Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ОТЧЕТ

ПО ОБЩЕПРОФЕССИОНАЛЬНОЙ ДИСЦИПЛИНЕ: МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

Создание палитр и процедурной генерации

Листов: 14

|  |  |
| --- | --- |
| Выполнил студент  Группы П50-4-21  Игошев Р. В.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 года | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_А.Ю.Бушин  «\_\_\_\_» \_\_\_\_\_\_\_2024 года |

Москва 2024

Цель работы: Создать несколько платформ при помощи палитр и разработать скрипт для процедурной генерации.

Начать следует с рассмотрения ресурсов проекта:

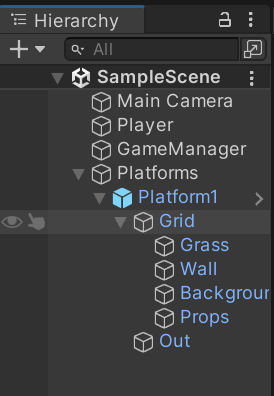


Рисунок 1 – Иерархия сцены

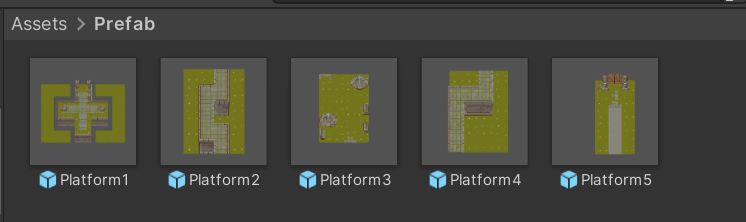


Рисунок 2 – Префабы платформ



Рисунок 3 – Расположение объектов на платформе

Для обозначения входа/выхода платформы используются определенные маркеры, обладающие определенным тегом для логики обработки.

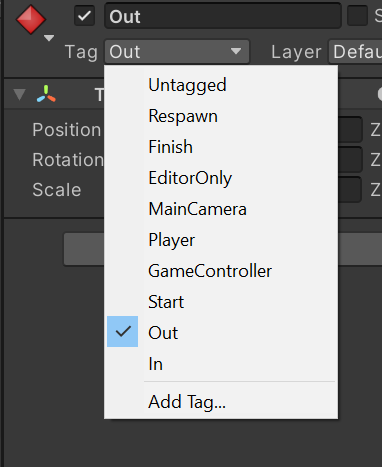


Рисунок 4 – Теги проекта

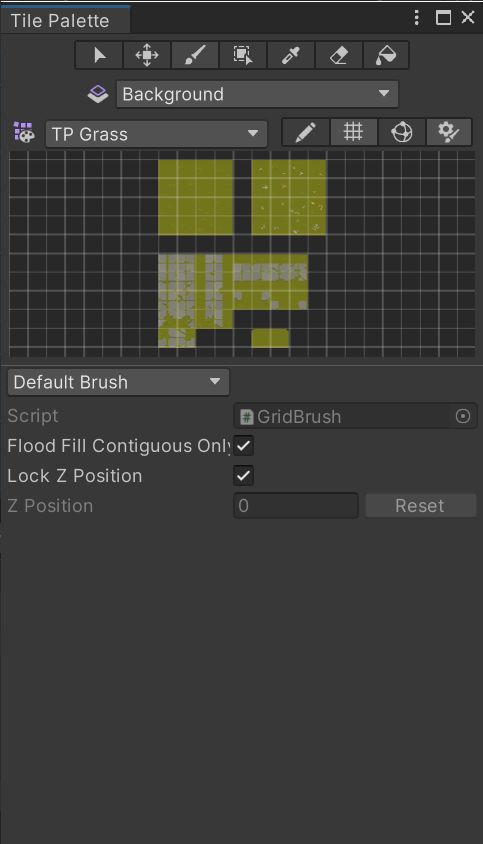


Рисунок 5 – Палетка клеток

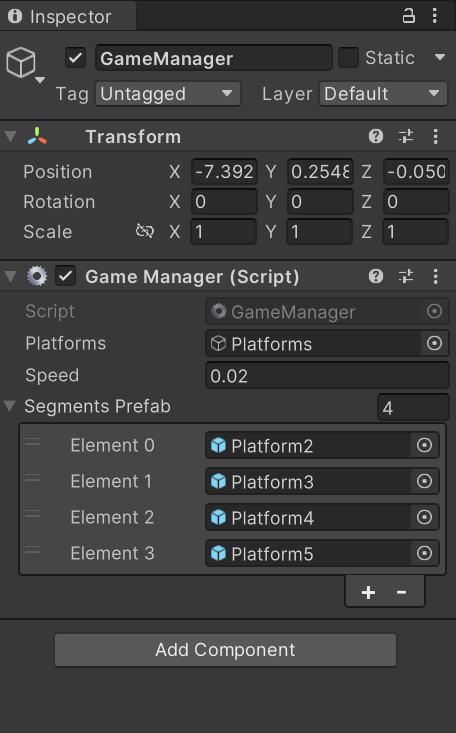


Рисунок 6 – Менеджер игрового процесса

Результаты работы:



Рисунок 7 – Отображение первой платформы



Рисунок 8 – Отображение второй платформы



Рисунок 9 – Отображение третьей платформы



Рисунок 10 – Отображение четвертой платформы



Рисунок 11 – Отображение пятой платформы

Код работы:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class GameManager : MonoBehaviour

{

public GameObject Platforms;

public float Speed = 0.05f;

public GameObject[] SegmentsPrefab;

private int Tick = 0;

private List<GameObject> Segments = new List<GameObject>();

private GameObject LastSegment;

private int currentSegmentIndex = 0;

public void Start()

{

LastSegment = GameObject.FindGameObjectWithTag("Start");

Segments.Add(LastSegment);

for (int i = 0; i < 5; i++) GenerateSegment();

}

private void FixedUpdate()

{

Platforms.transform.Translate(Vector2.down \* Speed);

if (Tick == 150)

{

GenerateSegment();

Tick = 0;

}

else Tick++;

if (Segments.Count > 50)

{

Destroy(Segments[0]);

Segments.RemoveAt(0);

}

}

private void GenerateSegment()

{

GameObject SelectedSegment = SegmentsPrefab[currentSegmentIndex];

currentSegmentIndex = (currentSegmentIndex + 1) % SegmentsPrefab.Length;

Transform OutTransform = LastSegment.transform.Find("Out");

if (OutTransform != null)

{

GameObject CurrentSegment = Instantiate(SelectedSegment);

Transform InTransform = CurrentSegment.transform.Find("In");

if (InTransform != null)

{

Vector3 newPosition = OutTransform.position - InTransform.localPosition;

CurrentSegment.transform.position = newPosition;

CurrentSegment.transform.parent = Platforms.transform;

Segments.Add(CurrentSegment);

LastSegment = CurrentSegment;

}

else

{

Destroy(CurrentSegment);

}

}

}

}

Вывод: