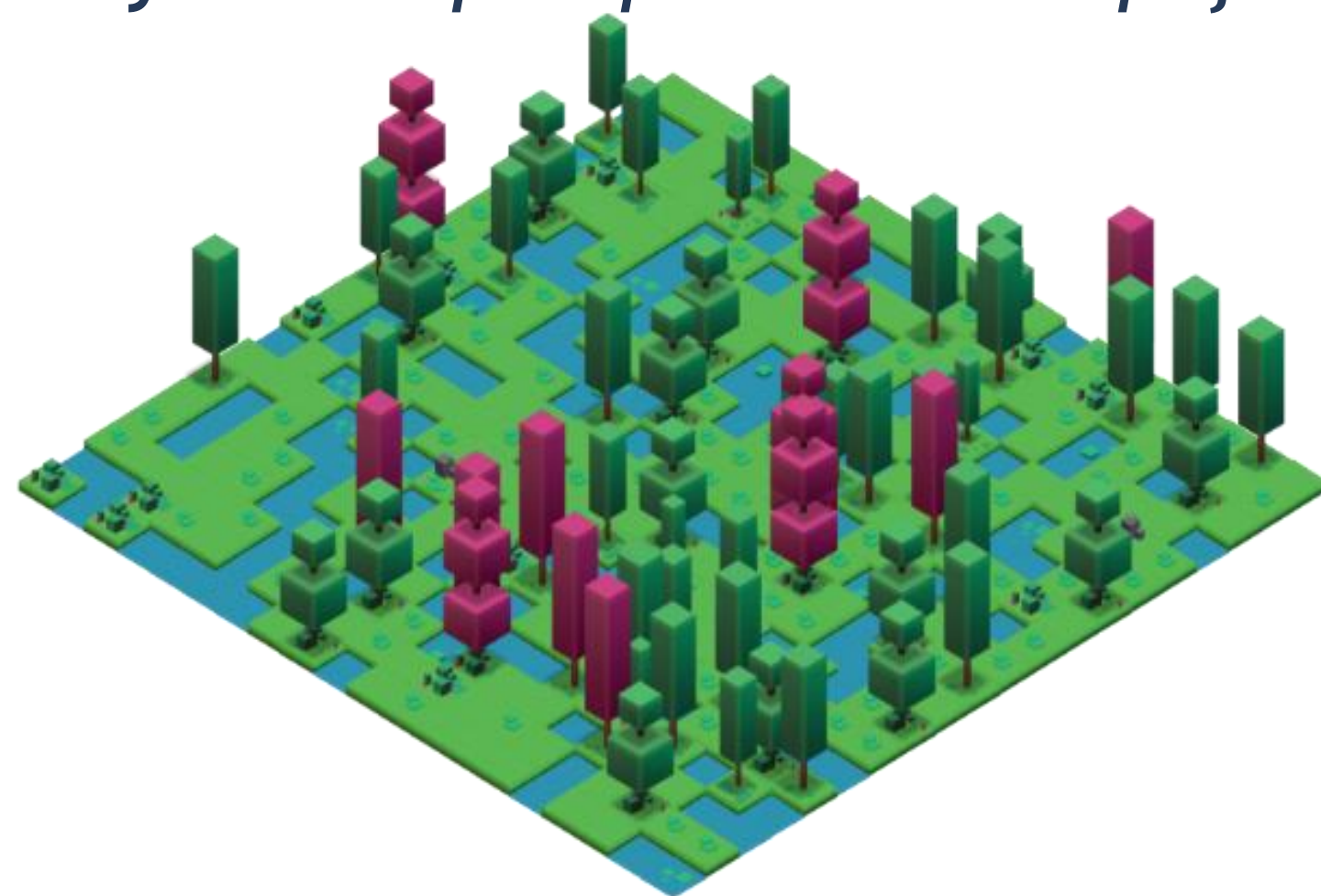


Рисунок 5 - Пример изометрической графики

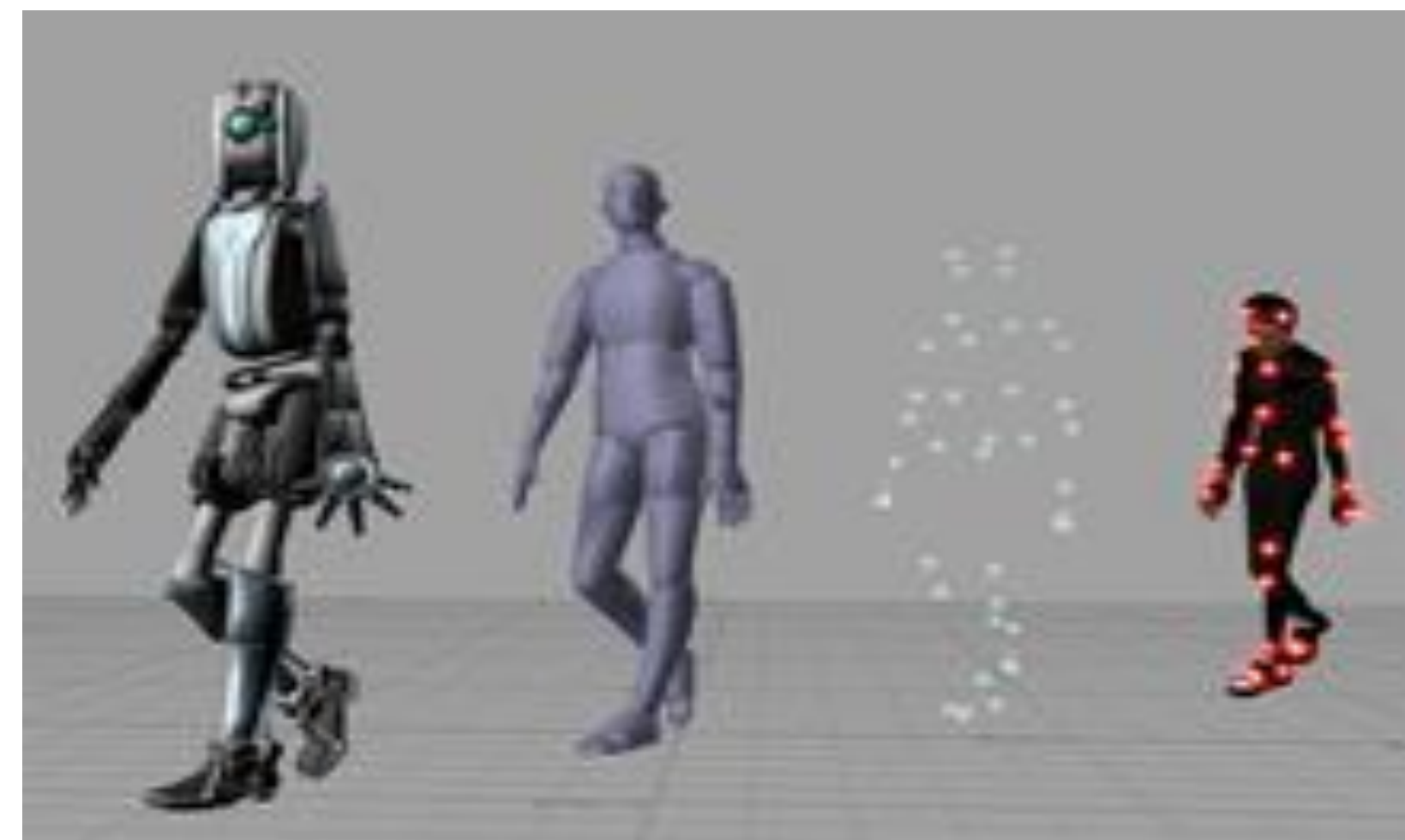


Рисунок 6 - Пример трехмерной графики

Enter the Gungeon

Движок: Unity

Преимущества:

- Низкая требовательность к устройству
- Интерфейс игры
- Игровой процесс

Недостатки:

- Отсутствие многопользовательской игры
- Стоимость игры



Рисунок 7 - Интерфейс игры Enter the Gungeon

Darkest Dungeon

Движок: Unity

Преимущества:

- Низкая требовательность к устройству
- Стилистика игры
- Полное соответствие жанру РПГ

Недостатки:

- Малая вариативность игрового процесса
- Стоимость игры
- Отсутствие персонализации
- Отсутствие многопользовательской игры



Рисунок 8 - Интерфейс игры Darkest Dungeon

World of Warcraft

Движок: движок компании Blizzard

Преимущества:

- Многопользовательская игра
- Обширная персонализация
- Большое количество игрового контента

Недостатки:

- Условно бесплатная игра
- Требовательность к устройству



Рисунок 9 - Интерфейс игры World of Warcraft

Genshin Impact

Движок: Unity

Преимущества:

- Бесплатная игра
- Кроссплатформенность
- Нетребовательна к устройству

Недостатки:

- Урезанная возможность многопользовательской игры
- Отсутствие персонализации персонажа



Рисунок 10 - Интерфейс игры Genshin Impact



СРАВНЕНИЕ АНАЛОГОВ

Таблица 1 - Критерии сравнения среды для написания игры

Среда	Enter the Gungeon	Darkest Dungeon	World of Warcraft	Genshin Impact	Разрабатываемая игра
Возможность играть онлайн через интернет	-	-	+	+	+
Возможность запускать игру в оффлайн режиме	+	+	-	-	+
Возможность многопользовательской игры	+	-	+	+	+
Возможность персонализации персонажа	+	-	+	+	+
Наличие сюжета в игре	-	+	+	+	+
Возможность запуска на мобильных устройствах	-	-	-	+	+
Возможность запуска на слабых компьютерах	+	+	-	-	+
Игры без продвижения по сюжету	-	-	+	+	+
Возможность бесплатной игры	-	-	+	+	+



ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОЙ ИГРЕ

1. Возможность запускать игру в оффлайн и онлайн режиме;
2. Возможность многопользовательской игры;
3. Возможность персонализации персонажа;
4. Возможность запуска на мобильных устройствах и слабых компьютерах;
5. Наличие сюжета в игре;
6. Возможность бесплатной игры;
7. Наличие игр без продвижения по сюжету;



ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1. Система должна загружать выбранные пользователем игровые пространства;
2. Система должна предоставлять возможность передвижения пользователя по игровому пространству;
3. Система должна взаимодействовать с пользователем и откликаться на его действия;
4. Система должна предоставлять пользователю возможность подключения к сети интернет;
5. Система должна оповещать пользователя при возникновении ошибок;
6. Система должна давать пользователю возможность персонализации персонажа;
7. Система должна предоставлять возможность выхода из системы;
8. Уровни игры должны быть спроектированы с помощью фиксированной генерации;



НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1. Интерфейс игры должен быть простым для восприятия пользователя;
2. Архитектура системы должна позволять добавлять к нему дополнительные модули не нарушая работоспособность других модулей;
3. Система должна отвечать на запрос пользователя не более 5 секунд;
4. Функции ввода и вывода данных должны быть спроектированы таким образом, чтобы их удавалось повторно использовать на уровне объектного кода в других приложениях;
5. Архитектура игровых пространств должна позволять добавлять к нему новые игровые пространства не нарушая целостность других игровых пространств;
6. Система должна предоставлять интуитивность в использовании пользователем;
7. Система должна предоставлять безопасность данных пользователей;
8. Система должна своевременно обновляться;
9. Система должна давать пользователю возможность изменять настройки системы;
10. Система должна предоставлять возможность авторизации пользователя;
11. Интерфейс системы должен быть читабельным и понятным для восприятия;
12. Система должна давать пользователю возможность принимать решения в игровом пространстве;
13. Система должна предоставлять возможность запуска на слабых компьютерах;



ОСНОВАНИЕ ВЫБОРА СРЕДСТВ РАЗРАБОТКИ

Программное обеспечение для визуализации: Unity

Программное обеспечение для написания кода: Visual Studio

Язык программирования: C#

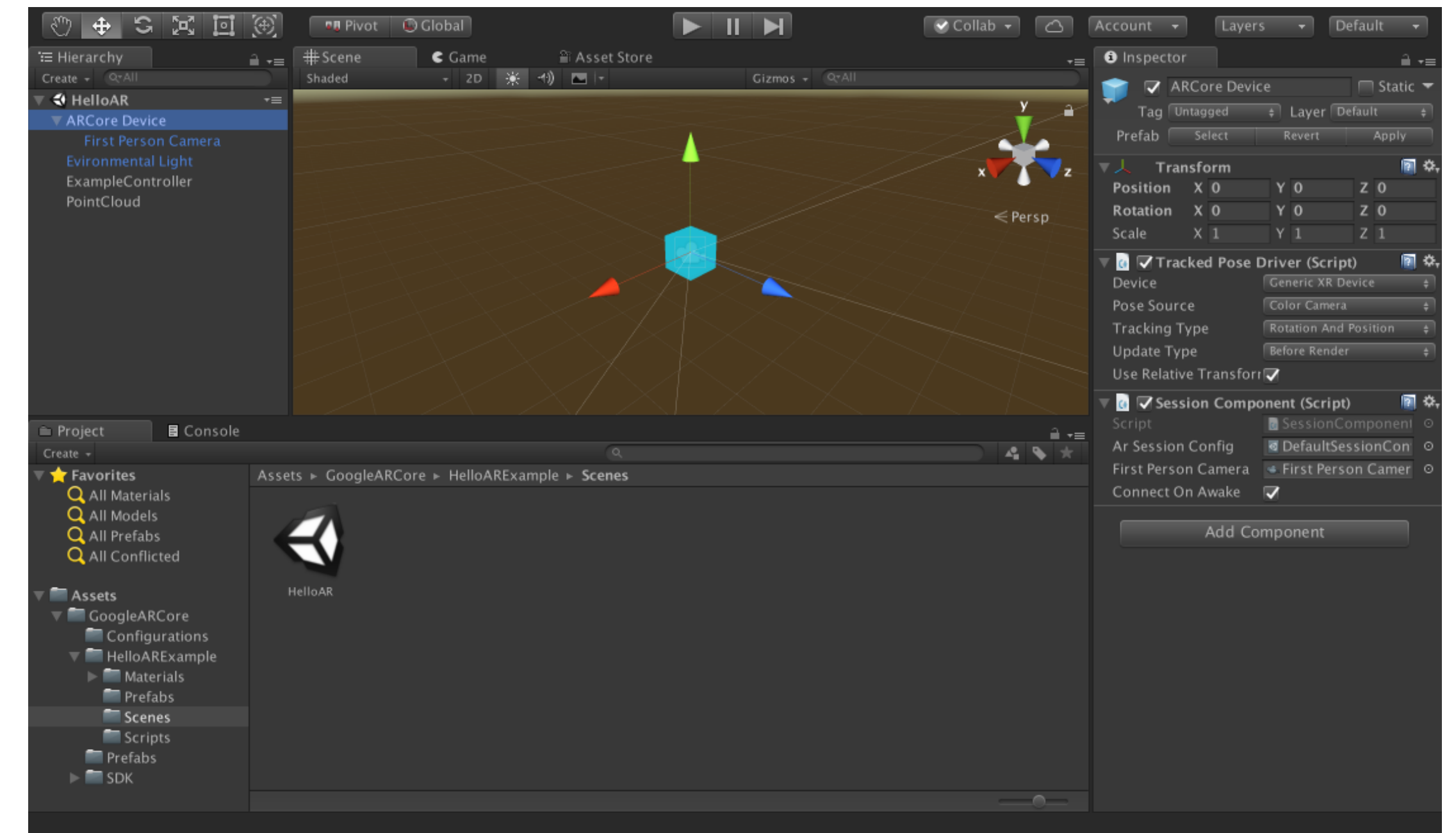


Рисунок 11 - Интерфейс Unity



ПРОЕКТИРОВАНИЕ АЛГОРИТМОВ

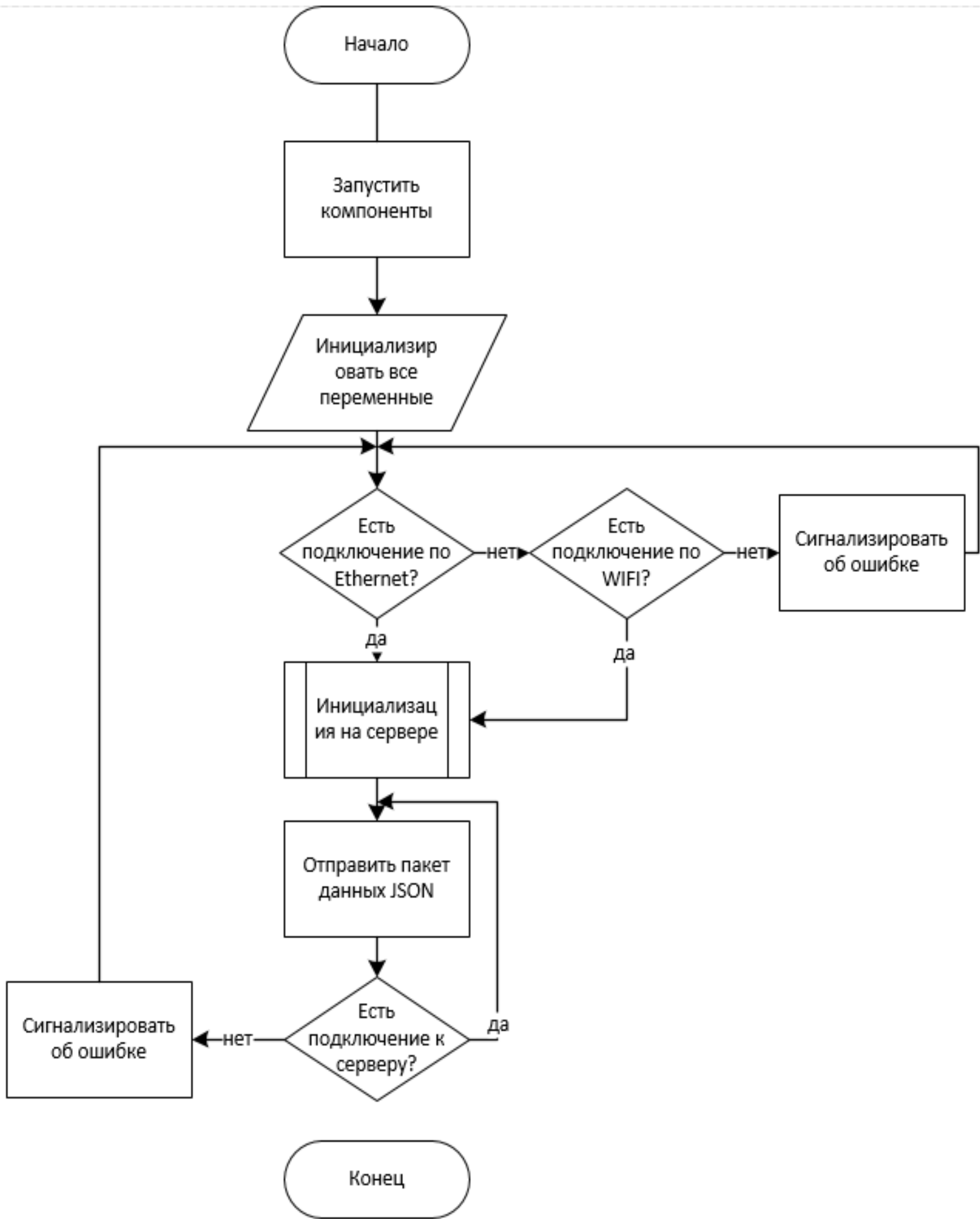


Рисунок 12 - Алгоритм подключения

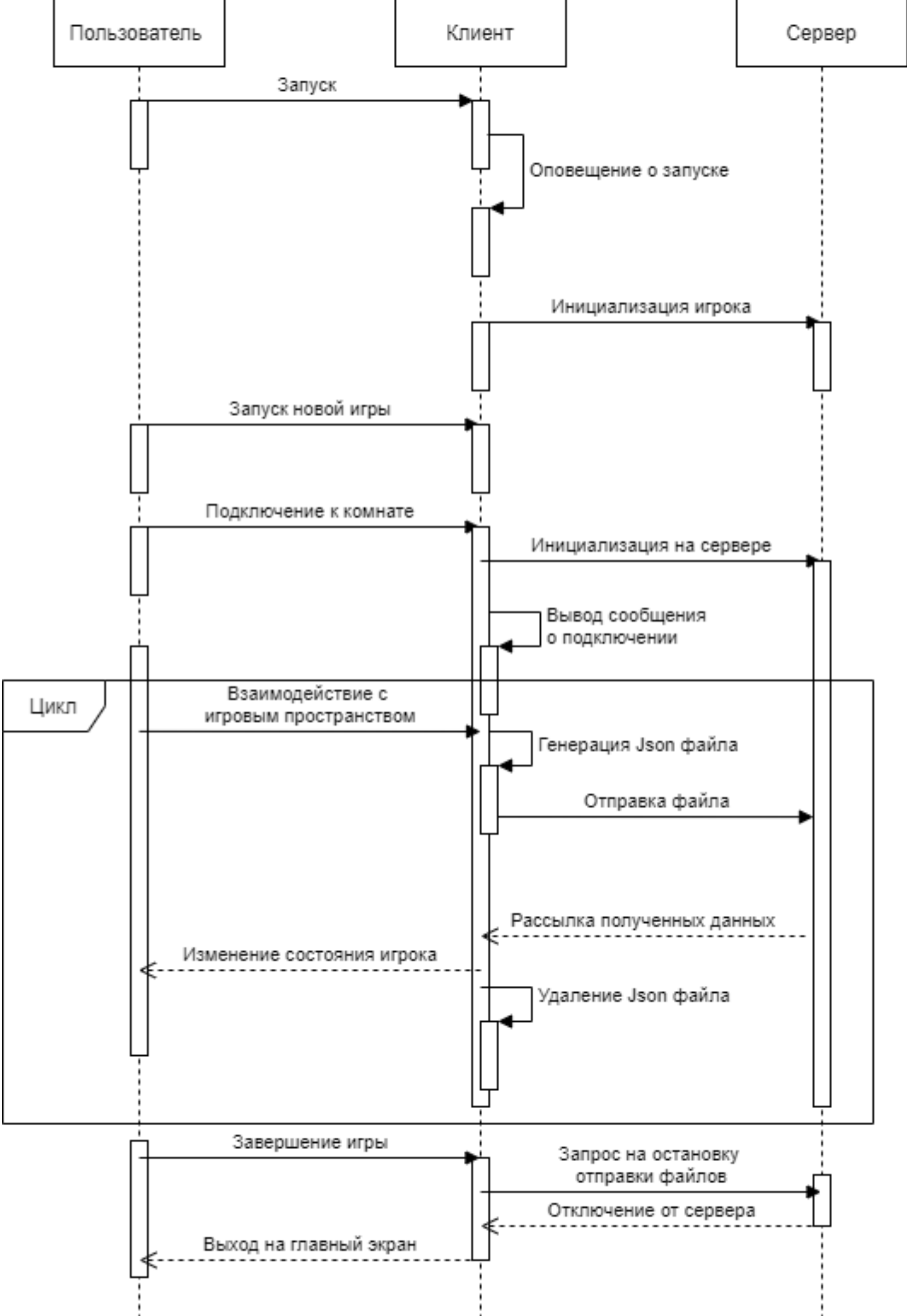


Рисунок 13 - Диаграмма последовательности

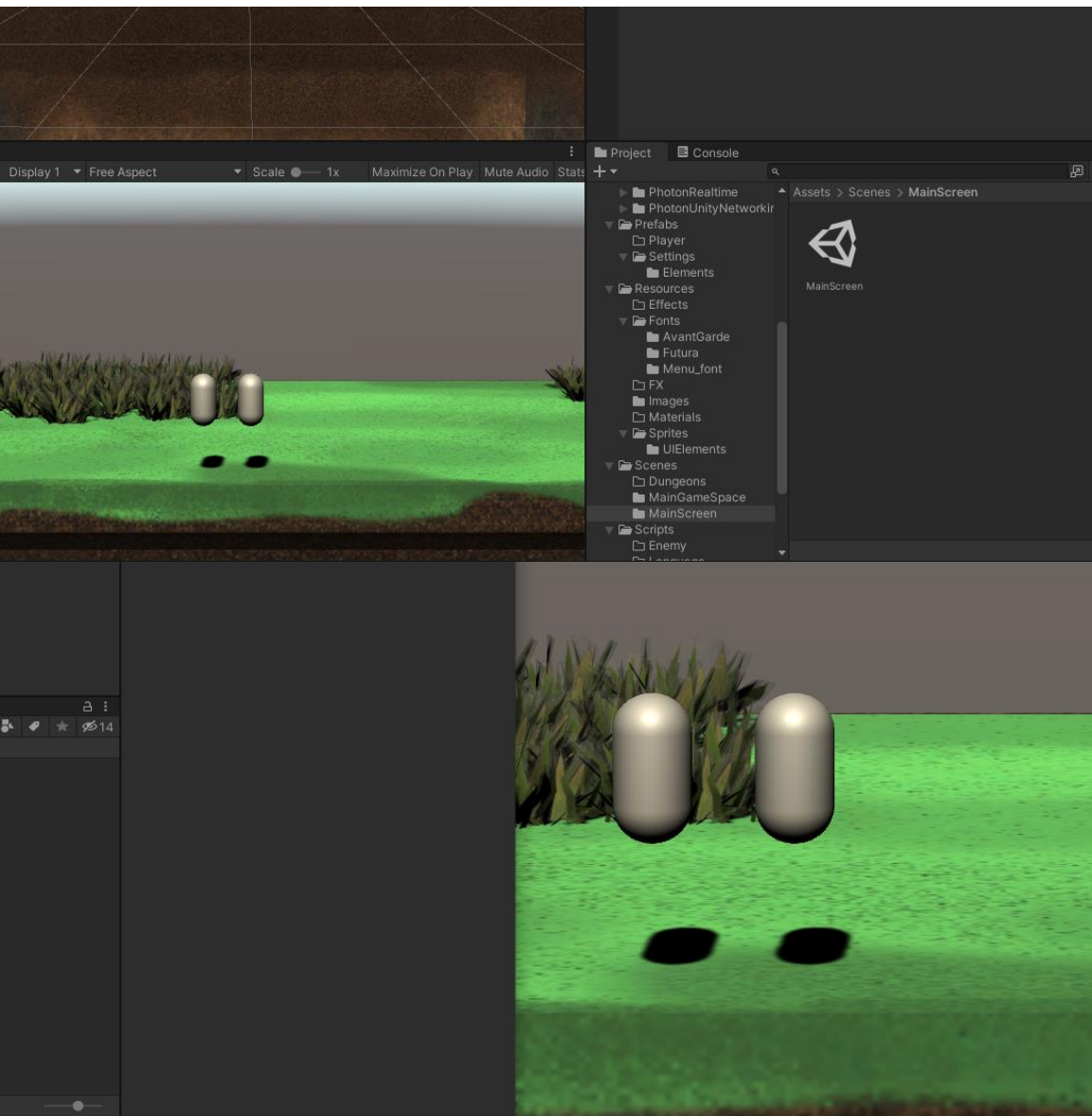


Рисунок 14 - Результат подключения

Сокращенный скрипт передвижения

```
public class Character : MonoBehaviour
{
    [SerializeField] private int life = 1;          // объявление количества хп
    [SerializeField] private float speed = 3.0F;    // начальная скорость
    public int Level;
    new private Rigidbody rigidbody;
    private Animator animation;
    private SpriteRenderer sprite;
    public bool ready = false;
    private CharacterState State {get {return
(CharacterState)animation.GetInteger("State");} set
{animation.SetInteger("State", (int)value);}}
    public void OnCollisionEnter2D(Collision2D collision) {if
(collision.gameObject.tag == "EndLevel") {ready = true;}}

    private void Start() {
        canv = GetComponent<Canvas>();
        rigidbody = GetComponent<Rigidbody>();
        animation = GetComponent<Animator>();
        sprite = GetComponentInChildren<SpriteRenderer>();
        //Cursor.lockState = CursorLockMode.Locked; }

    private void FixedUpdate() {
        State = CharacterState.Idle;
        float v = Input.GetAxis("Vertical");
        if (Input.GetButton("Horizontal") {RunLR(); // Движение влево/вправо}

        if (Input.GetButton("Vertical")) // ЕСЛИ же входные данные - кнопки W
или S
        {RunUD(v); // Движение вверх/вниз }
    }
}
```

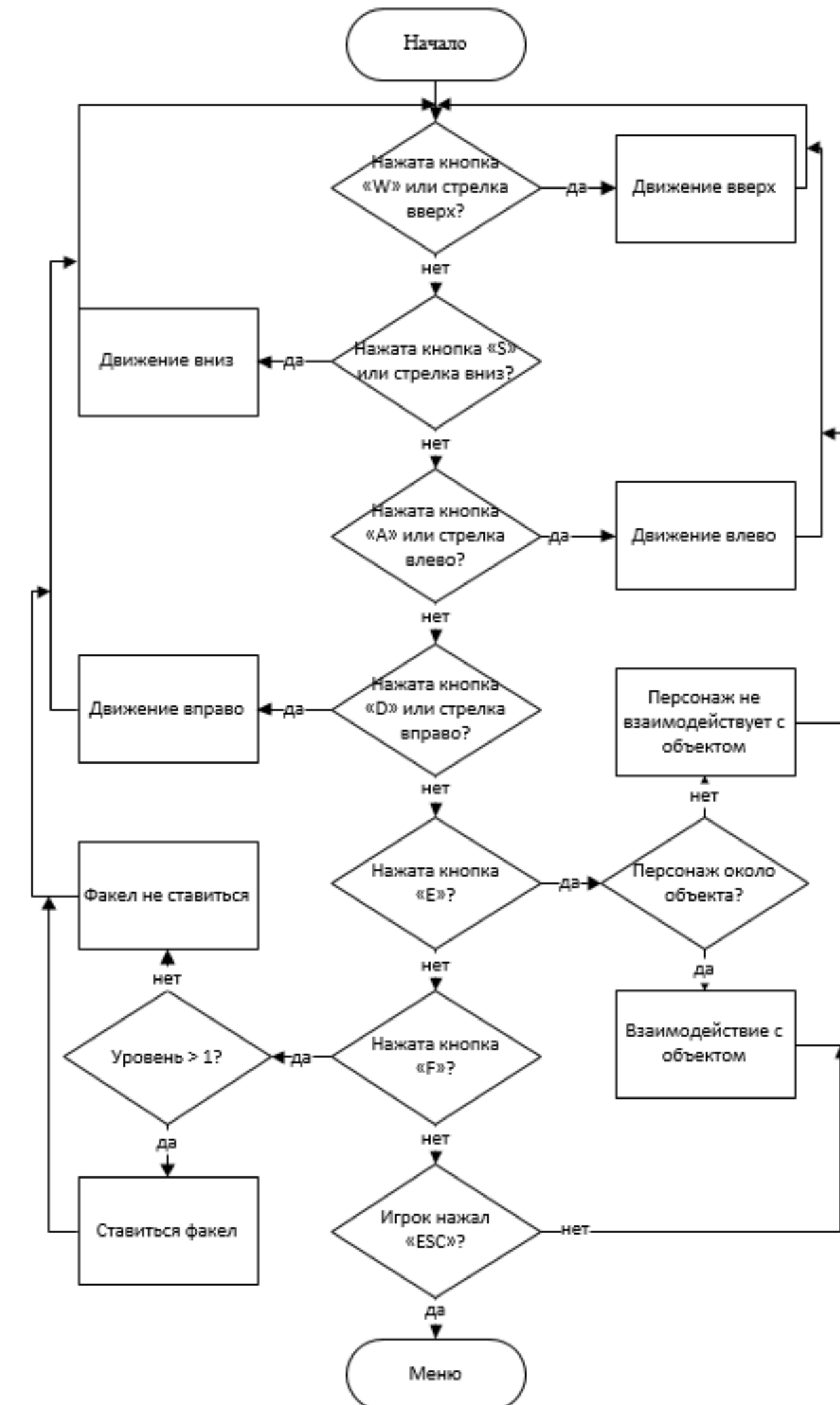


Рисунок 15 - алгоритм взаимодействия



ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА

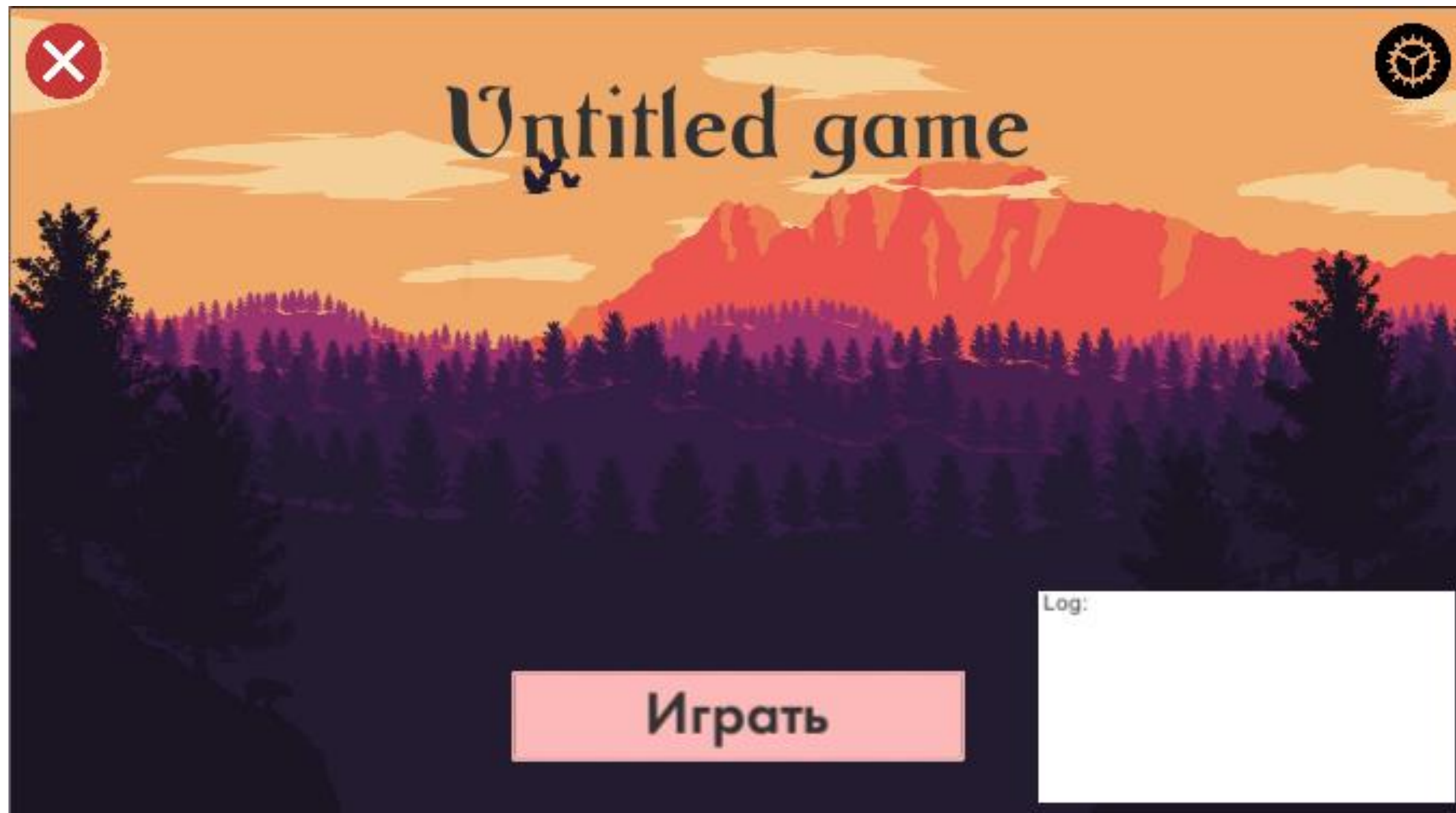


Рисунок 16 - Интерфейс главного меню

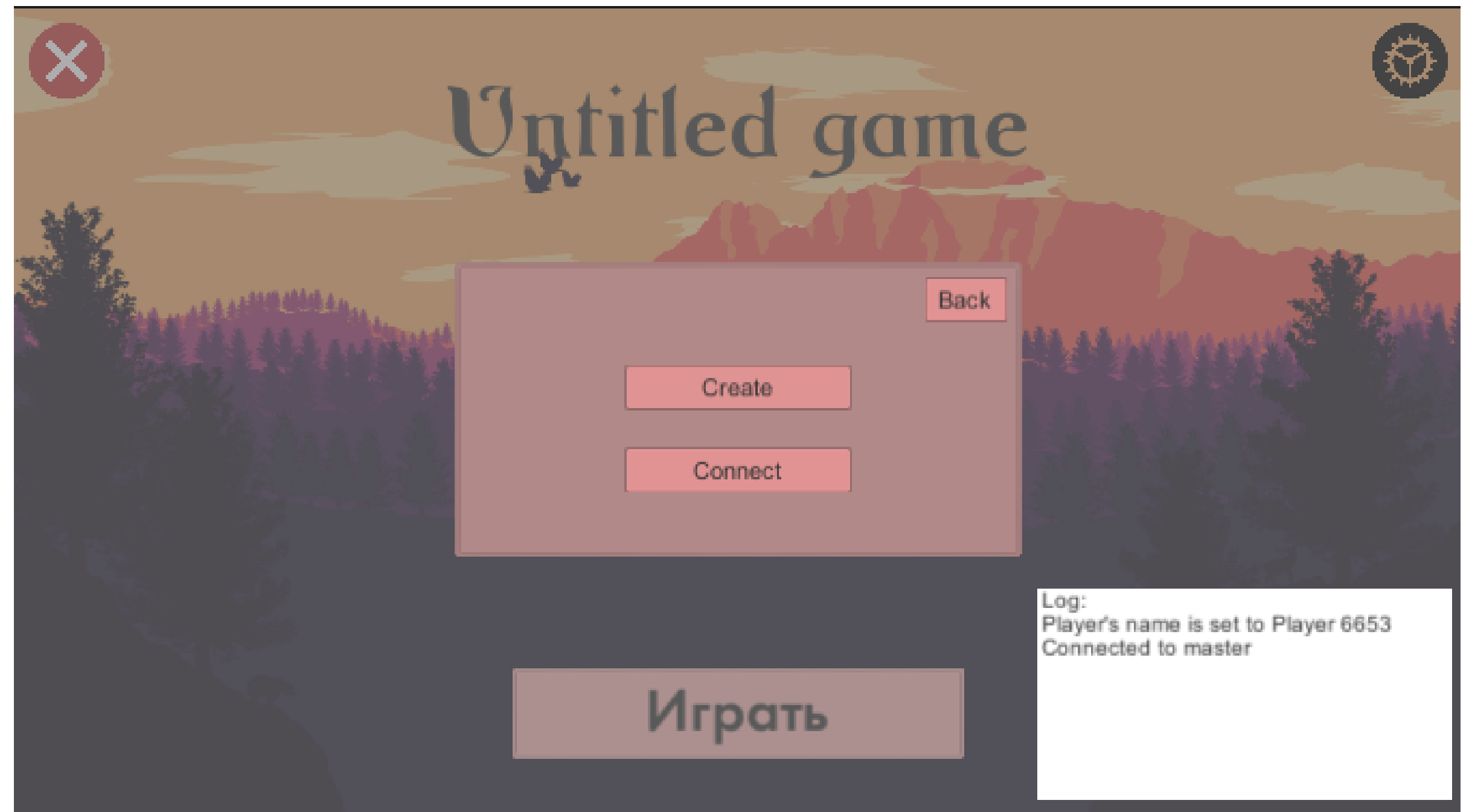


Рисунок 17 - Интерфейс запуска игры



ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА

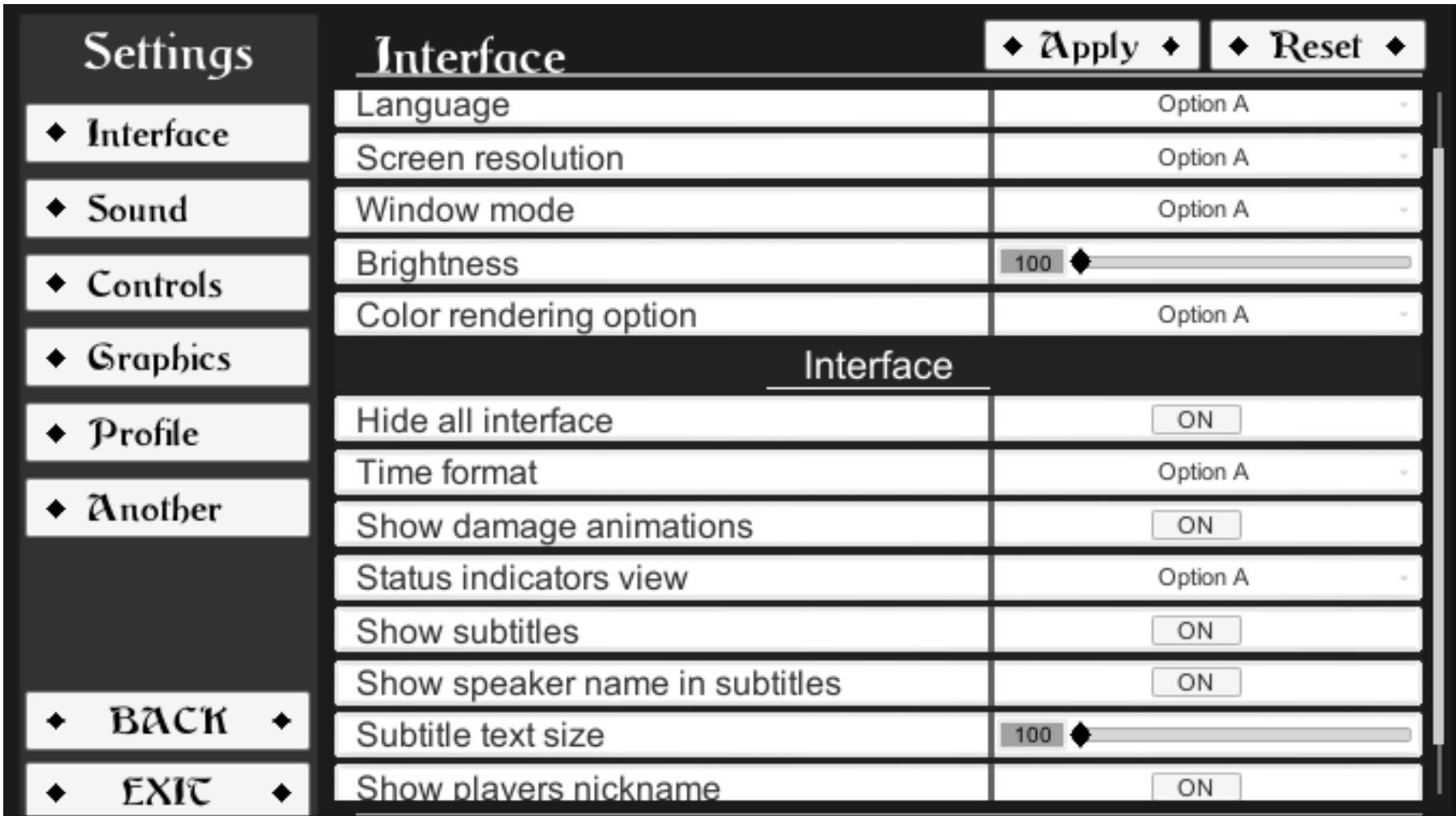


Рисунок 18 - Интерфейс меню настроек интерфейса

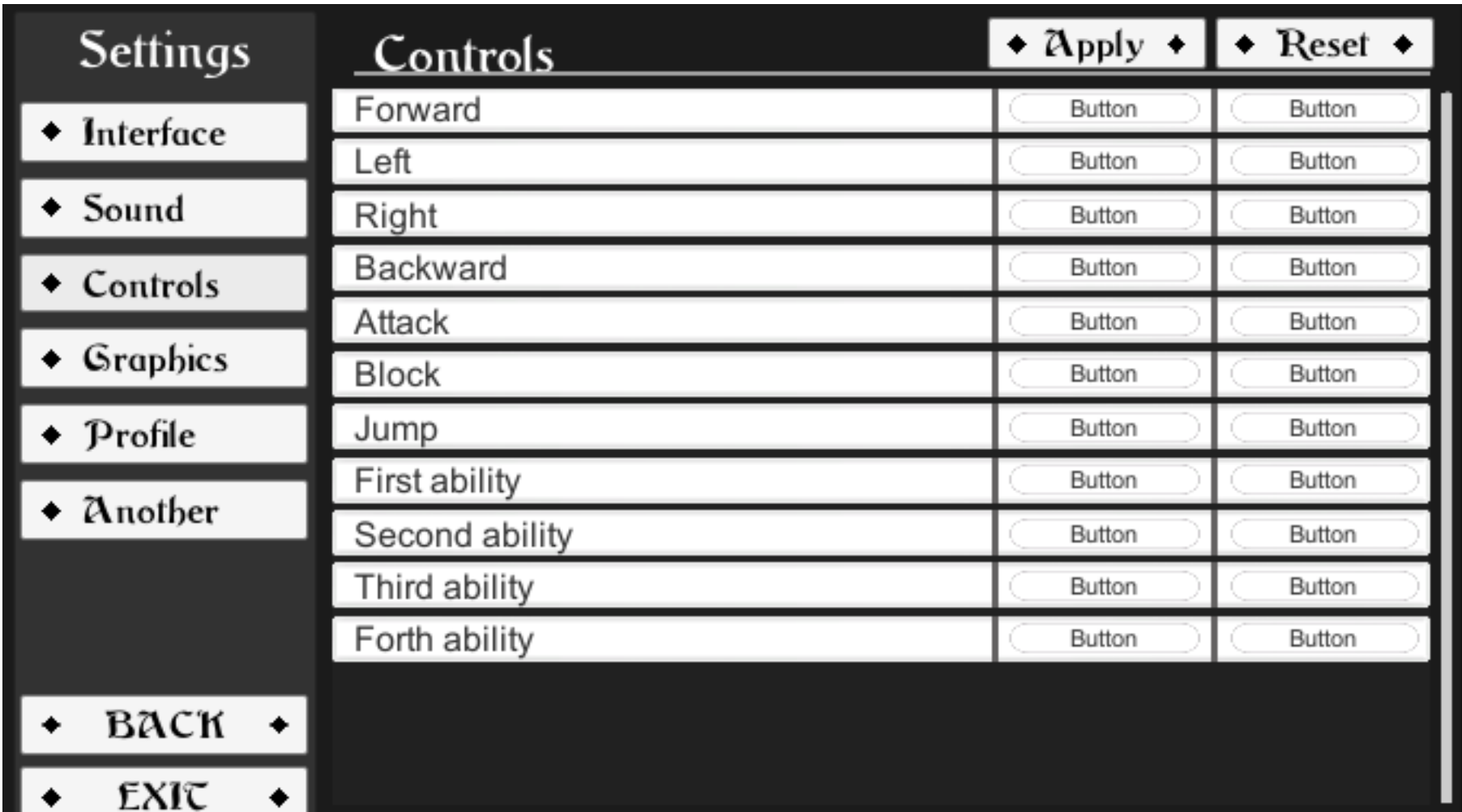


Рисунок 19 - Интерфейс меню настроек клавиш

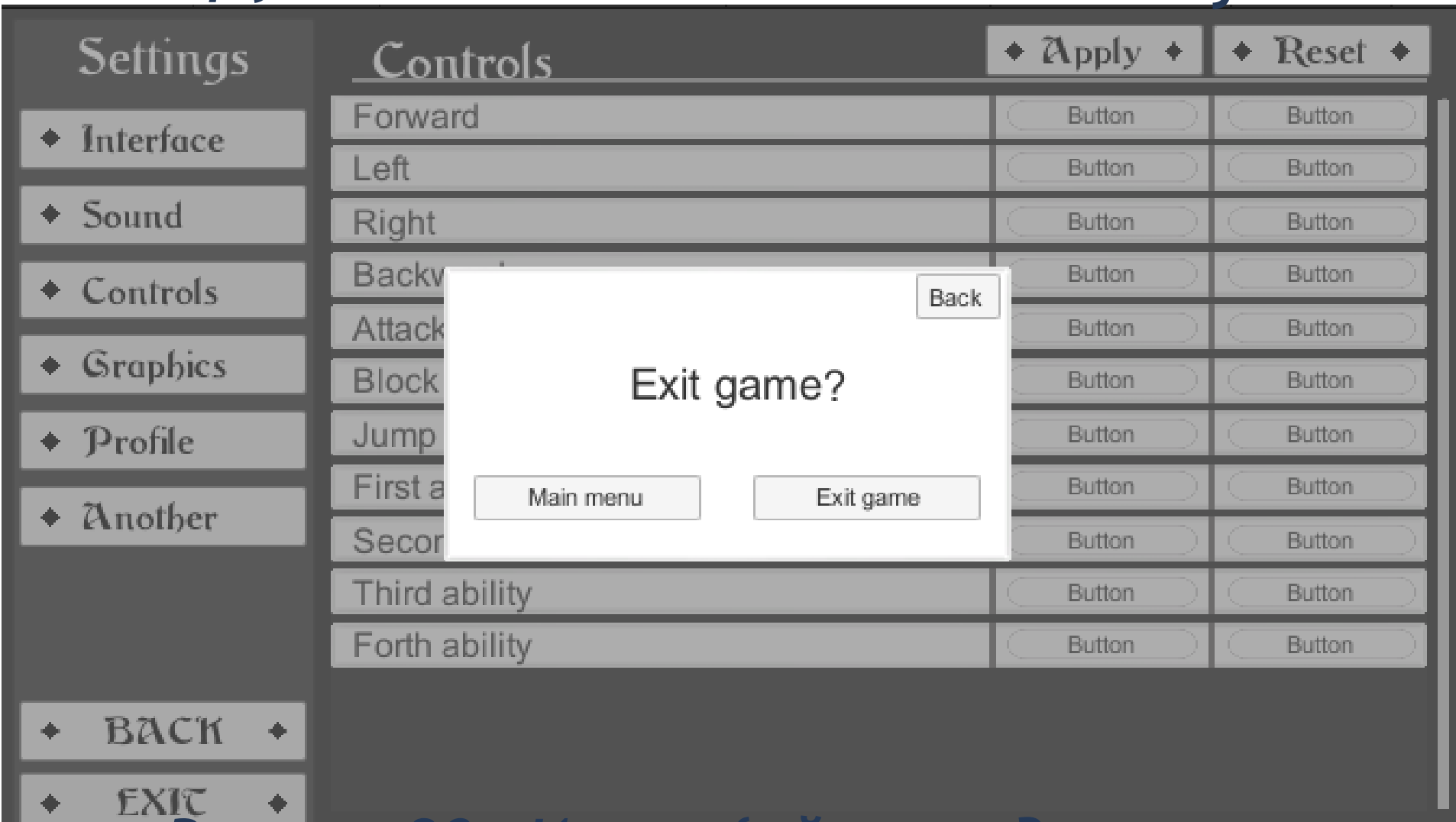


Рисунок 20 - Интерфейс выхода из игры



ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА

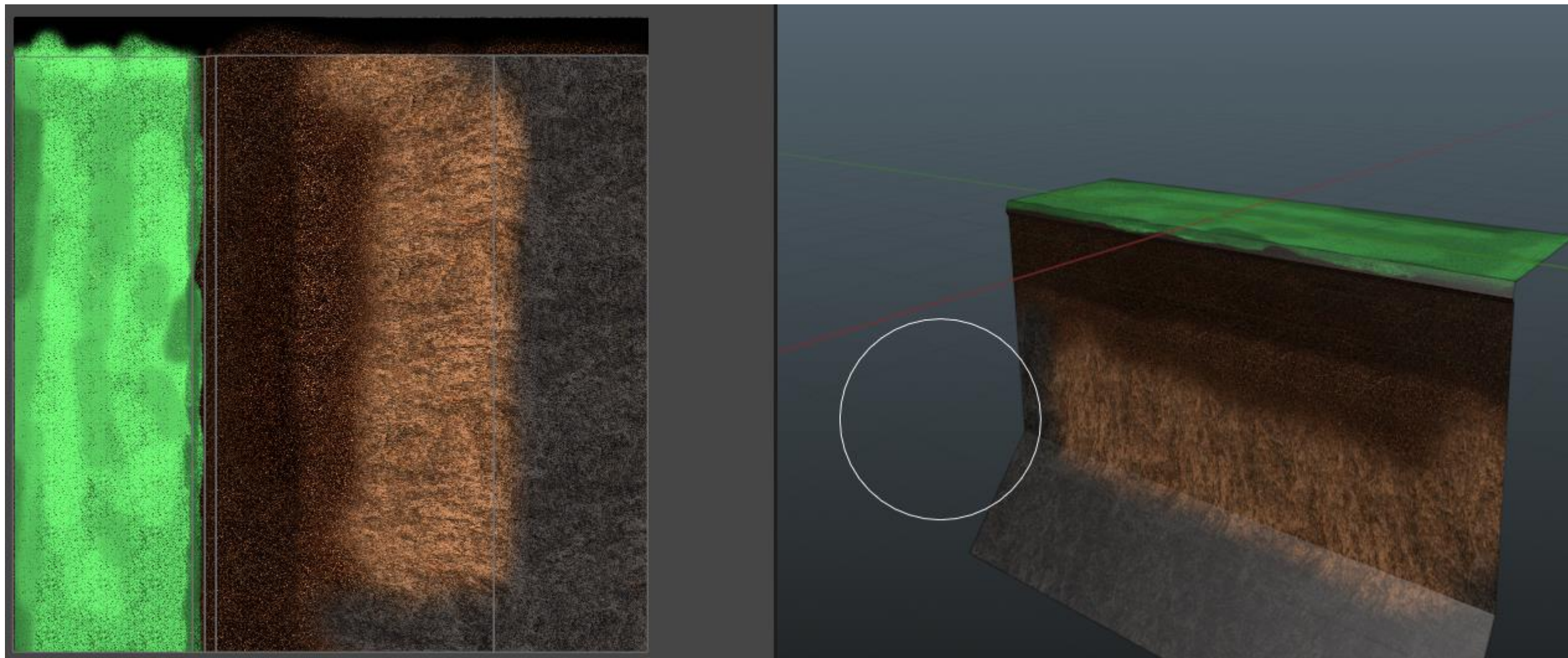


Рисунок 21 - Визуализация части игрового пространства



Рисунок 22 - Визуализация персонажа игры



РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

№ Теста	Действие	Ожидаемый результат	Реальный результат
T1	Нажатие кнопки «Продолжить» без сохранения	Кнопка не нажимается	Кнопка не нажимается
T2	Нажатие кнопки «Продолжить» с сохранением	Загрузка сохраненного уровня	Загрузка сохраненного уровня
T3	Нажатие кнопки «Новая игра»	Загрузка первого уровня	Загрузка первого уровня
T4	Нажатие кнопки «Настройки»	Открытие панели настроек	Открытие панели настроек
T5	Нажатие кнопки «Выход»	Сообщение о выходе	Сообщение о выходе
T6	Изменение настроек аудио и кнопка «Принять изменения»	Изменения принимаются	Изменения принимаются
T7	Изменение настроек видео и кнопка «Принять изменения»	Изменения принимаются	Изменения принимаются
T8	Изменение настроек игры и кнопка «Принять изменения»	Изменения принимаются	Изменения принимаются
T9	Изменение настроек аудио без кнопки «Принять изменения»	Изменения не принимаются	Изменения принимаются
T10	Изменение настроек видео и кнопки «Принять изменения»	Изменения не принимаются	Изменения не принимаются
T11	Изменение настроек игры и кнопки «Принять изменения»	Изменения не принимаются	Изменения не принимаются
T12	Движение персонажем через стены	Персонаж упирается в стену	Персонаж упирается в стену
T13	Беспорядочное нажатие на все кнопки	Сообщение об ошибке	Залипание клавиш
T14	Подключение геймпада	С него можно управлять	С него можно управлять
T15	Закрытие через диспетчер задач	Сохранение было произведено	Сохранение было произведено

Было проведено бета тестирование с участием учащихся НИУ «ВШЭ-Пермь» с применением технологий черного и белого ящиков

Так же было проведено интеграционное тестирование

После глобальных изменений программы добавлялись и использовались юнит тесты

Рисунок 23 - Критерии черного ящика



ЗАКЛЮЧЕНИЕ

1. Был проведен анализ действующих алгоритмов.
2. Были проанализированы, сопоставлены и выбран визуализатор для игры.
3. Были изучены способы разработки многопользовательских игр.
4. Был спроектирован алгоритм многопользовательской игры.
5. Был спроектирован интерфейс для игры.
6. Был проведен опрос в ходе тестирования и проанализированы ответы по поводу интерфейса, внести изменения.
7. Был реализован код программы и предварительный интерфейс.
8. Был реализован проект с использованием алгоритма многопользовательской игры.
9. Была написана пользовательская документация для программы.
10. Было проведено тестирование, для выявления недочетов и сбора критики тестировщиков.
11. Были внесены изменения на основании выявленных недочетов на этапе тестирования.

СПАСИБО ЗА ВНИМАНИЕ
РАД ОТВЕТИТЬ НА ВАШИ ВОПРОСЫ



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Автор: Чепоков Е.С.

Телефон.: +7 (951) 95-94-666

Электронная почта: eschepokov@edu.hse.ru

