

Разработка игры на Unity

Андрюенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описывается процесс разработки компьютерной 3D игры в жанре гоночной игры. Для разработки игры использовалась межплатформенная среда разработки компьютерных игр Unity. Созданную игру можно скачать на GitHub.

Ключевые слова: Unity, разработка игр, ассет, коллайдер.

Game development on Unity

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of developing a 3D computer game in the genre of a racing game. To develop the game, the Unity cross-platform computer game development environment was used. The created game can be downloaded on GitHub.

Keywords: Unity, game development, asset, collider.

1 Введение

1.1 Актуальность

Актуальность темы исследования связана с интенсивным развитием средств разработки компьютерных игр и сильно возросшим спросом на компьютерные игры. Сегодня любой желающий может попробовать себя в роли разработчика игр в связи с доступностью инструментов для разработки игр. Также разработка игр стала доходным занятием в современном мире

1.2 Обзор исследований

В своей работе И.В. Бахтин рассмотрел взаимодействие языка программирования C# и среды разработки игр Unity. [1]. Е.М. Курумбаев, Т.М. Марданова рассмотрели пример разработки игрового приложения в среде разработки Unity [2]. В.А. Рогачев, О.А. Шабалина подробно описали движок Unity, рекомендации по его использованию и пример разработки компьютерной игры [3]. В своей работе А.Д. Рейн, Н.А. Шолохов рассмотрели процессы разработки компьютерной 2D игры в жанре «Shooter» с использованием межплатформенной среды разработки компьютерных игр

Unity [4]. А.В. Шубин, Б.А. Козар в своей научной статье описали возможность создания различных трёхмерных геометрических фигур и их модификации, задавая требуемые параметры, при помощи среды разработки компьютерных игр Unity3D [5]. Т.Ю. Семенцова в своей работе описала процесса создания прототипа развивающей детской компьютерной игры. Предмет исследования условно разбит на несколько частей, каждая из которых прорабатывается в отдельности [6].

1.3 Цель исследования

Цель исследования – создать компьютерную 3D игру в жанре гоночной игры с помощью межплатформенной среды разработки компьютерных игр Unity и загрузить её в общий доступ на GitHub.

2 Материалы и методы

Для разработки игры использовалась межплатформенная среда разработки компьютерных игр Unity и готовые ассеты.

3 Результаты и обсуждения

В данной работе будет использована карта, созданная в Unity, и ассеты из «Asset store». Так как карта уже создана необходимо импортировать лишь пакет машины. Для этого заходим на сайт «Unity Asset Store» и ищем бесплатный ассет машины. В данной работе был использован ассет «Simple Cars Pack» [7]. Добавляем выбранный ассет себе, нажав кнопку «Add to My Assets». После импортируем пакет в используем проект Unity, где находится карта (рис. 1).

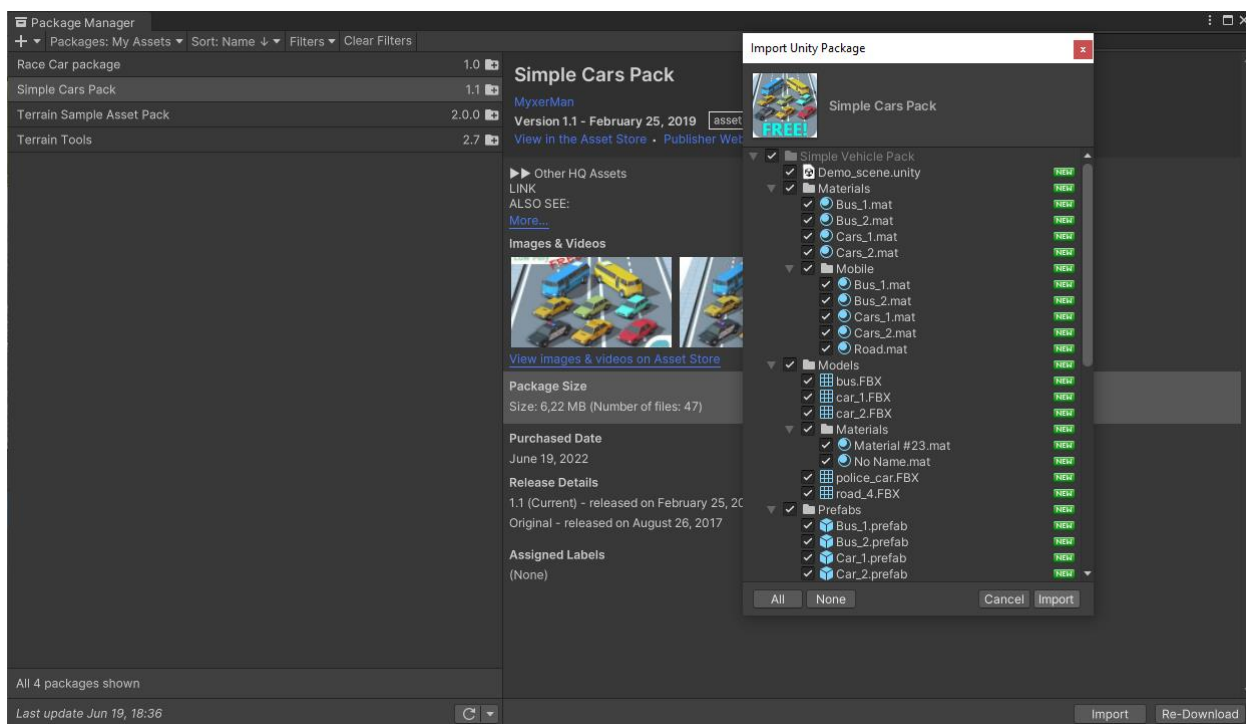


Рисунок 1 – Импорт ассета

Добавляем машину на сцену. Для этого перетаскиваем любую машину из папки «All Prefabs», находящийся в нижней части экрана, на сцену (рис. 2).



Рисунок 2 – Добавление машины на сцену

После добавления машины на сцену, в иерархии появилась модель машины, а в модели отобразились текстуры машины и её четырех колес. Для удобства рекомендуется переименовать колеса, так как дальнейшая работа будет только с ними. Теперь необходимо создать коллайдеры для колес. Для этого необходимо нажать на иерархию правой кнопкой мыши и выбрать «Create Empty». После в правом меню добавить созданному объекту компонент и выбрать «Wheels Collider» (рис. 3).



Рисунок 3 – Добавление коллайдера колесу

После создания коллайдера можно заметить небольшой зеленый круг на сцене. Его необходимо переместить ровно на колесо машины и уменьшить до размера колеса (рис. 4). С помощью коллайдеров пользователь будет управлять авто в игре.



Рисунок 4 – Конечное расположение коллайдеров колес

Теперь добавим компонент «Rigidbody» машине. В его свойствах задаем машине параметры массы и тяги 750 и 0,01 соответственно. Так же необходимо добавить коллайдер машине. Добавляем авто компонент «Box Collider» и подгоняем его по размеру машины (рис. 5).



Рисунок 5 – Добавление компонентов «Rigidbody» и «BoxCollider» авто

Далее прописываем скрипт, с помощью которого пользователь будет управлять машиной. Создаем C# Script в папке проекта в окне «Project», нажав правой кнопкой мыши и выбрав «Create». Называем скрипт carControl, и пишем код (рис. 6,7).

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class CarController : MonoBehaviour
6. {
7.
8.     [SerializeField] private Transform _transformFL;
9.     [SerializeField] private Transform _transformFR;
10.    [SerializeField] private Transform _transformBL;
11.    [SerializeField] private Transform _transformBR;
12.
13.    [SerializeField] private WheelCollider _colliderFL;
14.    [SerializeField] private WheelCollider _colliderFR;
15.    [SerializeField] private WheelCollider _colliderBL;
16.    [SerializeField] private WheelCollider _colliderBR;
17.
18.    [SerializeField] private float _force;
19.    [SerializeField] private float _maxAngle;
20.    // Start is called before the first frame update
21.
22.    private void FixedUpdate()
23.    {
24.        _colliderFL.motorTorque = Input.GetAxis("Vertical") * _force;
25.        _colliderFR.motorTorque = Input.GetAxis("Vertical") * _force;
26.
27.        if (Input.GetKey(KeyCode.Space))
28.        {
29.            _colliderFL.brakeTorque = 3000f;
30.            _colliderFR.brakeTorque = 3000f;
31.            _colliderBL.brakeTorque = 3000f;
32.            _colliderBR.brakeTorque = 3000f;
33.        }
```

Рисунок 6 – Код скрипта

```
34.     else
35.     {
36.         _colliderFL.brakeTorque = 0f;
37.         _colliderFR.brakeTorque = 0f;
38.         _colliderBL.brakeTorque = 0f;
39.         _colliderBR.brakeTorque = 0f;
40.     }
41.
42.     _colliderFL.steerAngle = _maxAngle * Input.GetAxis("Horizontal");
43.     _colliderFR.steerAngle = _maxAngle * Input.GetAxis("Horizontal");
44.
45.     RotateWheel(_colliderFL, _transformFL);
46.     RotateWheel(_colliderFR, _transformFR);
47.     RotateWheel(_colliderBL, _transformBL);
48.     RotateWheel(_colliderBR, _transformBR);
49. }
50.
51. private void RotateWheel(WheelCollider collider, Transform transform)
52. {
53.     Vector3 position;
54.     Quaternion rotation;
55.
56.     collider.GetWorldPose(out position, out rotation);
57.
58.     transform.rotation = rotation;
59.     transform.position = position;
60. }
61. }
62.
```

Рисунок 7 – Код скрипта

Данный скрипт необходимо привязать к машине. Перетаскиваем скрипт на машину. После этого действия в свойствах машины появится окно свойств скрипта. Его необходимо заполнить, привязав колеса и коллайдеры к скрипту (рис. 8). В окна «Transform» добавляем текстуры колеса, которые были вместе с моделью автомобиля. В окна «Collider» добавляем коллайдеры колес, созданные ранее. Так же указываем силу и угол в свойствах скрипта.

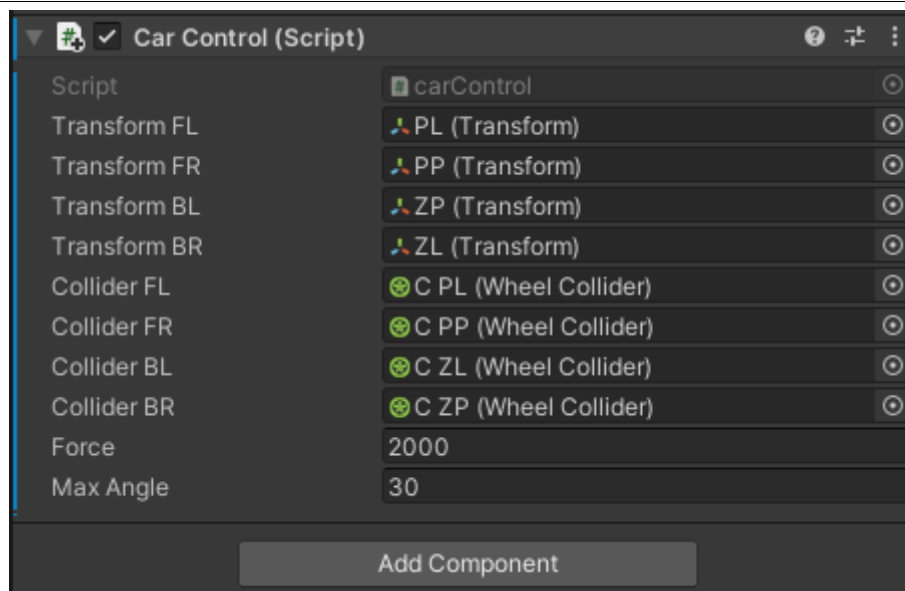


Рисунок 8 – Свойства скрипта

Осталось добавить камеру вида от третьего лица машины. Добавляем на сцену камеру и располагаем ее сзади машины, после чего привязываем, перетащив её в иерархию авто (рис. 9).



Рисунок 9 – Вид камеры при запуске игры

Выводы

В данной работе была создана компьютерная 3D игра в жанре гоночной игры с помощью межплатформенной среды разработки компьютерных игр Unity. Данную игру можно протестировать, скачав с сайта GitHub по ссылке <https://github.com/IvanAndrienko/RaceUnity3D>.

Библиографический список

1. Бахтин И.В. Создание игрового приложения с использованием среды разработки игр Unity. // Форум молодых ученых. 2019. № 2 (30). С. 250-267.
2. Курумбаев Е.М., Марданова Т.М. Созание игрового приложения в среде разработки Unity. // Наука и современность. Материалы Региональной научно-практической конференции. 2019. С. 67-68.
3. Рогачев В.А., Шабалина О.А. Разработка компьютерных игр в среде Unity: основные приемы работы и примеры их применения. // Учебное пособие / Волгоград, 2016.
4. Рейн А.Д., Шолохов Н.А. Разработка 2d игры с использованием межплатформенной среды разработки компьютерных игр Unity. // Актуальные направления развития техники и технологий в России и за рубежом – реалии, возможности, перспективы. Материалы и доклады. 2021. С. 240-241.
5. Шубин А.В., Козар Б.А. Процедурная генерация геометрических фигур при помощи среды разработки видеоигр Unity3D. // Modern Science. 2022. № 1-1. С. 430-434.
6. Семенцова Т.Ю. Разработка детских игр на движке Unity. // XLVII итоговая студенческая научная конференция УдГУ. Материалы всероссийской конференции. Ответственный редактор А.М. Макаров. 2019. С. 25-26.
7. URL: <https://assetstore.unity.com/packages/3d/vehicles/land/simple-cars-pack-97669#description> (дата обращения 08.06.2022)