



Budapest University of Technology and Economics

Car driving game by Unity engine

Supervisor: Hideg Attila

Author: Dong Sizhe

2021.5

Content

目录

Content	2
Motivation	3
Unity Engine	4
Study Process	4
1. Get the setup wizard	4
Minimum System Requirement	4
Install Unity and the Hub	5
Explore Unity Editor	7
Review the Package Manager	11
2. Work with GameObjects in 3D Scene	13
Create a new 3D project and Scene	13
Create simple 3D objects	14
Organize GameObjects in the Hierarchy	18
Master the tools	19
3. Add components to 3D GameObjects	21
Create sphere in scene	21
Position the Main Camera	22
Make the sphere solid	23
4. Create effects for 3D GameObjects	25
Import an image file	25
Create the Brick Material	28
Adding a Physic Material	30
Adjusting the Lighting in the scene	32
5. Get 3D effect	33
Get assets from the Asset Store	33
Import assets from the Asset Store	35
Develop my own project	37
1. Introduce of the texture	37
2. Introduce the model	37
Summary	40

Motivation

I played many car driving game on different platforms , for example Euro Truck Simulator , Dirt Rally , Real Racing In this case , I'm very interesting how it is designed and developed , so I decide to study game development of Unity engine in this semester as my project lab.

Unity Engine

Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Inc.'s Worldwide Developers Conference as a Mac OS X-exclusive game engine. The engine has since been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms. It is particularly popular for iOS and Android mobile game development and used for games such as Pokémon Go, Monument Valley, Call of Duty: Mobile, Beat Saber and Cuphead. It is cited to be easy to use for beginner developers and is popular for Indie game development.

The engine can be used to create three-dimensional (3D) and two-dimensional (2D) games, as well as interactive simulations and other experiences. The engine has been adopted by industries outside video gaming, such as film, automotive, architecture, engineering and construction.

Study Process

1. Get the setup wizard

Minimum System Requirement

Mobile			
Operating system	Android	iOS	tvOS
Version	4.4 (API 19)+	11+	11+
CPU	ARMv7 with Neon Support (32-bit) or ARM64	A7 SoC+	A8 SoC+
Graphics API	OpenGL ES 2.0+, OpenGL ES 3.0+, Vulkan	Metal	Metal
Additional requirements	1GB+ RAM. Supported hardware devices must meet or exceed Google's Android Compatibility Definition (Version 9.0) limited to the following Device Types: 1. Handheld (Section 2.2) 2. Television (Section 2.3) 3. Tablets (Section 2.6) Hardware must natively be running Android OS. Android within a Container or Emulator is not supported. For Development: Android SDK (10/API 29), Android NDK (r19) and OpenJDK, which are installed by default with Unity Hub	For development: Mac computer running minimum macOS 10.12.6 and Xcode 9.4 or higher.	Apple TV 4th generation+

Desktop

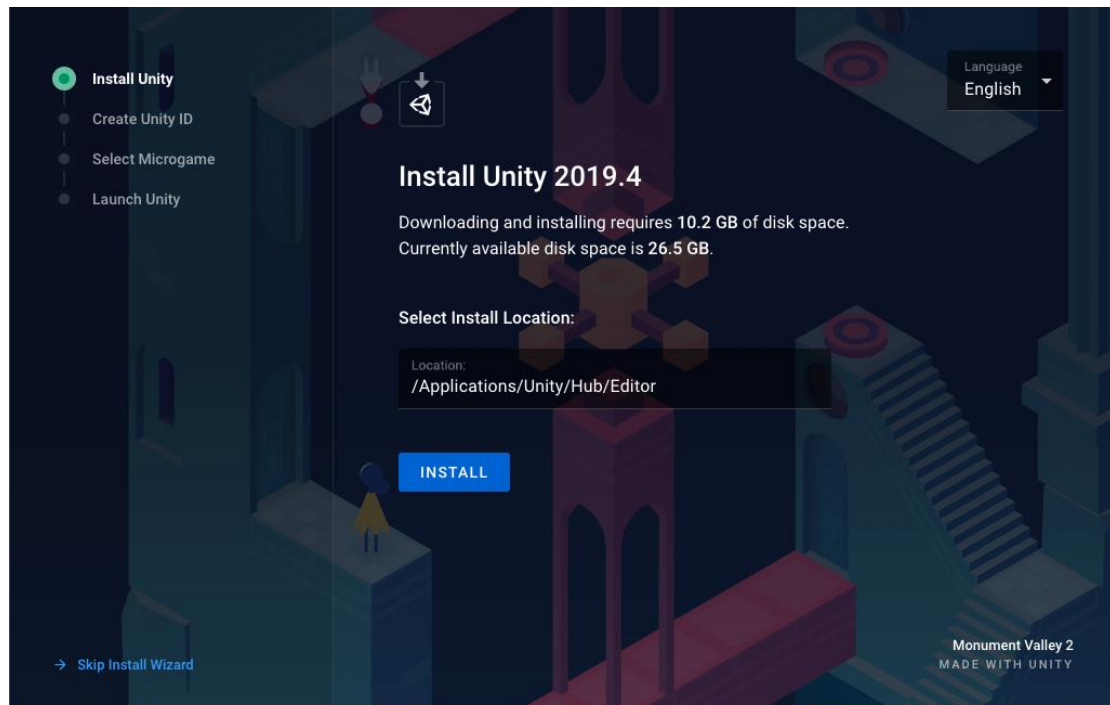
Operating system	Windows	Universal Windows Platform	macOS	Linux
Operating system version	Windows 7 (SP1+) and Windows 10	Windows 10, Xbox One, HoloLens	High Sierra 10.13+	Ubuntu 20.04, Ubuntu 18.04, and CentOS 7
CPU	x86, x64 architecture with SSE2 instruction set support.	x86, x64 architecture with SSE2 instruction set support, ARM, ARM64.	x64 architecture with SSE2.	x64 architecture with SSE2 instruction set support.
Graphics API	DX10, DX11, DX12 capable.	DX10, DX11, DX12 capable GPUs.	Metal capable Intel and AMD GPUs	OpenGL 3.2+, Vulkan capable.
Additional requirements	Hardware vendor officially supported drivers. For development: IL2CPP scripting backend requires Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK.	Hardware vendor officially supported drivers. For development: Windows 10 (64-bit), Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK.	Apple officially supported drivers. For development: IL2CPP scripting backend requires Xcode. Targeting Apple Silicon with IL2CPP scripting backend requires macOS Catalina 10.15.4 and Xcode 12.2 or newer.	Gnome desktop environment running on top of X11 windowing system Other configuration and user environment as provided stock with the supported distribution (such as Kernel or Compositor) Nvidia and AMD GPUs using Nvidia official proprietary graphics driver or AMD Mesa graphics driver.
For all operating systems, the Unity Player is supported on workstations, laptop or tablet form factors, running without emulation, container or compatibility layer.				

Install Unity and the Hub

If you're a new user, you'll use the installation wizard to install the latest long term support (LTS) version of Unity.

Install Unity 2019.4 LTS

1. Locate the UnityHubSetup.exe file you downloaded previously.
2. Launch the **UnityHubSetup** installer.
3. Follow your platform instructions to install the Unity Hub.
4. Open the Unity Hub application.
5. Choose an install location then select **Install**.



Create your Unity ID

To create your Unity ID:

1. Complete the required fields.

Create Unity ID

Username

Email

Password

☐ I have read and agree to the [Unity Terms of Service](#), which include the Additional Terms for Unity Software (the "[Software Terms](#)"), and I acknowledge the [Unity Privacy Policy](#).

☐ I understand that by checking this box, I am agreeing to receive promotional materials from Unity

CREATE UNITY ID

[I already have a Unity ID](#)

OR:

Sign in with google

Sign in with facebook

Downloading Unity ... 5.1%

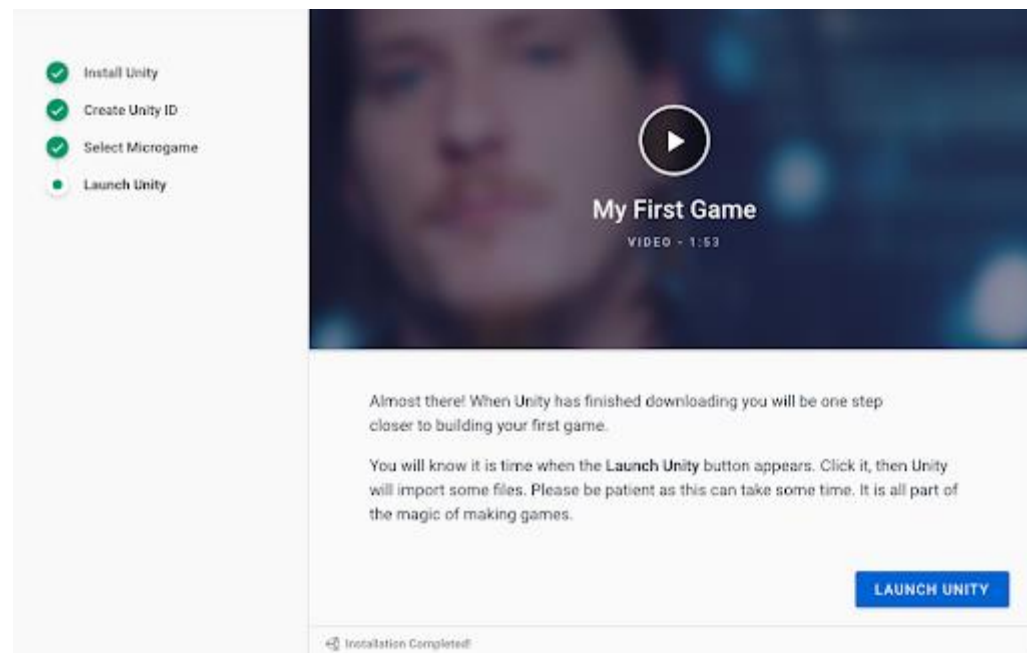
2. Review and agree to the Terms of Service.

Note: The promotional materials are useful, with lots of tips to help you on your learning journey.

3. Select **Create Unity ID**.

Launch Unity Editor

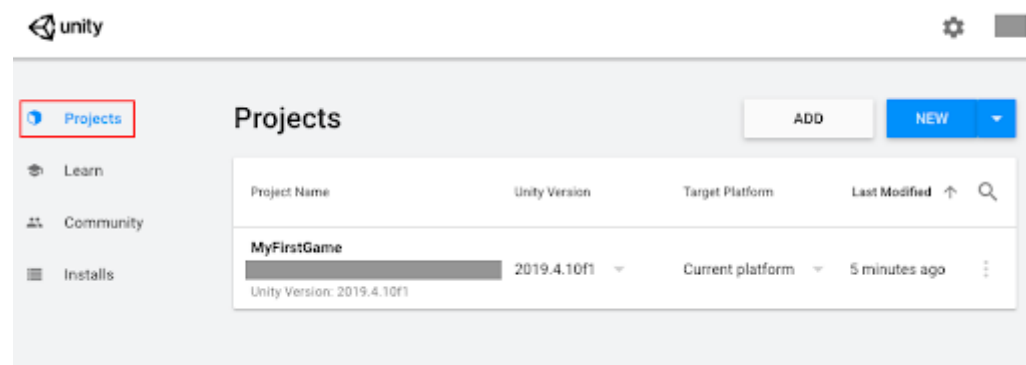
When Unity has finished installing, select **Launch Unity**. Unity might also take some time to launch, the first time you open it. That's also completely normal.



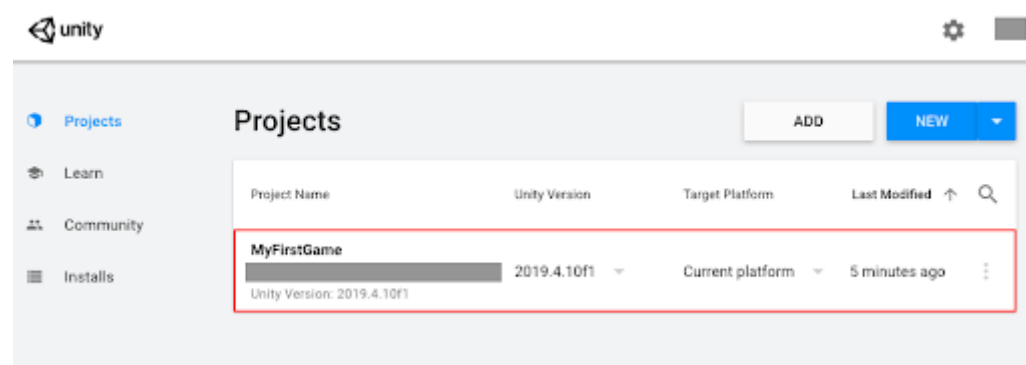
Explore Unity Editor

Before you begin

1. Open the Unity Hub and select the **Projects** tab.



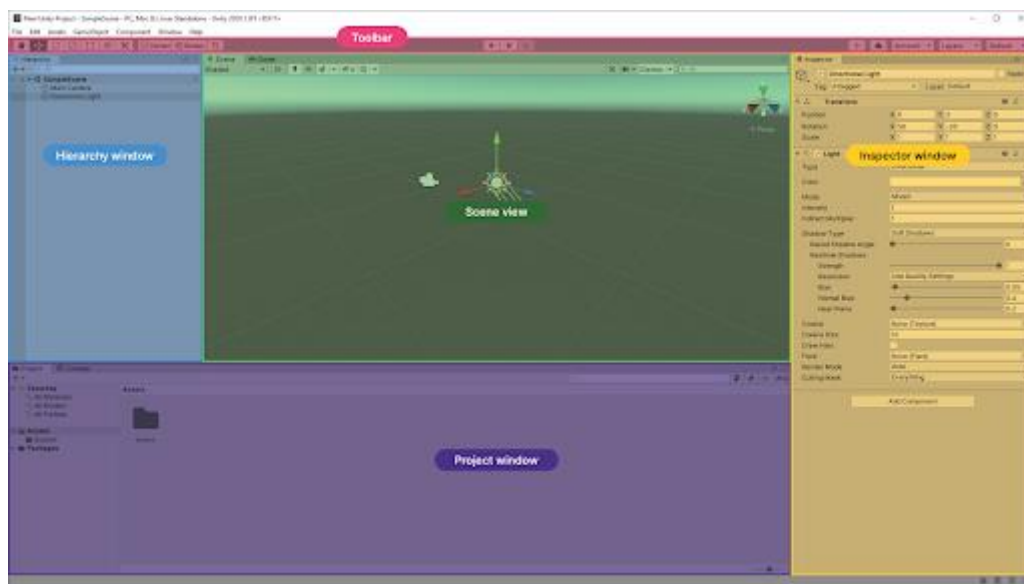
2. Select the project name in the list available to open it.



Introduction to the Unity Editor

As you saw in the microgame tutorials, different areas of the Unity Editor interface are used to complete different tasks. You've already used the main windows of the Editor interface, but let's quickly recap them here and review the different layouts available.

Unity Editor interface



There are five key areas of the basic Editor interface.

Scene view and Game view

In the center of the default Unity Editor layout is the **Scene view**. This is your interactive window into the world you are creating. You'll use the Scene view to manipulate objects and view them from various angles.

In the default layout, the **Game view** also appears in this area; you'll use the Game view to playtest your game.

Hierarchy window

The **Hierarchy** is where you can organize all the things in your game world (and the world itself). These things are called **GameObjects**.

If you add a GameObject to your project in Scene view, it will be listed in the Hierarchy.

If you delete a GameObject from the Scene, it will no longer be listed.

Project window

The **Project window** is where you can find all the files (assets) available for use in your project, whether you use them or not.

The Project window works like a file explorer, organized in folders. You can drag assets directly from the Project window into the Scene view to add them to the Scene.

Note the difference between the Project and Hierarchy windows: the Hierarchy contains all the GameObjects in the current Scene, and the Project window contains all the assets available to your entire project.

Inspector window

The **Inspector** is where you'll find and configure detailed information about GameObjects.

When you select a GameObject in Scene view or in the Hierarchy, you'll see its **components** in the Inspector. Components describe the properties and behaviors of GameObjects.

Toolbar

The Toolbar is always at the top of the Unity Editor interface. Use the toolbar buttons to select and adjust GameObjects, change your point of view in the Scene, and start and stop Play Mode.

Editor layouts

Use the **Layout** menu in the Toolbar to change the arrangement of the windows in the Editor. There are several layouts provided, and you can save your own layouts.

The best layout for the Editor depends on what you are doing and your own personal preferences. For example:

- If you're spending a lot of time configuring components, you might want to dock the Inspector next to the Hierarchy.
- If you're doing a lot of environment design, you might want to make the Scene view as large as possible.
- If you're editing the user interface, you'll want to make sure the Game view is visible.

Using Scenes in your project

Projects in Unity Editor are organized into **Scenes**. Scenes are containers for everything in the experience you are creating.

One way to think about Scenes is as discrete experiences. For example, each level in a game could be a separate Scene, and the game's main menu could be another.

A Unity project can have one Scene or more than a hundred, depending on its scope and complexity. There aren't strict rules about exactly how you should organize a Unity project into Scenes, except that a project in Unity must have at least one Scene.

Review the Scenes in the Microgame

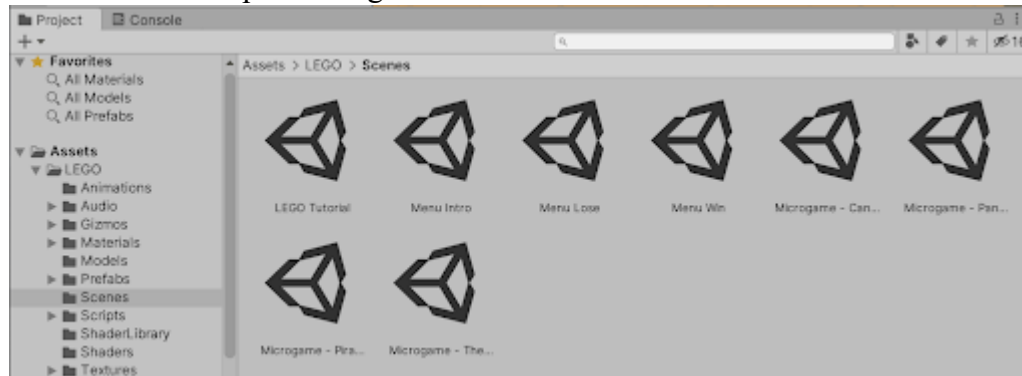
Let's review the Scenes in your Microgame project:

1. In the Project window, go to **Assets** > **[name of your Microgame]** > **Scenes**. This folder contains all the Scenes in the project.

2. Review the names of the Scenes in the folder.

In the LEGO® Microgame, there are eight Scenes:

- One Scene for the in-Editor tutorial experience
- Three menu Scenes
- Four example Microgame Scenes



Navigating the Scene

When working in Scenes, you'll want to navigate the Scene view easily. Navigating the Scene is like operating a drone camera — it lets you examine your GameObjects from any angle or distance.

In the Microgame, you navigated your Scene by changing position, angle, and zoom of the Scene view. With practice, you can learn to navigate with ease. There are also more general settings you can use to configure the Scene view.

Let's quickly review the basics:

- **Pan:** Select the **Hand** tool in the Toolbar, and click and drag in the Scene view to move your view.
- **Zoom:** Holding **Alt** (Windows) or **Option** (macOS), right-click and drag in the Scene view to zoom.
- **Orbit:** Holding **Alt** (Windows) or **Option** (macOS), left-click and drag to orbit around the current pivot point. Note: this option is not available in 2D mode.
- **Focus (Frame Select):** When a GameObject is selected, select **F** with your cursor in the Scene view to focus your view on that GameObject. Note: if your cursor is not in the Scene view, Frame Select will not work.

You can also use Flythrough mode to navigate in the Scene view by flying around in first person, which is common in many games. To do this:

- Click and hold the right mouse button.
- Use **WASD** to move the view left/right/forward/backward.
- Use **Q** and **E** to move the view up and down.
- Select and hold **Shift** to move faster.

Review the Package Manager

You will use the **Package Manager** window to install, remove, and update packages that add functionality to the Unity Editor.

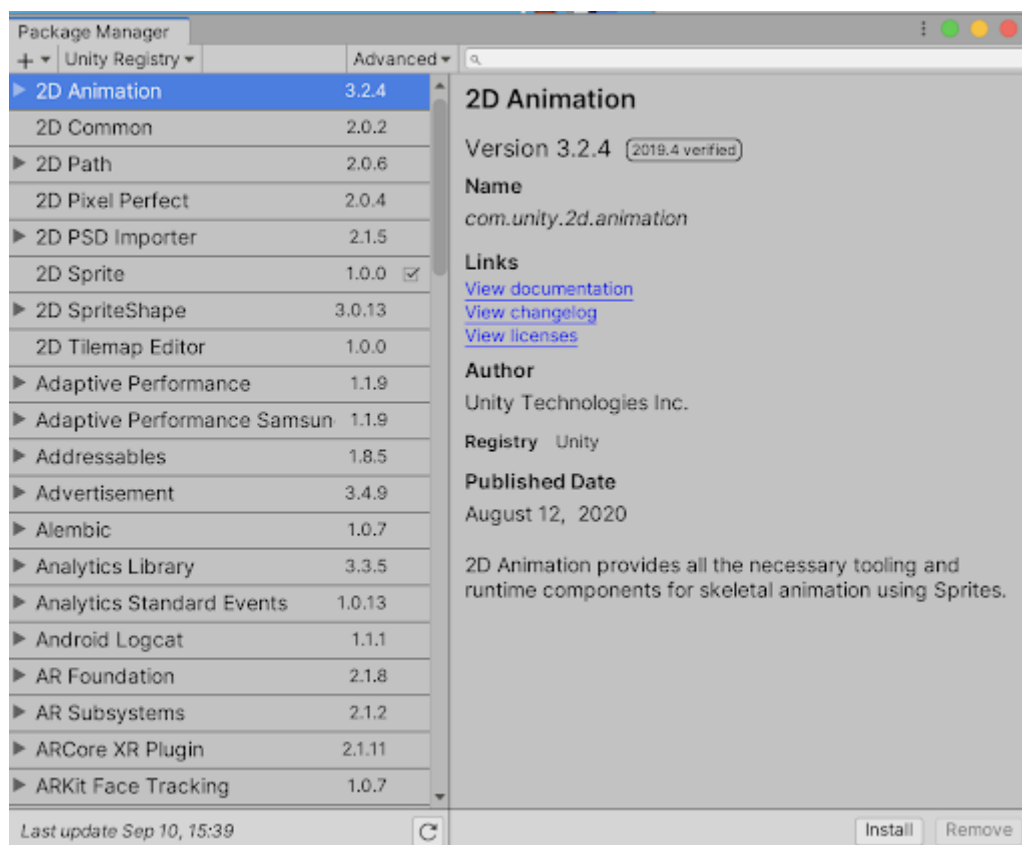
To open the Package Manager:

1. In the top menu, go to **Window > Package Manager**.



2. By default, the window will show the **Unity Registry**, listing all the available packages you can use to enhance Unity Editor.

There are quite a lot of packages!



Should you ever need specific information about a particular package, you can find each package's documentation in the Unity Manual.

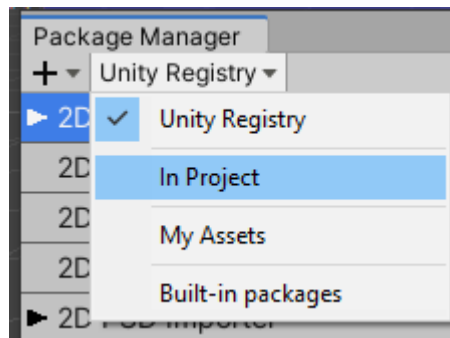
Beyond the functionality provided by the core Editor and packages, as you gain experience, you might need more custom tools to achieve your goals. This is possible through both third-party plugin integrations and by creating your own in-Editor tools.

Keep this in mind as you become more experienced — your possibilities are limitless!

Packages in your project

Let's review the specific packages in your Microgame project.

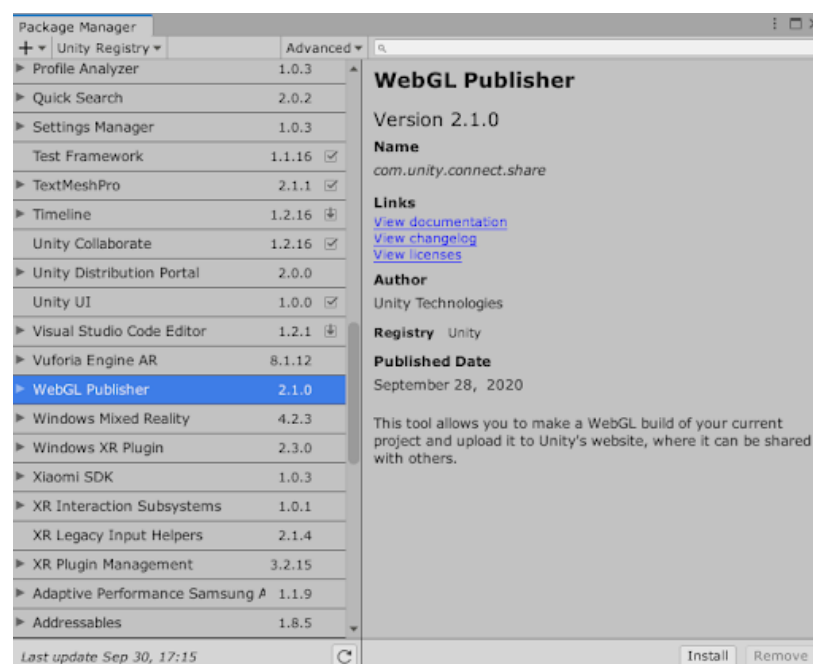
3. In the left-hand corner of the window, select the package scope drop-down menu and choose In Project.



These are the additional packages that the team which created the Microgame used to craft the player experience that you customized.

4. Let's double-check that you have a package to help you to share your work with others later in this course. Scroll the list of in-project packages and check if you have the WebGL Publisher package. If you do, skip the next step.

5. If you do not have the WebGL Publisher package as indicated by a checkmark, return to the Unity Registry and select it. Then select the Install button to add it to the Unity Editor.

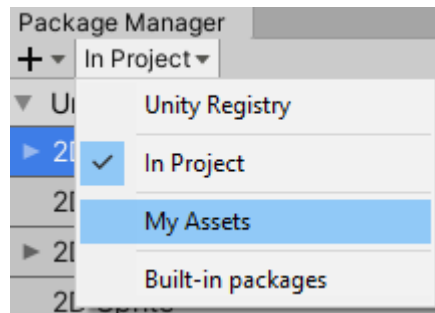


6. When installation is complete, close the Package Manager window.

Assets in the Package Manager

Another useful feature of the Package Manager is the My Assets section, which catalogs assets you have imported from other sources — including the Unity Asset Store, which you will visit later in this course.

In the left-hand corner of the window, select the package scope drop-down menu and choose My Assets.

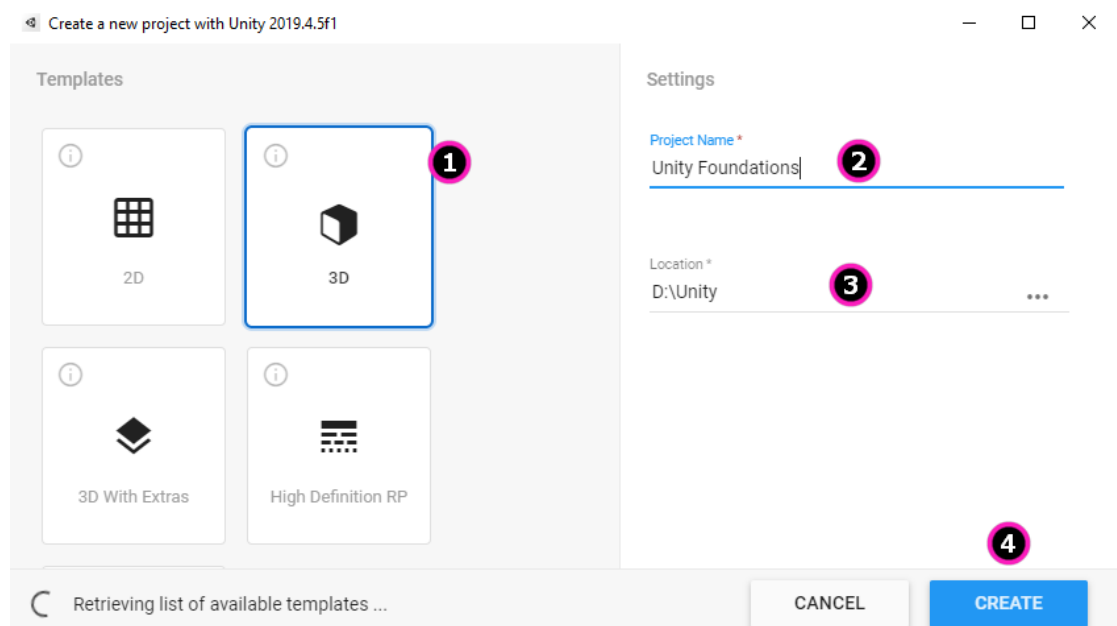


You will see a list of assets used in your Microgame, plus any assets you might have imported already. Later, you will use this window to bring new assets into Unity to use in your own projects.

2. Work with GameObjects in 3D Scene

Create a new 3D project and Scene

In the Unity Hub, create a new Unity project using the 3D template.

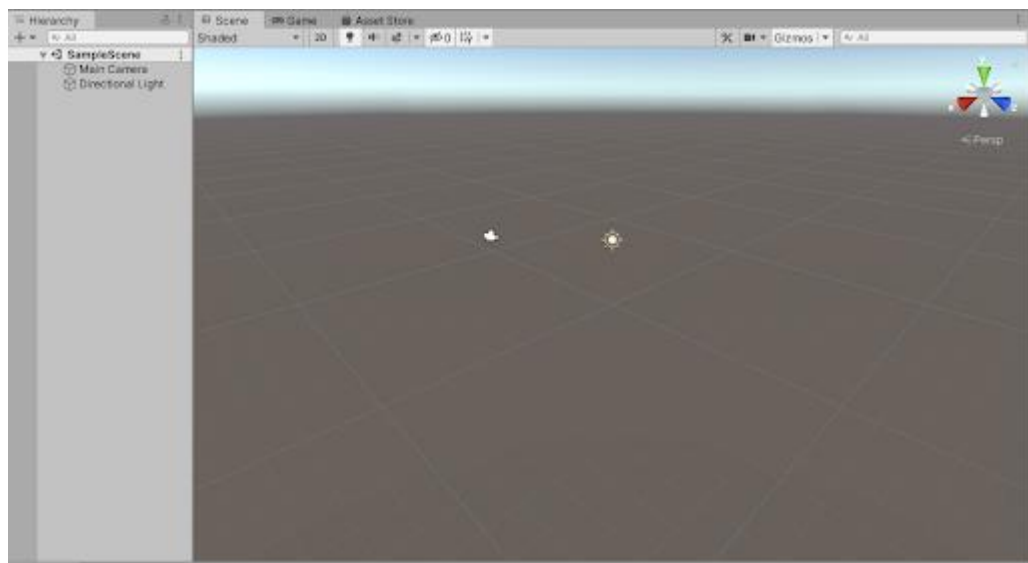


Select the 3D template option (1), give your project a name (2), select the location you wish to save the project (3), and select Create (4). Your new project will contain one Scene.

Note: You might see a prompt to download the 3D Template if this is the first time you have used it. Look for this prompt on the 3D card when you select it.

The default 3D Scene

In the Microgames, you started in a Scene that was already built for you, with characters that moved and responded to your commands, a rigid ground on which they traveled, and goals and obstacles to make the game interesting. Now, however, you are starting from the very beginning. The default Scene in the 3D Template only gives you vast, boundless, empty space. A plane that looks like the ground is just for reference — it is not even a rigid surface.



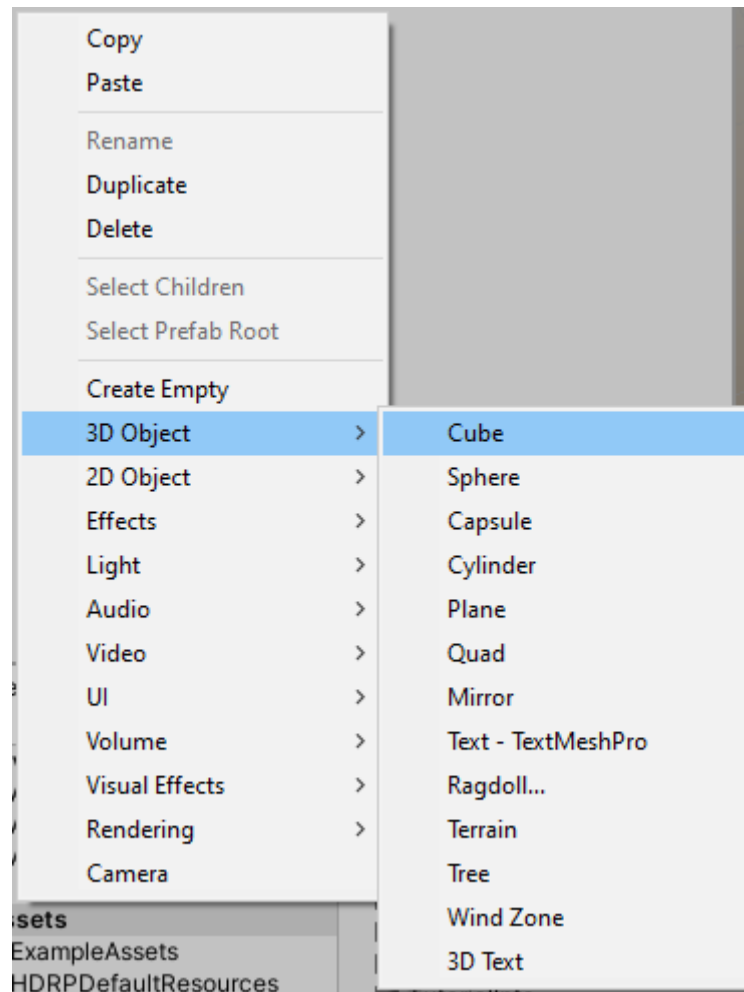
The default 3D Scene comes equipped with two important GameObjects, which are listed in the Hierarchy window:

Main Camera, which controls what your players will see in the Game view (Play mode)

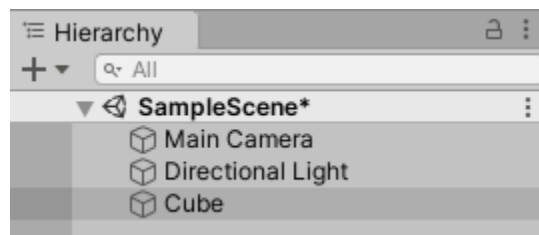
Directional Light, which simulates the sun and provides light that will reflect off 3D your GameObjects to create realistic visual effects

Create simple 3D objects

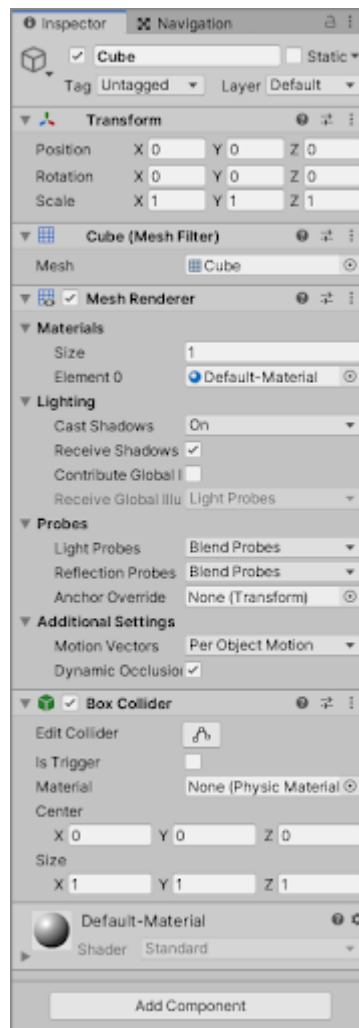
1. To create a cube primitive in the Scene, right-click an empty section in the Hierarchy and select 3D Object > Cube.



The GameObject named Cube now appears in the Hierarchy, and a Cube appears in the Scene view.



2. With Cube selected, look at the Inspector window, which displays the GameObject's properties.

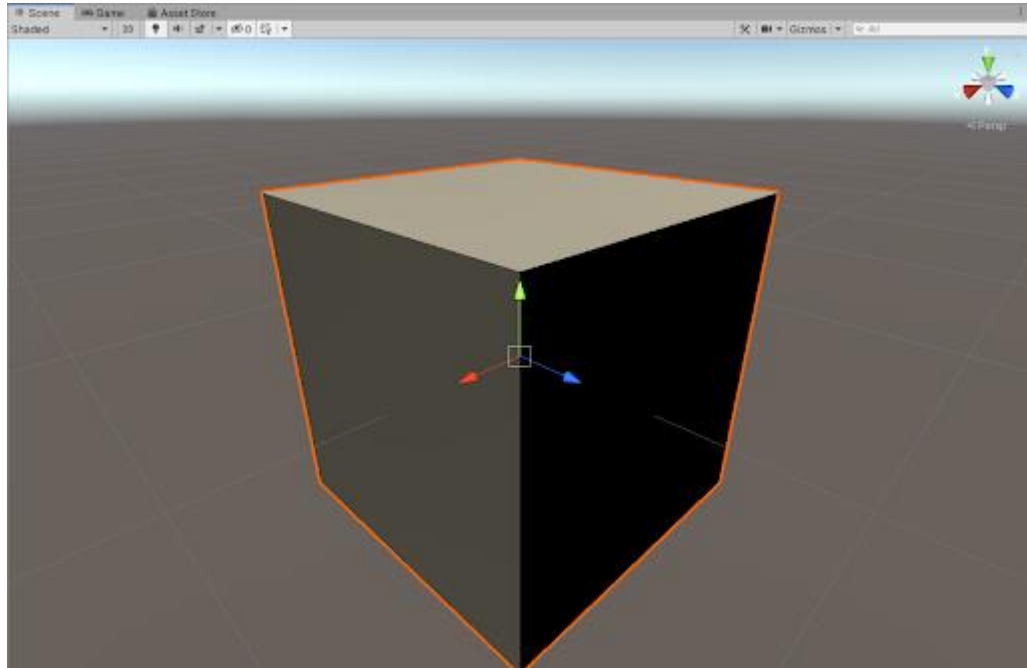


Each section of the Inspector represents a component, which is a set of properties and behaviors of the selected GameObject. Some components are built-in to primitives like the ones you see here. Later, you will add more components to give GameObjects more sophisticated properties and behaviors.

With the Cube selected, the Transform Component displays the size, rotation, and position of the Cube in the Scene using X,Y, and Z values. By default, these values are meters.

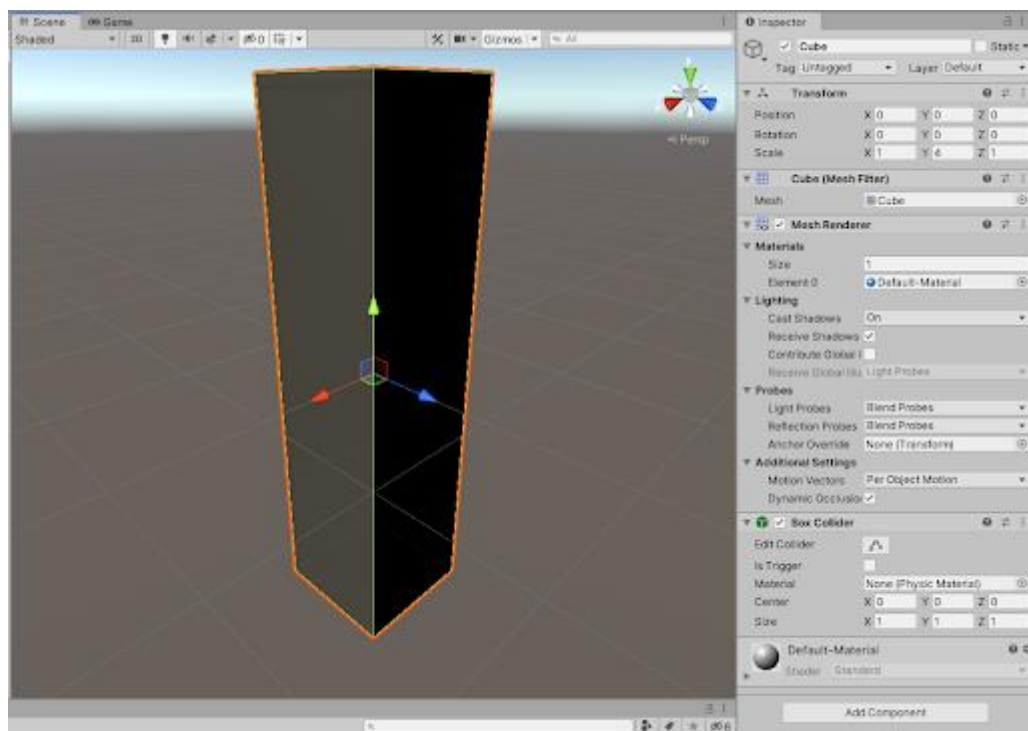
Note: Unity uses a Y-up coordinate system. This means that in the Editor screen space, the Y direction is vertical and X and Z represent the horizontal plane (similar to north, south, east, and west). Some 3D applications, such as 3DsMax, use Z as the vertical.

3. Select F on the keyboard to focus on the Cube. (Tip: The Scene view window must be activated; you can select a GameObject in Scene view, or right-click anywhere, to activate the window.)



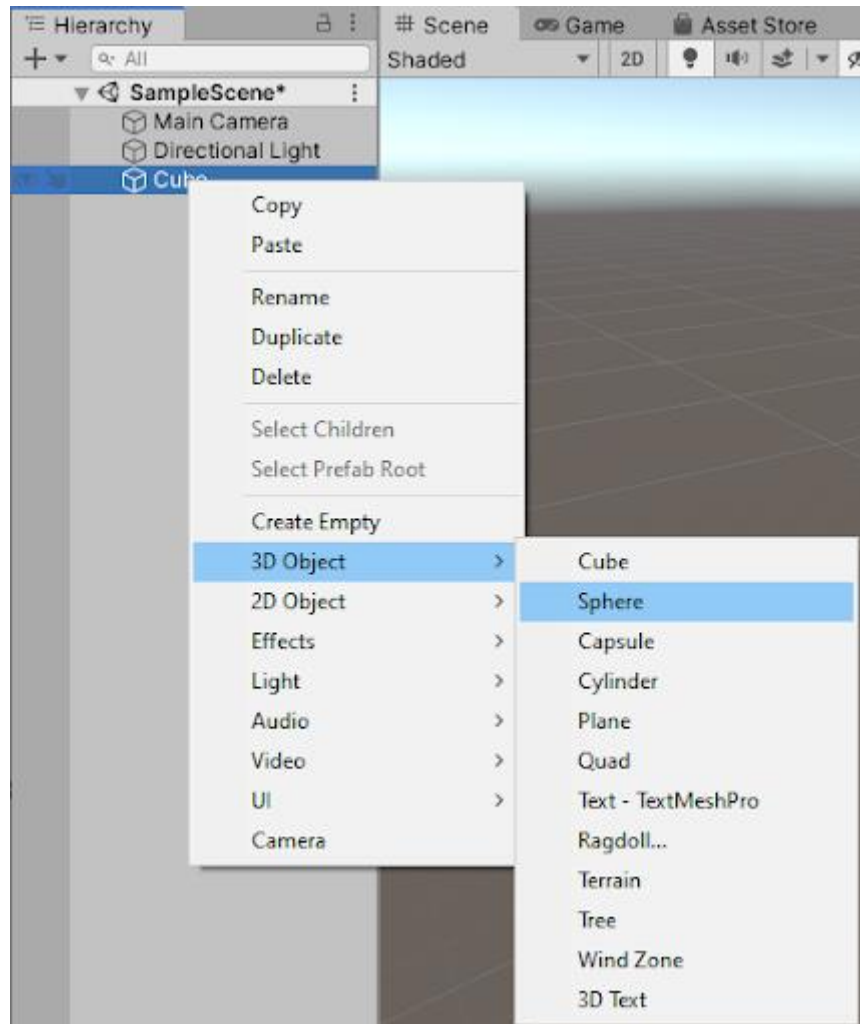
By default, the Move tool is selected, and its Gizmo is located in the center of the cube. The arrows point in the positive directions in each dimension. The dimensions are color coded. You can always get your bearings by looking at the Scene Gizmo in the upper right corner of the Scene view.

4. With the Cube selected, change the Y value of the Scale in the Transform Component to 4. The Cube height will change to 4. Zoom out as needed.

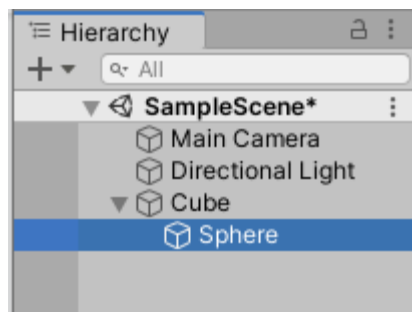


Organize GameObjects in the Hierarchy

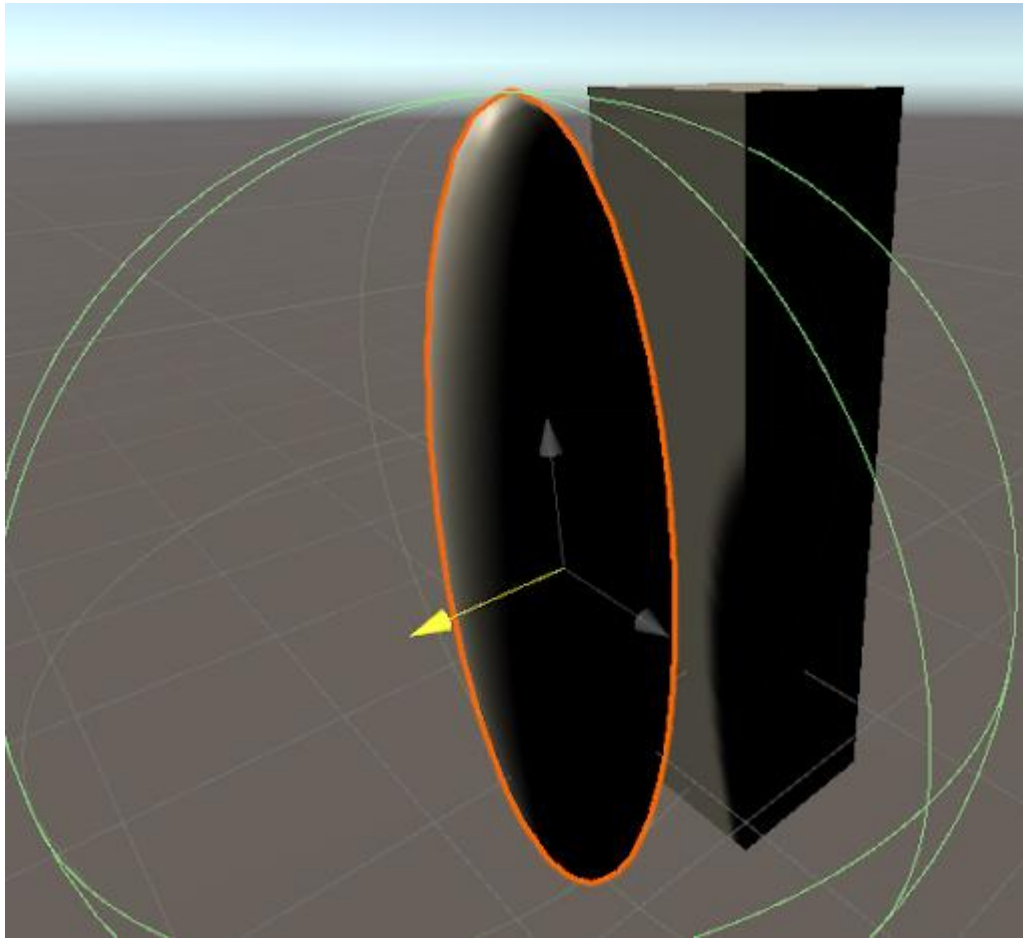
1. Right-click the Cube in the Hierarchy window so that it is selected.
2. Select 3D Object > Sphere.



As shown in the Hierarchy, the Sphere is a child GameObject of the Cube. These two objects are now linked. You can operate on them as a unit.

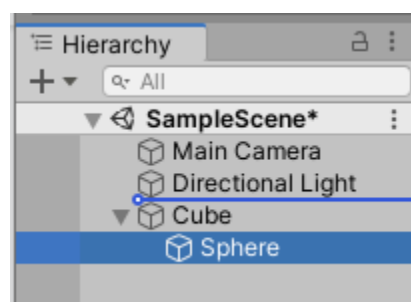


3. Sphere is inside the Cube where you can't see it. Select Sphere in the Hierarchy, and the Gizmo will appear so that you can move it in the Scene view to a location at least partially visible outside of Cube. It will be tall, not spherical, because its initial dimensions were based on its parent.



4. Select the parent object, Cube, and move it. Cube and Sphere will move together. Try scaling and rotating your object, too.

To remove the parent-child relationship, drag the Sphere in the Hierarchy up and all the way to the left, to indicate the root level. Both GameObjects now appear as independent GameObjects in the Hierarchy.

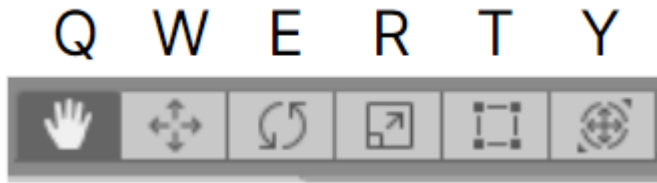


Master the tools

Mastery of the transform and Scene navigation tools is an essential skill in Unity, which allows you to position, scale, and view your GameObjects efficiently in 3D space. With these tools and some practice, you can develop habits to get the results you want quickly and easily.

The keyboard shortcuts for the toolbars correspond to the QWERTY keys at the

top left of a standard keyboard. Using these keys, you can switch quickly between the tools and keep your mouse in the Scene view.



Q: Hand tool, to pan your view

W: Move tool, to select and change position

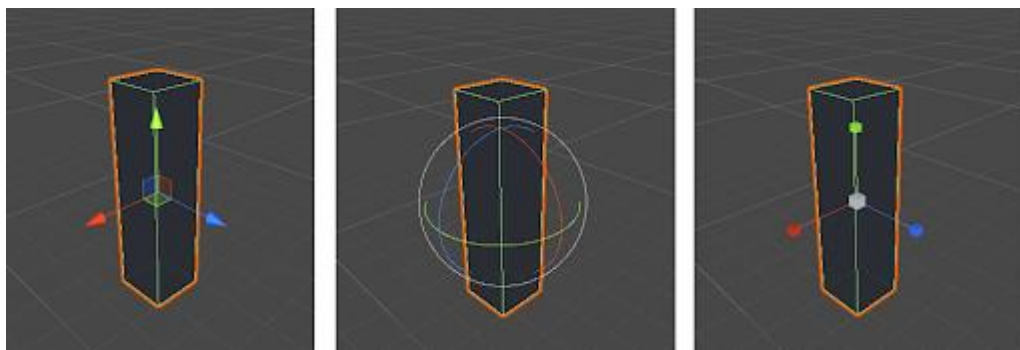
E: Rotate tool, to select and rotate

R: Scale tool, to select and change size

T: Rect Transform tool, to scale in 2D

Y: Transform tool, to move, scale, and rotate with one Gizmo

For each of the transform tools, a Gizmo appears that allows you to manipulate the GameObject along each specific axis. As you manipulate these controls, the values in the Transform Component change accordingly.



Scene view navigation

Select a GameObject in the Scene view	Left-click the GameObject
Move the Scene view camera	Middle-click and drag to your desired view
Rotate the Scene view camera	Right-click and drag to your desired view

Useful shortcuts

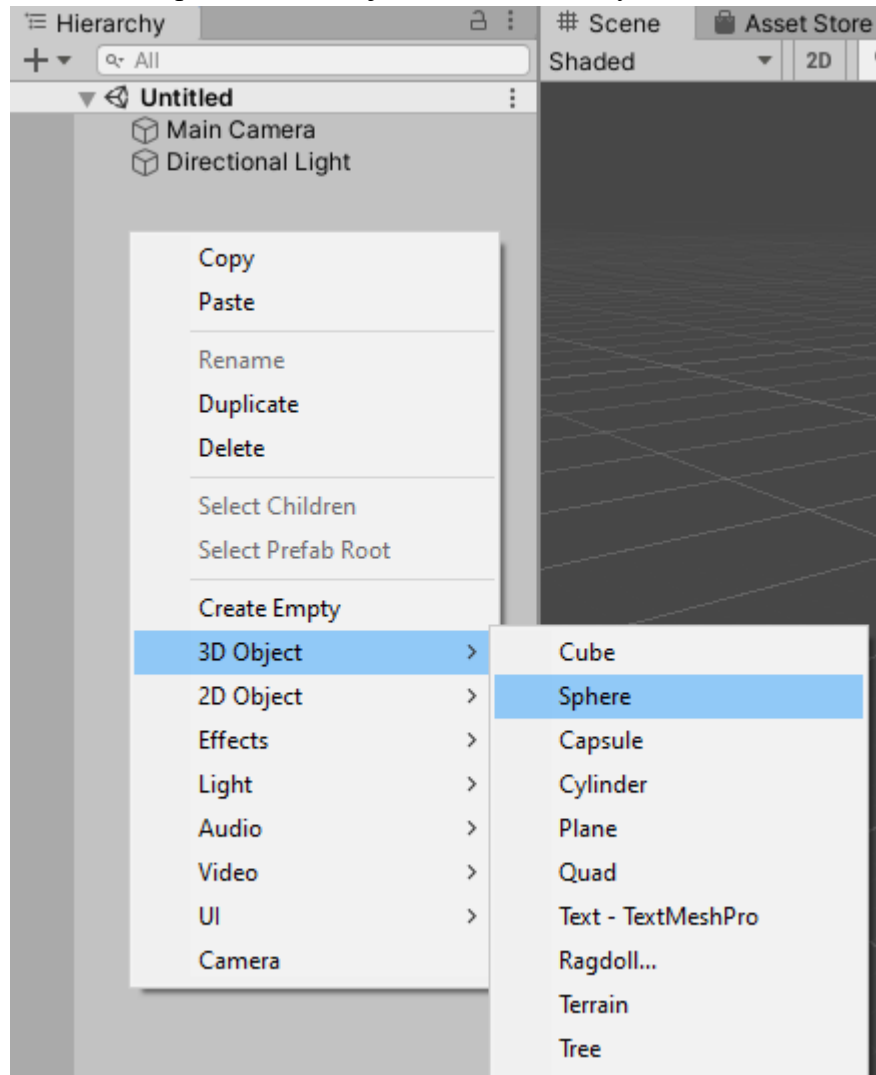
Tool	Icon	Shortcut	Purpose
Hand Tool		Q	Move your Scene around in the window
Move Tool		W	Select items and move them individually.
Rotate Tool		E	Select items and rotate them
Scale Tool		R	Scale your <u>GameObjects</u> up and down
Rect Transform Tool		T	Move, rotate, or scale 2D GameObjects or UI
Rotate, Move or Scale		Y	Move, rotate, or scale 3D GameObjects

You will discover your own system for navigating the Scene view and manipulating GameObjects efficiently. For example, your system might be to rest the fingers of your non-mouse hand on the QWER keys to change tools, rest your thumb on the ALT key to orbit your view of the Scene, and move your index finger to the F key to focus on a GameObject as needed.

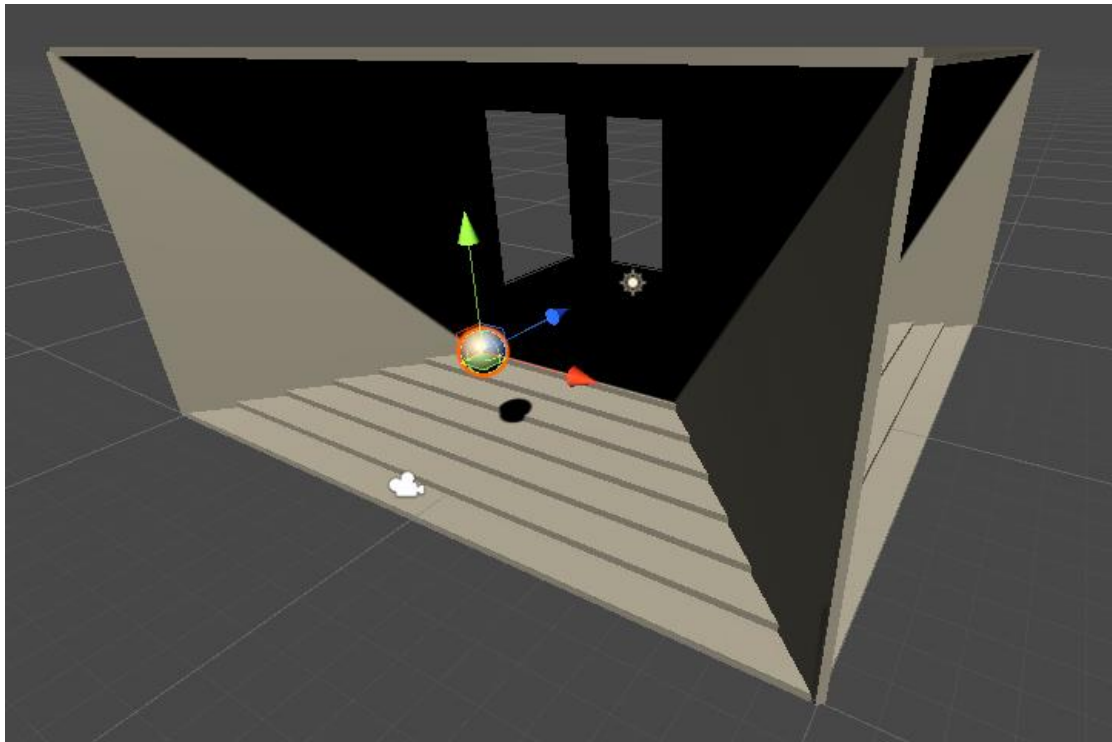
3.Add components to 3D GameObjects

Create sphere in scene

1. Create a new Sphere GameObject in the Hierarchy.

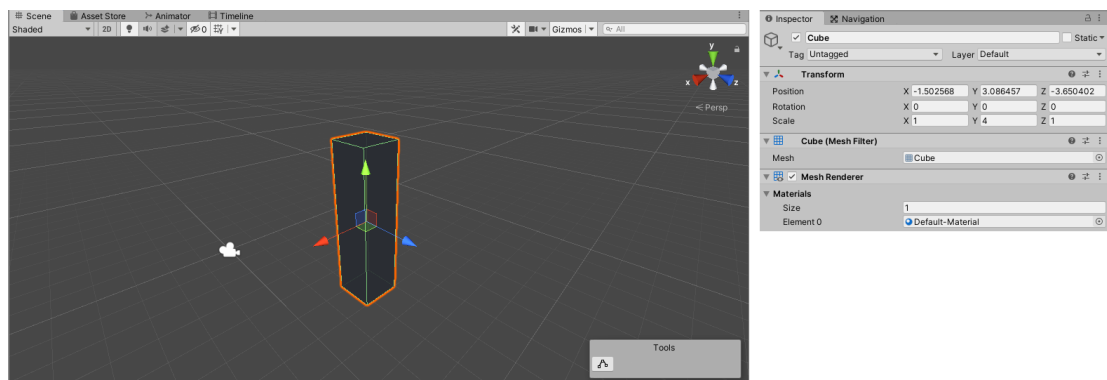


2. Move the sphere to the space inside the Monument, so that it is positioned in “mid-air” above the steps. You can use the Transform tools, the Transform Component in the Inspector, or both.



Position the Main Camera

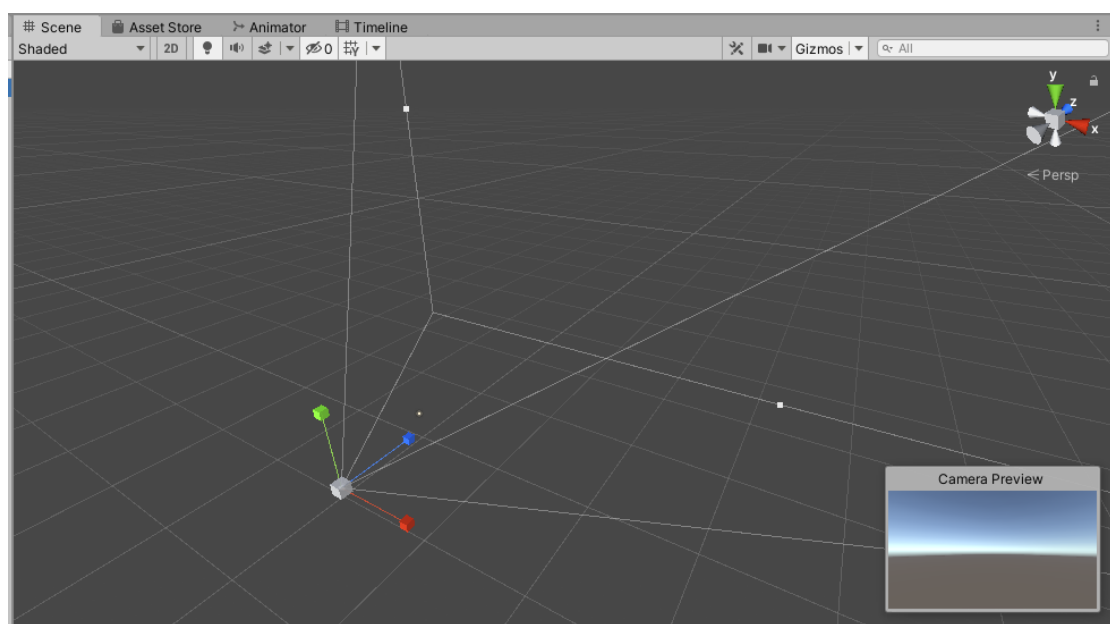
In every new Scene, there is a camera like the one shown below.



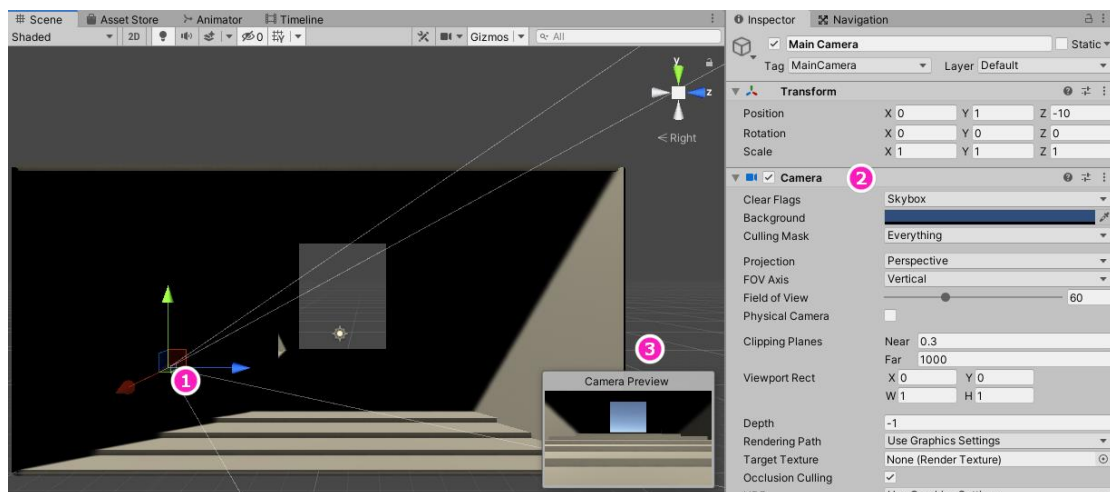
This camera, named Main Camera in the Hierarchy, captures and displays your Scene to the player as it appears in the Game view. While the Scene view is where you build Scenes, the Game view is a preview of the way your game appears to the player.

The camera in your Scene is a GameObject like any other. When you select it, you will see its Transform Component in the Inspector window. You can change these settings as you would with any other GameObject. Note that scaling a camera will have no effect.

When you select a camera, you see the outlines of a pyramid-like shape called the frustum. The frustum shows you what part of your Scene the camera is viewing. Anything outside the frustum is not visible to the camera.



When a camera is selected (1), you can expand the Camera Component in the Inspector (2) to open a **Camera Preview** window (3) that shows you what is visible to the camera in the Scene.

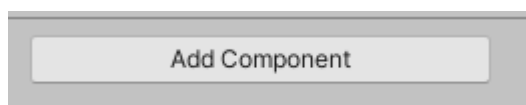


Move the camera to a position where it can “see” the ball and the steps below it.

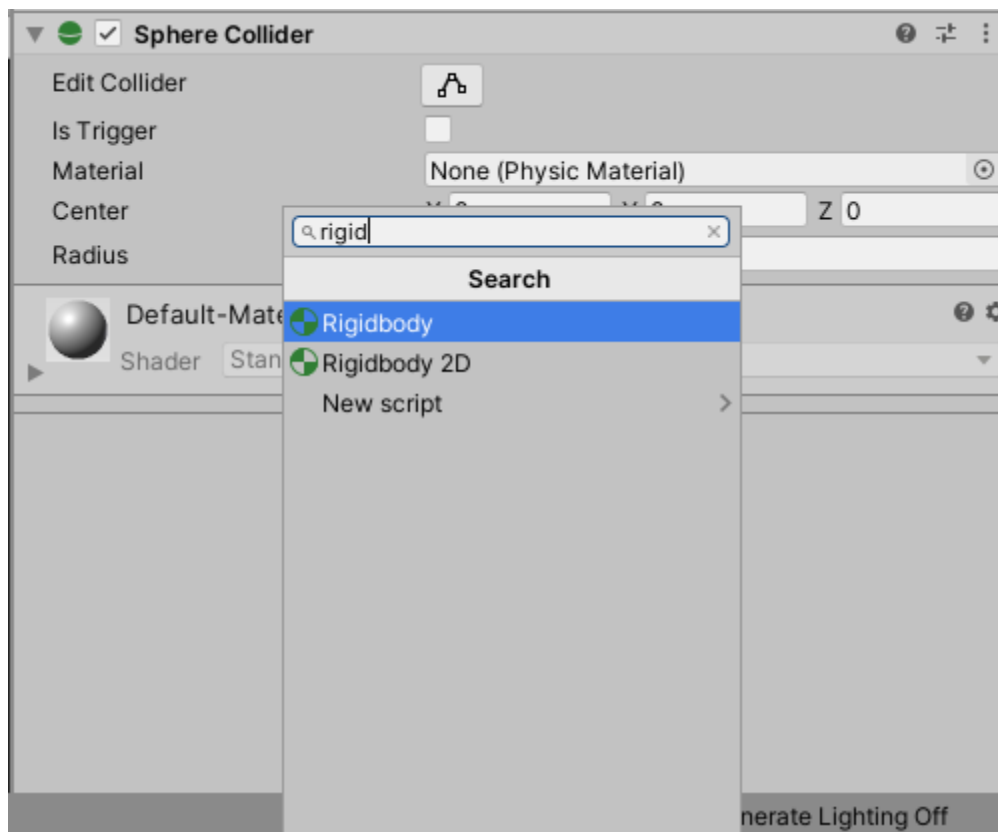
Make the sphere solid

To add the Rigidbody Component to the Sphere GameObject:

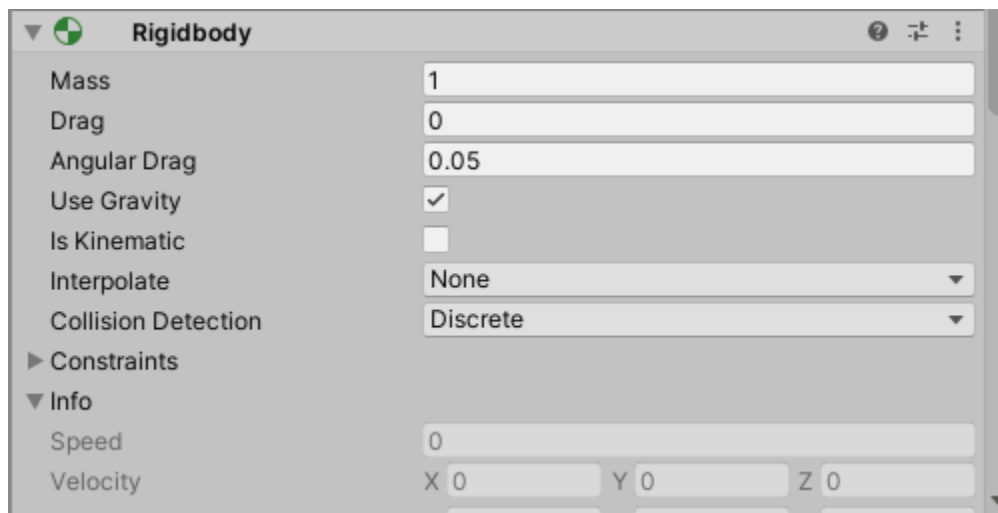
1. In the Hierarchy window, select the Sphere.
2. Select the Add Component button in the Inspector window.



3. Use the search bar to find a Rigidbody component and select it. Note that there are two options available, Rigidbody and Rigidbody 2D. Make sure you select the Rigidbody component, not the 2D one.



4. In the Rigidbody Component in the Inspector, select Use Gravity if it is not already selected.



5. To see the effect of these properties, start the game by pressing the Play button, which will display the Game view.

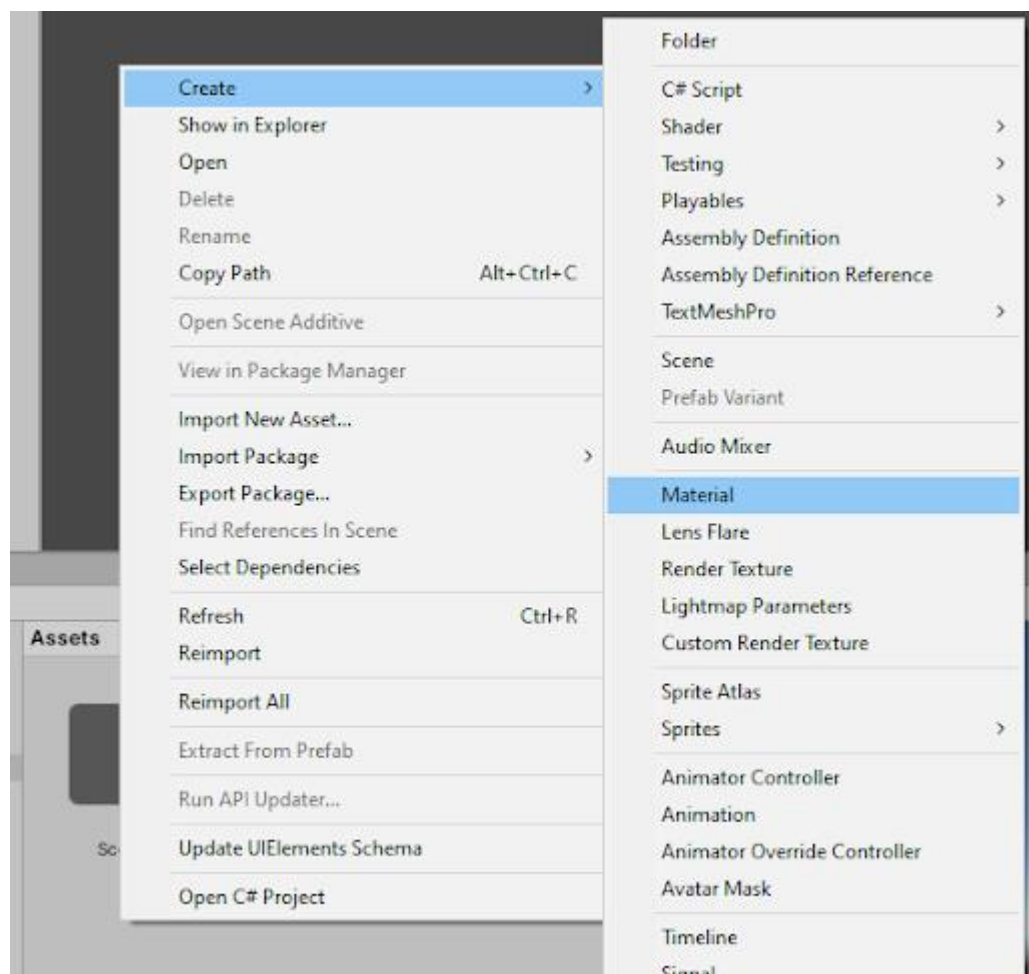
The ball falls and rolls down the steps. If it does not roll, try moving it to a new position using the transform tools and run the game again.

4. Create effects for 3D GameObjects

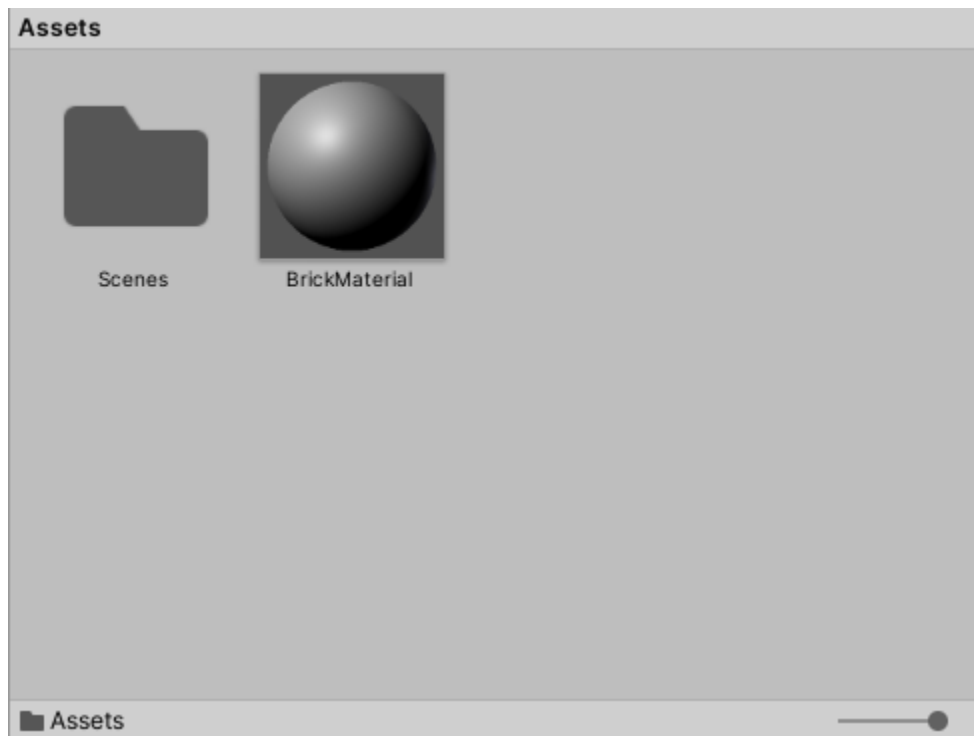
Import an image file

1. To make your Unity windows match the examples below, right-click the tab of the Project window and select Two Column Layout if it is not selected already. The folders of your project appear in the left column, and the contents of the selected folder appear in the right column.

2. Right-click the Assets folder in the Project window and select Create > Material.

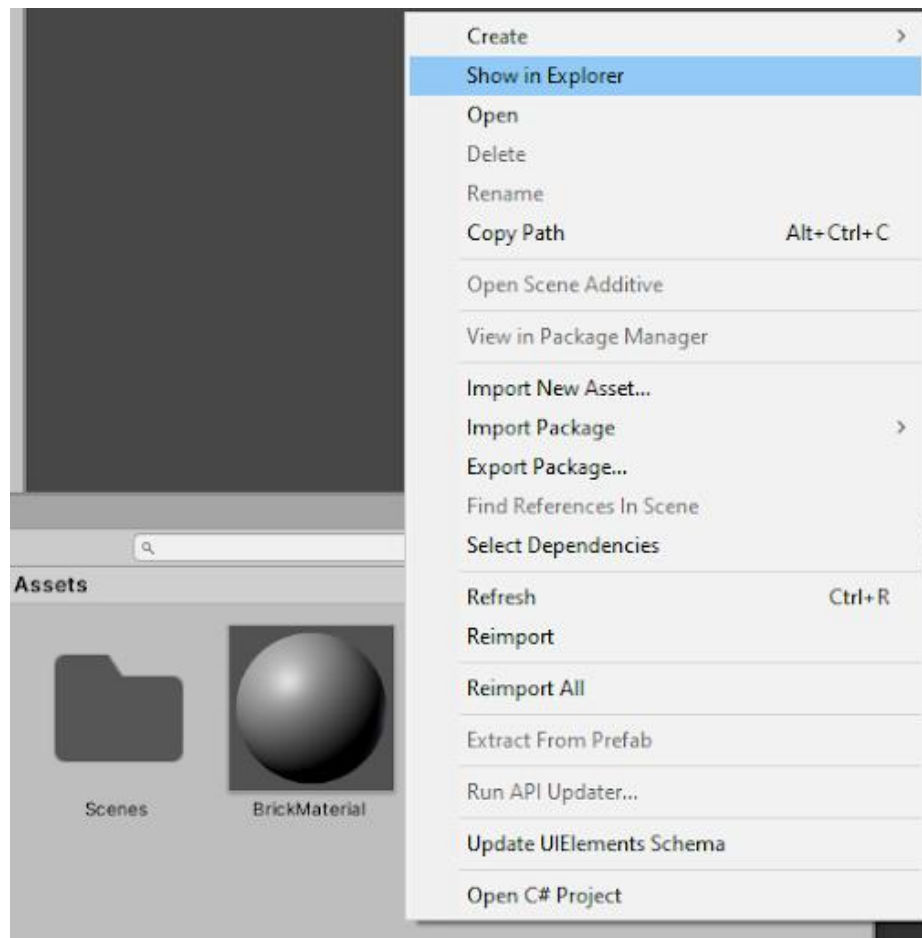


3. Rename the new Material as BrickMaterial. The preview of the new Material is a 3D sphere, so that you can see how light interacts with the Material at many angles. When you select the Material in the Project window, a more accurate preview appears in a resizable pane of the Inspector.

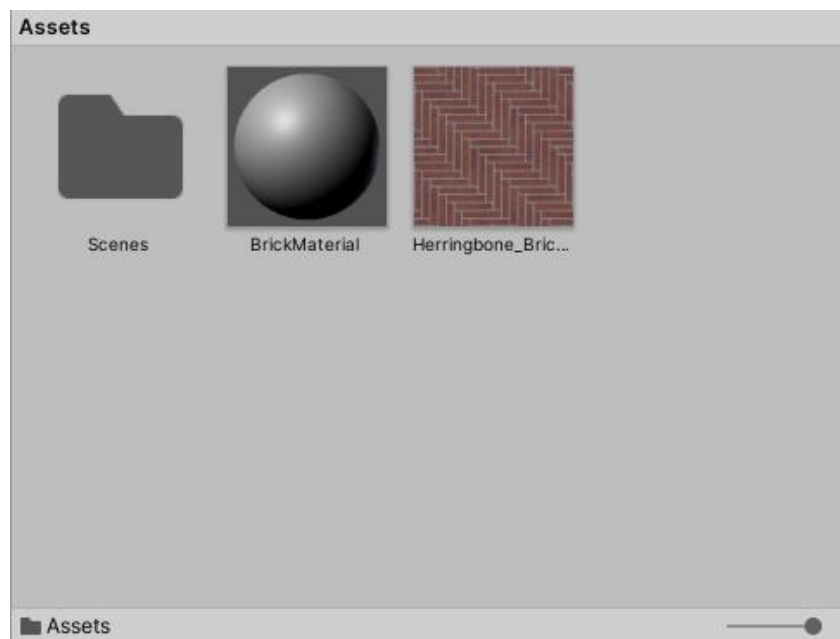


4. Download the file `Herringbone_Brick_baseColor.png` from the Files listed at the top of this tutorial.

5. To import the file into the project, you can simply move the downloaded file to the Assets folder for this Unity project, which is represented in the Project window. To see the contents of the Assets folder in your file explorer, right-click the Assets folder and select Show in Explorer.

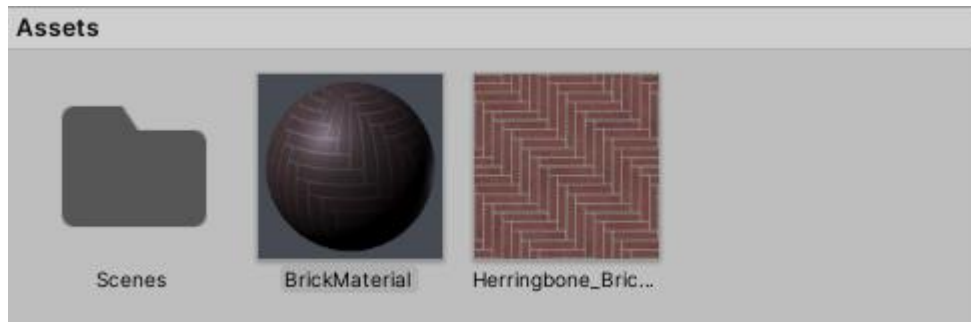


6. In your computer's file explorer, move the `Herringbone_Brick_baseColor.png` file into your Assets folder. When you return to Unity, you will see its icon.

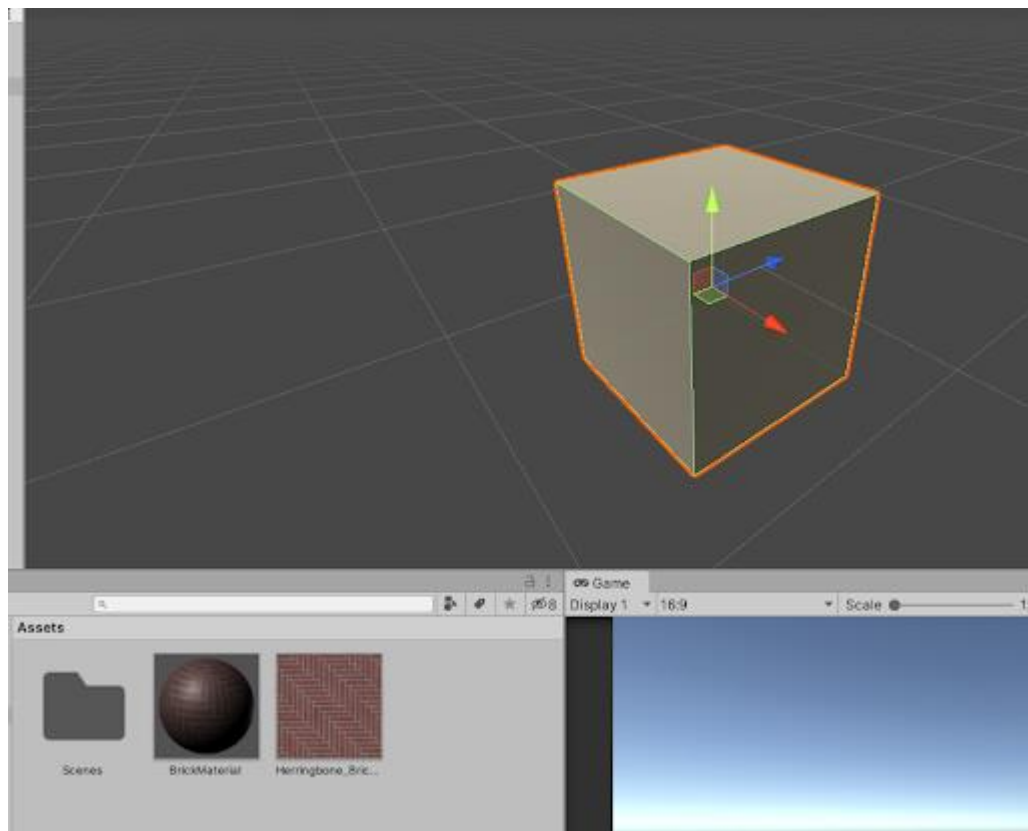


Create the Brick Material

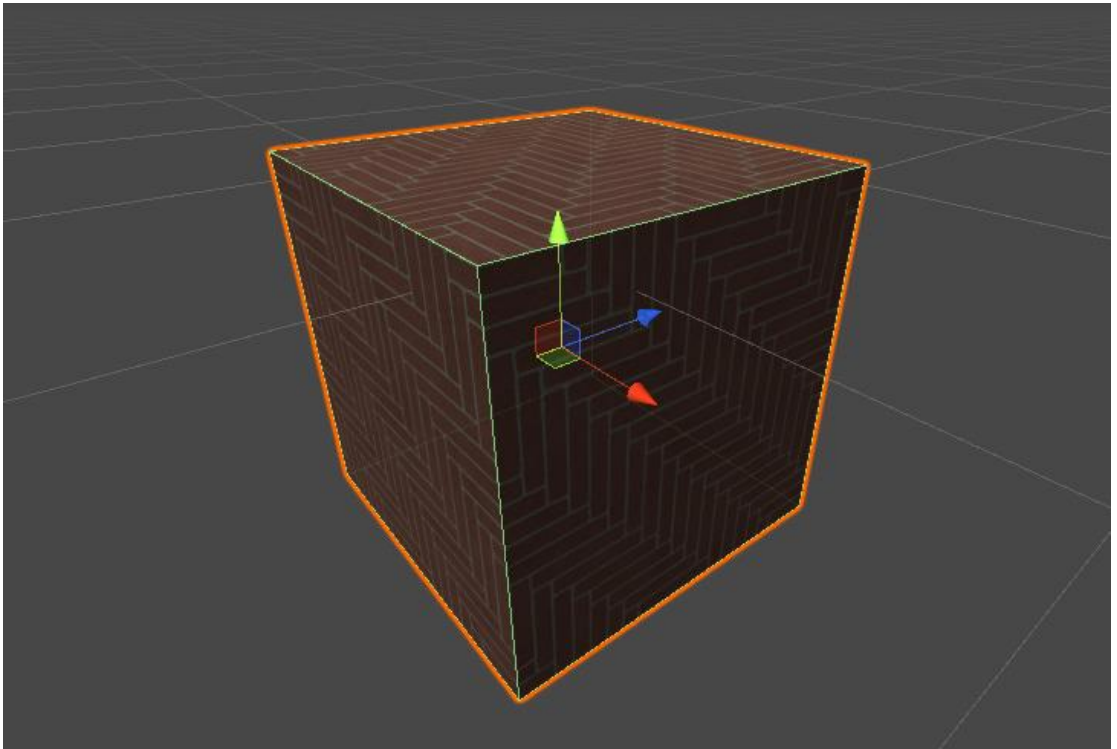
1. Select the BrickMaterial.
2. In the Inspector window, select the circle icon next to the Albedo property. Albedo encompasses the reflective properties of the Material, based on a solid color or an image.
3. Select Herringbone_Brick_baseColor from the list of images in your project.
The image is now applied to the material as the Albedo map, and will appear in the Material preview.



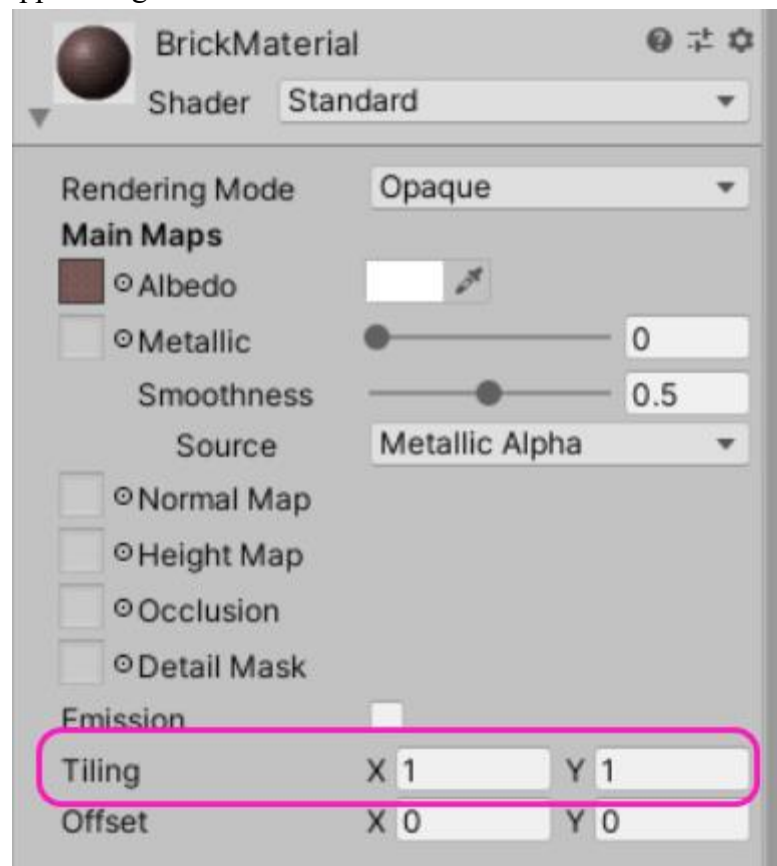
4. Create a new cube GameObject in the Scene.
5. Drag the material onto the cube in the Scene view.



6. The cube now displays the brick Material.



7. To change the scale of the Material, select the cube, open the BrickMaterial Component in the Inspector, and change the Tiling X and Y properties. Smaller numbers make the bricks appear larger.

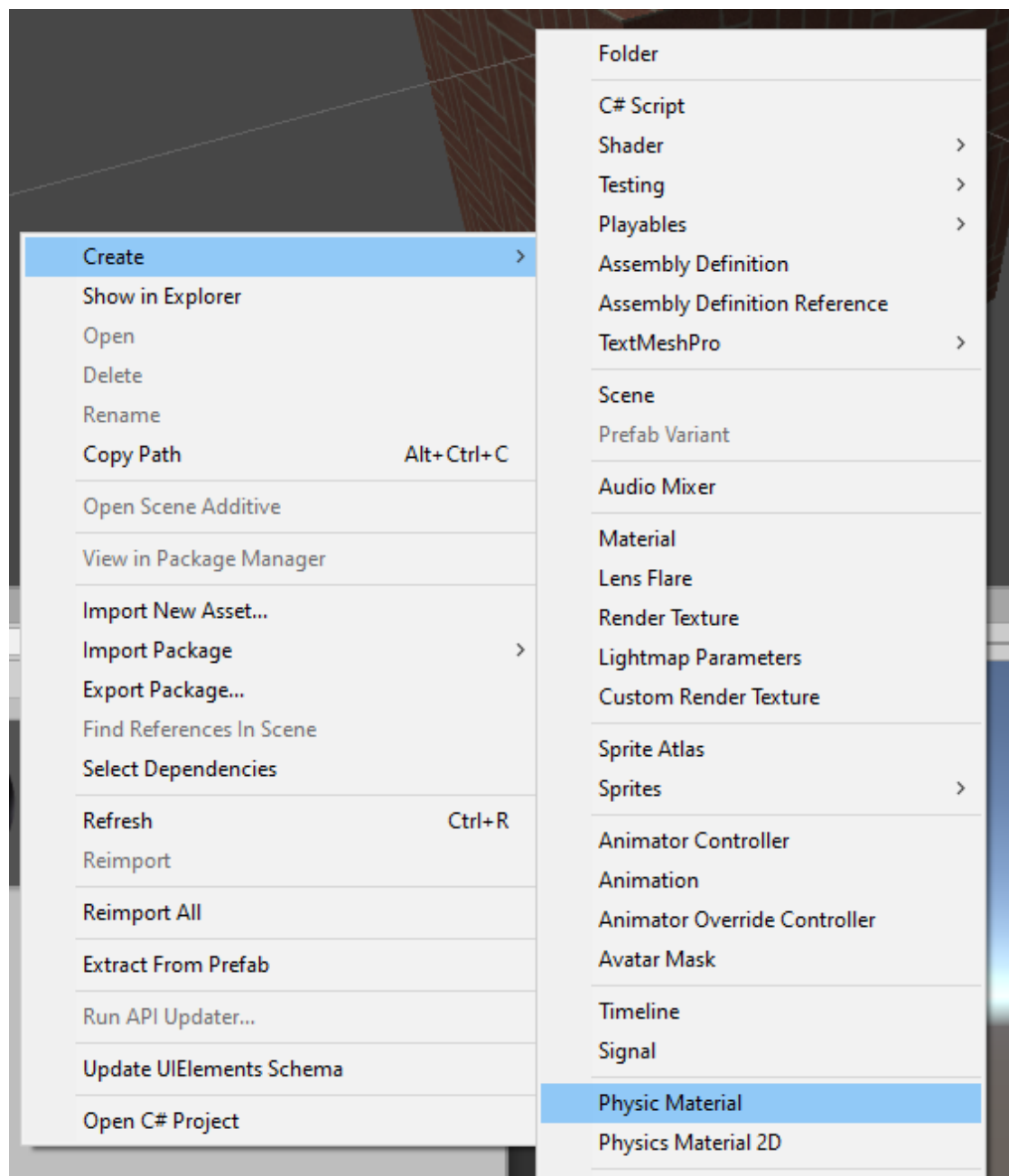


Adding a Physic Material

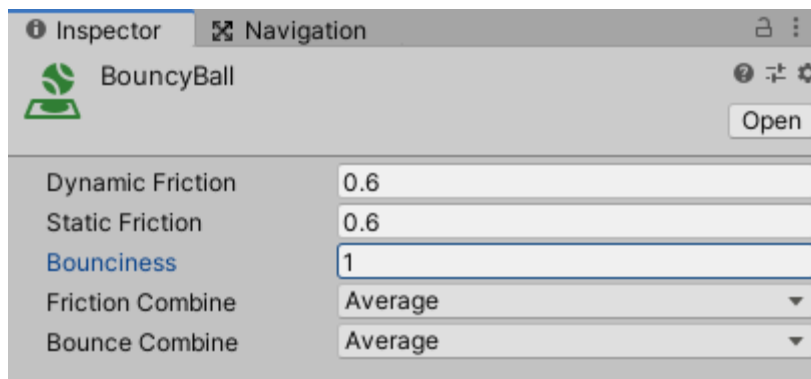
Earlier, you added a Rigidbody component to a GameObject to give it physical properties. You can also add additional physical characteristics with another type of Material called Physic Materials. With Physic Materials, you can make an object bounce and change its friction and drag properties. These properties take effect when the object is under the effects of gravity. Physic Materials don't change the visual properties of the GameObject's mesh.

1. In the Project window, create a new folder called Materials. Right-Click in the Materials folder in the Assets window and select Create > Physic Material.

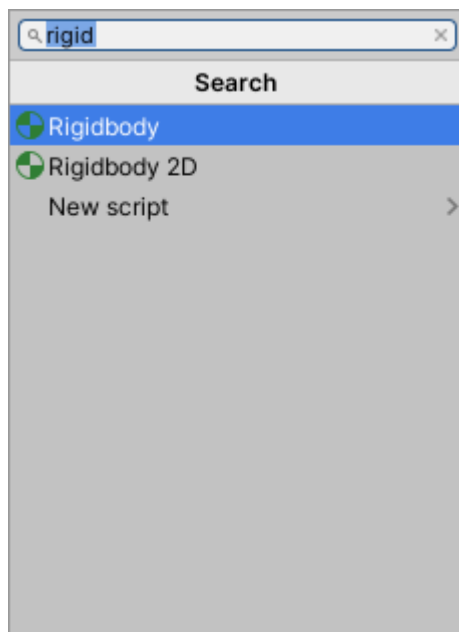
2. Rename the material to BouncyBall.



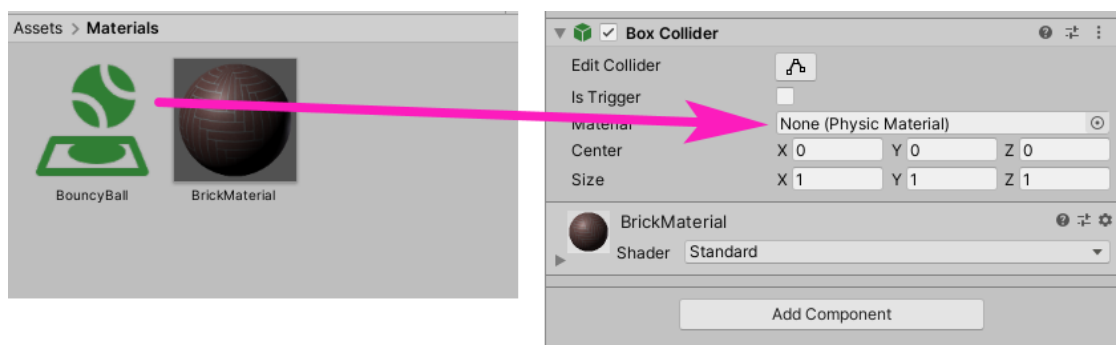
3. Select the new Physic Material.
4. Change the Bounciness value to 1 in the Inspector window.



5. Select the cube. In the Inspector window, select Add Component and add a Rigidbody component.



6. Your cube should already have a Box Collider component added to it. This component is automatically added when you create the cube in the scene. Drag the new Bounce physics material you created earlier into the Material slot in the Box Collider component.



7. Run the game. Your cube should now fall to the platform and have a bit of bounce

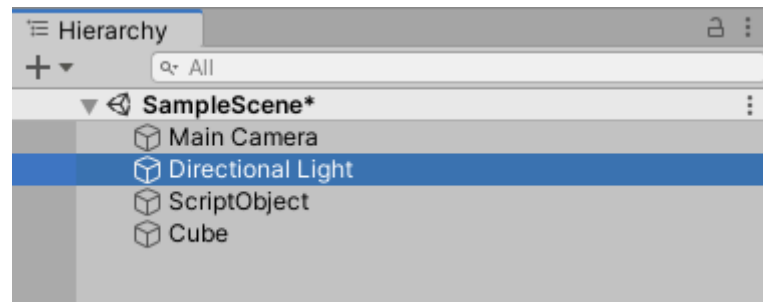
when it hits the floor.

8. Experiment with your bouncy cube: try rotating it, duplicating it, and making cubes fall on top of each other!

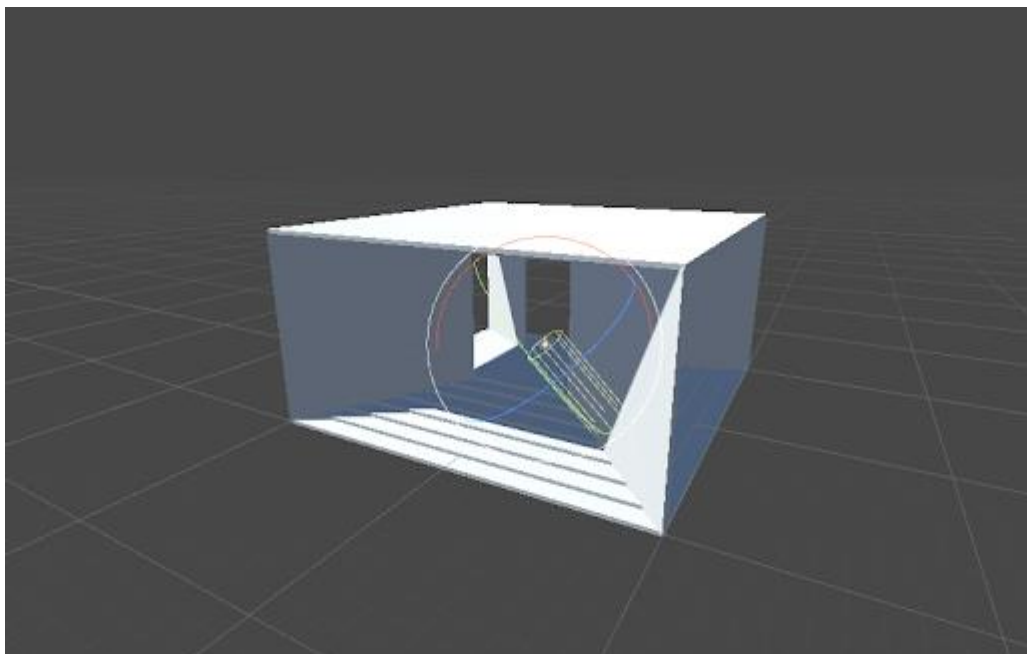
Adjusting the Lighting in the scene

In a new 3D Scene, an object named Directional Light is included to simulate sunlight. You can manipulate this light with transform tools like any other GameObject.

1. Select the Directional Light in the Hierarchy window.



2. Change the rotation of the light using the transform tools Move (shortcut key “w”) and Rotate (shortcut key “e”). Directional lights simulate a light source that bathes the entire scene in light coming from a specific direction. Changing the position of the light will not have an influence on the scene but changing the rotation will change the angle of the light in the Scene. Take note of how the sky and ambient light changes as you change the angle of the light.



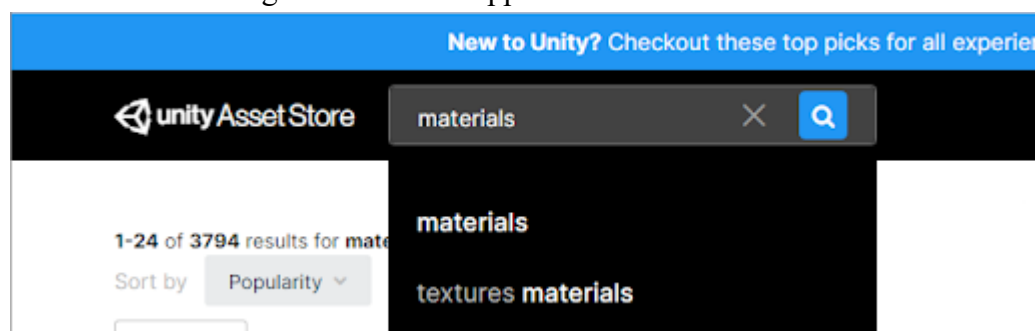
5. Get 3D effect

Get assets from the Asset Store

1. Go to the Unity Asset store (this link will open in your default browser) and log in with your Unity ID.

Hint: Anything you acquire through the Asset Store will be linked to your Unity account and will be available in the Editor as long as you are signed in with the same Unity ID.

2. In the Asset Store search tab, type “materials” to search for available Material assets. All assets with the tag of “materials” appear.



3. Use the checkbox on the right of the store screen to view only the free assets.



4. Select the Yughues Free Ground Materials (or something similar).



NOBIAX / YUGHUES

Yughues Free Ground Materia...

★★★★★ (1581)

FREE

5. Check the Unity version of the assets you have selected. It is critical to make sure that the assets will be compatible with your version of Unity. On the Details page, you will see all the information about the asset, including the Unity version compatibility.

Yughues Free Ground Materials



Nobias / Yughues



5 | 129 Reviews

FREE[Add to My Assets](#)

License

Extension Asset

File size

132.1 MB

Latest version

1.0

Latest release date

Mar 31, 2015

Support Unity versions

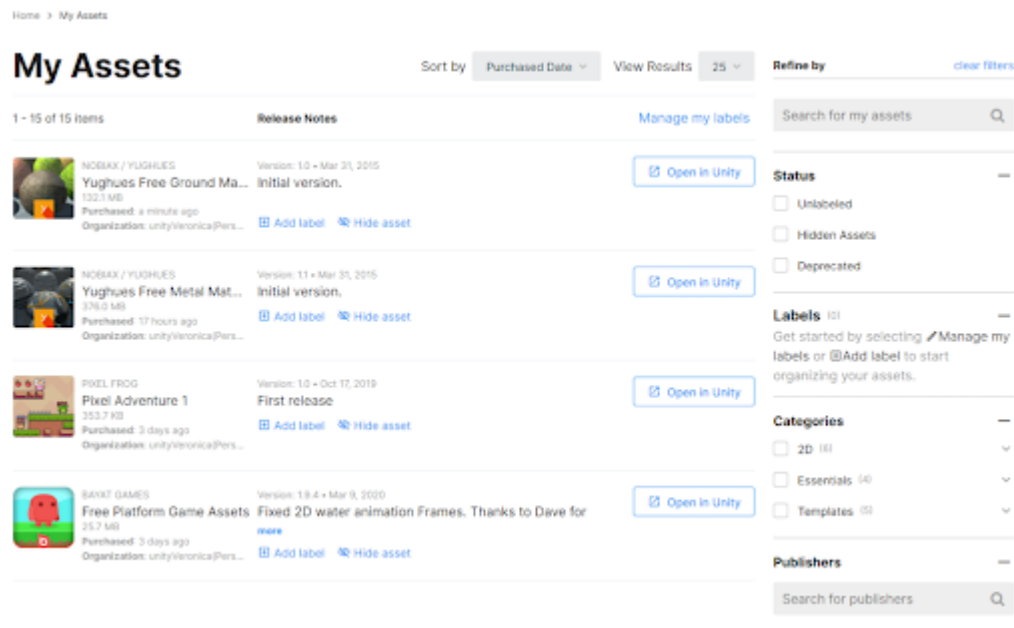
4.2.2 or higher

6. Select Add to My Assets to begin the process of procuring the asset package you have chosen. You will see a confirmation message like the one below.



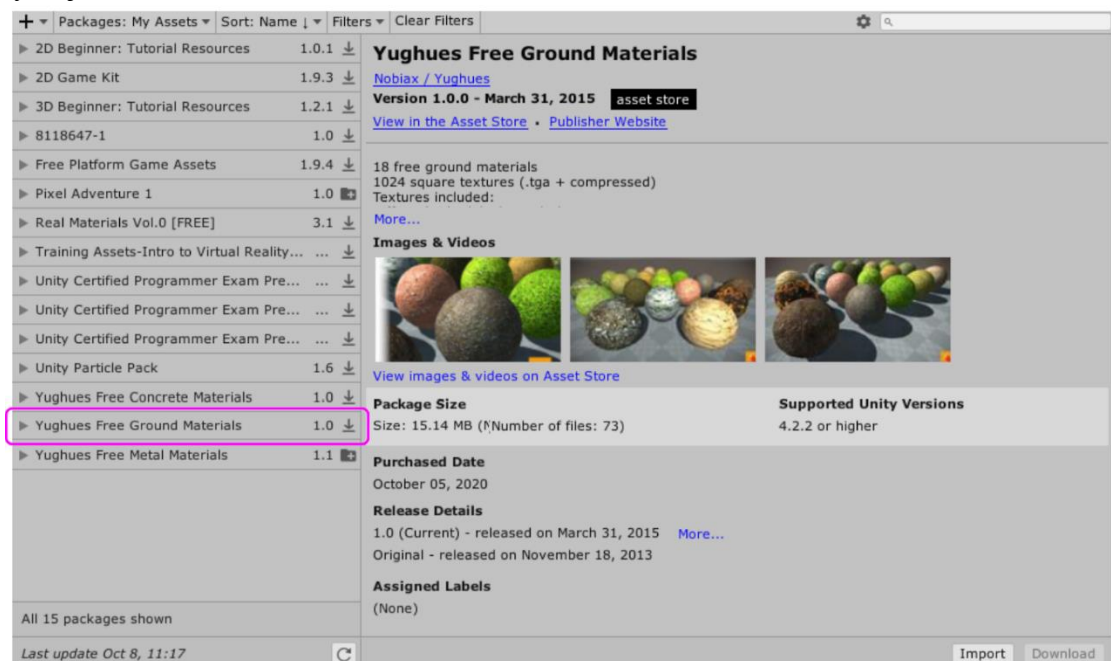
Import assets from the Asset Store

1. Select Go to My Assets to view your My Assets page.



You can always return to the My Assets page in the Asset Store to view the assets you have selected and purchased

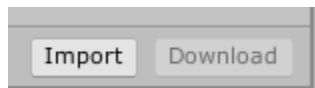
2. Select the Open in Unity button for the assets you just selected. If your computer prompts you to open the Unity Editor, select the option to do so. The Unity Editor will open with the My Assets section of the Package Manager displayed, including the asset package you just selected in the Asset Store.



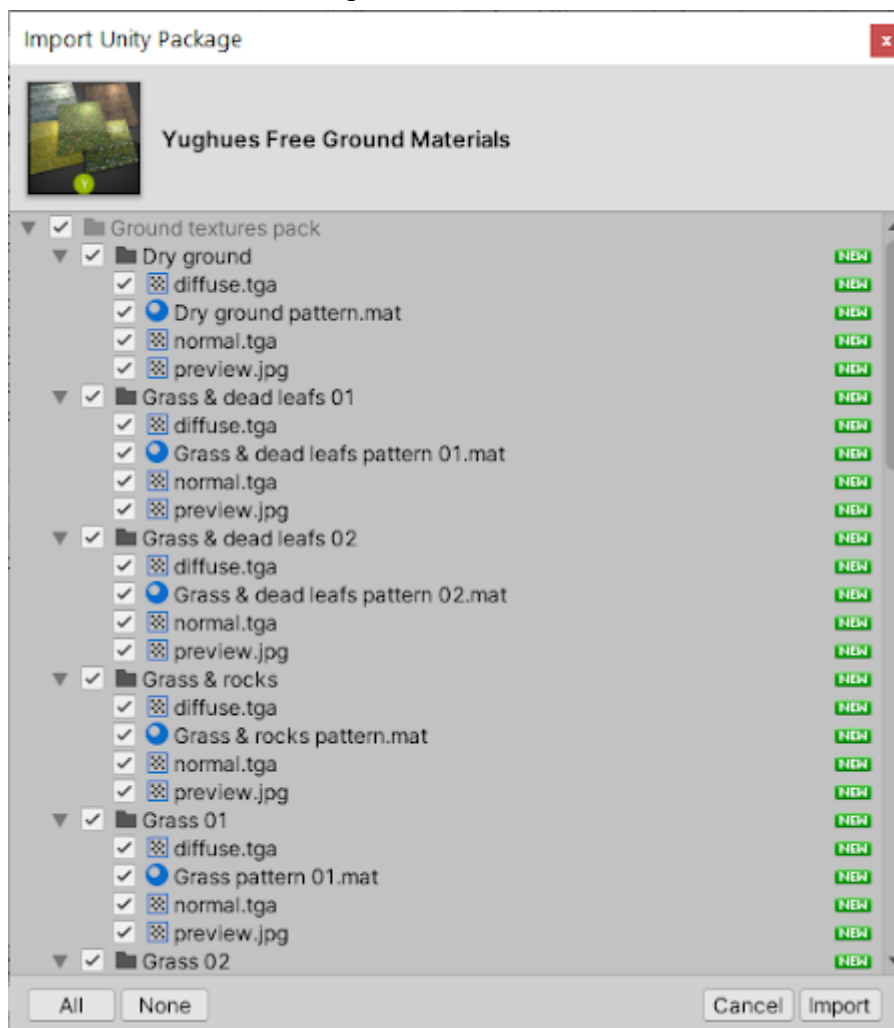
3. Select the package from the Asset Store, and select the Download button at the lower right corner. Once you have downloaded an asset package from any project, you do not have to download it again for use in another project, and this button will not be highlighted.



4. When downloading is complete, select the Import button.

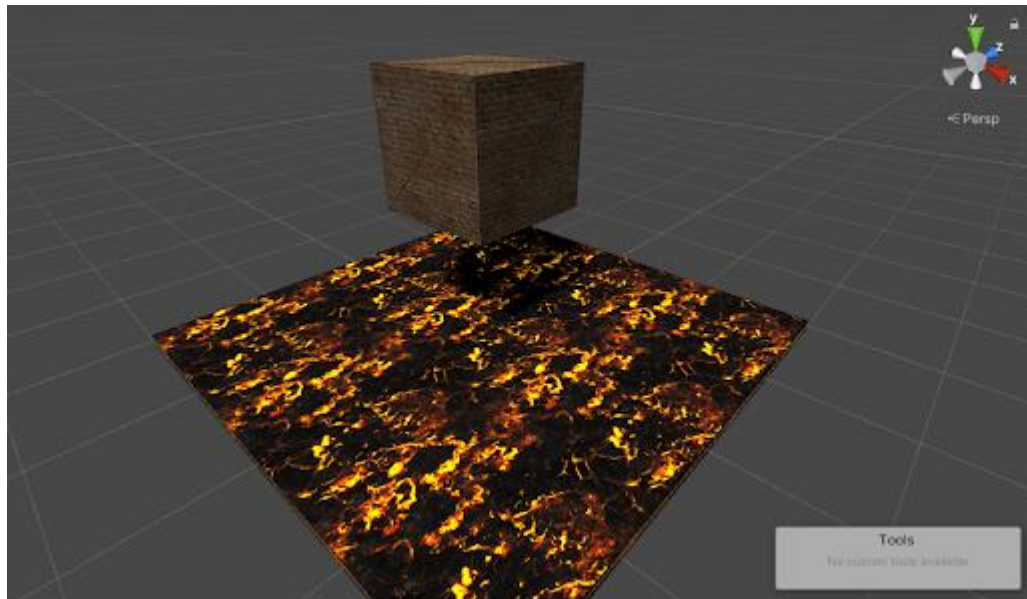


5. An import options panel will open. Use the checkboxes to import everything or only specific assets in the package. For this example, select everything and select the Import button at the bottom of the panel.



The package will be extracted and placed in your project's Assets folder.

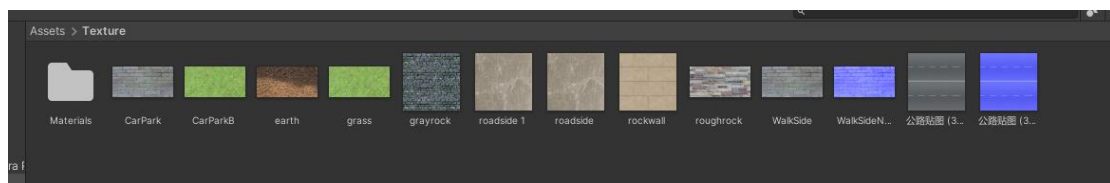
6. Create a Plane to act as the ground for your Scene. Select one of the Materials and drag it onto the ground platform. The Material is now added to your GameObject.



Develop my own project

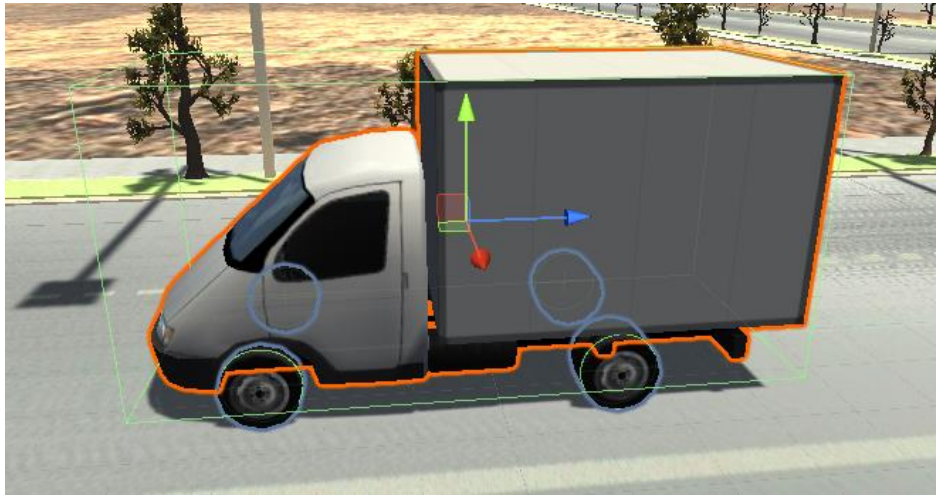
1.Introduce of the texture

This is the texture I used in my program:



2.Introduce the model

The model of the truck:



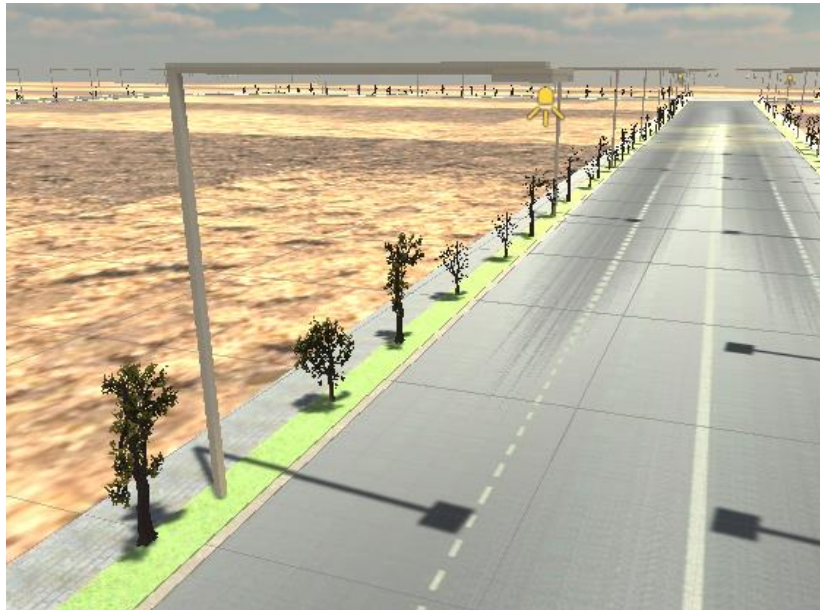
The model of the trees:



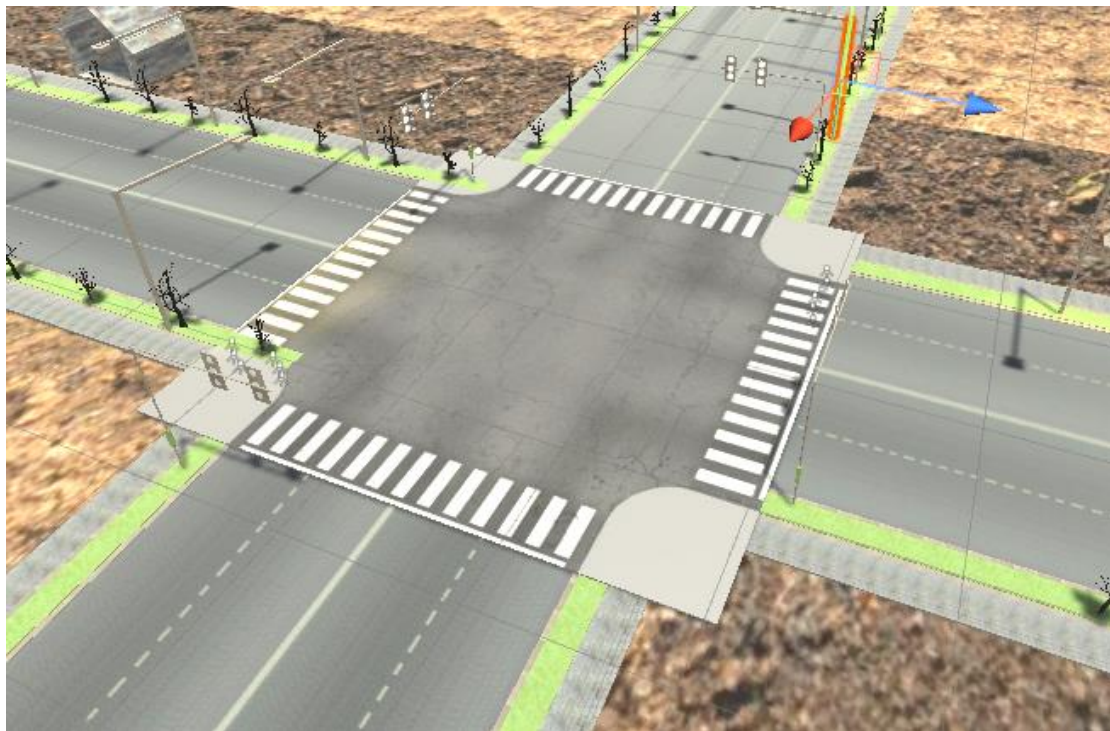
The model of the traffic light:



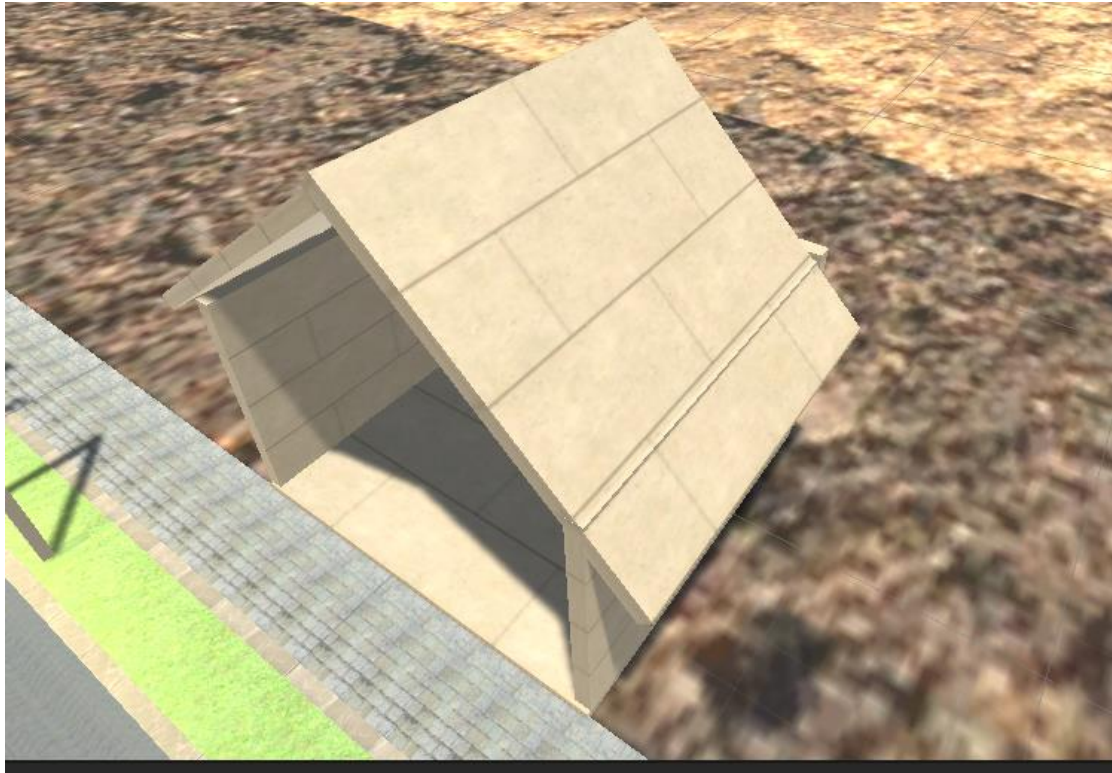
The traffic of the light:



The model of the cross:



The model of the small garage:



Summary

During the study of the development , I realize the Unity engine is a very powerful 3D and 2D development tool , and there are very much technical point to learn and achieve

Also the script of the development is the soul of a game , properly use the trigger and script can make the game functional .