

Game Programming Using Unity 3D

Lesson 1: Introduction To Unity



Ferdinand Joseph Fernandez

Chief Technological Officer, Dreamlords Digital Inc.

Admin and Co-founder, Unity Philippines Users Group

September 2011



This document except code snippets is licensed with
Creative Commons Attribution-NonCommercial 3.0 Unported



All code snippets are licensed under CC0 (public domain)

What Is Unity 3D

Unity 3d is a complete authoring solution to create video-games, or other interactive content (training simulations, architectural visualizations, etc.) created by the company known as Unity Technologies.

Unity is available for indie developers, hobbyists, and students with a free version. The free version is usable for commercial purposes with no upfront costs or royalty fees whatsoever. Unity is also available with a paid premium version, designated “Unity Pro”, meant for “AAA” style, high quality games such as those found in console games.

Unity has 500,000+ registered users worldwide, including companies such as Cartoon Network, Coca-Cola, Disney, Electronic Arts, Microsoft, NASA, the US Army, Warner Bros. and more.



Information

Unity Pro comes with all the “bells and whistles” that the free version doesn't have, like post-processing effects, real-time shadows, video playback, streaming assets (background loading), the ability to load C/C++ plugins, and more.

However, the license comes with a hefty cost of USD 1,500.00 per programmer in your team.

Find out more on <http://unity3d.com/unity/licenses>

That being said, you can just as well create a fun and enjoyable game without needing the Pro features.

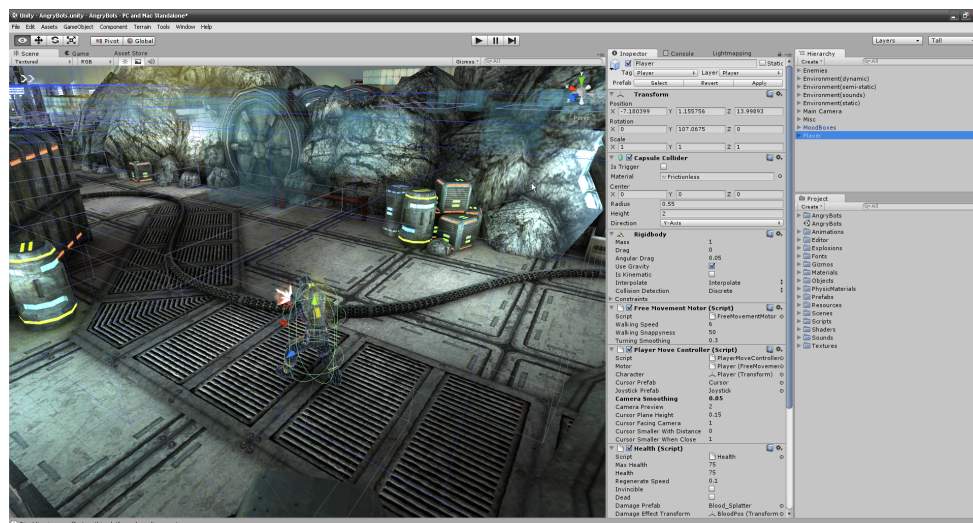


Illustration 1: Unity comes with an IDE for developing games visually, in a rapid-prototyping approach.

Where To Get Unity 3D

The latest version of Unity is always available at <http://unity3d.com/unity/download/>. When you have Unity installed and are connected to the Internet, it will always notify you at startup when a new version is available.

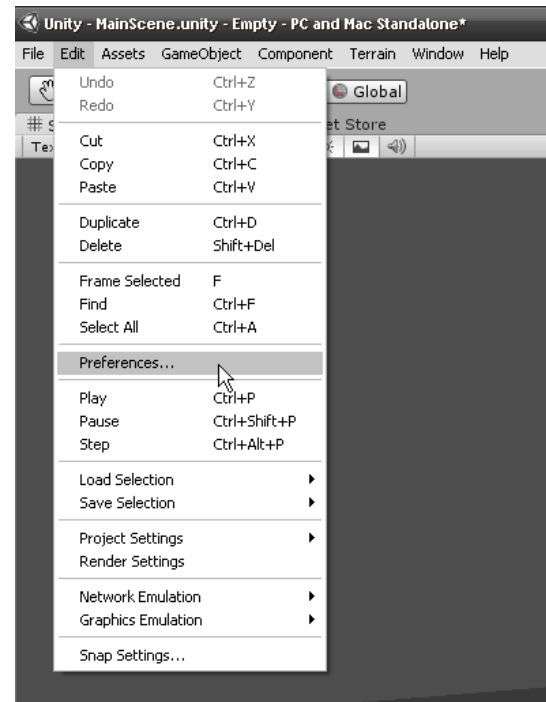
Opening a Unity Project

Start Unity. As of version 3.4, by default it will open the sample project “Angry Bots”.

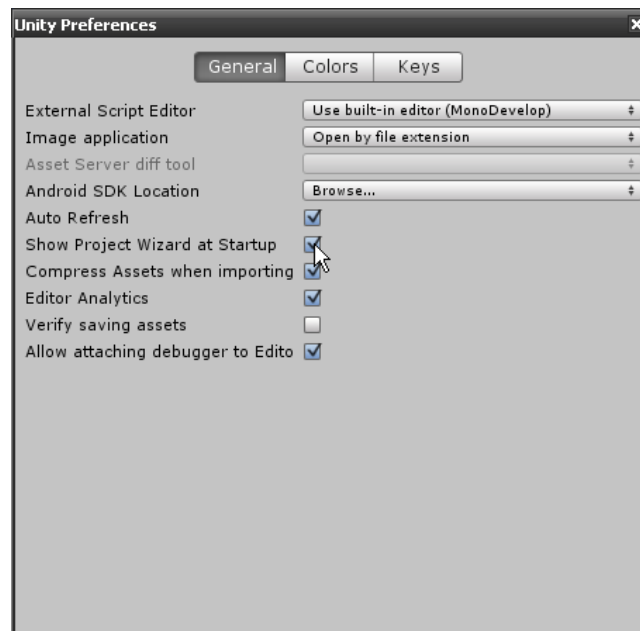
You may be greeted with the “Welcome To Unity” Welcome Screen. We won't be needing this, so uncheck “Show at Startup” at its lower-right corner and close that window.

We don't want Unity to open a project immediately upon startup, so we change that in the preferences.

Go to the preferences window by choosing it from **Edit > Preferences...**



Then check “Show Project Wizard at Startup”.



After checking it, close the Preferences Window. Now, whenever you start Unity, it will open a dialog window to let you choose which project you want opened.

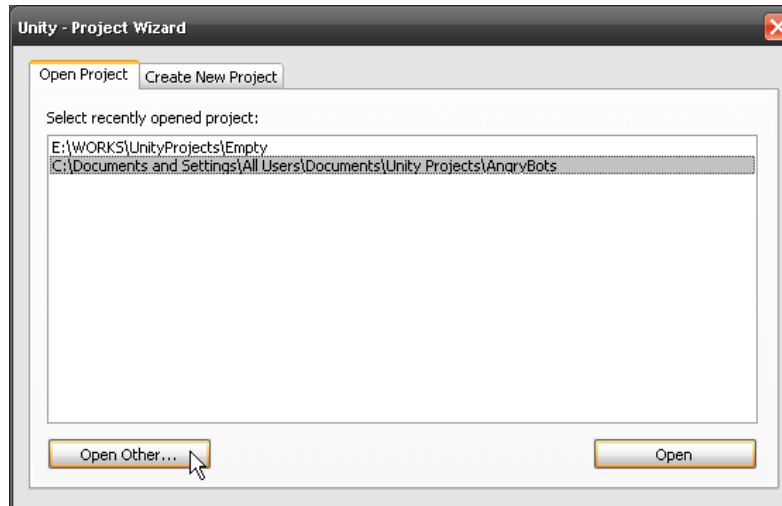
Now we'll open the Lesson 1 project. If you haven't downloaded the LessonAssets zip file, do so now. (Can be found at <https://dl.dropboxusercontent.com/u/25260770/UnityLessons/index.htm>). It contains all the necessary files we'll be needing for all the lessons. Inside that zip file is the Lesson 1 folder that we need to open.

There are two ways to open a Unity project.

Opening a project within Unity

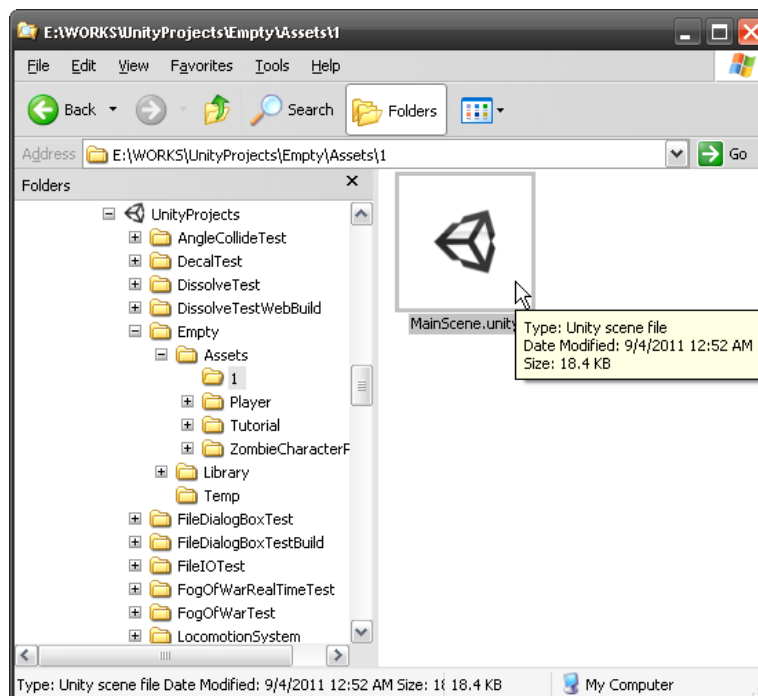
From the top menu bar, choose **File > Open Project...** This will open the Project Wizard Dialog.

Choose the “Open Other...” button and locate the Lesson 1 project within your system. **Just choose the root folder of the project**, not a file within the project. After clicking the OK button, Unity will momentarily close and open your selected project. The next time you open Unity, your recently opened projects will be listed out so you can easily open them.



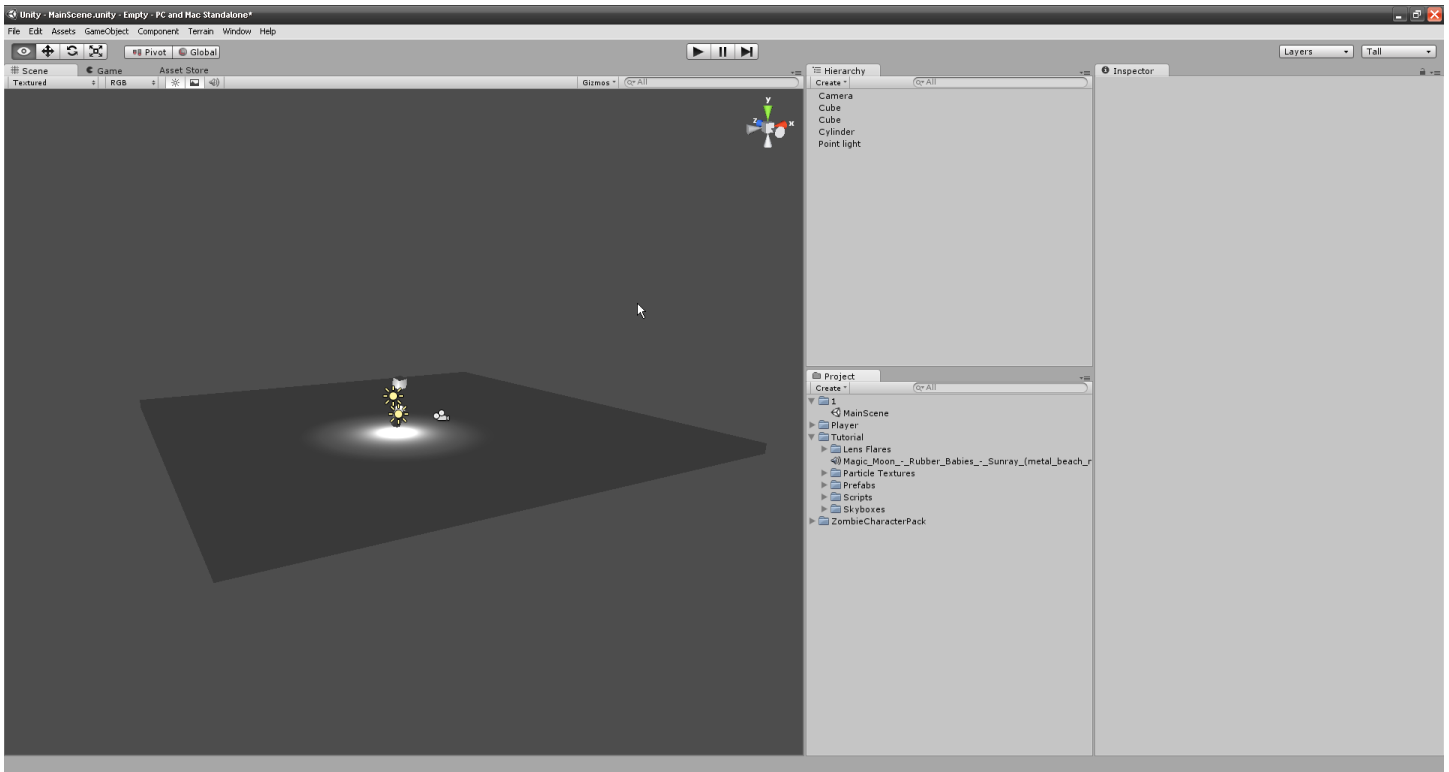
Opening a project from your OS

Open Windows Explorer (Windows) or Finder (Mac). Go to the Unity project you want opened. Inside should be at least two folders named “Assets” and “Library”. Go inside “Assets”. If you are opening the Lesson 1 project, you should see a file named “MainScene.unity”. **Simply double-click this file** and Unity will launch with this project opened.

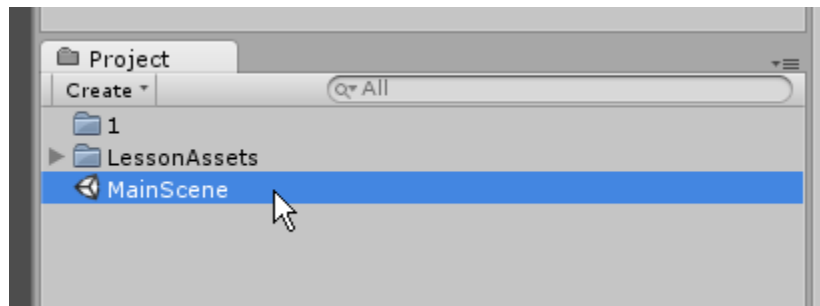


Familiarizing Yourself With The Unity Editor GUI

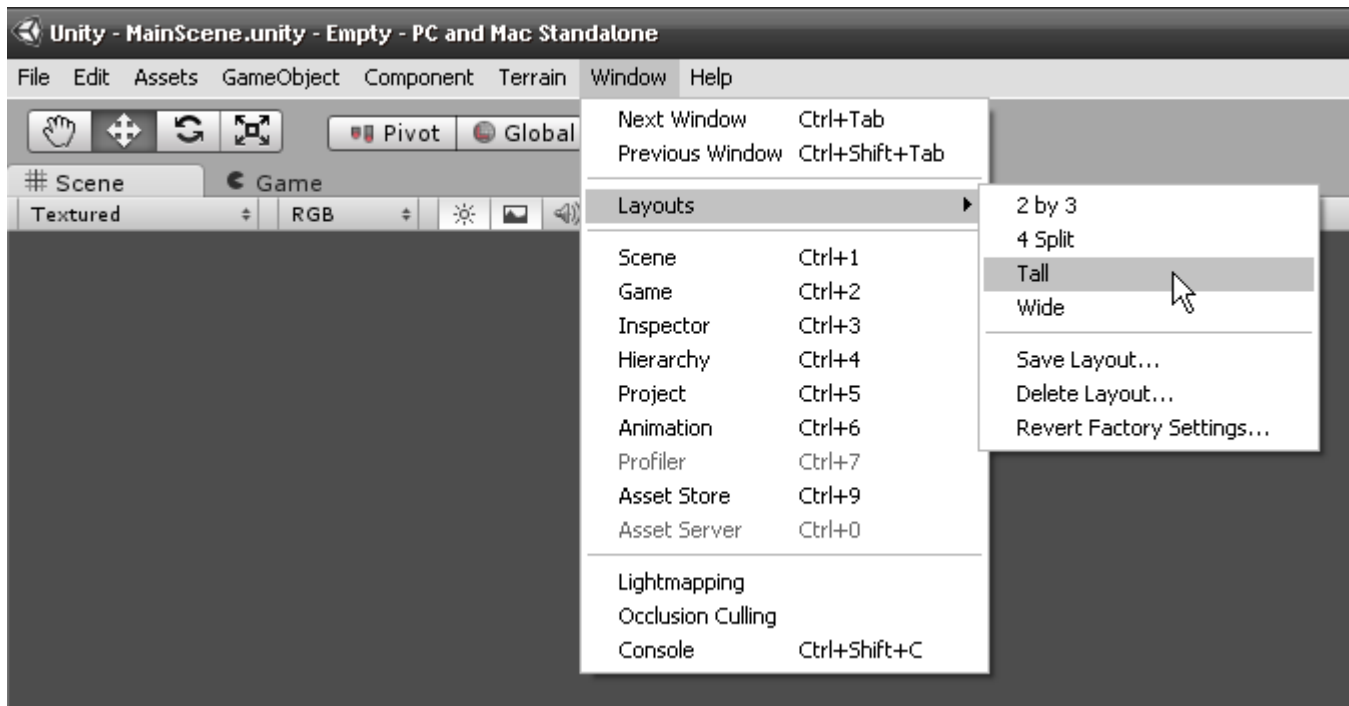
After opening the Lesson 1 project, your screen should look familiar with the picture below.



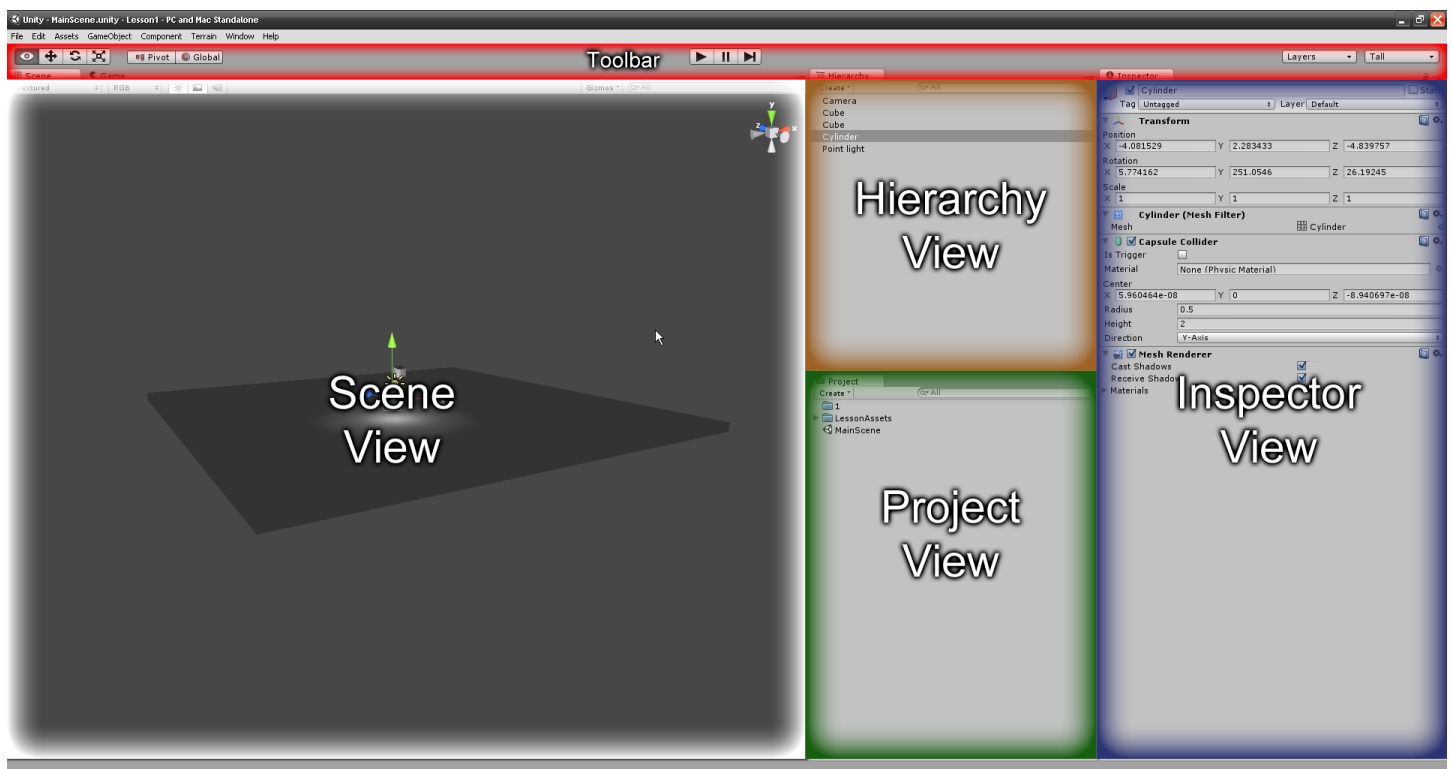
If your Unity Editor's window title says Untitled, make sure to open the MainScene file from the tab that says "Project". Double-click that file.



Now to ensure the layout of your Unity Editor is the same with the rest of this lesson, change your layout by choosing **Window > Layouts > Tall** in the top menu.



The Unity Editor comprises of five main areas:



1. **Scene View:** This shows a view of your 3d world. In Unity, the 3d world is called a “scene”.
2. **Hierarchy View:** This shows a list of all the objects in your scene. In Unity, objects are called “game objects”.
3. **Inspector View:** This shows detailed information about the currently selected game object.

4. **Project View:** This shows all the resources that your project can make use of. This ranges from source code to 3d objects, images, sounds, fonts, and other files.
5. **Toolbar:** Various other buttons to interact with your project.

Moving around your scene

First off, you need to be familiar with the controls.

Move your mouse cursor to the Scene View. **Hold the middle mouse button** and move the mouse. This will move the camera. Moving the camera is also known as “panning” the camera.

Hold Alt and the left mouse button, then move the mouse. This rotates the camera.

To zoom in and out, you have two options: **use the mouse wheel, or hold Alt and the right mouse button** then move the mouse.

Practice using these controls until you get comfortable with them.

Focusing On An Object

In your Hierarchy View, find the object named “Simple Cylinder” and click on it. Now move your mouse cursor to the Scene View. Press F on your keyboard. This will center the camera on the selected object.

While that object is centered, **rotating the camera will rotate around that object**. Once you pan the camera though, the object won't be centered anymore.

Moving Around Flythrough Style

An alternative way to navigate the Scene is to use WASD style movement. Move your mouse cursor to the Scene View. Hold the right mouse button. Use the W and S to move forward and backward. Use A and D to move left and right. Use Q and E to move up and down. Moving the mouse while the right mouse button is held will rotate the camera.

Introduction To Game Objects

In Unity, everything you see in the 3d world is a game object. This includes the player, enemies, floors, walls, lights, and even the buttons of your game.

The place where you put game objects is called a scene. You can have multiple scenes in your game, for example, one to hold the main menu, and several others for the levels or stages of your game.

Creating Game Objects

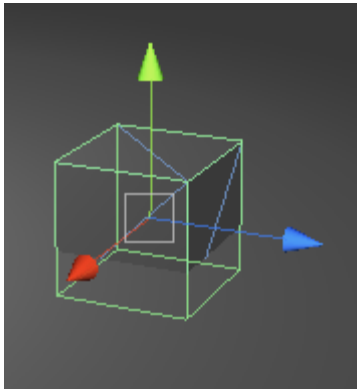
In the top menu, choose **GameObject > Create Other > Cube**. This will create a default cube at the center of your vision in the Scene View. You'll also find in that menu other default objects that you can create, but we'll concentrate on the cube for now.

Moving Game Objects

Click on your created cube in the Scene View. Center on it with F. You'll see three colored arrows on your cube. If not, press W on your keyboard, or click on the Move Tool in the upper-left of the screen.



The three arrows are handles to let you move the cube. The red arrow lets you move the cube left and right. The green arrow lets you move the cube up and down. And the blue arrow lets you move the cube forward and backward.



Simply drag these arrows to move your cube back-and-forth in that direction.

To move the cube freely in any direction, drag from the white box located at the center of these three arrows.

When you move your game object, look at the Inspector View. You'll see there the numerical values that denote the x, y, and z position of the game object.



Distance in Unity is not measured in any particular unit of measurement, but by default, the physics engine assumes the distances are measured in meters.

Those are text boxes, so you can type the exact value you want. You can also cut/copy/paste them.

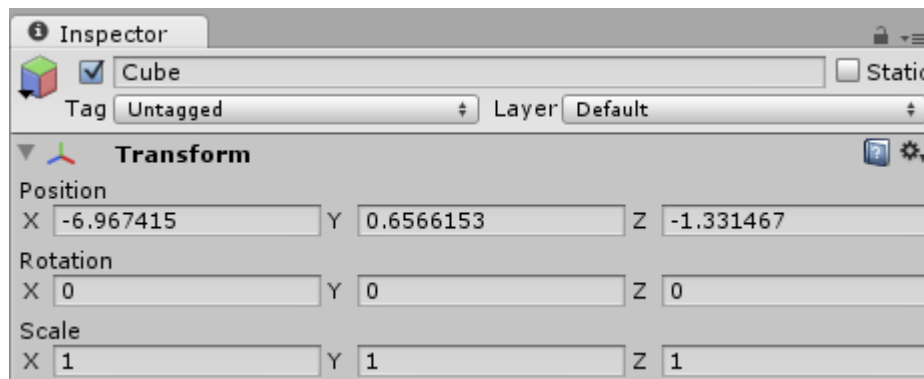
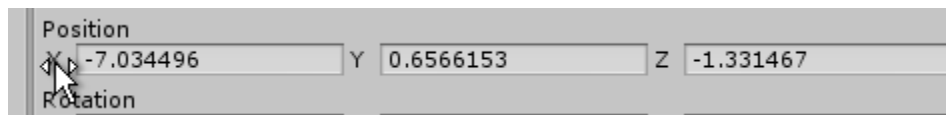


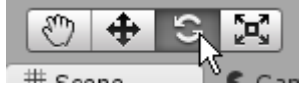
Illustration 5: The Inspector shows the x, y, and z position of your selected game object, among other things.

You can also edit the x, y, and z values with your mouse. Just move your mouse on the label and then drag.



Rotating Game Objects

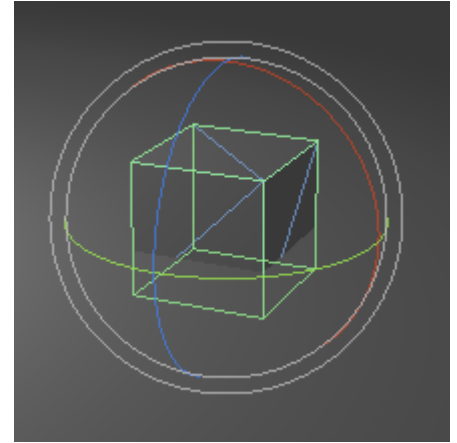
Rotating game objects work the same. Click on your cube to ensure the Scene View can accept keyboard commands, then press E. You can also just click on the Rotate Tool in the toolbar.



You'll see the three arrows change to three arcs of the same colors. The idea is the same: the three arcs let you rotate the game object in the three axes, x, y, and z.

The white outer circle lets you rotate the game object based on the view. Try it out and see.

You'll find the Inspector also lists out the rotation values of your game object. This is in Euler angles. Same as before, you can type exact values or drag the labels to edit them.



Scaling Game Objects

Lastly, you can scale game objects to do resizing, enlarging,

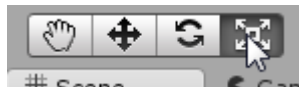
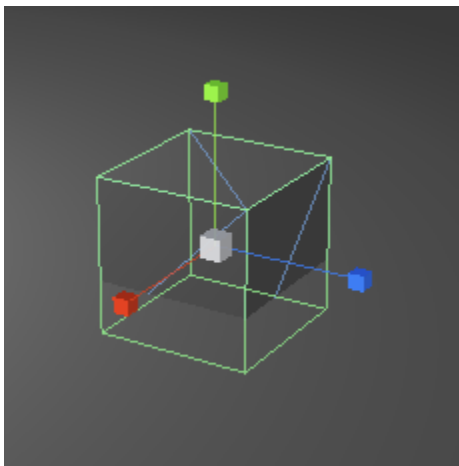


Illustration 7: The Scale Tool selected.

squashing, flattening and the like.



Scaling uses three arrows ending with cubes instead of arrowheads. Use the Scale Tool button, or press R on your keyboard to activate the Scale Tool.

Like before, simply drag back and forth to resize in that direction. Drag the white cube at the center to resize proportionally in all dimensions.

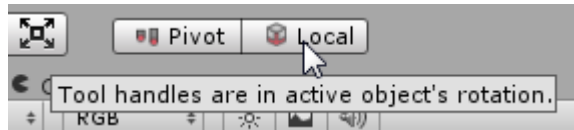
World Space and Local Space

Now that you know how to rotate, you need to learn something about spaces.

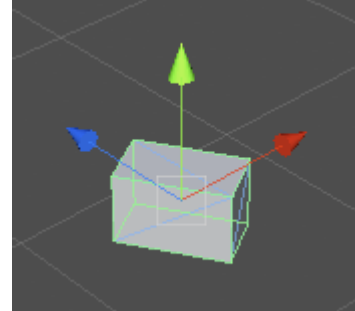
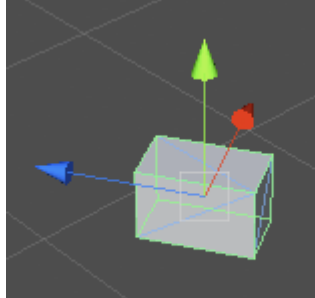
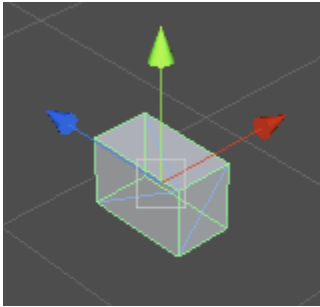
When you rotate an object, you are changing where its facing. You could say its “forward” direction is now changed to where you have pointed it.

So, for example, let's say you rotated a game object 45 degrees in the y-axis, its now pointing diagonally. If you want that object to move forward with respect to where its facing now, you are said to be moving it in its local space, or local axis.

If you want to do such a thing, make sure you set Unity to work in local space. You can do this by clicking on the corresponding button in the upper left area of the Unity GUI:



This will toggle between Global and Local. You can also do this by pressing X on your keyboard.



Snapping

You can make movement, rotation, or scaling done in precise increments by holding Ctrl while moving/rotating/scaling.

Selecting Multiple Game Objects

Hold Shift or Ctrl when selecting a new game object in the Scene View to add it to your selection. You can then move/rotate/scale them simultaneously.

In the Hierarchy View, hold Ctrl and click a game object's name to selectively add or remove it from your selection.

Hold Shift, and click a game object's name to select multiple game objects at once.

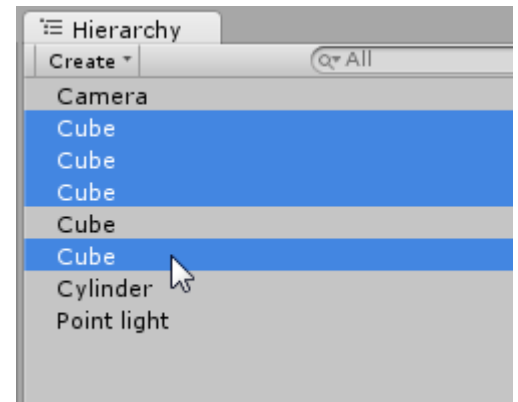


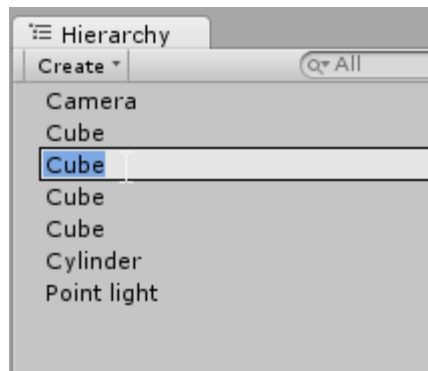
Illustration 11: Having multiple cubes selected.

Renaming Game Objects

There are two ways to change the names of your Game Objects.

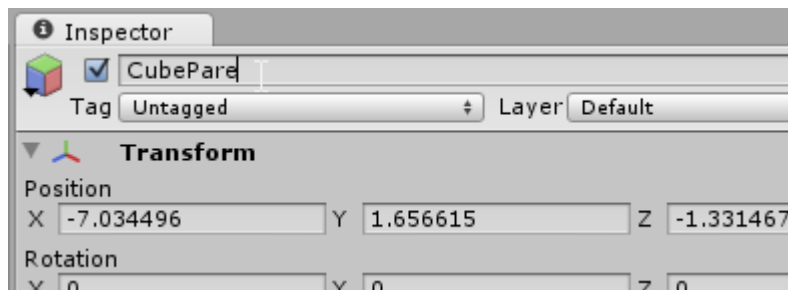
Renaming From Hierarchy

Click on the game object's name in the Hierarchy then press F2 (on Windows), or Enter (on Mac), to be able to change the name. Press Enter to confirm.



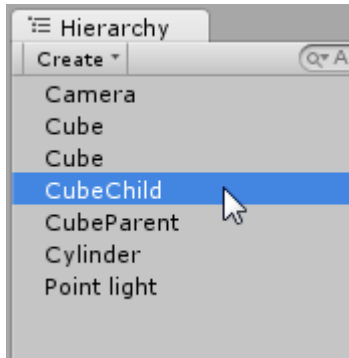
Renaming From Inspector

Select your game object, then go to the Inspector. At the top, you can change the name by clicking on it. Press Enter to confirm.

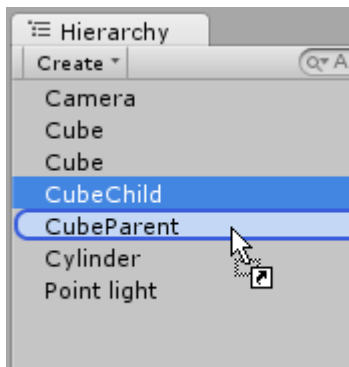


Parenting Game Objects

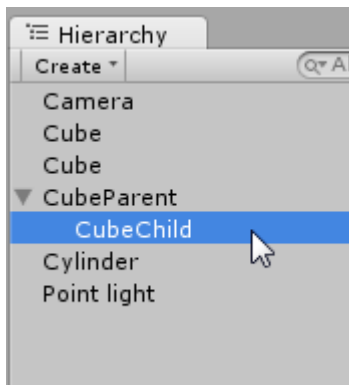
Parenting is Unity's process of grouping many game objects together.



1. Select the game object that you want to be grouped inside another one. This will be the child.

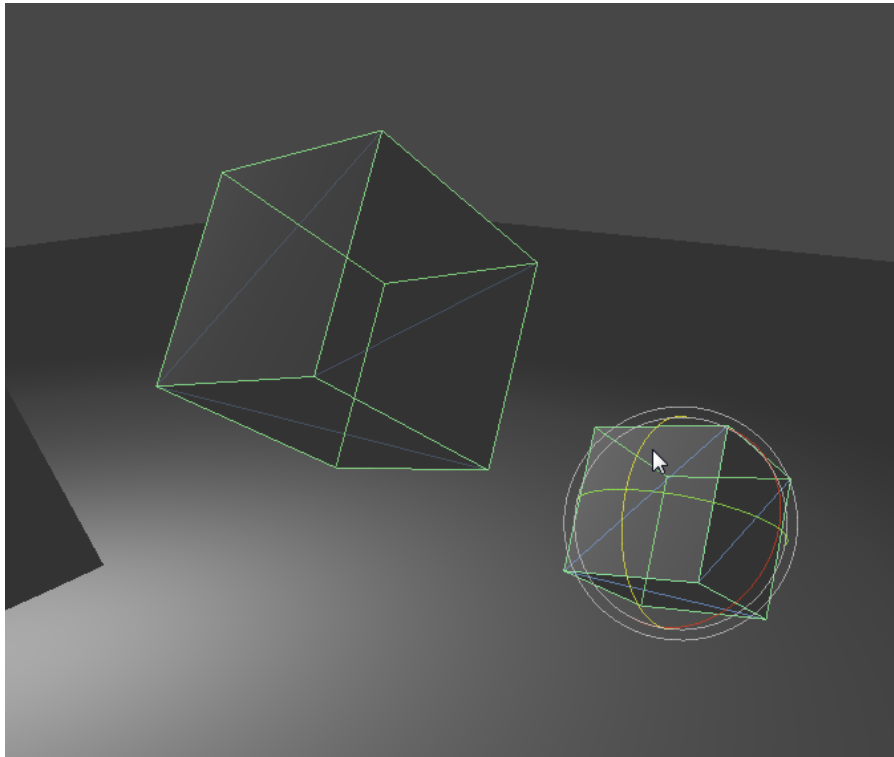


2. Drag it to the other game object that will become the parent.



3. After that, the dragged game object is now parented to the other game object.

Parented game objects follow the parent's position, rotation, and scale.



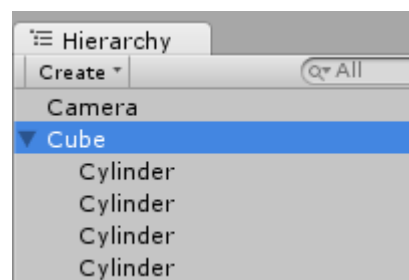
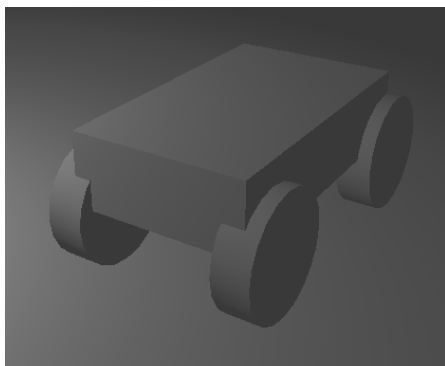
Duplicating Game Objects

To create an exact copy of a game object, press Ctrl+D. The duplicate will have the same scale and rotation as the original.

Your duplicate will also end up occupying the same space of the original object, so just move it afterwards.

Exercise

Create a simple collection of game objects looking like the picture below. Make sure to parent all the cylinders to the box.



Introduction To Unity Components

In Unity, you can give your game objects any kind of properties. These properties are called components. **Components are the things that define what a game object is: what it has, and how it behaves.**

You've already encountered one of them: the “transform” component. This transform component defines the position, rotation, and scale of a game object.

If you select a game object, **the Inspector view will show all the components the selected game object has.**

To display a 3d shape, your game object also needs other components. To define its shape to the physics engine, it also needs another different type of component.

In the game objects you've encountered, the component “Mesh Filter” specifies which 3d shape the game object uses. The component “Mesh Renderer” takes care of displaying that 3d shape. “Box Collider” specifies its shape (as a box) to the physics engine.

Interacting With The Physics: The Rigidbody Component

Try this: In the Lesson 1 project, locate the game object called “Simple Cylinder”. Select it. Focus on it by pressing F.

Add a rigidbody component by going to the top menu and choosing **Component > Physics > Rigidbody**.



Now in the toolbar, you will see three buttons in the middle like the picture to the left. These buttons control when to start running your game.

The leftmost button starts the game. The middle button pauses your game. And the rightmost button lets you advance the game by one step at a time.

Go ahead and press the leftmost button to start your game.

When your game is being run, Unity calls this “Play Mode”.

If you did everything correctly, you'll see the cylinder fall to the ground like a real world object. **This is what the rigidbody component does: it makes your game object react to the physics.**

Now, to stop your game from running, click on the play button again.

When the game is not running, Unity calls this “Edit Mode”.

Removing A Component

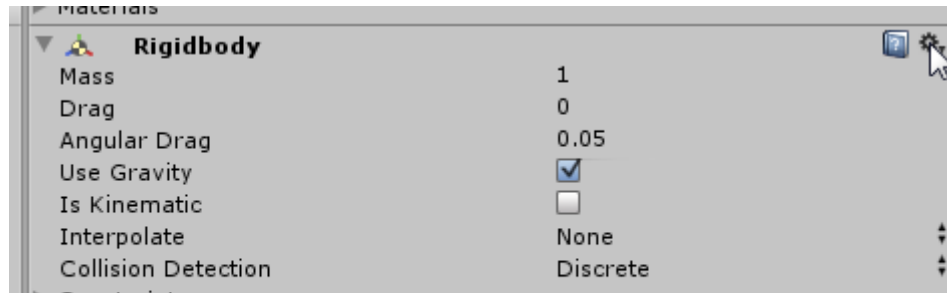
In contrast, see what happens when you remove the rigidbody that you previously added. To remove the rigidbody, look at the Inspector view and locate the rigidbody component. You'll see a gear icon at its upper-right hand corner.



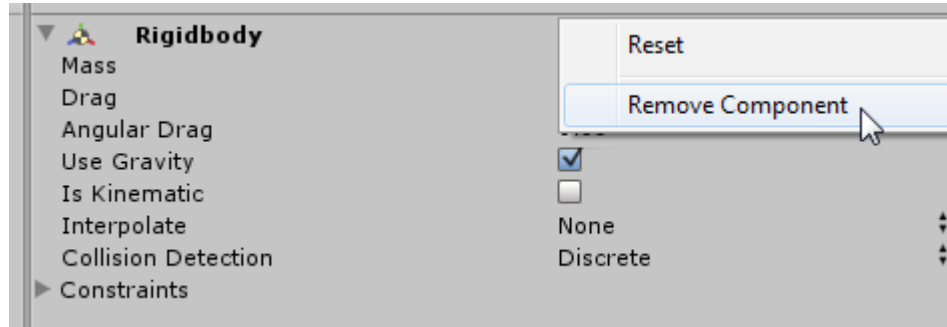
Information

For moving or animated objects, most physics engines don't use the actual shape seen to calculate the physics of a displayed object. This is because it would consume too much computer resources that way.

Usually it takes a simpler shape, like a box, or a capsule. That simple shape would encompass the displayed shape to compute the physics. Its not going to be as accurate as using the exact shape, but as a result, it becomes a lighter load on the computer.



Click on it and select “Remove Component” from the menu that appears.



Now start your game. You'll see that instead of falling, the cylinder remains floating in the air. Since the rigidbody is not present, the cylinder isn't reacting like a real-world physical object anymore.

Adding Illumination: The Light Component

With the same cylinder selected, add a light component by choosing **Component > Rendering > Light**. If you look at the light's properties in the Inspector, you'll see some interesting stuff.

Click on the white bar for the color. You'll see a color chooser dialog window. Choose any color you want. You'll see that the Scene View automatically updates in real-time the color that you choose.

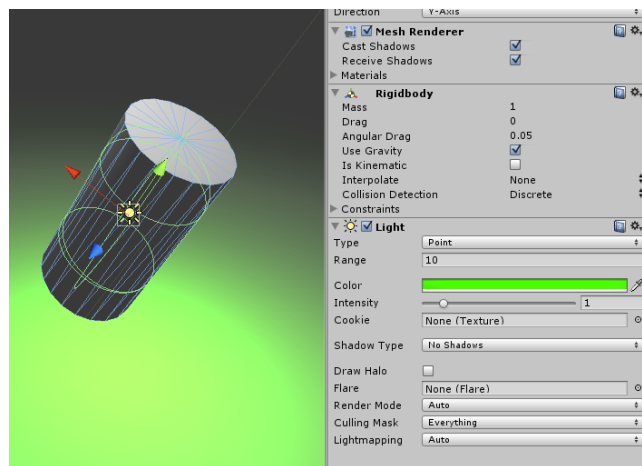


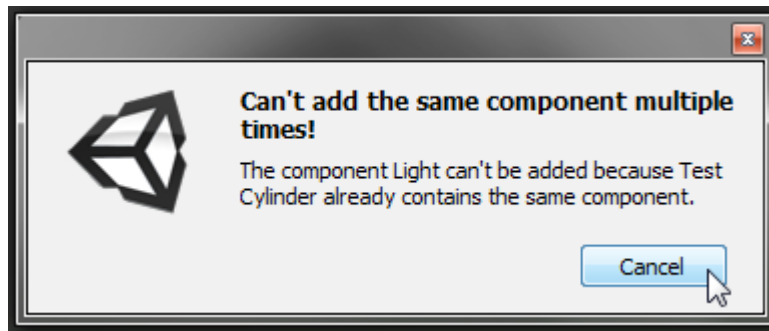
Illustration 16: Unity shows you the color changes in real-time while you edit.

If you add a rigidbody once more, you'll find your game object has both a rigidbody and a light component. Your game object would fall on the ground *and* shine a light *at the same time*.

This is the advantage of the component-based design that Unity makes use of. You can give any game object any combination of components you want.

Adding More Than One Light

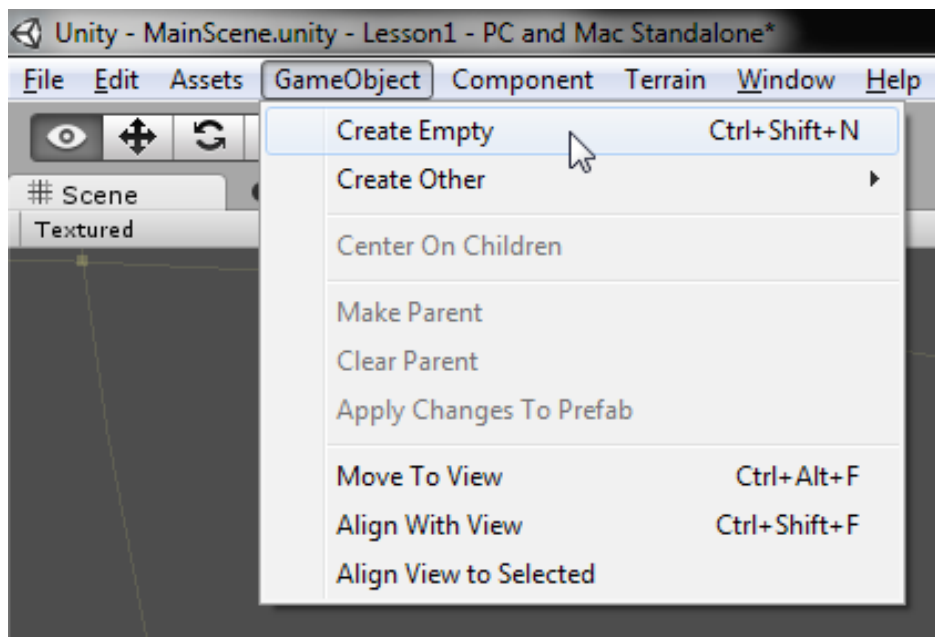
Now what if you want to add two lights to your cylinder? Go ahead and try to add another one. Unfortunately, you'll be greeted with an error:



This is because you can only add one Light component to each game object at most. This holds true for most component types. Like what the error message says, **you can't add the same component to a single game object multiple times.**

So how do we fix this? **What we do is we create an invisible game object that will hold the second light, and parent that invisible game object to the cylinder.** Your cylinder can have as many children game objects as you want it to.

In Unity, you can create invisible game objects by going to **GameObject > Create Empty**. You can also press Ctrl + Shift + N.



In Conclusion...

If you've come this far, congratulations! You're on your way in creating your own games with Unity. You learned the basic gist of what Unity is, how to open a project, what game objects are all about and how to move them around. You've learned what components are: the things that describe what a game object is and how they behave.

In the next lesson, you'll learn more components that you can use, like terrain, basic visual effects like particle effects and lens flares, and even music.

You'll be asked to create your own level and compile your application into an .exe file. You'll also learn how to deploy your game to a web browser!