

УДК 004.946

<https://doi.org/10.33619/2414-2948/115/25>

РАЗРАБОТКА НА ЯЗЫКЕ C# В РЕДАКТОРЕ UNITY ЛОГИЧЕСКОЙ ИГРЫ – ШУТЕРА ОТ ПЕРВОГО ЛИЦА

©Сафина Г.Ф., ORCID: 0000-0002-7326-0896, SPIN-код: 4562-2453, канд. физ.-мат. наук,
Уфимский университет науки и технологий, г. Нефтекамск, Россия, safinagf@mail.ru

©Кириллова Е. А., Уфимский университет науки и технологий,
г. Нефтекамск, Россия, elize.none@yandex.ru

©Коняев Ю. С., Уфимский университет науки и технологий,
г. Нефтекамск, Россия, iury.conyaev2016@yandex.ru

C # DEVELOPMENT IN UNITY LOGIC GAME EDITOR – FIRST PERSON SHOOTER

©Safina G., ORCID: 0000-0002-7326-0896, SPIN-code: 4562-2453, Ph.D.,
Ufa University of Science and Technology, Neftekamsk, Russia, safinagf@mail.ru

©Kirillova E., Ufa University of Science and Technology,
Neftekamsk, Russia, elize.none@yandex.ru

©Konayev Yu., Ufa University of Science and Technology,
Neftekamsk, Russia, iury.conyaev2016@yandex.ru

Аннотация. Приведены алгоритмы процесса разработки компьютерной игры-шутера от первого лица с помощью редактора Unity. Игра позволяет видеть окружающий мир глазами самого игрока, что значимо для обучающих игр с возможностями прохождения логических головоломок или выполнения конкретных заданий. Для реализации программной части игры привлечены библиотеки и функционал языка программирования C#. К визуализации персонажей и объектов игры применены приемы 3d-моделирования с помощью Blender.

Abstract. The algorithms of the process of developing a computer game-shooter from the first person using the Unity editor are given. The game allows you to see the world around you through the eyes of the player himself, which is significant for educational games with the ability to complete logical puzzles or complete specific tasks. To implement the software part of the game, libraries and functionality of the C # programming language are involved. 3D modeling techniques using Blender have been applied to the visualization of characters and objects in the game.

Ключевые слова: движок Unity, разработка игр, язык программирования C#, компьютерные игры, 3d-моделирование, игровое обучение.

Keywords: Unity engine, game development, C # programming language, computer games, 3d - modeling, game learning.

Работа посвящена алгоритмической разработке компьютерной игры в виде прохождения обучающимся (игроком) определенных логических заданий. Игра разработана в формке шутера от первого лица, что позволяет игроку в полной мере находиться в окружающем пространстве и видеть прохождение этапов игры «своими глазами». Такие игры не только развивают практические когнитивные навыки, но позволяют развивать скорость реагирования на меняющиеся ситуации вокруг игрока, тренируя его мышление, развивая адекватную реакцию и логическое принятие быстрых и обдуманных решений [1–2].

В качестве среды разработки игры была выбрана редакторская среда Unity, которая широко применяется разработчиками компьютерных игр, так как обладает рядом преимуществ [3, 4]: может применяться для разных платформ и операционных систем (мобильные устройства, компьютеры, интернет-приложения и т.д.): имеет широкие приемы визуализации, в том числе математических моделей; содержит модульную систему готовых компонент, наборов встроенных ресурсов (<https://clck.ru/3MNmgC>). Основные шаги реализации проекта состояли из решения следующих задач разработки игры — шутера от первого лица: разработка внешнего вида и сюжета игры; создание необходимых для проекта моделей и персонажей; написание программного кода для реализации механик, задействованных в проекте; создание игровой локации и расстановки в ней враждебных ботов; сбор (компановка) приложения, тестирование и исправление возможных ошибок. Целевой аудиторией проекта является детская и подростковая аудитория, которая проявляет интерес к быстрым и активным действиям при прохождении в игре логических заданий. Разработка игры проведена в редакторе Unity. Для написания программного кода на языке C# была использована программа Microsoft Visual Studio. Создание игровых объектов и объектов окружения осуществлялось с помощью Blender [3].

Также в проекте задействованы ассеты из Unity Asset Store, помогающие сформировать окружение, такие как SimplePoly — Town Pack, Low poly city vehicle pack, Simple Town Lite — Cartoon Assets, CITY package, Handpainted Grass & Ground Textures. В качестве основной локации был выбран город. Выполнение логических заданий игры предполагает визуальный сбор предметов — «кусочков хлеба», которые необходимо найти в разных частях локации игры. Препятствием для достижения цели являются враждебные mobs — «голуби», наносящие игроку (шутеру от первого лица) контактный урон. В проекте созданы модели «врагов-голубей», «кусочков хлеба», используемого в качестве монет, а также часть деревьев и кустарников. Предметы окружения взяты также из библиотек Unity Asset Store. Следующим шагом была разработка программного кода для реализации игровых механик и интерфейса. На Рисунках 1 и 2 представлены фрагменты кода Move.cs для управления движением игрока и примененные настройки.

```
Сообщение Unity | Ссылка 0
void Update()
{
    rotationX -= Input.GetAxis("Mouse Y") * Settings.sensitivityVert;
    rotationX = Mathf.Clamp(rotationX, Settings.minVert, Settings.maxVert);
    rotationY += Input.GetAxis("Mouse X") * Settings.sensitivityHor;

    transform.localEulerAngles = new Vector3(0f, rotationY, 0f);

    playerCamera.transform.localEulerAngles = new Vector3(rotationX, 0f, 0f);
}

Сообщение Unity | Ссылка 0
private void FixedUpdate()
{
    if (canMove)
    {
        Vector3 movement = new Vector3(0, rb.velocity.y - 0.5f, 0);

        if (Input.GetKey(KeyCode.UpArrow) || Input.GetKey(KeyCode.W))
            movement += transform.forward * Settings.speedPlayer;

        if (Input.GetKey(KeyCode.DownArrow) || Input.GetKey(KeyCode.S))
            movement -= transform.forward * Settings.speedPlayer;

        if (Input.GetKey(KeyCode.LeftArrow) || Input.GetKey(KeyCode.A))
            movement -= transform.right * Settings.speedPlayer;

        if (Input.GetKey(KeyCode.RightArrow) || Input.GetKey(KeyCode.D))
            movement += transform.right * Settings.speedPlayer;
    }
}
```

Рисунок 1. Фрагмент кода Move.cs управления движением игрока

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Ссылка: 11
public static class Settings
{
    public static float sensitivityHor = 7f;
    public static float sensitivityVert = 7f;
    public static float minVert = -45f;
    public static float maxVert = 45f;

    public static float speedPlayer = 15;
    public static int jumpPlayerForce = 1100;
    public static int maxCountEnemy = 10;
    public static int countEnemy = 0;
    public static float intervalFirePlayer = 0.0f;
}
```

Рисунок 2. Настройки

Также были разработаны коды проекта, отвечающие за соответствующие действия: Fire.cs — прохождение этапов и здоровье (выживание) игрока; DestroyObject.cs — удаление объектов («врагов») через заданное время; CoinScript.cs — сбор объектов; UIManager.cs — вывод нужной информации на экран; StartMessage.cs — появление и исчезновение приветственного сообщения при входе в игру (код стартового сообщения); Win.cs — появление и исчезновение сообщения об успешном завершении миссии (код победного сообщения); Defeat.cs — вывод сообщения о поражении и отключения у игрока возможности перемещения при «гибели» (код, срабатывающий в случае гибели игрока); Win.cs — появление и исчезновение сообщения об успешном завершении миссии (код завершающего сообщения игры). На Рисунках 3–5 представлены фрагменты демонстрации разработанной игры шутера от первого лица.



Рисунок 3. Демонстрация стартового сообщения



Рисунок 4. Локация и враждебные mobs



Рисунок 5. «Хлеб» (объекты для сбора) на локации

Таким образом, разработана с яркой графикой и интуитивно понятным интерфейсом позволяющая игроку поэтапно проходить логические задания, находясь в окружающем пространстве, просматривая результаты прохождения игры «своими глазами», принимая решения с соответствующей визуальной и мыслительной реакцией.

Игра прошла успешное тестирование в своей целевой аудитории. Имеет возможность развития — актуализации и дополнения новыми логическими заданиями, игровыми объектами и т.д.

Может дополняться и другими проектировщиками — разработчиками игр — шутеров от первого лица, так как использованный для разработки программный и алгоритмический стек проекта (Unity, C#, Microsoft Visual Studio, Blender, Unity Asset Store) понятен всем разработчикам, работающим в аналогичной сфере игр.

Список литературы:

1. Мэннинг Д. Unity и для разработчика. СПб: Питер, 2018. 352 с.
2. Digital educational resource - computer game. research. benefits and harms // Modern problems of science and education. 2021. №6. P. 189-189. <https://doi.org/10.17513/spno.31407>
3. Корнилов А. В. Unity. Полное руководство. СПб.: Наука и техника, 2021. 496 с.
4. Ларкович С. Н. Unity на практике. Создаем 3D-игры и 3D-миры. М.: Наука и техника, 2022. 384 с.

References:

1. Menning, D. (2018). Unity i dlya razrabotchika. St. Petersburg. (in Russian).
2. Digital educational resource - computer game. research. benefits and harms (2021). *Modern problems of science and education*, (6), 189-189. (in Russian). <https://doi.org/10.17513/spno.31407>
3. Kornilov, A. V. (2021). Unity. Polnoe rukovodstvo. St. Petersburg. (in Russian).
4. Larkovich, S. N. (2022). Unity na praktike. Sozdaem 3D-igry i 3D-miry. Moscow. (in Russian).

*Работа поступила
в редакцию 21.03.2025 г.*

*Принята к публикации
30.03.2025 г.*

Ссылка для цитирования:

Сафина Г.Ф., Кириллова Е. А., Коняев Ю. С. Разработка на языке C# в редакторе Unity логической игры – шутера от первого лица // Бюллетень науки и практики. 2025. Т. 11. №6. С. 182-185. <https://doi.org/10.33619/2414-2948/115/25>

Cite as (APA):

Safina, G., Kirillova, E., & Konyaev, Yu. (2025). C# Development in Unity Logic Game Editor – First Person Shooter. *Bulletin of Science and Practice*, 11(6), 182-185. (in Russian). <https://doi.org/10.33619/2414-2948/115/25>