УДК 004

Разработка сетевой игры «Новый год»

Матвеева Алёна Сергеевна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
Целью данного исследования является разработать сетевую игру «Новый год». Для реализации использовалась программы Unity, Visual Studio и язык программирования C#. Данная статья может быть использована в качестве основы для дальнейших исследований и разработок в области информационно-коммуникационных систем и сетей, а также в сфере развлекательной индустрии.
Ключевые слова: Игра, Unity, объект.

Development of the online game "New Year"

Matveeva Alyona Sergeevna
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this study is to develop a network game "New Year". Unity, Visual Studio, and the C# programming language were used for implementation. This article can be used as a basis for further research and development in the field of information and communication systems and networks, as well as in the entertainment industry.
Keywords: Game, Unity, object.

1 Введение

1.1 Актуальность

В современном мире информационно-коммуникационные системы и сети играют важную роль во многих сферах деятельности. Они позволяют людям обмениваться информацией, взаимодействовать друг с другом и создавать новые возможности для развития. Поэтому необходимо развивать навыки программирования и работы с компьютером.

1.2 Обзор исследований

В статье Н. А. Базеевой и Д. С. Лебедевой описано про игровую индустрию и рассмотрены особенности языков программирования для разработки игр [1]. В работе Р.Ф. Гайнуллина, В.А. Захарова, Е.А. Аксеновой изучен инструмент для разработки двухмерных и трёхмерных игр на Unity [2]. С.А. Суродин представил сценарий изучения одного из основных движков, для

создания красивых 2D и 3D игр [3]. В статье Э.С. Окороков, Н.Н. Сивцев, Г.Ю. Протодьякова описали процессы разработки игр [4]. И. Ю. Просвирнина в свой работе описала процесс разработки игры «Морской бой» [5]. В англоязычной статье Boyraz G., Kirci P. был рассмотрен процесс разработки игры на Unity [6]. Prasetyo M.A., Pamungkas Ch.G., Budi Luhur G. разработали дизайн для мобильной 2D игры [7].

1.3 Цель исследования

Целью данного исследования является разработать сетевую игру «Новый год».

2 Материалы и методы

В данной работе использовалась версия Unity 2018.1.1f1 – это мощный инструмент для разработки игр, который позволяет создавать игры для различных платформ, включая компьютеры, мобильные устройства и консоли.

C# – это объектно-ориентированный язык программирования, который широко используется для разработки игр в Unity.

Visual Studio – это интегрированная среда разработки (IDE), которая предоставляет широкий набор инструментов для разработки приложений на различных языках программирования.

3 Результаты

Для создания игры в программе Unity создадим новый проект и добавим на сцену игровое поле будущей игры файл «Board.png» (Рис. 1).

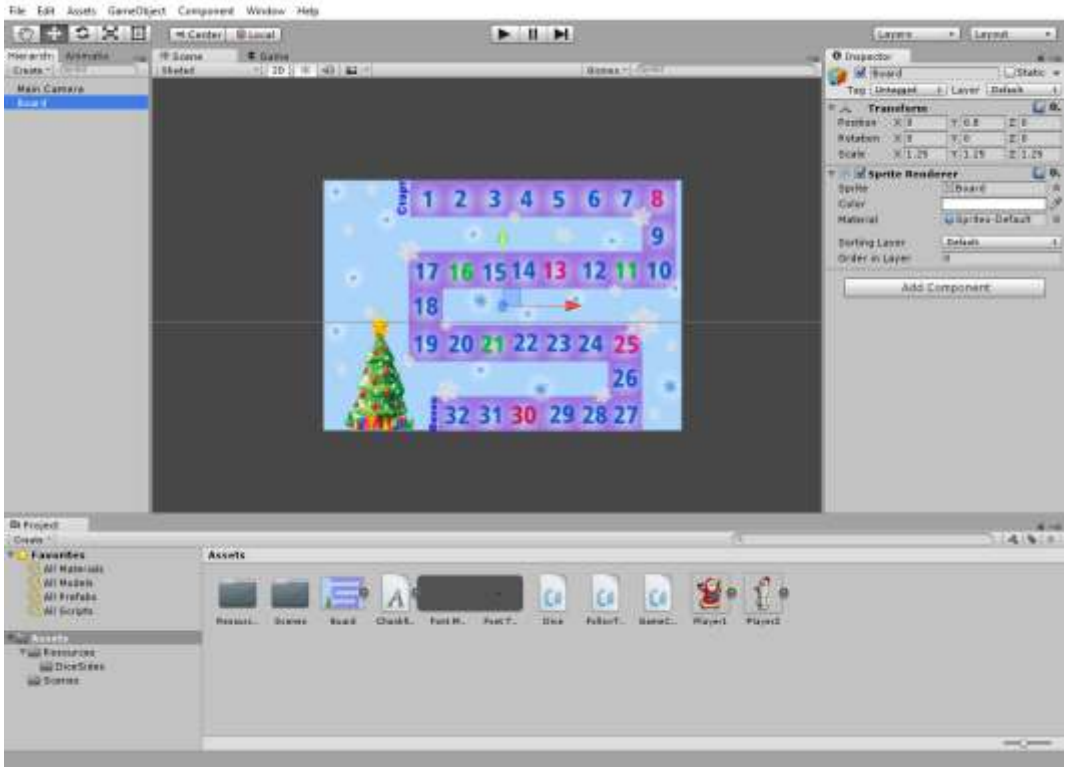

Рисунок 1 – Создание проекта в программе Unity

Добавляем пустой объект, переименовываем его в «Игровое поле», затем добавляем новый объект и выбираем значок фиолетовый ромб (Рис. 2).

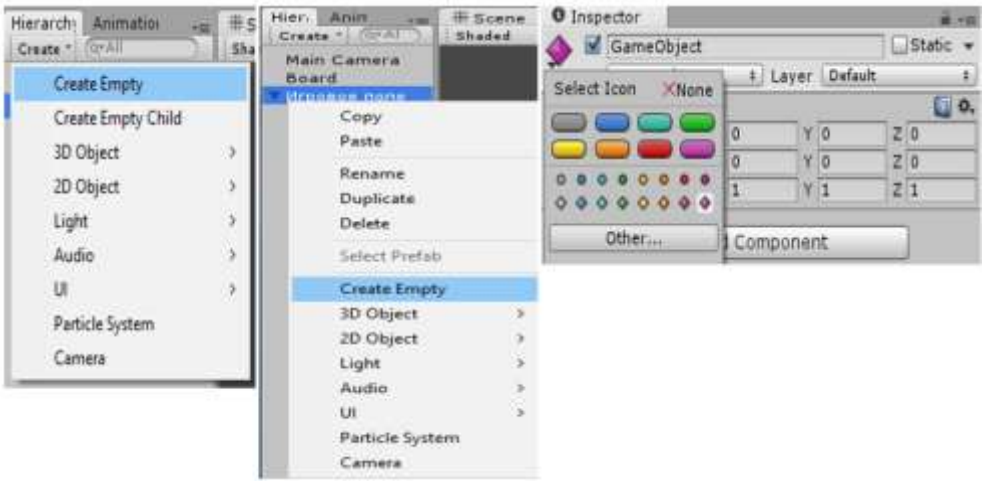

Рисунок 2 – Добавление объектов

Переименовываем объект «Плита 1», создаём его 32 копии с нумерацией и размещаем по порядку на игровом поле (Рис.3).

Рисунок 3 – Размещение объектов на игровое поле

Файл «FollowThePath.cs» с кодом для перемещения игроков по каждой плитке на игровом поле:

```
using UnityEngine;
public class FollowThePath :MonoBehaviour {
    public Transform[] waypoints;
    [SerializeField]
    private float moveSpeed = 1f;
    [HideInInspector]
    public int waypointIndex = 0;
    public bool moveAllowed = false;
      // Используйте это для инициализации
```

```
      private void Start () {
transform.position = waypoints[waypointIndex].transform.position;}
      // Обновление вызывается один раз за кадр
      private void Update () {
        if (moveAllowed)
Move();}
    private void Move()
{if (waypointIndex<= waypoints.Length - 1)
{transform.position = Vector2.MoveTowards(transform.position,
            waypoints[waypointIndex].transform.position,
moveSpeed * Time.deltaTime);
            if                    (transform.position                    ==
waypoints[waypointIndex].transform.position)
{ waypointIndex += 1;}}}}
```

Добавляем файлы «Player1.png» – игрок 1 (ДедМороз) и «Player2.png» - игрок 2 (Снеговик), размещаем на игровом поле. Игрокам добавляем скрипт «FollowThePath» в параметрах указываем скорость перемещения (Move Speed) – 5 и добавляем все 32 плитки в «Waypoints» (Рис. 4).

Рисунок 4 – Добавление игроков на игровом поле

Добавляем файл «Side6.png», размещаем на игровом поле и добавляем к нему компонент «BoxCollider 2D» (Рис. 5).

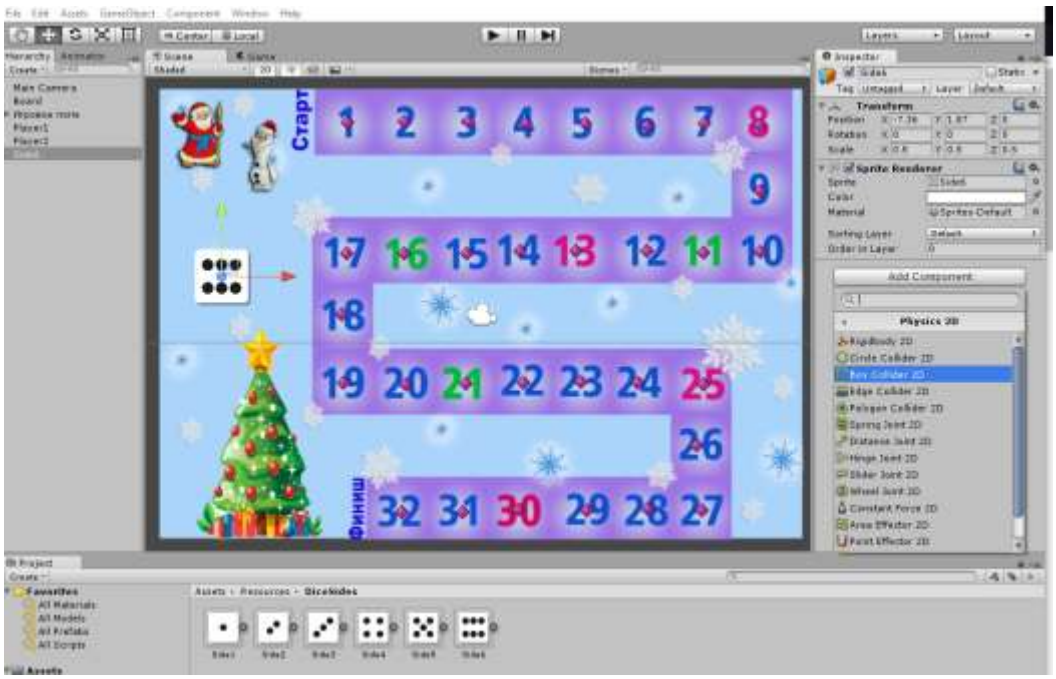

Рисунок 5 – Добавление объекта «Side6»

Файл «Dice.cs» с кодом, который при нажатии на игральные кости, будет выдавать число от 1 до 6 – количество на которое будет ходить игрок:

```
using System.Collections;
using UnityEngine;
public class Dice :MonoBehaviour {
    private Sprite[] diceSides;
    private SpriteRenderer rend;
    private int whosTurn = 1;
    private bool coroutineAllowed = true;
// Используйте это для инициализации
      private void Start () {
        rend = GetComponent<SpriteRenderer>();
diceSides = Resources.LoadAll<Sprite>("DiceSides/");
rend.sprite = diceSides[5]; }
    private void OnMouseDown()
    {if (!GameControl.gameOver&&coroutineAllowed)
StartCoroutine("RollTheDice");}
    private IEnumeratorRollTheDice(){
coroutineAllowed = false;
        int randomDiceSide = 0;
        for (int i = 0; i<= 20; i++){
randomDiceSide = Random.Range(0, 6);
rend.sprite = diceSides[randomDiceSide];
            yield return new WaitForSeconds(0.05f); }
GameControl.diceSideThrown = randomDiceSide + 1;
        if (whosTurn == 1) {
GameControl.MovePlayer(1);
        } else if (whosTurn == -1) {
GameControl.MovePlayer(2); }
whosTurn *= -1;
coroutineAllowed = true;}}
```

Объекта «Side6» переименовываем в «Кубик» и к нему добавляем скрипт «Dice» (Рис. 6).

Рисунок 6 – Объект «Кубик»

Файл «GameControl.cs» с кодом игры:

```
using UnityEngine;
using UnityEngine.UI;
public class GameControl :MonoBehaviour {
    private static GameObjectwhoWinsTextShadow, player1MoveText,
player2MoveText;
    private static GameObject player1, player2;
    public static int diceSideThrown = 0;
    public static int player1StartWaypoint = 0;
    public static int player2StartWaypoint = 0;
    public static bool gameOver = false;
    // Use this for initialization
    void Start () {
whoWinsTextShadow = GameObject.Find("WhoWinsText");
        player1MoveText = GameObject.Find("Player1MoveText");
        player2MoveText = GameObject.Find("Player2MoveText");
        player1 = GameObject.Find("Player1");
        player2 = GameObject.Find("Player2");
        player1.GetComponent<FollowThePath>().moveAllowed = false;
        player2.GetComponent<FollowThePath>().moveAllowed = false;
whoWinsTextShadow.gameObject.SetActive(false);
player1MoveText.gameObject.SetActive(true);
player2MoveText.gameObject.SetActive(false); }
    // Update is called once per frame
    void Update(){
        if (player1.GetComponent<FollowThePath>().waypointIndex>
            player1StartWaypoint + diceSideThrown)        {
            player1.GetComponent<FollowThePath>().moveAllowed = false;
player1MoveText.gameObject.SetActive(false);
player2MoveText.gameObject.SetActive(true);
            player1StartWaypoint =
player1.GetComponent<FollowThePath>().waypointIndex - 1;  }
        if (player2.GetComponent<FollowThePath>().waypointIndex>
player2StartWaypoint + diceSideThrown) {
            player2.GetComponent<FollowThePath>().moveAllowed = false;
player2MoveText.gameObject.SetActive(false);
player1MoveText.gameObject.SetActive(true);
            player2StartWaypoint =
player2.GetComponent<FollowThePath>().waypointIndex - 1;  }
        if (player1.GetComponent<FollowThePath>().waypointIndex ==
            player1.GetComponent<FollowThePath>().waypoints.Length) {
```

```
whoWinsTextShadow.gameObject.SetActive(true);
whoWinsTextShadow.GetComponent<Text>().text = "Player 1 Wins";
gameOver = true; }
        if (player2.GetComponent<FollowThePath>().waypointIndex ==
            player2.GetComponent<FollowThePath>().waypoints.Length){
whoWinsTextShadow.gameObject.SetActive(true);
player1MoveText.gameObject.SetActive(false);
player2MoveText.gameObject.SetActive(false);
whoWinsTextShadow.GetComponent<Text>().text = "Player 2 Wins";
gameOver = true; } }
    public static void MovePlayer(int playerToMove) {
        switch (playerToMove) {
            case 1:
                player1.GetComponent<FollowThePath>().moveAllowed = true;
                break;
            case 2:
                player2.GetComponent<FollowThePath>().moveAllowed = true;
break; } } }
```

Данный код представляет собой основной контроллер игры. Игра представляет собой гонку между двумя игроками на игровом поле, состоящем из пути с определенными индексами.

Игра начинается с того, что оба игрока находятся на начальной позиции пути. Когда игроки бросают кубик и получают результат (diceSideThrown), код обновляется каждый кадр (Update) и проверяет, достиг ли каждый игрок своей целевой позиции на пути. Когда один из игроков достигает последней позиции на игровом поле, игра заканчивается, появляется текст, указывающий победителя (whoWinsTextShadow), становится видимым, и игра больше не продолжается (gameOver = true).

Добавляем объект-UI «Canvas» и у скрипта «Canvas Scaler» выбираем параметр «Scaler With Screen Size» (Рис. 7).

Рисунок 7 – Объект-UI «Canvas»

Добавляем объект-UI «Image» и переименовываем в «Player1Icon» в скрипт «Image» добавляем картинку игрока 1и редактируем размеры, те же действия выполняем для игрока 2 (Рис. 8).

Рисунок 8 – Объекты-UI «Player1Icon» и «Player2Icon»

Добавляем объект-UI «Text» и переименовываем в «Player1MoveText» в скрипт «Text» добавляем текст к игроку 1 и редактируем шрифт, те же действия выполняем для игрока 2 (Рис. 9).

Рисунок 9 – Объекты-UI « Player1MoveText » и « Player2MoveText »

Добавляем объект-UI «Text» и переименовываем в «WhoWinsText» в скрипт «Text» добавляем текст «Кто победит в игре?» и редактируем шрифт. Этот объект-UI «WhoWinsText» появляется по завершению игры и называет победителя (Рис. 10).

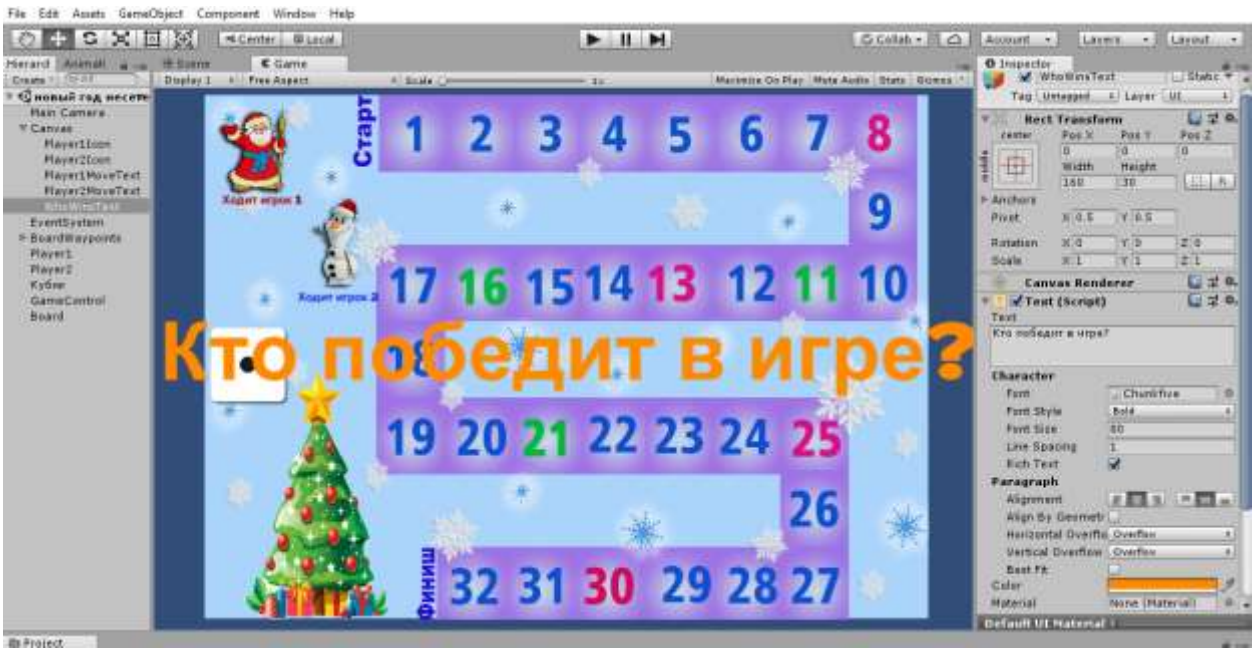

Рисунок 10 – Объект-UI «WhoWinsText»

Создана игра «Новый год» на основе настольных игр-ходилок (Рис. 11).

Рисунок 11 – Процесс игры «Новый год»

4 Выводы

В данной работе была разработана сетевая игра под названием "Новый год". В процессе разработки были применены современные технологии и инструменты для создания игры, включая язык программирования и среду разработки Unity.

Игра "Новый год" представляет собой гонку между двумя игроками на игровом поле, состоящем из пути с определенными индексами. Игроки бросают кубик и двигаются по пути в соответствии с результатом броска. Победителем становится игрок, первым достигший последней позиции на пути. Были реализованы функции для перемещения игроков, обновления состояния игры и определения победителя.

В дальнейшем можно рассмотреть возможности расширения функциональности игры, добавление новых уровней сложности, внедрение мультиплеерных режимов игры, а также улучшение графики и звукового сопровождения.

В целом, разработка сетевой игры "Новый год" позволила применить полученные знания и навыки в области информационно-коммуникационных систем и сетей, а также получить практический опыт в разработке игр. Данная статья может быть использована в качестве основы для дальнейших исследований и разработок в области информационно-коммуникационных систем и сетей, а также в сфере развлекательной индустрии.

Библиографический список

1. Базеева Н. А., Лебедев Д. С. Языки программирования для создания игр//E-Scio. 2019. №4. С. 31-39.
2. Гайнуллин Р.Ф., Захаров В.А., Аксенова Е.А. Создание 2d игры на Unity3D 5.4 // Вестник современных исследований. 2018. №4. С. 78-82.
3. Суродин С.А. Unity 3D. разработка сценария проектирования в среде Unity 3D// Информатика и вычислительная техника. 2015. №3. С. 504-511.
4. Окороков Э.С., Сивцев Н.Н., Протодьякова Г.Ю. Разработка инди игр // Автоматика. Вычислительная техника. 2018. №9. С. 6-10.
5. Просвирнина И. Ю., Егунова А. И., Аббакумов А. А. Среда разработки Microsoft Visual Studio на примере создания игры "морской бой" //Интеграционные процессы в науке в современных условиях. 2017. С. 123-125.
6. Boyraz G., Kirci P.//Constructing a 3D game with Unity 3D game engine//Conference of Open Innovations Association, FRUCT. 2021. № 28. С. 554-557.
7. Prasetyo M.A., Pamungkas Ch.G., Budi Luhur G.//Game design for an environmental-themed 2D adventure mobile game//International Journal of Research and Applied Technology. 2022. Т. 2. № 2. С. 49-57.