

Процесс разработки runner-игры на движке Unity

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье исследуется процесс разработки компьютерной игры жанра runner на игровом движке Unity. Актуальность работы обусловлена высокой популярностью жанра, его простотой механик и широкой аудиторией игроков. Рассматриваются этапы создания игрового прототипа, включая проектирование игровой логики, настройку физических компонентов, работу с анимацией и интерфейсом пользователя, а также реализацию основных механик взаимодействия. Практическая значимость заключается в систематизации навыков разработки игр и демонстрации возможностей Unity для создания интерактивных приложений. Проведённый анализ и тестирование подтвердили корректность работы игровых элементов и стабильность функционирования прототипа, что делает предложенный подход применимым для образовательных и прикладных целей.

Ключевые слова: Unity, runner-игра, игровая механика, C#, физика в играх, интерфейс пользователя, игровой прототип.

The Process of Developing a Runner Game on the Unity Engine

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article explores the process of developing a runner-type computer game using the Unity game engine. The relevance of the work is determined by the genre's high popularity, its simple mechanics, and wide player audience. The stages of creating a game prototype are examined, including the design of game logic, configuration of physical components, working with animation and user interface, as well as the implementation of core gameplay mechanics. The practical significance lies in systematizing game development skills and demonstrating Unity's capabilities for creating interactive applications. The conducted analysis and testing confirmed the correctness of the game elements and the stability of the prototype, making the proposed approach applicable for educational and practical purposes.

Keywords: Unity, runner game, game mechanics, C#, game physics, user interface, game prototype.

1 Введение

1.1 Актуальность

В последние годы индустрия видеоигр демонстрирует устойчивый рост, а мобильные и инди-игры занимают значительную долю рынка цифровых развлечений. Одним из наиболее востребованных и популярных жанров является runner-игра, отличающаяся простотой механик, высокой динамичностью и ориентированностью на широкую аудиторию. Разработка игр данного жанра представляет интерес как с точки зрения программной реализации, так и с позиции проектирования игрового процесса и пользовательского опыта.

Игровой движок Unity является одним из самых распространённых инструментов для создания 2D и 3D-игр благодаря кроссплатформенности, развитой экосистеме, поддержке языка C# и обширной документации. Это делает Unity особенно актуальным для обучения разработке игр и прототипировании игровых проектов. Несмотря на большое количество готовых решений и обучающих материалов, процесс разработки runner-игры требует комплексного подхода, включающего проектирование архитектуры, реализацию игровой логики, работу с анимацией, физикой и оптимизацией производительности.

В связи с этим изучение процесса разработки runner-игры на движке Unity является актуальным, так как позволяет систематизировать практические навыки геймдева, продемонстрировать применение современных технологий и инструментов разработки, а также может быть использовано в образовательных и прикладных целях.

1.2 Обзор исследований

Д.А. Дашиев в своей статье рассматривает разработку интерактивной среды для создания адаптивных игр с анимированными персонажами, позволяющей в режиме проектирования и запуска гибко изменять игровые миры и правила без написания программного кода, что повышает удобство разработки, снижает нагрузку на разработчиков и расширяет возможности применения геймификации в образовательных и прикладных целях [1]. М. Р. Абляев, Ф. Р. Аметов в своей работе анализируют процесс разработки игр в интегрированной среде Unity, рассматривая условия её использования, поддерживаемые платформы, основные среды программирования, структуру и функционал пользовательского интерфейса, а также методы оптимизации графики и возможности встроенного интернет-магазина компании Unity [2]. Н.В. Гоголев в своей статье рассматривает основные жанры и механики видеоигр, включая шутеры, стратегии и ролевые игры, анализирует взаимодействие игрока с игровым миром и приводит примеры кода на C++, ориентированные на разработчиков и заинтересованных в игровой индустрии [3]. Д. Рахмани, Д.А. Кубатин, И.С. Гераськин в своей статье проводят сравнительный анализ сетевых библиотек и фреймворков ENet C#, Unity NetCode, Mirror и Riptide для разработки многопользовательских игр на Unity и C#, оценивая их по критериям стабильности, производительности,

масштабируемости и удобства использования, а также формулируют практические рекомендации по выбору оптимального решения для игровых проектов различного масштаба [4]. Д.Г. Евсеев в своей статье проводит сравнительный анализ игровых движков Godot Engine, GameMaker, Unity и Unreal Engine по функциональности, сложности освоения, поддержке сообществ, платформ, языков программирования и стоимости лицензий, а также предоставляет рекомендации по выбору оптимальной среды разработки, соответствующей целям и требованиям проекта [5].

1.3 Цель исследования

Целью исследования является практическая реализация процесса разработки runner-игры с использованием игрового движка Unity, включая проектирование игровой логики, реализацию основных механик, работу с графическими и анимационными элементами, а также анализ применяемых инструментов и технологий.

2 Материалы и методы

В ходе выполнения исследования использованы программные, технические и информационные материалы, необходимые для разработки компьютерной игры жанра runner. В качестве основного инструмента разработки применялся игровой движок Unity, обеспечивающий создание 2D/3D-игр, управление сценами, объектами, физикой. Для реализации игровой логики использовался язык программирования C#, являющийся основным языком скриптинга в Unity.

В качестве графических материалов применялись спрайты, модели, созданные с использованием встроенных средств Unity, а также сторонних графических редакторов.

В процессе разработки применялись методы объектно-ориентированного программирования, модульного проектирования и событийного управления. Проектирование структуры игры осуществлялось на основе анализа жанра runner и существующих игровых решений. Для реализации игровых механик использовались методы обработки пользовательского ввода, управления игровыми объектами, подсчёта игровых очков.

Методами исследования также являлись анализ и обобщение теоретических источников по разработке игр, изучение официальной документации Unity, а также практическое экспериментирование и тестирование игрового прототипа. Для оценки корректности работы игры применялись методы функционального тестирования и отладки, включая использование встроенных средств профилирования Unity.

3 Результаты

Для начала создается игровое поле (рис. 1).

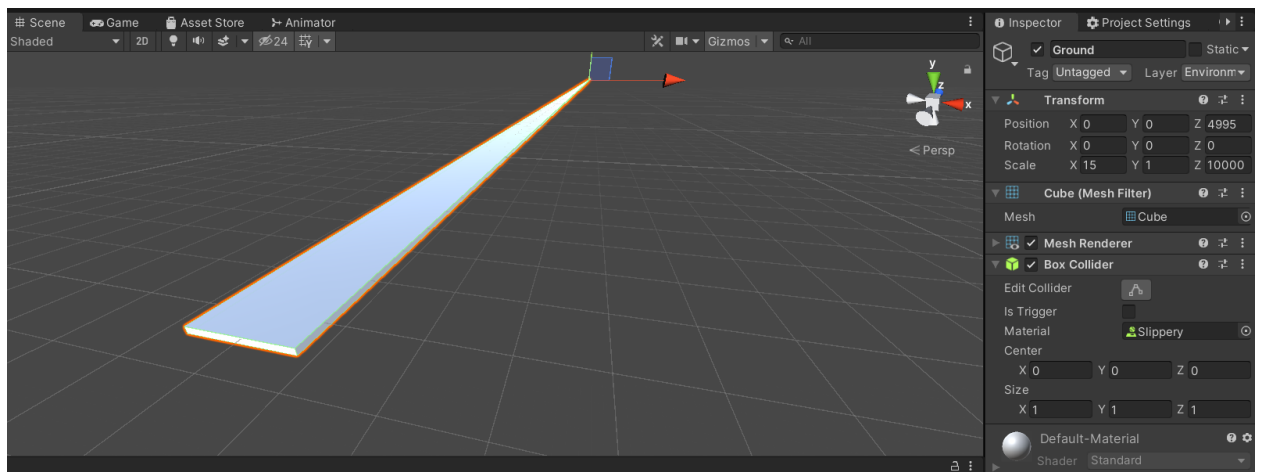


Рисунок 1 – Игровое поле

Потом создадим препятствия и добавим к ним компонент «Box Collider», чтобы, когда игрок к ним прикасается он не проходил объекты на сквозь (рис. 2).

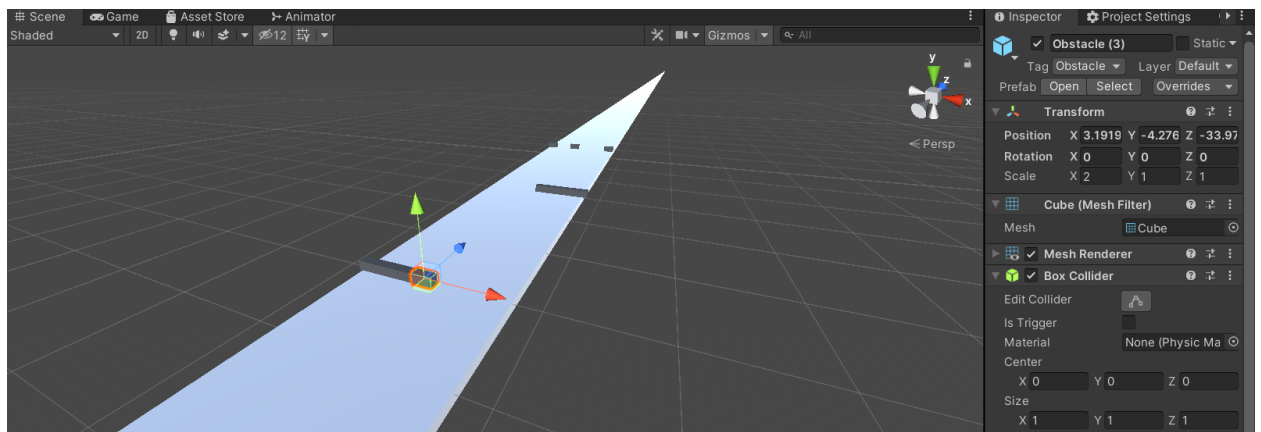


Рисунок 2 – Препятствия

Добавим игрока, и так же добавим ему компонент «Box Collider», чтобы он не проходил объекты на сквозь, и компонент «Rigidbody» для того что бы добавить, игроку физику (рис. 3).

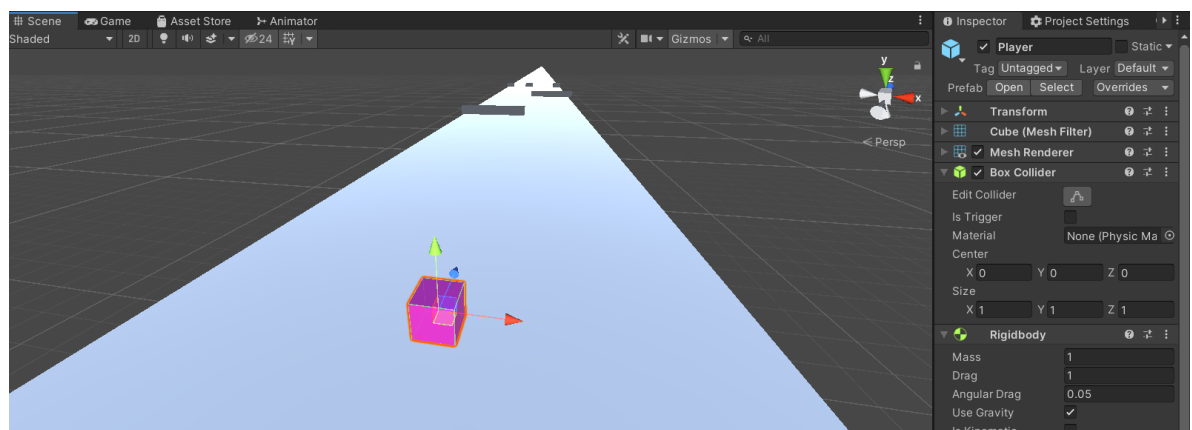
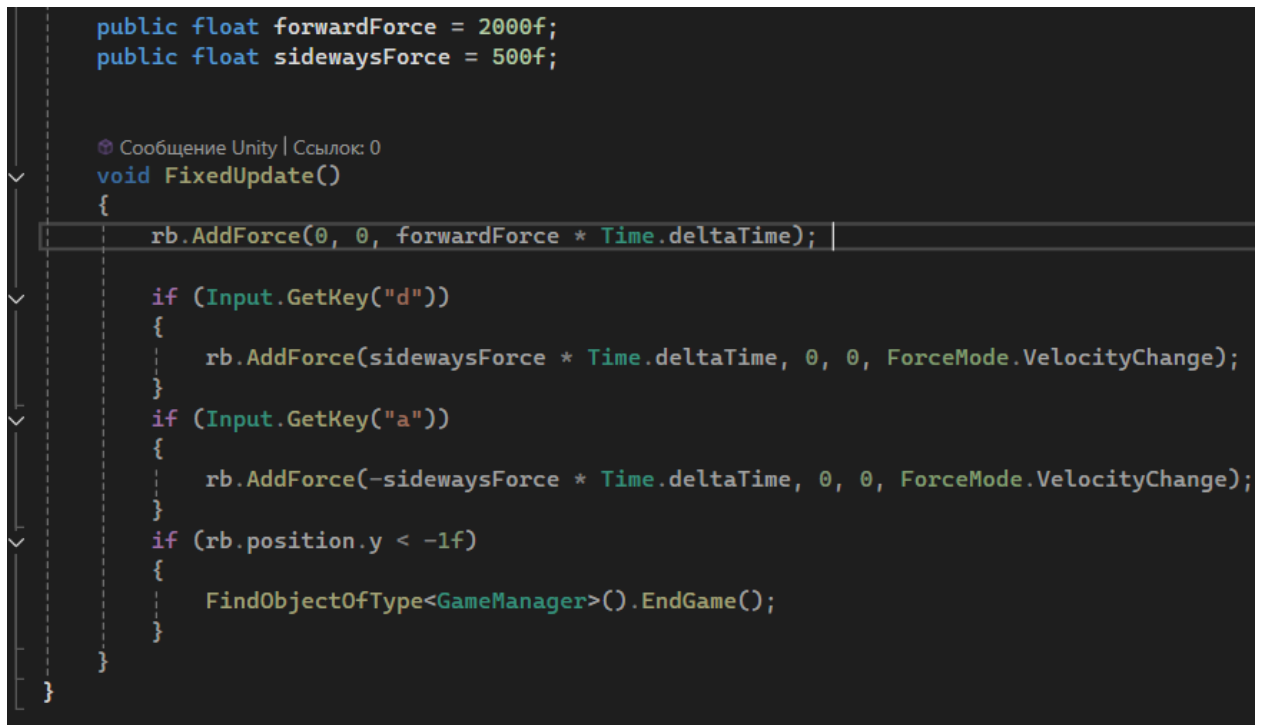


Рисунок 3 – Игрок

Скрипт «PlayerMovement» для перемещения игрока (рис. 4).



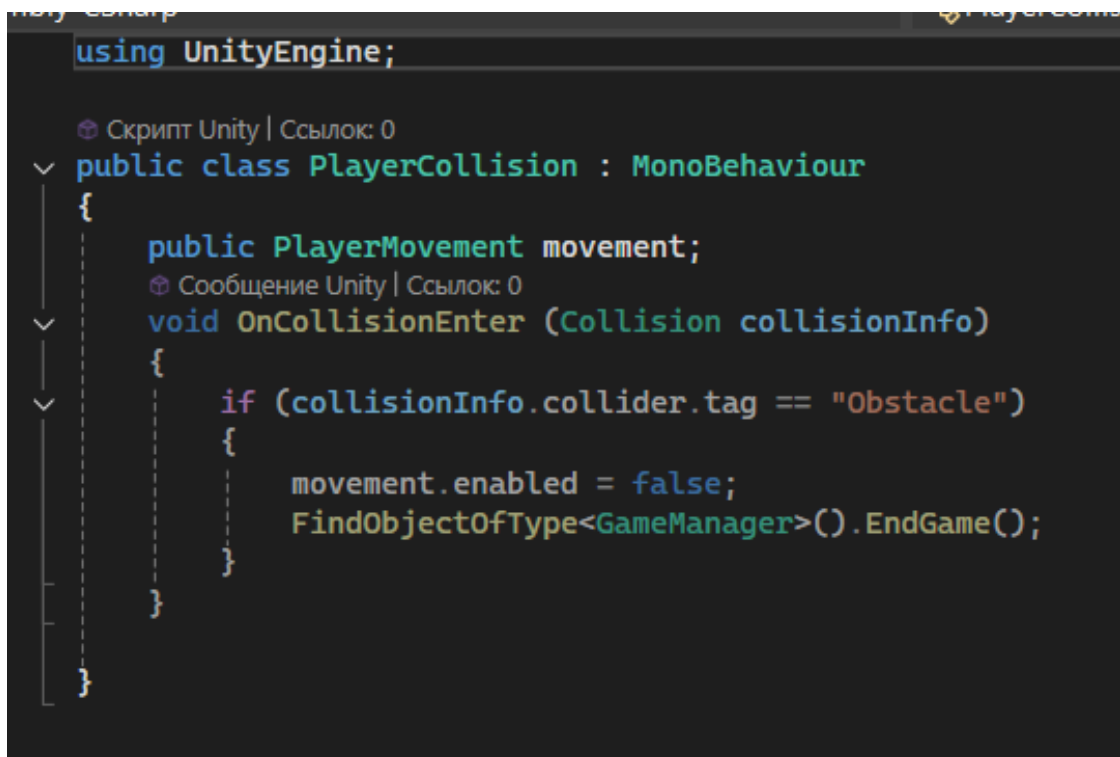
```
public float forwardForce = 2000f;
public float sidewaysForce = 500f;

Сообщение Unity | Ссылок: 0
void FixedUpdate()
{
    rb.AddForce(0, 0, forwardForce * Time.deltaTime);

    if (Input.GetKey("d"))
    {
        rb.AddForce(sidewaysForce * Time.deltaTime, 0, 0, ForceMode.VelocityChange);
    }
    if (Input.GetKey("a"))
    {
        rb.AddForce(-sidewaysForce * Time.deltaTime, 0, 0, ForceMode.VelocityChange);
    }
    if (rb.position.y < -1f)
    {
        FindObjectOfType<GameManager>().EndGame();
    }
}
```

Рисунок 4 – PlayerMovement

Скрипт «PlayerCollision» для проигрыша игрока если он упадет с платформы или коснется препятствия (рис. 5).



```
using UnityEngine;

Сообщение Unity | Ссылок: 0
public class PlayerCollision : MonoBehaviour
{
    public PlayerMovement movement;
    Сообщение Unity | Ссылок: 0
    void OnCollisionEnter (Collision collisionInfo)
    {
        if (collisionInfo.collider.tag == "Obstacle")
        {
            movement.enabled = false;
            FindObjectOfType<GameManager>().EndGame();
        }
    }
}
```

Рисунок 5 – PlayerCollision

Создаем объект «Canvas» и к нему подключаем камеру, что бы он выводил пользовательский интерфейс (рис.6).

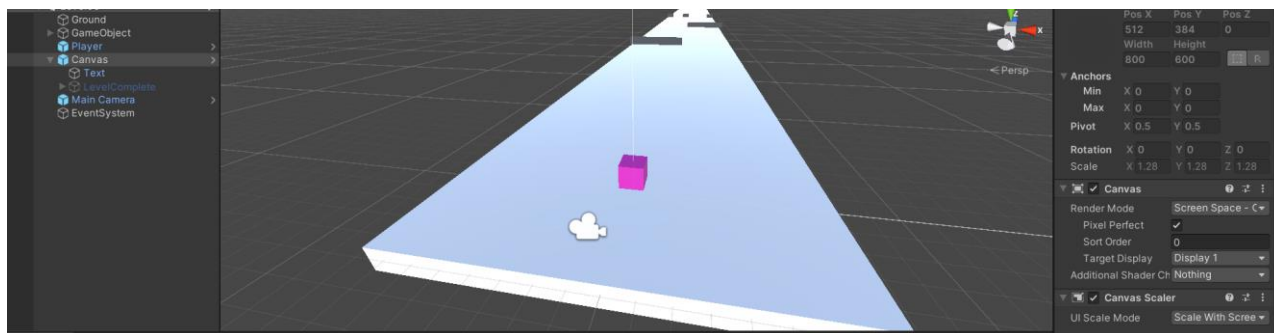


Рисунок 6 – Canvas

Далее в «Canvas» создаем объект «Text» и пишем небольшой скрипт что бы он изменял число в зависимости от пройденного расстояния (рис. 7).

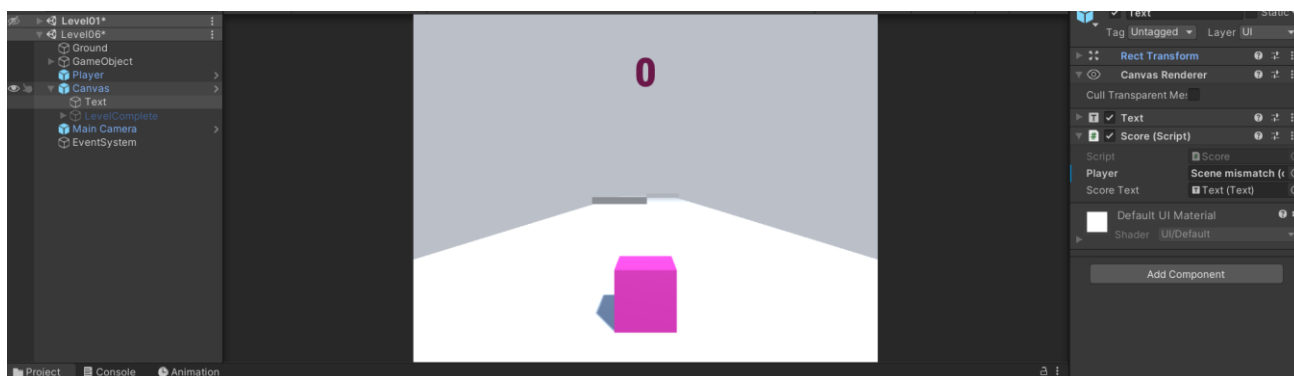


Рисунок 7 – Настройки интерфейса

Создаем меню для этого на отдельной сцене, создаем 2 кнопки, «Начать игру» и «Выход», а также текстовый объект для вывода текста (рис. 8).

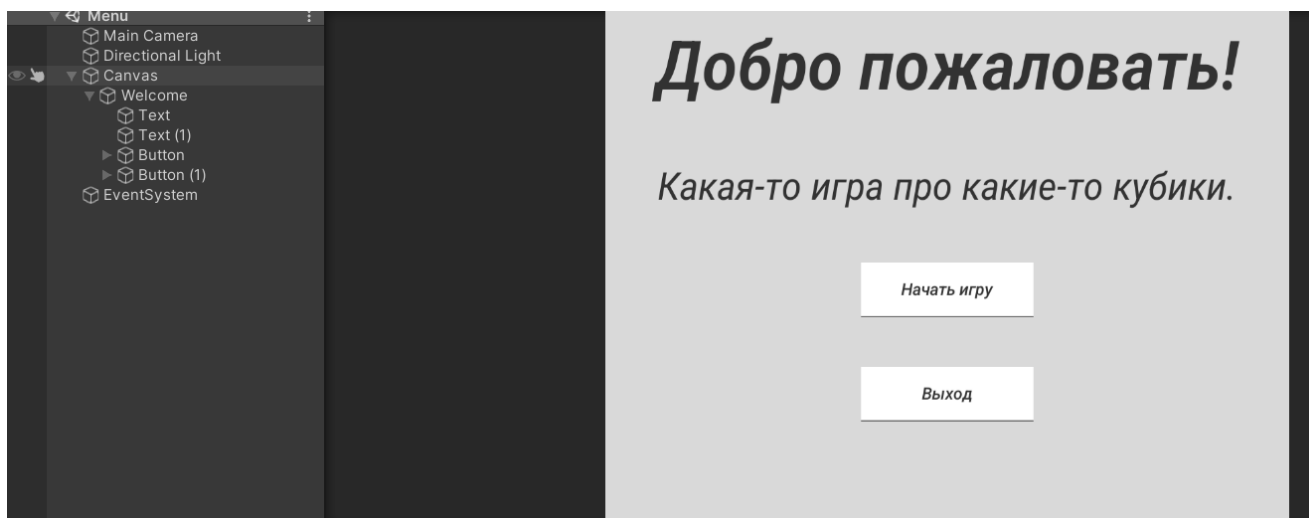


Рисунок 8 – Меню

На левой части показанная как выглядит интерфейс в движке Unity, а на правой как у пользователя, во время игры (рис. 9).

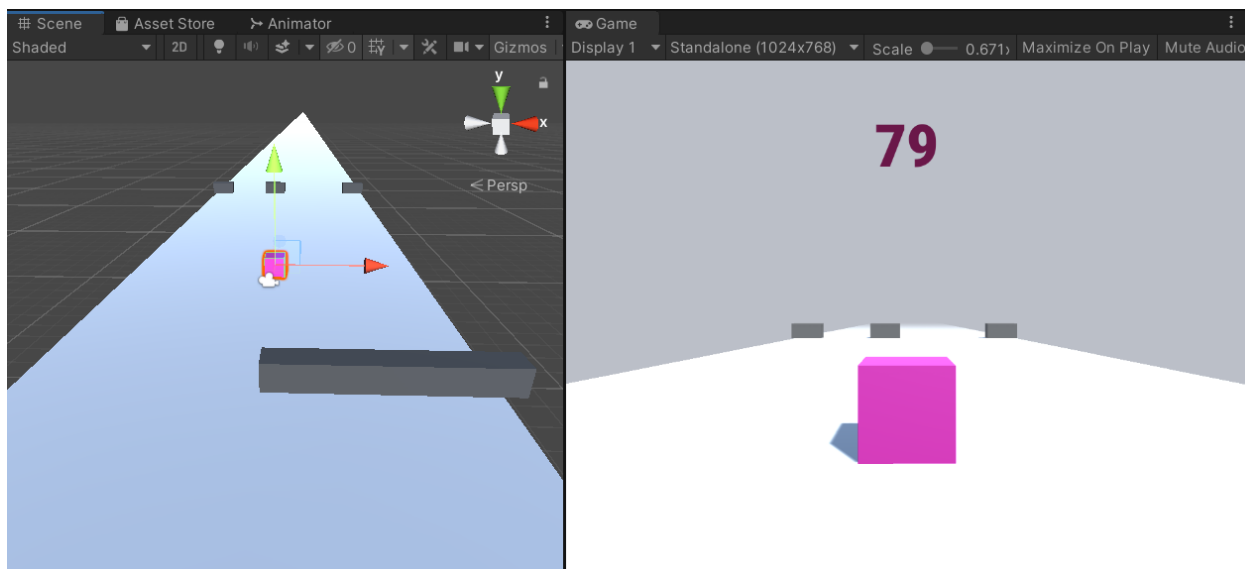


Рисунок 9 – Интерфейс игры

4 Выводы

В ходе данного исследования был рассмотрен и проанализирован процесс разработки компьютерной игры жанра runner с использованием игрового движка Unity. Рассмотрены особенности жанра, определены основные требования к игровым механикам и структуре проекта, а также реализован игровой прототип.

В процессе работы применены современные инструменты и технологии разработки игр, включая язык программирования C#. Использование объектно-ориентированного подхода и модульного проектирования позволило обеспечить гибкость и расширяемость игровой архитектуры. Проведённое тестирование подтвердило корректность работы основных игровых механик и стабильность функционирования разработанного приложения.

Результаты исследования подтверждают целесообразность использования игрового движка Unity для разработки игр жанра runner и демонстрируют его широкие возможности при создании интерактивных приложений. Полученные в ходе работы знания и практические навыки могут быть использованы в образовательных целях, а также послужить основой для дальнейшего развития и усложнения игрового проекта.

Библиографический список

1. Дашиев Д. А. Разработка игр на Unity // Информационные системы и технологии в образовании, науке и бизнесе: Материалы всероссийской научно-практической конференции, Улан-Удэ, 09–10 июня 2022 года / Научный редактор А.А. Тонхоноева. Улан-Удэ: Бурятский

- государственный университет имени Доржи Банзарова, 2022. С. 29-37. URL: <https://elibrary.ru/item.asp?id=48716850>
2. Абляев М. Р., Аметов Ф. Р. Разработка игр с использованием UNITY // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2016. №. 2. С. 42-52. URL: http://www.cepulib.ru/documents/ukazateli/iktVobraz/2016_2.pdf#page=41
 3. Гоголев Н. В. Основные жанры и механики видеоигр // Научное сообщество студентов: междисциплинарные науки: Материалы II Всероссийской молодёжной научно-практической конференции, Орёл, 14 мая 2025 года. Орёл: 2025. С. 195-204 URL: <https://elibrary.ru/item.asp?id=82561411>
 4. Рахмани Д., Кубатин Д. А., Гераськин И. С. Сравнительный анализ сетевых библиотек и фреймворков, используемых при разработке игр на игровом движке Unity и языке программирования C# // Международный журнал гуманитарных и естественных наук. 2025. № 9-1. С. 208-213. URL: <https://elibrary.ru/item.asp?id=83085376>
 5. Евсеев, Д. Г. Выбор среды разработки для создания компьютерных игр // Актуальные проблемы современной науки: взгляд молодых учёных: Материалы Национальной научно-практической студенческой конференции, Брянск, 13–14 декабря 2023 года. Брянск: 2023. С. 556-562. URL: <https://elibrary.ru/item.asp?id=65360223>