УДК 004

Реализация механики перетаскивания и поворота игровых объектов прикосновением с помощью программирования на C#

Ульянов Егор Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается и описывается процесс реализации механики перетаскивания и поворота игровых объектов прикосновением с помощью программирования в Unity 3D. Работа механики осуществляется посредством стандартных средств среды разработки. Практическим результатом является созданная и протестированная механика.
Ключевые слова: Unity 3D, механика, перетаскивание, прикосновение

Implementation of the mechanics of dragging and rotating game objects by touch using C# programming

Ulianov Egor Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses and describes the process of implementing the mechanics of dragging and turning game objects by touch using Unity 3D programming. The mechanics work is carried out through standard development environment tools. The practical result is created and tested mechanics.
Keywords: Unity 3D, mechanics, drag and drop, touch

Реализация механики перетаскивания и поворота игровых объектов прикосновением с помощью программирования на C# - это актуальная тема для исследования в области разработки игр и мобильных приложений.

Сейчас на мобильных устройствах игры, которые можно управлять на прикосновениях, являются одними из самых популярных и востребованных. Механика перетаскивания и поворота игровых объектов является критически важным элементом геймплея таких игр.

Разработчики мобильных приложений имеют потребность в использовании удобных и простых способов управления, и реализация настраиваемой механики перетаскивания и поворота игровых объектов позволяет сделать приложение более доступным для пользователя.

Цель данной статьи рассмотреть возможности игрового движка Unity 3D в реализации различных игровых механик на сцене.

И. А. Савин, О. В. Батенькина рассмотрели процесс написания скриптовых сценариев при разработке виртуального тренажера [1]. 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 [2]. В своей работе Р. Ф. Гайнуллин, В. А. Захаров, Е. А. Аксенова изучили инструмент для разработки двух- и трёхмерных игр – Unity 3D[3]. К. В. Богданов, П. Р. Михеев, И. Н. Суворов в своей работе описали развитие игровых движков, а именно провели обзор от примитивной графики до высокоуровневых инструментариев [4].

Начинаем реализацию механики с создания нового 3D проекта см. рисунок 1.

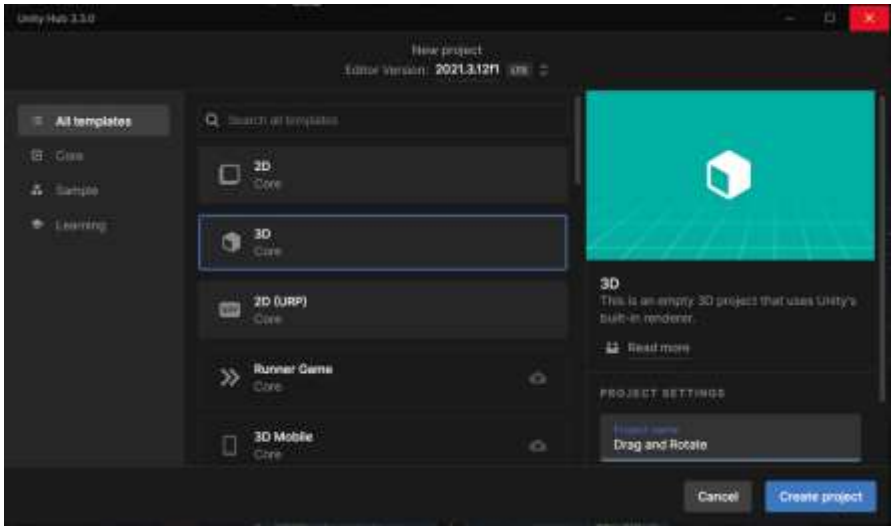

Рис. 1. Создание проекта

Далее в настройках необходимо сменить платформу проекта на android см. рисунок 2-3.

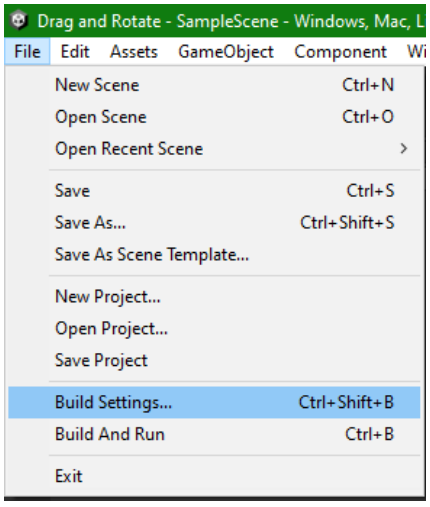

Рис. 2. Переходим в настройки

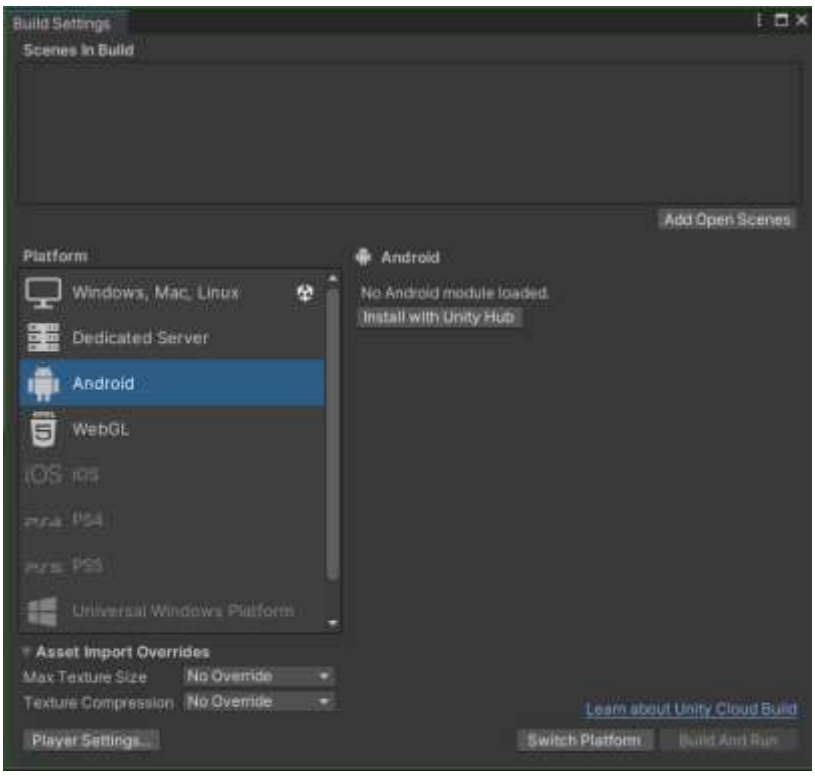

Рис. 3. Смена платформы

Также меняем разрешение проекта на портретное см. рисунок 4.

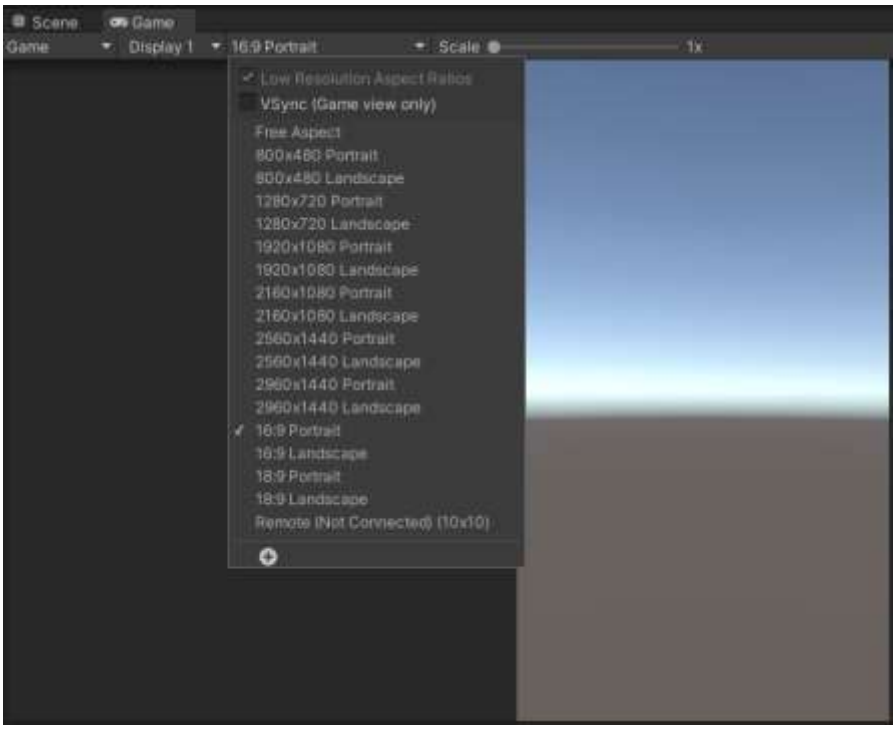

Рис. 4. Смена разрешения

Для подключения к телефону необходимо в настройках проекта разрешить доступ к любому android-смартфону см. рисунок 5-6.

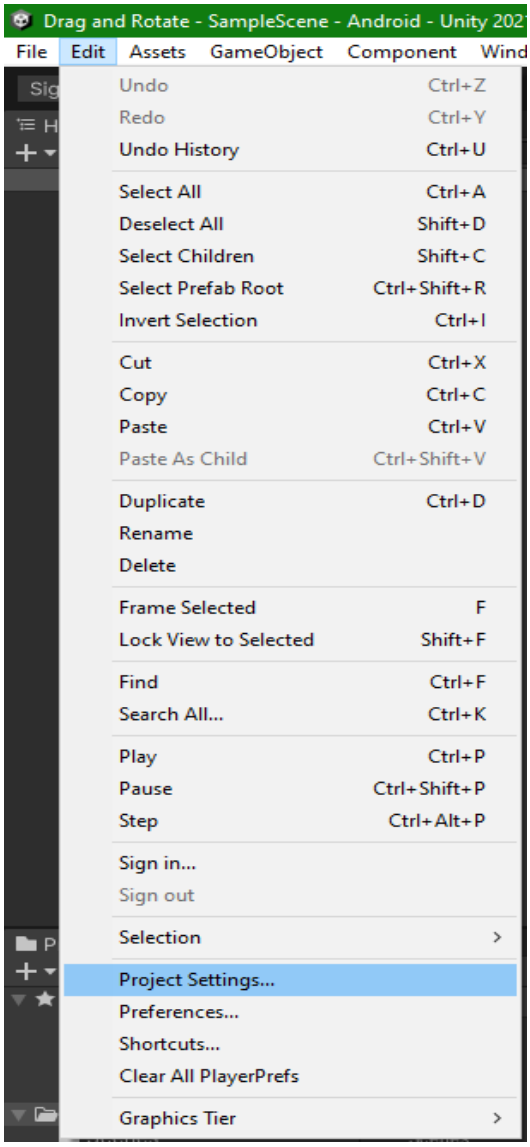

Рис. 5. Переход в настройки проекта

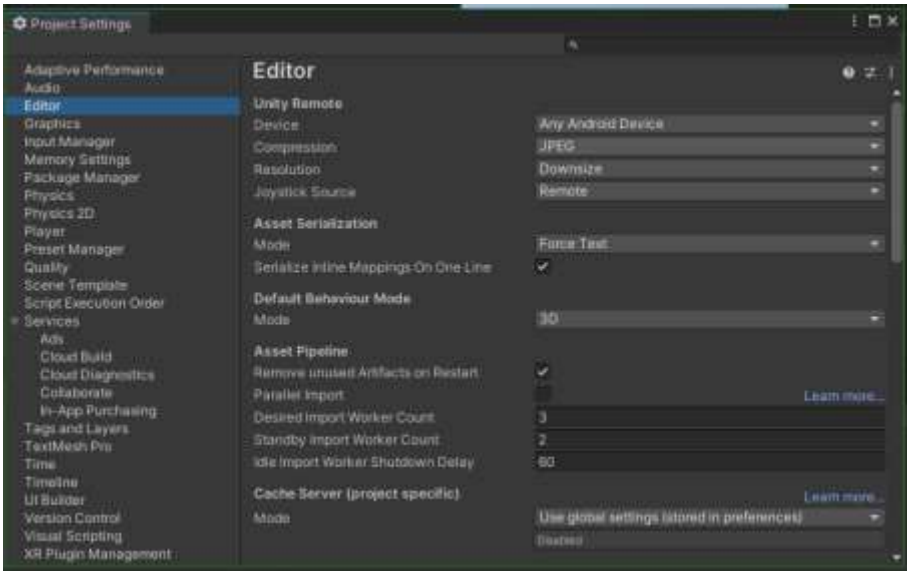

Рис. 6. Разрешение доступа

Далее создаем на сцене два куба с тегом cube см. рисунок 7.

Рис. 7. Создание игровых объектов

Скайбокс изменим на любой сплошной цвет см. рисунок 8.

Рис. 8. Смена заднего фона

Добавляем к камере скрипт «ActivateObjects» см. рисунок 9.

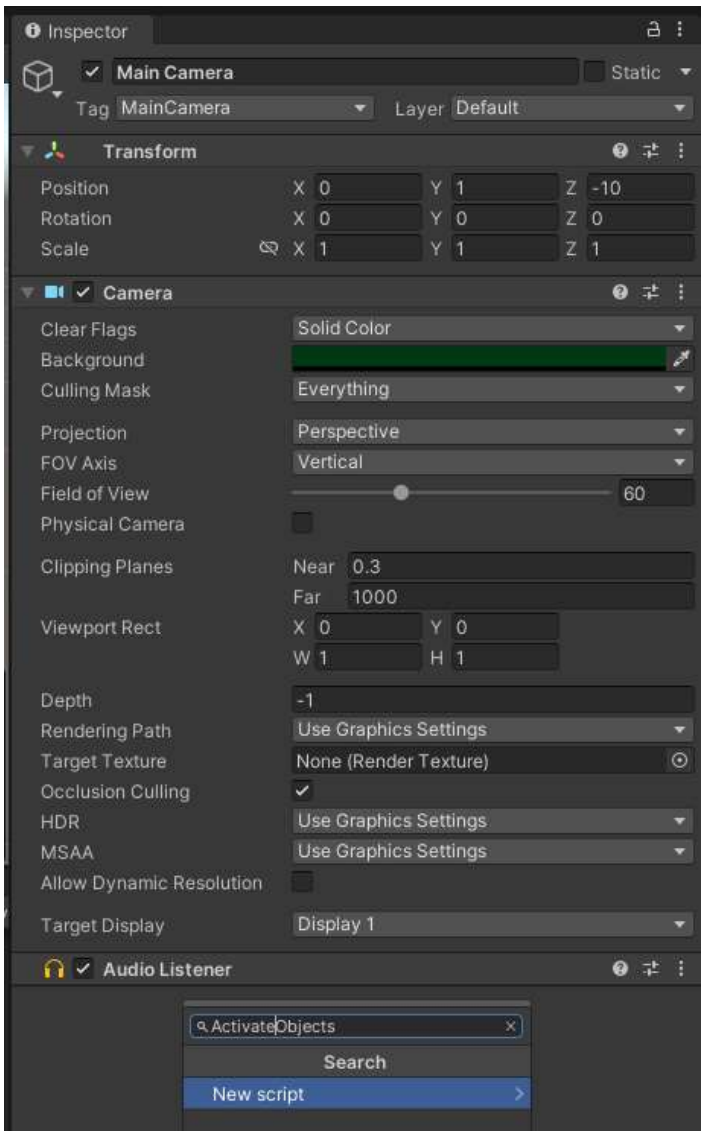

Рис. 9. Добавление скрипта

К обоим кубам добавляем скрипт «DragAndRotate» см. рисунок 10.

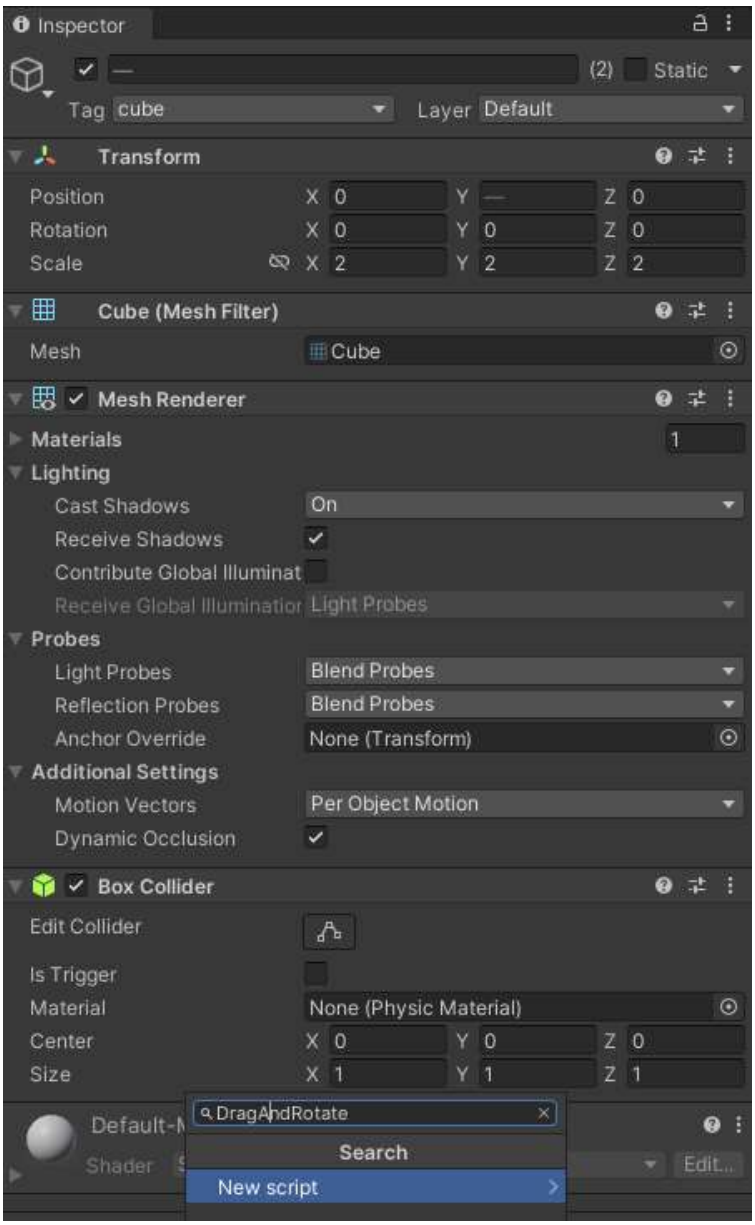

Рис. 10. Добавление скрипта

Переходим к написанию кода, «ActivateObject» будет отвечать за активацию объекта при касании к экрану мобильного устройства. Метод «Update» будет производить проверку, что к экрану прикоснулись и при этом коснулись зарегистрированных приложением объектов. Если касание произошло на объекте с тэгом "cube", то вызывается функция «GetComponent», которая получает компонент «DradAndRotate» этого объекта. Затем переменной «isActive» компонента присваивается значение, которое противоположно текущему значению. То есть, если объект уже был активирован, то после касания он будет дезактивирован, и наоборот см. рисунок 11.

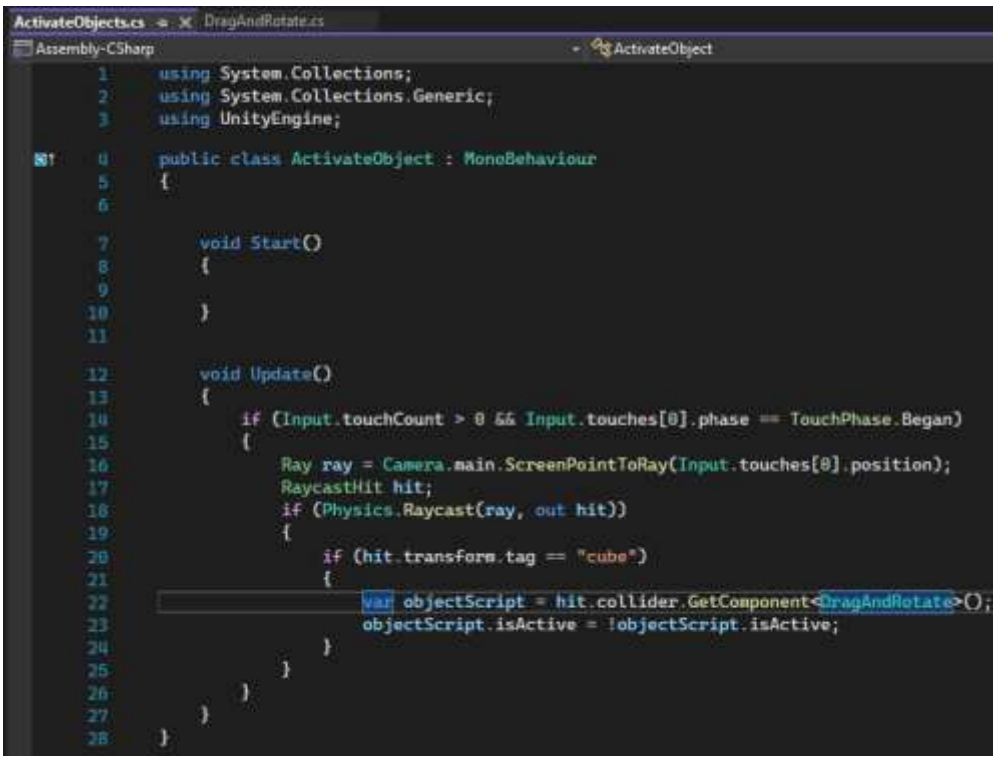

Рис. 11. Скрипт «ActivateObject»

Скрипт «DragAndRotate», будет позволять перемещать и вращать объект при касании экрана на мобильном устройстве. Переменная «isActive» отвечает за состояние объекта: если «isActive = true», то объект перемещается и вращается, если не равно «true», то объект не активен и не подвержен изменениям. Если «isActive = true», то в методе «Update» устанавливается цвет объекта - красный, а если «false» - то белый. Это сделано для того, чтобы пользователь мог заметить, активен ли объект. Далее код проверяет, что на экране одно касание, и если происходит перемещение, то объект вращается. Когда касание заканчивается, «isActive» становится равным «false», что означает, что объект находится в неактивном состоянии. В конце метода «Update» изменяется цвет объекта в материале доступном через «MeshRenderer», которые также зависит от значения «isActive» см. рисунок 12.

ActivateObjects.cs DragAndRotate.cs

Assembly-CSharp DragAndRotate

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class DragAndRotate : MonoBehaviour
6   {
7       public bool isActive = false;
8       Color activeColor = new Color();
9
10
11      void Start()
12      {
13
14      }
15
16
17      void Update()
18      {
19          if (isActive)
20          {
21              activeColor = Color.red;
22              if (Input.touchCount == 1)
23              {
24                  Touch screenTouch = Input.GetTouch(0);
25                  if (screenTouch.phase == TouchPhase.Moved)
26                  {
27                      transform.Rotate(0f, screenTouch.deltaPosition.x, 0f);
28                  }
29                  if (screenTouch.phase == TouchPhase.Ended)
30                  {
31                      isActive = false;
32                  }
33              }
34          }
35          else
36          {
37              activeColor = Color.white;
38          }
39          GetComponent<MeshRenderer>().material.color = activeColor;
40      }
41  }
```

Рис. 12. Скрипт «DragAndRotate»

Теперь необходимо проверить работу скриптов см. рисунок 13-15.

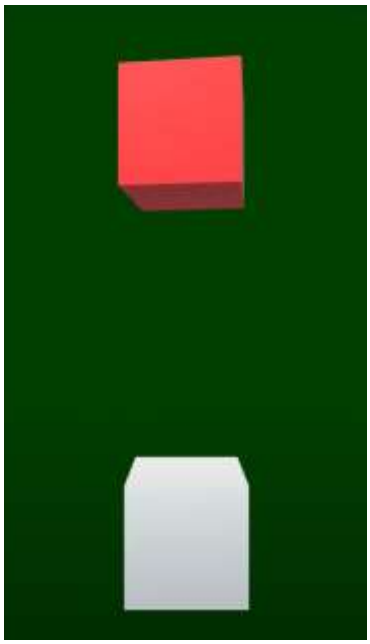

Рис. 13. Вращение первого куба

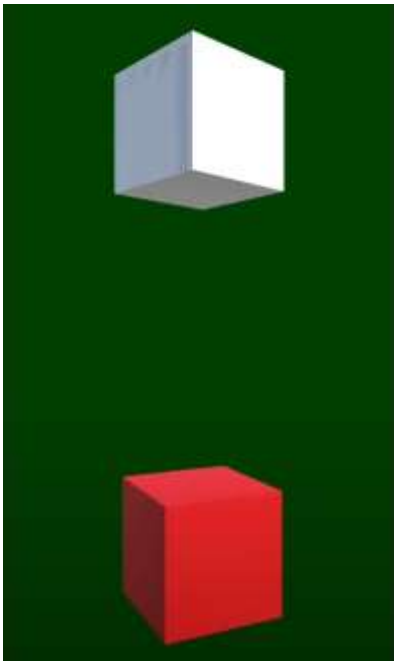

Рис. 14. Вращение второго куба

Как показано на скриншотах механика перетаскивания и поворота игровых объектов работает в соответствии с указанными настройками.

Библиографический список

1. Савин И. А., Батенькина О. В. Написание скриптов для трехмерного графического движка // Визуальная культура: дизайн, реклама, информационные технологии. 2018. № 12-7 (28). С. 7-15.
2. Суродин С. А. Unity 3D. разработка сценария проектирования в среде Unity 3D// Информатика и вычислительная техника. 2015. №3. С. 504-511.
3. 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.
4. Богданов К. В., Михеев П. Р., Суворов И. Н. Развитие игровых движков// Актуальные научные исследования в современном мире. 2021. №4. С. 24-29.