УДК 004

Реализация простой системы строительства в Unity 3D

Ульянов Егор Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В настоящей статье рассматривается и описывается реализация базовой механики игр в жанре стратегия, механика строительства. Данная механика будет реализовываться на языке программирования C# с помощью игрового движка Unity 3D. Практическим результатом является реализованная и протестированная возможность строительства.
Ключевые слова: Unity 3D, механика строительства, программирование, скрипт

Implementation of a simple construction system in Unity 3D

Ulianov Egor Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses and describes the implementation of the basic mechanics of games in the genre of strategy, mechanics of construction. This mechanic will be implemented in the C# programming language using the Unity 3D game engine. The practical result is the realized and tested possibility of construction.
Keywords: Unity3D, construction mechanics, programming, script

Красивая визуальная составляющая, атмосферная музыка, хорошая оптимизация — большой плюс. Но без интересных механик игрок заскучает и выйдет несмотря на все плюсы. Слово «механика» часто употребляется при обучении геймдизайну, но не всегда объясняется, что это такое. Чтобы в этом разобраться, нужно понять, как работает игра. Игровой процесс основан на взаимодействии игрока с элементами игры: игрок прыгает на платформу, открывает инвентарь, ведёт машину и занимается другими делами. Каждое взаимодействие влияет на игровой мир и описывается правилами. Набор правил игры, который можно логически обособить, торжественно называется механикой. Чем игра больше взаимодействует с игроком, тем больше пользователь будет возвращаться в игру.

Цель данной статьи рассмотреть возможности игрового движка Unity 3D в реализация базовых механик игр жанра стратегия, а именно механика строительства.

С. А. Суродин в своей статье представил сценарий углубленного изучения одного из лучших движков, существующих на данный момент, для создания красивых 2D и 3D игр[1]. В своей работе Р. Ф. Гайнуллин, В. А. Захаров, Е. А. Аксенова изучили инструмент для разработки двух- и трёхмерных игр – Unity 3D[2]. А. В. Гришиным, Е. В. Бычковой проведен анализ средства взаимодействия пользователя с функциональной и эстетической частью во внутренней среде для разработки игр. Описаны преимущества использования C# и IDE unity[3].

Для начала создадим проект 3D и назовём его произвольным именем см. рисунок 1.

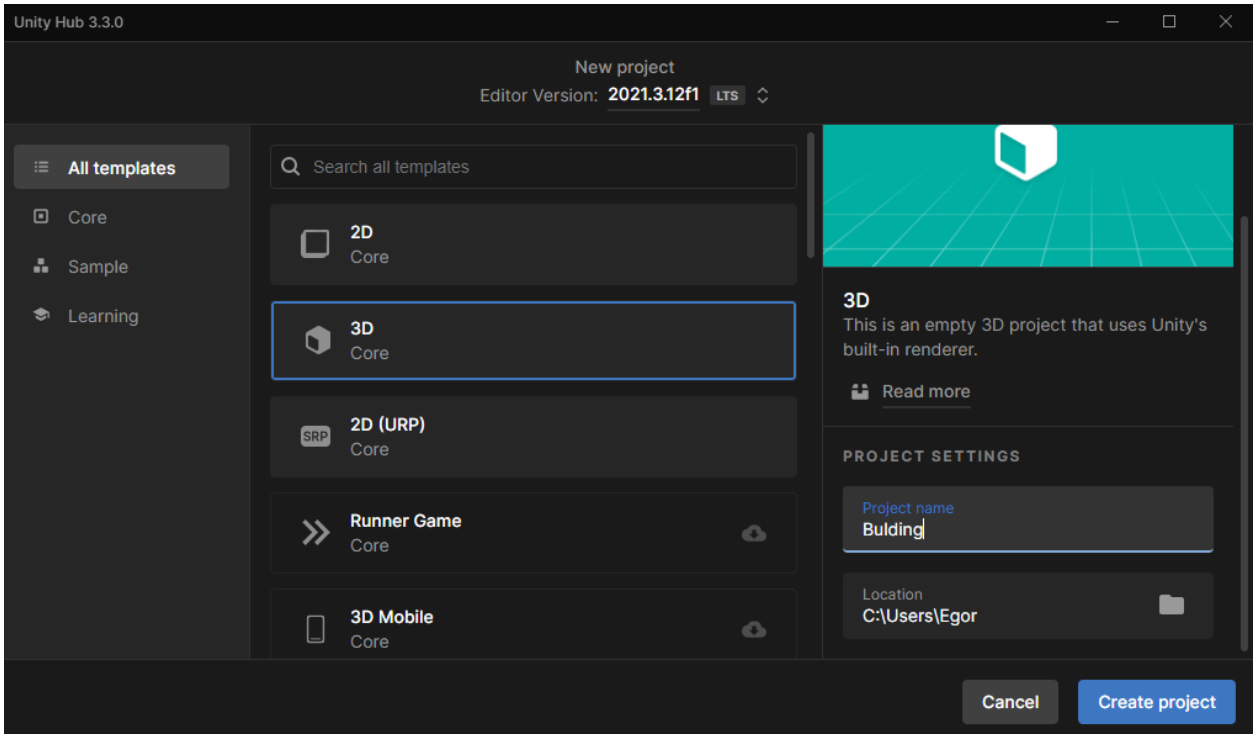

Рис. 1. Создание проекта

Далее создаем плоскость и пустой игровой объект, который понадобиться в дальнейшем. Перемещаем камеру так, чтобы находилась в удобном положении для построения. см. рисунок 2.

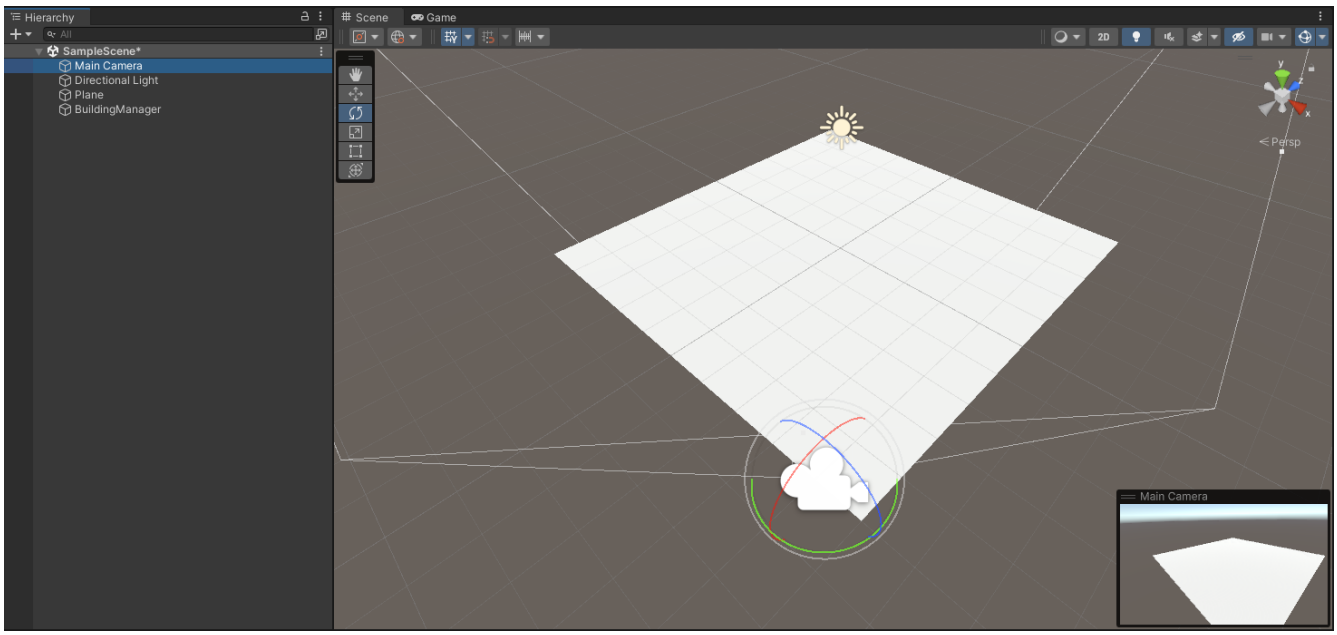

Рис. 2. Создание плоскости, пустого объекта и настройка камеры

Теперь добавим объекты, которые будем строить: куб, цилиндр, шар. При помощи созданного материала с произвольным цветом, окрашиваем объекты см. рисунок 3.

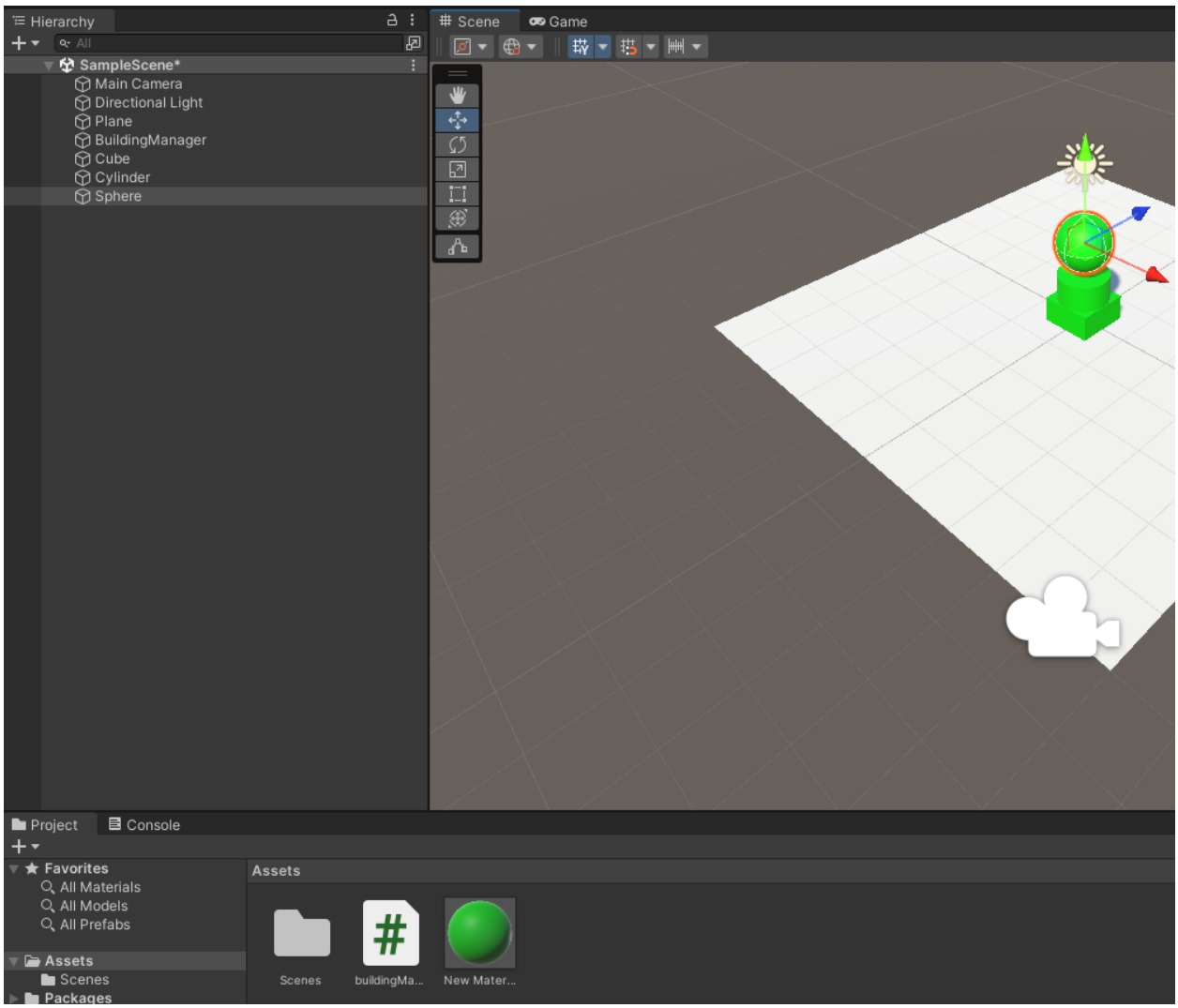

Рис. 3. Добавление и индивидуализация объектов

При помощи «Drag’n’Drop» переносим созданные объекты в папку «Assets», для создания префабов и удаляем со сцены см. рисунок 4.

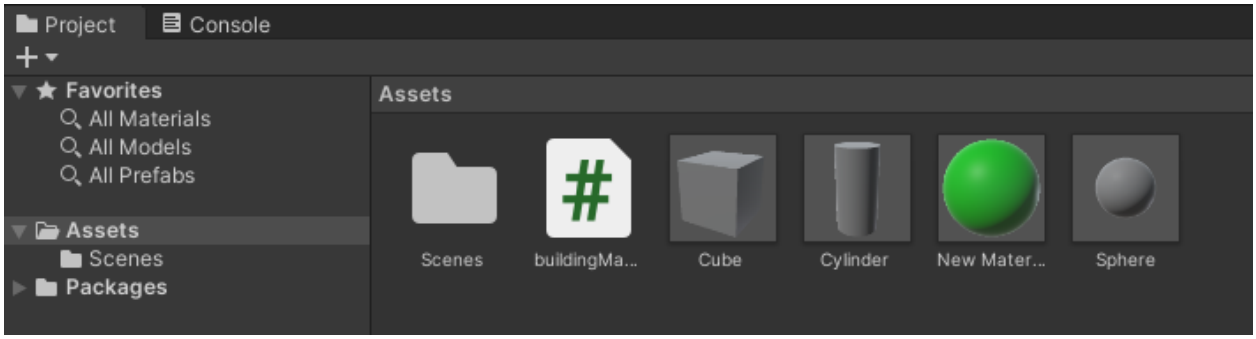

Рис. 4. Создания префабов объектов

Переходим к написанию кода, создаем скрипт «buildingManager» в котором создаем необходимые переменные см. рисунок 5-6.

```
buildingManager.cs
Assembly-CSharp                                   buildingManage
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
    Скрипт Unity | Ссылок: 0
5   public class buildingManager : MonoBehaviour
6   {
7       public GameObject[] objects;
8       private GameObject pendingObj;
9
10      private Vector3 pos;
11
12      private RaycastHit hit;
13      [SerializeField] private LayerMask layerMask;
14
15
        Сообщение Unity | Ссылок: 0
16      void Update()
17      {
18          if (pendingObj != null)
19          {
20              pendingObj.transform.position = pos;
21              if (Input.GetMouseButton(0))
22              {
23                  PlaceObject();
24              }
25          }
26      }
27
        Ссылок: 1
28      void PlaceObject()
29      {
30          pendingObj = null;
31      }
32
        Сообщение Unity | Ссылок: 0
33      private void FixedUpdate()
34      {
35          Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
36
37          if (Physics.Raycast(ray, out hit, 1000, layerMask))
38          {
39              pos = hit.point;
40          }
41      }
```

Рис. 5. Основная логика программы

```
    Ссылок: 0
44  public void SelectObject(int index)
45  {
46      pendingObj = Instantiate(objects[index], pos, transform.rotation);
47  }
48
49  }
50
```

Рис. 6. Продолжение кода

Далее также посредством «Drag’n’Drop» переносим скрипт на созданный ранее пустой объект, выбираем слой «Ground» в пункте «Layer», и добавляем строительные объекты см. рисунок 7.

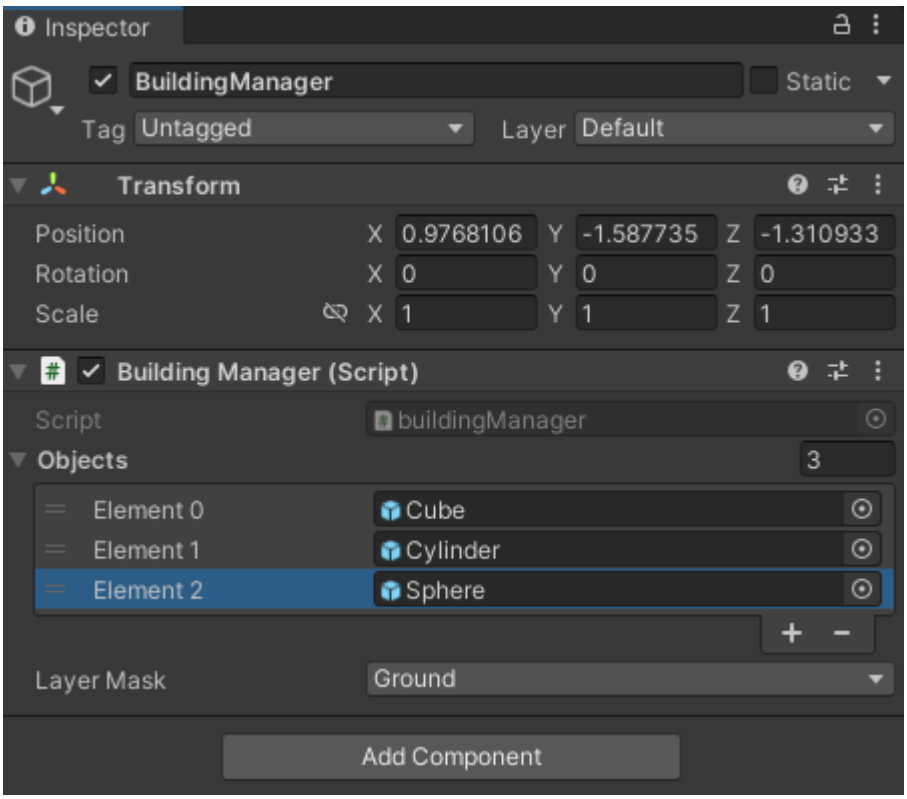

Рис. 7. Создание макроса

Для возможности выбора пользователем объекта строительства добавим кнопки, для этого кликаем правой кнопкой мыши по области «Hierarchy», выберем «UI» и «Button – TextMeshPro» см. рисунок 8-10.

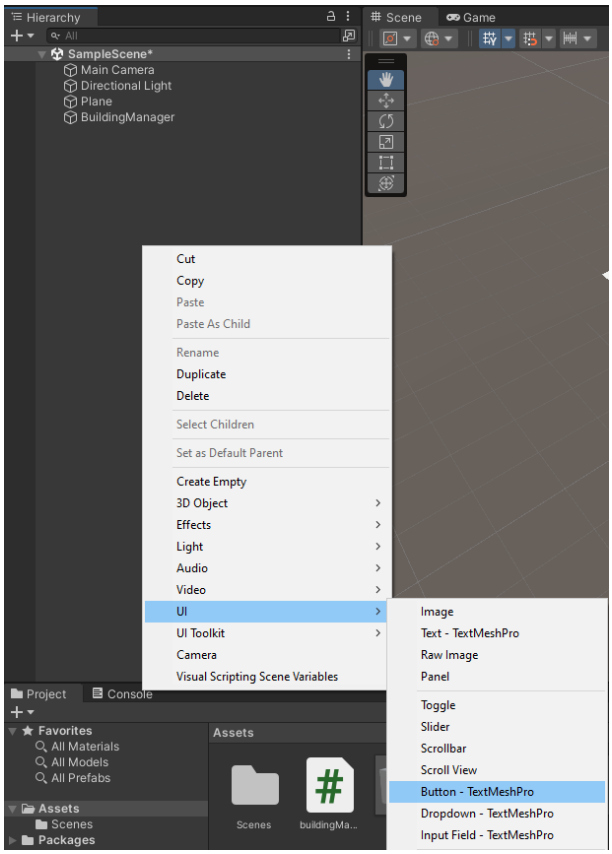

Рис. 8. Добавление кнопок

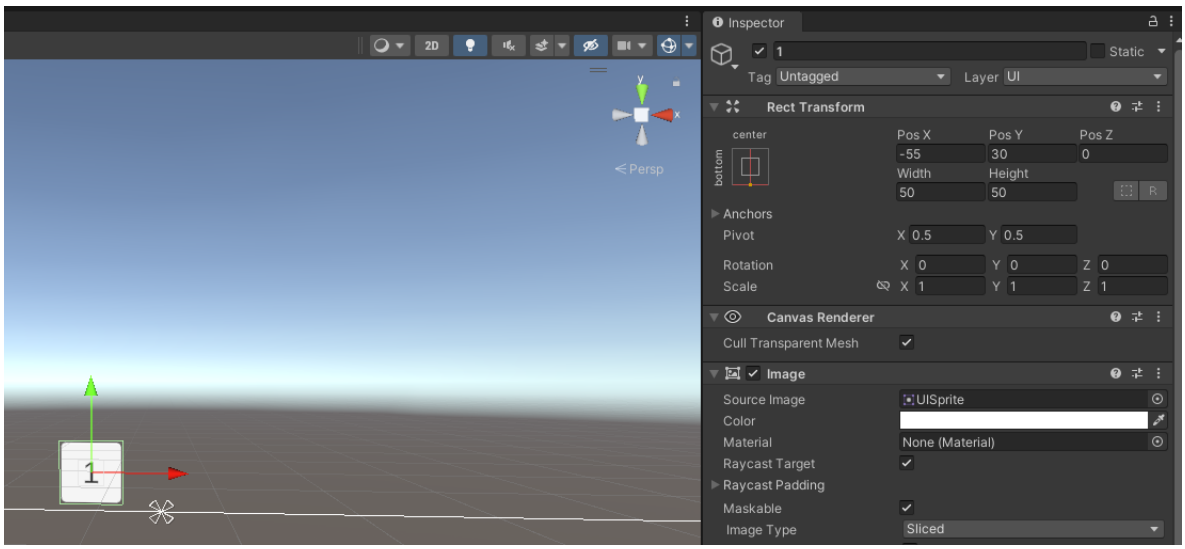

Рис. 9. Настройки кнопки

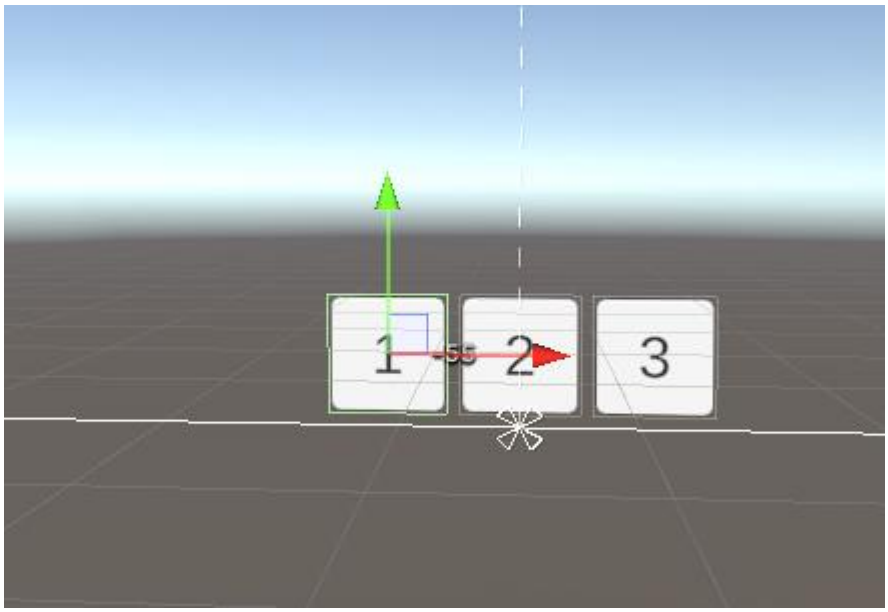

Рис. 10. Дублирование кнопок

Затем в каждой кнопке создаем новое событие «OnClick()», перетаскиваем скрипт «buildingManager», затем в выпадающем списке выбираем «Buildingmanager / SelectObject()». Вводим значения 0-3 для каждой кнопки по возрастанию см. рисунок 11.

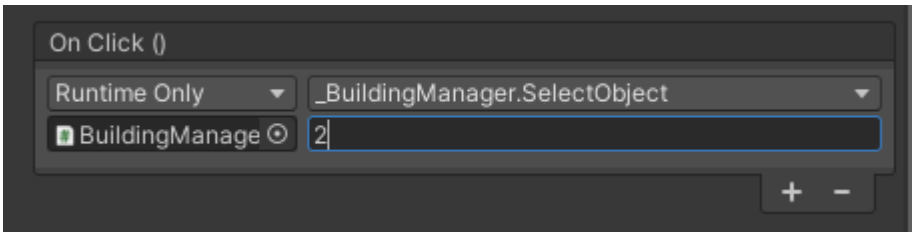

Рис. 11. Добавление события «OnClick ()»

Для проверки работы механики запускаем проект, и выбираем объект для постройки и перемещаем по плоскости мышкой. Кликом левой мыши расставляем объекты в любом месте см. рисунок 12.

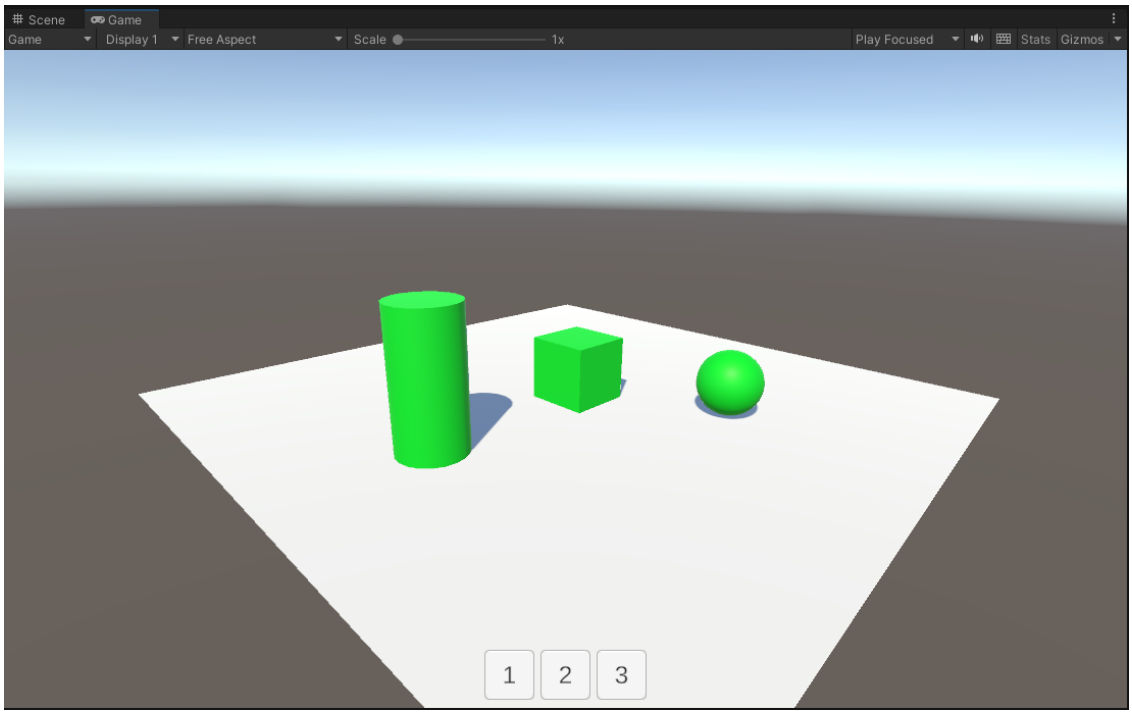

Рис. 12. Демонстрация рабочей механики

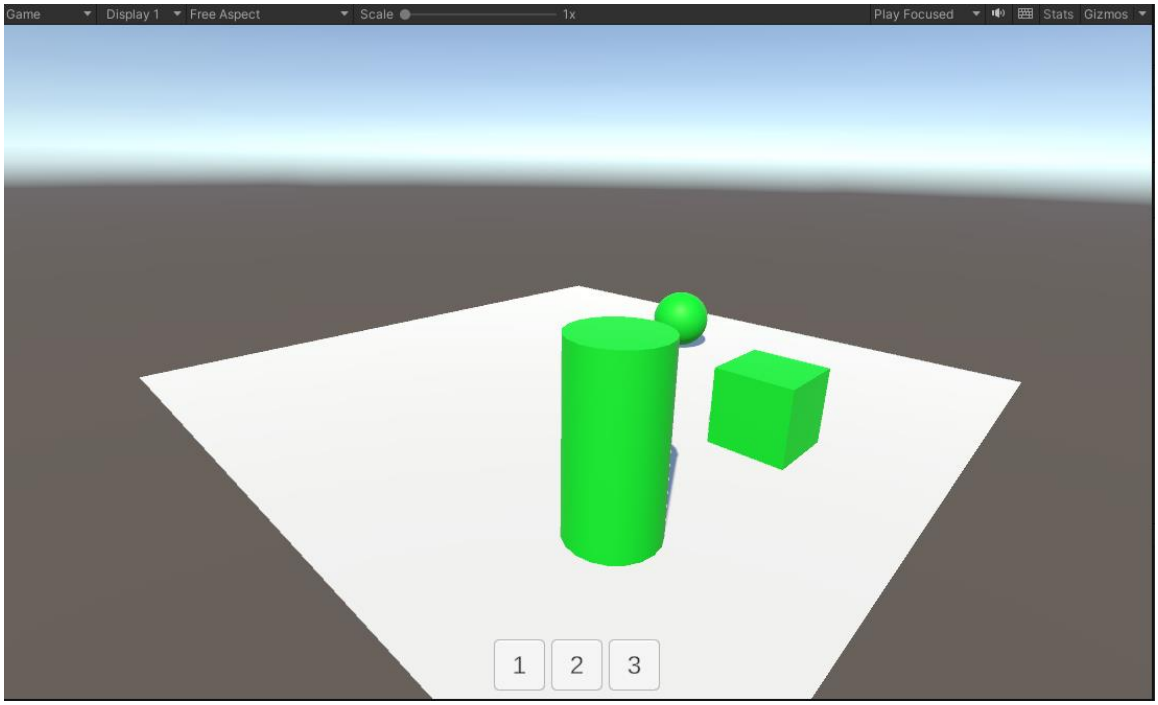

Рис. 13. Демонстрация рабочей механики

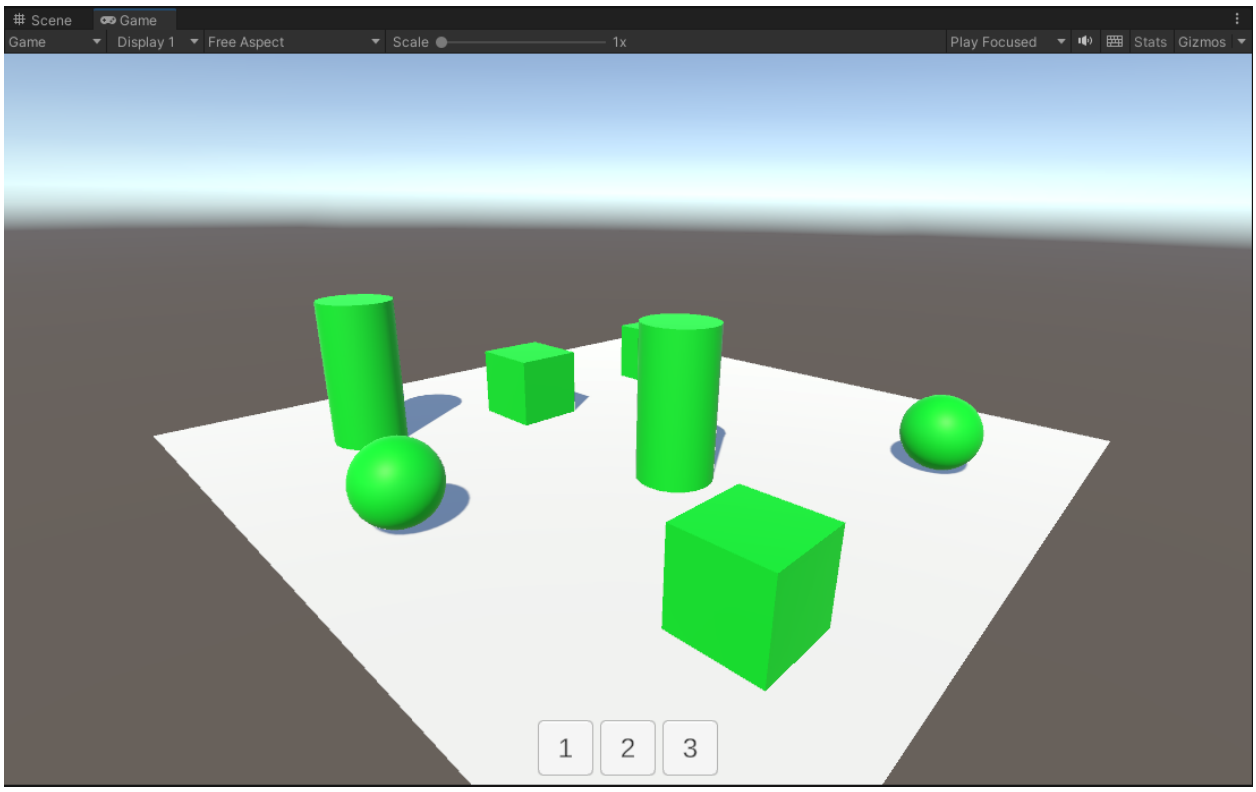

Рис. 14. Демонстрация рабочей механики

Таким образом была реализована простая механика строительства, которая в дальнейшем может быть доработана, а именно: добавление новых механик, графики и анимации. Были проанализированы существующие аналоги и методы разработки, а также выбрана среда разработки. Для реализации поставленной задачи отлично подошла разработка с помощью игрового движка «Unity 3D», среды разработки «Visual Studio», а также языка программирования C#. Такой выбор заметно упростил реализацию механики, так как в интернете имеется достаточное количество документации. Во время реализации механики строительства был полученный ценный опыт работы с этим средством разработки.

Библиографический список

1. Суродин С. А. Unity 3D. разработка сценария проектирования в среде Unity 3D //Информатика и вычислительная техника. 2015. №3. С. 504-511.
2. Гайнуллин Р. Ф., Захаров В. А., Аксенова Е. А. Создание 2d игры на Unity 3D 5.4 //Вестник современных исследований. 2018. №4. С. 78-82.
3. Гришин А. В., Бычкова Е. В. Анализ UNITY-разработки с использованием C# и IDE UNITY //Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2022). 2022. С. 424-426.