**Laboratory Work**

**«Multimedia Engine»**

**Made by student 8ВМ82 Bokijonov Asrorjon**

**Tomsk 2019**

**Introduction**

Description of the chosen engine or framework, its purpose.

- Tell where it can be downloaded, how it can be used.

- List its advantages and disadvantages compared to other modern multimedia frameworks/engines.

- Do some example work with this engine, describe the process with screenshots.

- List some source code and basic commands which you've used.

- Make a conclusion which should contain your opinion on this multimedia engine or framework.

**Unity Engine**

**Overview**

Unity is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) real-time engine developed by [Unity Technologies](https://en.wikipedia.org/wiki/Unity_Technologies), first announced and released in June 2005 at [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.) Unity gives users the ability to create games and interactive experiences in both [2D](https://en.wikipedia.org/wiki/2D_computer_graphics) and [3D](https://en.wikipedia.org/wiki/3D_computer_graphics), and the engine offers a primary scripting API in [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)). The intuitive workspace, the complete toolset and productive and rapid workflows that come in Unity framework enable the users to reduce the effort, time and cost drastically in making such kind of interactive content. The game can be published on Windows Executable, web browsers, Mac OSX App, Adobe Flash, iPad, iPhone, Android Phones, Xbox 360, Wii U, PS3 and tablets. As there are numerous resources available to come up with a clarity regarding the further step in the development of the game, Unity framework is the most preferred choice. [3D game development](http://www.theappguruz.com/3d-game-development)definitely can be accomplished in faster and shorter time span.

**You can also download Unity engine**

<https://unity3d.com/ru/get-unity/download>

**Unity engine advantages:**

* **Graphics.** The engine is highly preferred for its extended support to 27 platforms. The app developed and deployed can be easily shared between PC, web and mobile platforms. Besides, the agile methodology enables speedy prototyping and constant releases, which in turn speed up the game development.
* **IDE.** The text editor is provided by IDE to write the code, but sometimes a distinct code editor is also used by the developers to alleviate confusion. Additionally, the integrated development editor support JavaScript and C# for scripting, and also offers notable features that are ideal for the game development.
* **Graphics.** The high quality audio and visual effects are supported by the engine that eases the game development. The visuals are adaptable on every screen and device without any distortion or compromise with the image quality.
* **Documentation.** It’s a must have. The novice developers need the easy-to-understand documentation that’s provided in detail by the Unity engine. The detailed documentation includes the explanation of every small topic.
* **Debugging.** The debugging and tweaking is amazingly easier with Unity game development because all the game variables are displayed during gameplay, which in turn allow the developers to debug the process at runtime.

**Unity engine disadvantages:**

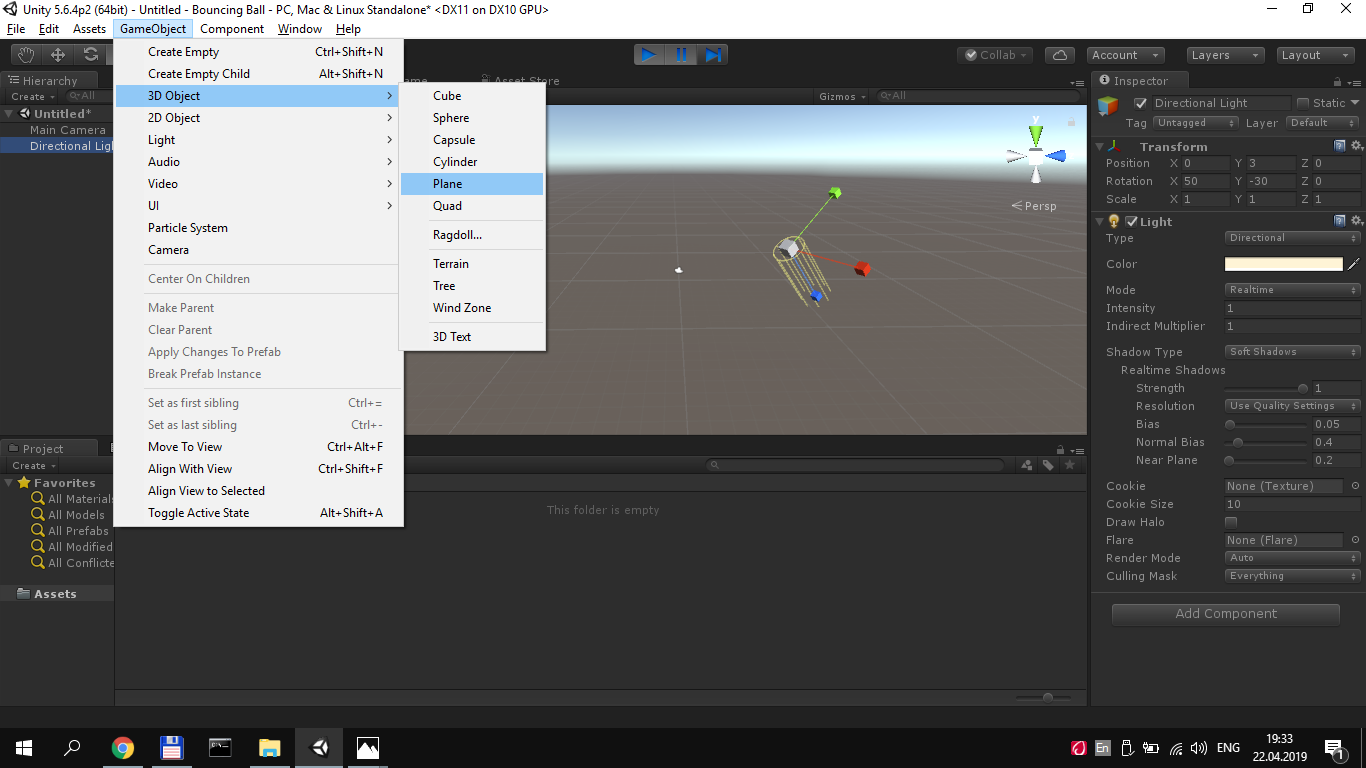
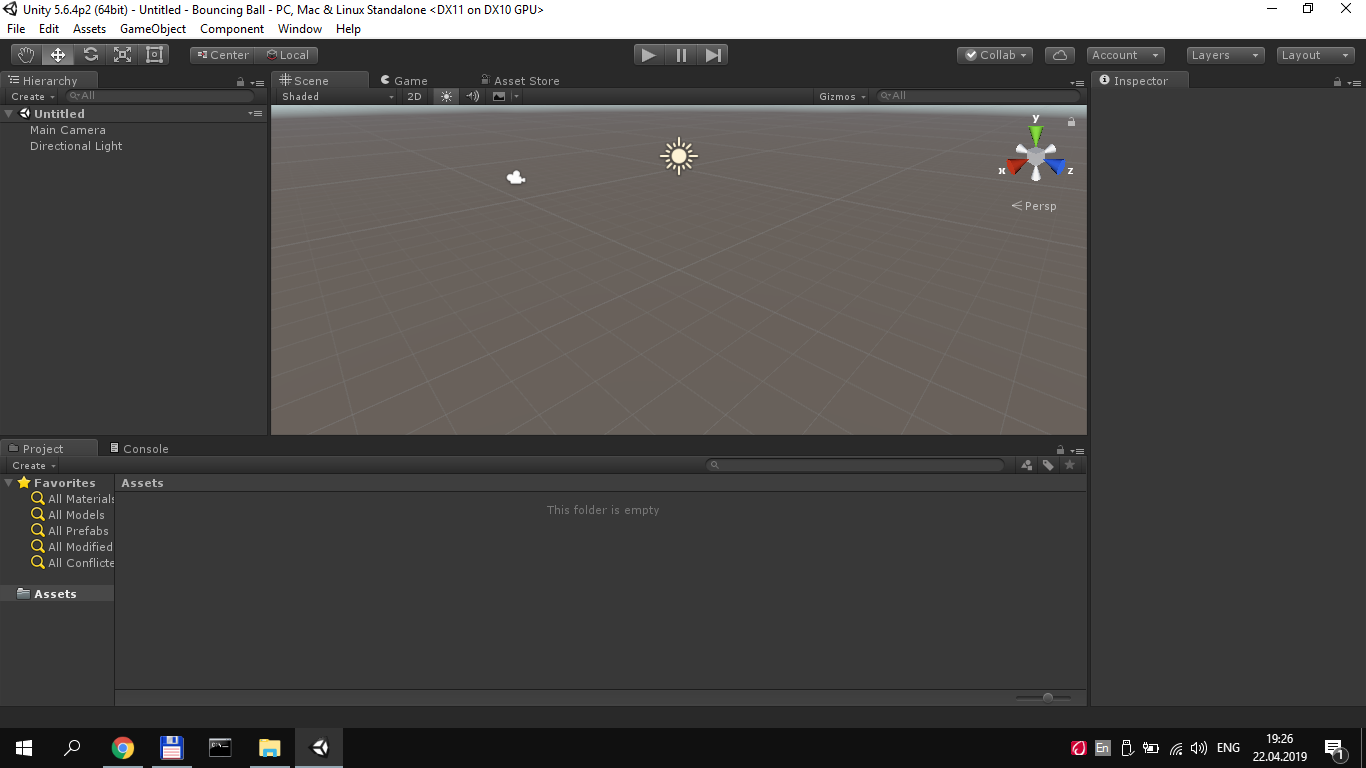
* **Graphics.** Not to say, but the engine lags behind from a graphical point of view. It does not offer an array of tools to create stupendous graphics as opposed to other game development engines.
* **Physics.** In Unity 5 engine, the built-in support for the PhysX physics engine has some performance issues and lacks some important functionalities which need to be added to craft the excellent game app.
* **License cost.** The developers need to have licenses for the best graphics, deployment and performance improvements. These licenses are expensive to purchase. Moreover, the use of rendering, buffer support, stencil support and pretty more features scale up the development costs due to expensive licenses.
* **Source code.** The code is stable in Unity as opposed to other engines and packed with a great architecture that improves the game app performance. But, unavailability of the source code makes finding, addressing and fixing the performance issues difficult.
* **Memory hogging.** The game developed leveraging Unity engine consumes more memory, which in turn creates OOM errors and debugging issues in the apps.

**My game in Unity Engine**

The game is called a Bouncing and rotating cube. Here I make a bouncing cube in fence around the field. I am going to show how I can bounce and rotate our game object, so first of all, I created a new project called the Bouncing cube.

I first of all, create a plane and cube in Unity (Figure 2).

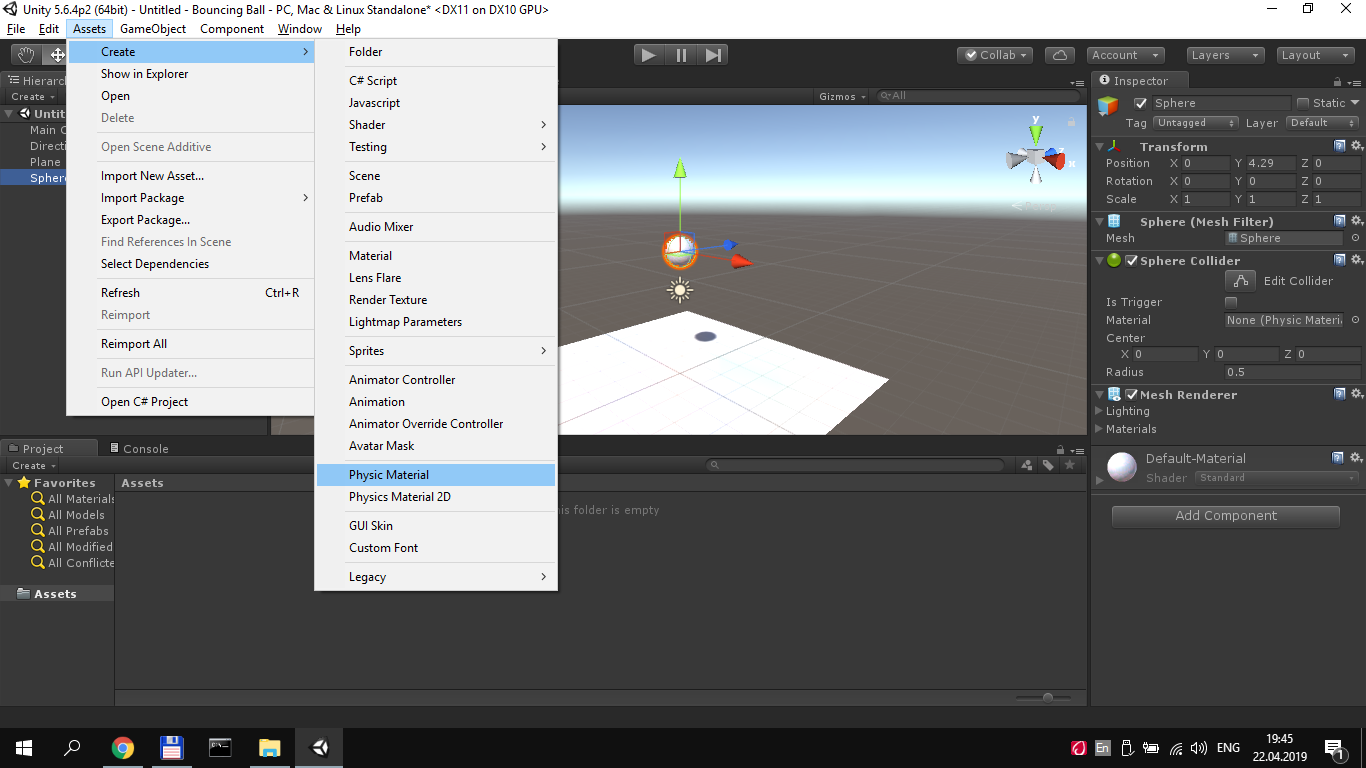
**Figure 1. The Main window of Unity Figure 2. The process of creating a plane and cube**



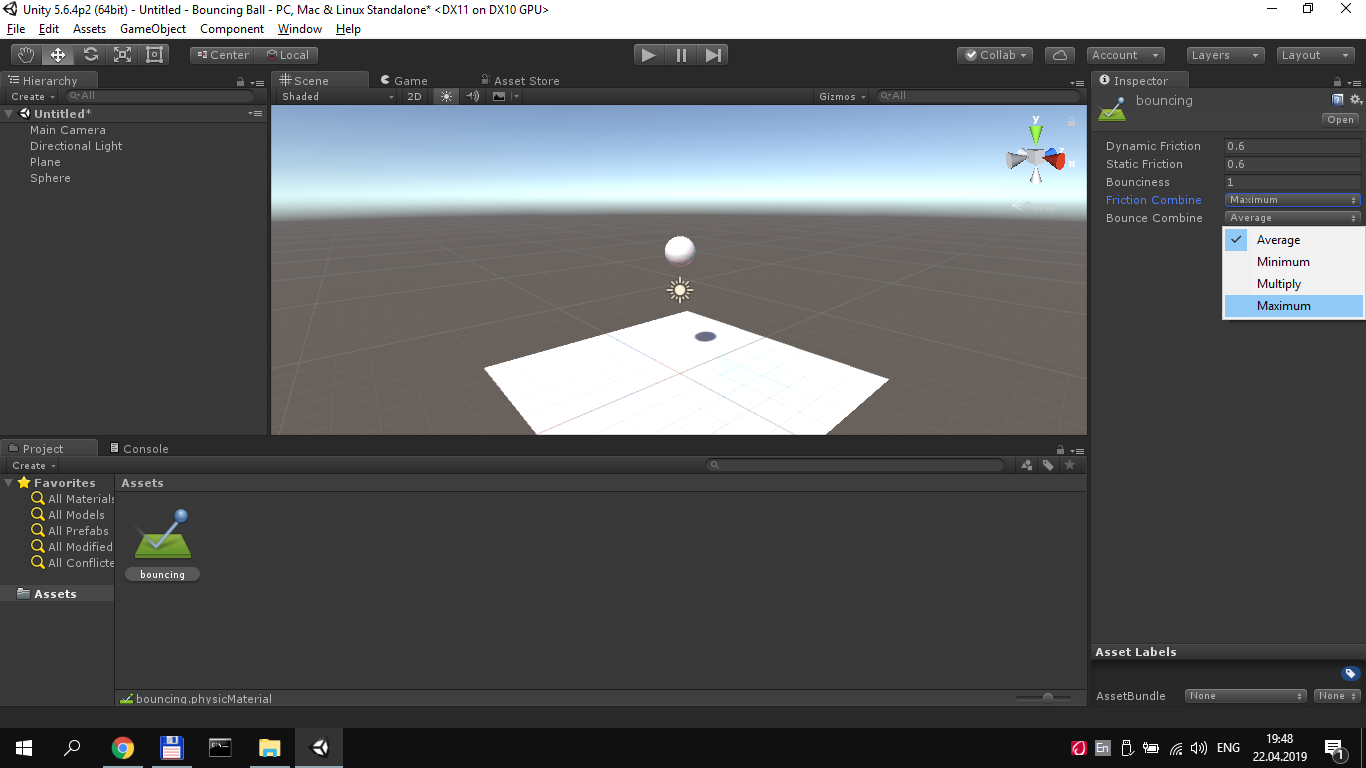
The Physic Material is used to adjust friction and bouncing effects of colliding objects.

To create a Physic Material select **Assets** > **Create** > **Physic Material** from the menu bar. Then drag the Physic Material from the Project View onto a Collider  
 in the scene.

**Figure 3. The process of creating of Physic Material**

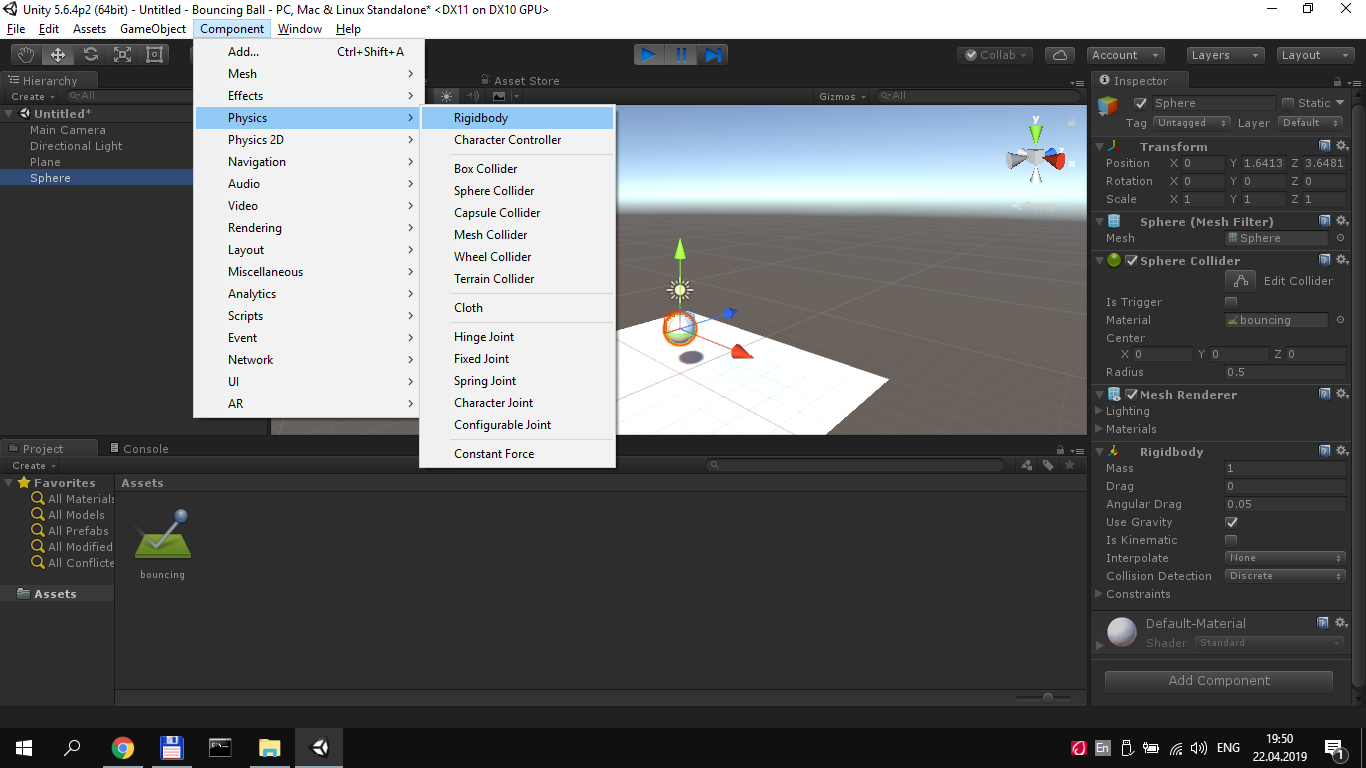


In figure 4 showed how I create a new **Physic Material** called **bouncing.** The **Physic Material** is used to adjust friction and bouncing effects of colliding objects. To create a Physic Material select **Assets > Create > Physic Material** from the menu bar. Then drag the Physic Material from the Project View onto a **Collider** in the scene.

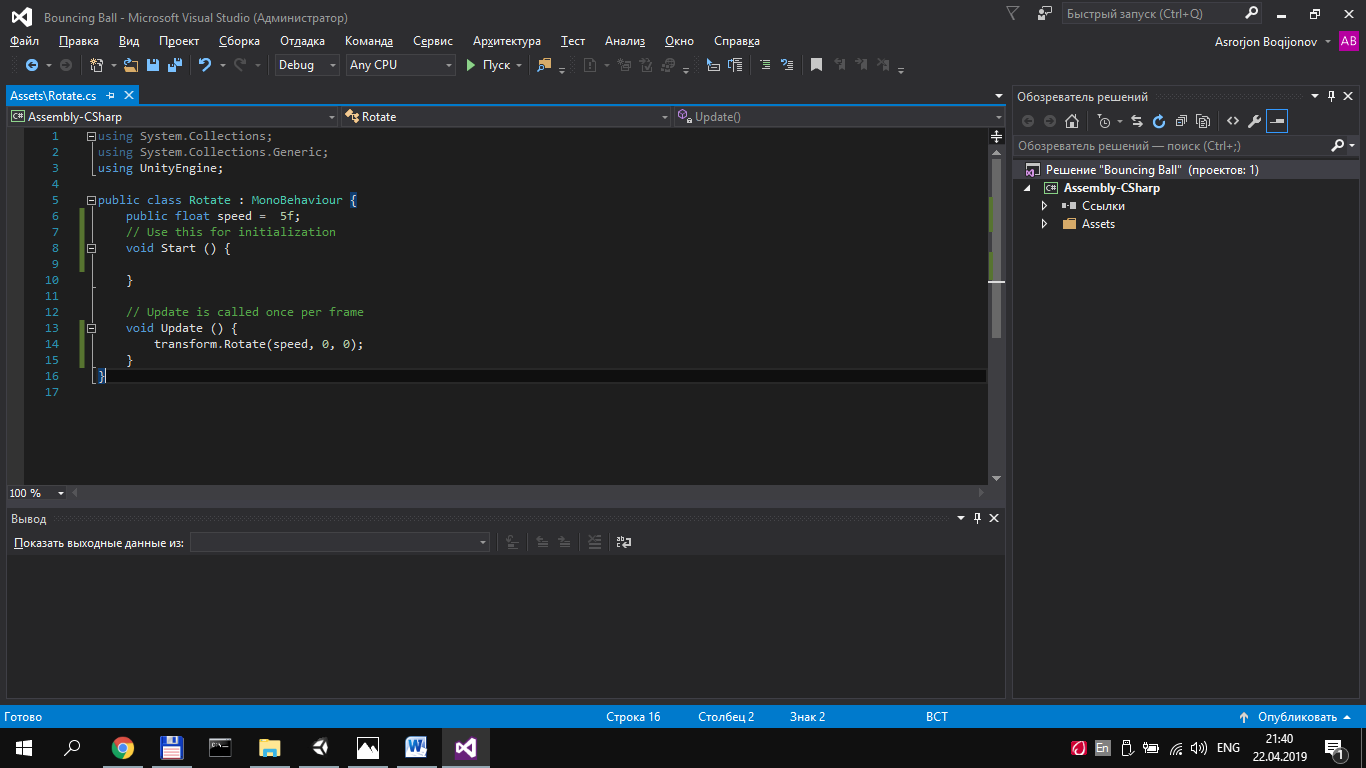


**Figure 4. The Physic Material called Bouncing**  **and it’s properties.**

Rigidbodies enable my GameObjects to act under the control of physics. The Rigidbody can receive forces and torque to make your objects move in a realistic way. Any GameObject must contain a Rigidbody to be influenced by gravity, act under added forces via scripting, or interact with other objects through the NVIDIA PhysX physics engine.  
.  **Figure 5. Creating Rigidbody of game object**



Then I created script to cube and called it rotate. So I have script Rotate open the script in Monodevelop(or Visual Studio)(Figure 6).



**Figure 6. The Script of Rotation Cube in Visual Studio**

**Script of the Rotation Cube in C#**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Rotate : MonoBehaviour {

public float speed = 5f;

// Use this for initialization

void Start () {

}

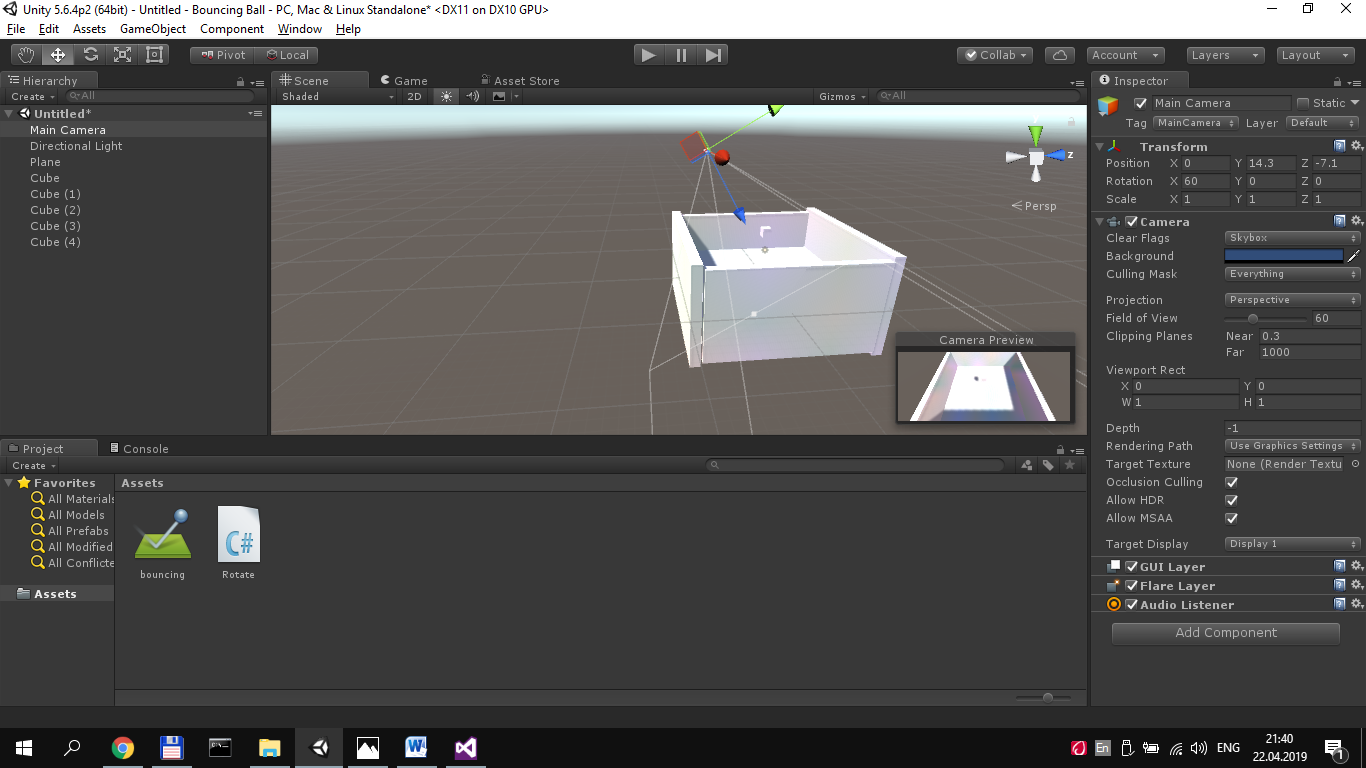
// Update is called once per frame

void Update () {

transform.Rotate(speed, 0, 0);

}

}

****

**Figure 7 Final stage.**

**Conclusion**

Unity is the ultimate and most successful platform for game development, it is used to build 2D and 3D games and deploy them on various operating system without any compromised with quality. Once you know why it works, you see that it is such a **brilliant solution** that you will hardly stop to explain anyone why it works: you will be now dedicating your time to creating games!