

ТРЕХМЕРНОЕ МОДЕЛИРОВАНИЕ В UNITY 3D. ТЕХНОЛОГИЯ СОЗДАНИЯ И ФИЗИКА МОДЕЛЕЙ

Бувич Н.В., Романова М.П.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Столер В.А. – канд.техн.наук, доцент, заведующий кафедрой ИКТ

Аннотация. Рассмотрен игровой движок Unity 3D, как инструмент для работы с трехмерными моделями, которые в дальнейшем можно использовать для создания и разработки компьютерных игр. Описаны этапы создания изображения с 3D-графикой, включая технологии создания, физику моделей, вплоть до использования определённых инструментов.

Ключевые слова: трехмерная модель, 3D-графика, Unity 3D, компьютерные игры, инструменты, движок, физика

Введение. В настоящее время игровая индустрия стала очень популярна и стремительно развивается, а ее экономический масштаб не меньше, чем в любой другой сфере Интернета. Игровая индустрия продолжает увеличивать размеры рынка и общие продажи с очень высокой скоростью. Игровые работы охватывают все большее количество видов, а отличным игровым произведениям нет конца. Основой для создания компьютерных игр является разработка игрового движка, используемого этими же играми. Движок Unity считается одним из самых популярных движков для разработки игр среди других программных устройств, предназначенных для создания игр. Среди тысяч лучших бесплатных мобильных игр в рейтинговой системе, 38% игр сделаны с использованием движка Unity [1]. Также Unity позволяет создавать не только игры, но и другие интерактивные развлечения или мультимедиа.

Целью работы является показать, как конструируется 3D-графика при помощи игрового движка Unity 3D. В данной статье авторами показаны этапы создания 3D-моделей, а также его технологии.

Основная часть. В этой статье мы разберём как используется движок Unity на примере солнечной системы, подключим физическое взаимодействие и анимацию к 3D-моделям, в нашем случае к планетам. Чтобы быстрее реализовать задумку, выберем основные задачи нашего проекта:

- создание компонентов;
- создание макета системы;
- настройка фонового изображения и освещения;
- разработка анимации;
- создание механики тел;
- тестирование программного продукта.

Unity предоставляет возможности создавать проекты любой сложности для различных платформ, как и для Windows, Linux, MacOS, IOS и Android-систем.

Вначале создаём пустой 3D-проект, который использует встроенный рендер Unity. На первом этапе мы создали компоненты для солнечной системы: Солнце и планеты. Для этого добавляем в проект сферы различных размеров, придаём объектам нужную текстуру. Unity поддерживает большинство известных форматов текстур, например, как PNG, JPG, PSD, TGA. Мы выбрали нужные и наложили их на объекты системы проекта.

Следующим шагом формируем макет солнечной системы: располагаем планеты по порядку от Солнца до Нептуна вдоль горизонтальной линии, учитывая при этом радиусы их орбит, что пропорциональны реальным значениям.

В проекте два источника света. Одним из которых выступает Солнце, представленное точечным источником света с теплотой свечения в 5700 К. Другой – свет далеких звезд космического пространства, глобальный источник света, что равномерно падает со всех сторон. Следующим этапом заменим фон, уставленный по умолчанию на изображение космического пространства. В меню настройки глобального света меняем интенсивность его свечения. Результат представлен на скриншоте (рисунок 1).

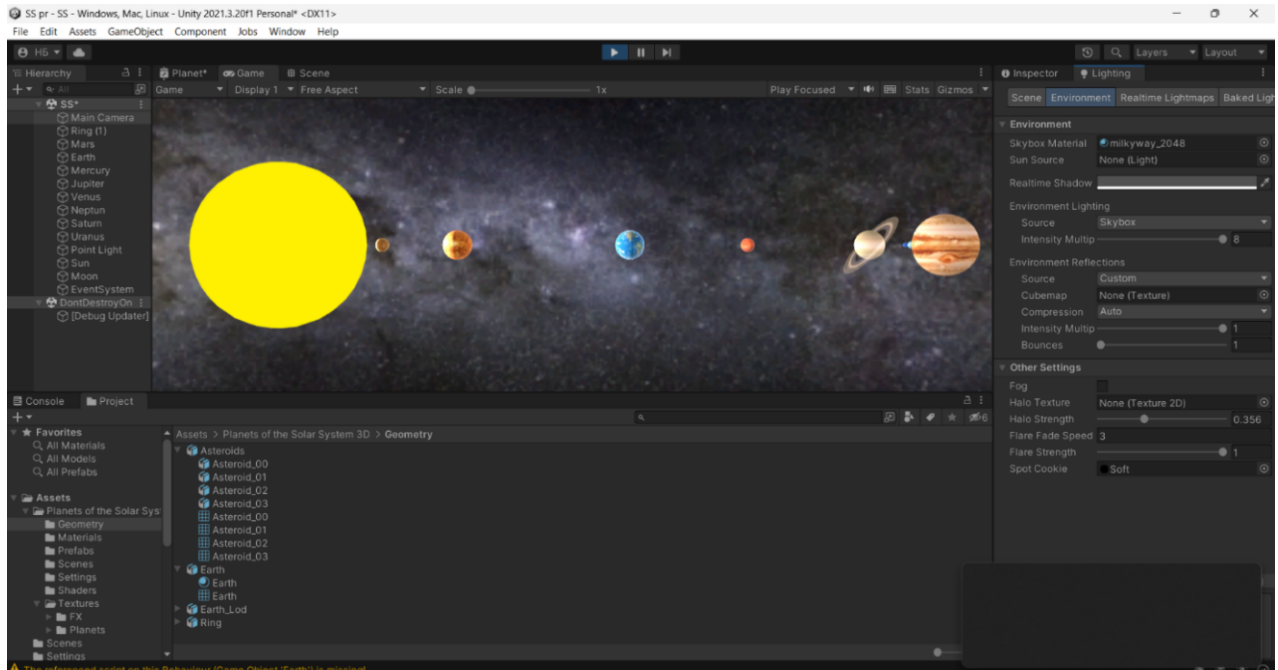


Рисунок 1 – Макет солнечной системы в окне Unity 3D

Следующим этапом будет добавление анимации вращения, которая заставит крутиться планеты вокруг своей оси. Для данной анимации создаем файл со скриптом, прописывая определенную функцию, и выбираем вокруг чего будет вращаться данный объект. То есть, к примеру, если мы хотим задать вращение для Земли, то нужно выбрать Землю как объект, вокруг которого она будет вращаться. Такую анимацию делаем для каждой планеты.

Теперь настроим физическое взаимодействие для наших объектов. Для данного пространства не свойственно ускорение свободного падения такое, каким оно стоит в проекте UNITY по умолчанию. Солнечная Система свободна от влияния внешних сил, то есть является замкнутой. Чтобы избавиться от земного ускорения свободного падения, в настройках физики проекта изменяем значение ускорения по вертикали с $-9,8$ на 0 . Теперь мы имеем независимую от внешних сил систему. Следующими действиями нужно настроить силы притяжения планет к солнцу. Первым шагом придадим нашим планетам массу при помощи модификатора Rigidbody, отвечающего за симуляцию твердого тела. После этого в настройках Collider [2] настроим расстояние, на котором будут взаимодействовать наши объекты. Создаем параметр, регулирующий силы притяжения, используя базовые навыки программирования на C#. Код представлен на скриншоте (рисунок 2).

Применим этот параметр для каждого объекта нашей системы. Если запустить проект, солнце притянет к себе другие составляющие объекты. Придадим нашим планетам скорость, чтобы удерживать их на своих орбитах. Теперь наша система работает исправно по законам физики, как и в реальности.

```

1  using System.Collections.Generic;
2  using UnityEngine;
3
4
5  public class Fiziks : MonoBehaviour
6  {
7      private HashSet<Rigidbody> affectedBodies = new HashSet<Rigidbody>();
8
9      private void Start()
10     {
11     }
12
13
14     private void OnTriggerEnter(Collider other)
15     {
16         if (other.attachedRigidbody != null)
17         {
18             affectedBodies.Add(other.attachedRigidbody);
19         }
20     }
21
22
23     private void OnTriggerEnter(Collider other)
24     {
25         if (other.attachedRigidbody != null)
26         {
27             affectedBodies.Remove(other.attachedRigidbody);
28         }
29     }
30
31     private void Update()
32     {
33         foreach (Rigidbody body in affectedBodies)
34         {
35             Vector3 directionToPlanet = (transform.position - body.position).normalized;
36             body.AddForce(directionToPlanet*1000);
37         }
38     }
39
40
41
42
43

```

Рисунок 2 – Пример кода для вращения тел вокруг другого тела

Заключение. В ходе работы мы рассмотрели основные инструменты Unity 3D для создания трехмерных моделей, их анимации и настройки физического взаимодействия, имея при этом базовые навыки программирования. Были сделаны модели планет, макет, собранный из них, поставлен космос в качестве фонового изображения, настроены освещение, вращение элементов системы вокруг своей оси и вращение вокруг одного элемента «Солнце» в одной игровой сцене. В итоге солнечная система была реализована с помощью популярной среды разработки игр Unity 3D, что позволяет убедиться в её выдающихся возможностях.

Список литературы

1. История развития движка Unity [Электронный ресурс]. – Режим доступа : <https://russianblogs.com/article/8243330706/>– Дата доступа : 21.03.2023
2. Работа с физикой Unity3D [Электронный ресурс] / AMS. – Режим доступа : <https://coderlessons.com/articles/mobilnaia-razrabotka-articles/rabota-s-fizikoi-unity3d> – Дата доступа : 21.03.2023.

UDC 004.925 – 047.58

THREE-DIMENSIONAL MODELING IN UNITY 3D. TECHNOLOGY OF CREATION AND PHYSICS OF MODELS

Buyevich N.V., Romanova M.P.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Stoler V.A. – PhD, associate professor, Head of the Department of ECG

Annotation. The Unity 3D-game engine is considered as a tool for working with three-dimensional models, which can later be used to create and develop computer games. The stages of creating an image with 3D-graphics are described, including creation technologies, model physics, up to the use of certain tools.

Keywords: 3D-model, 3D-graphics, Unity 3D, computer games, tools, engine, physics