

Introduction to Unity

Beste F. Yuksel Some figures taken from Unity's award winning tutorials - rest of my own making :)

What is Unity

Unity is a 2D and 3D game engine that can be applied to a variety of platforms including phones and VR/AR (XR) devices.

A game engine is a framework that allows you to import art and assets (2D and 3D) from other software such as Maya, assemble those assets into scenes, add lighting, audio, physics, animation, interactivity, and logic.

The scripting language that we will be using with Unity is C# (very similar to Java) but JavaScript and Boo can also be used.

Unity has fantastic tutorials and documentation which I strongly recommend you peruse.

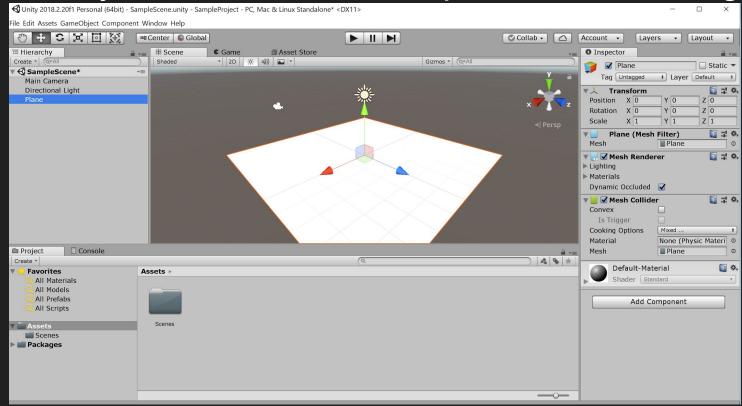
Create a new project in Unity called SampleProject (3D)

G Unity 2018.2.20f1				×	(
Projects	Learn	H New	Dpen	My Account	
	Project name	Template			
	SampleProject	3D	~		
	Location				
	C:\Users\Beste\UnityProjects	Add Asset Package			
	Organization				
	BesteFiliz ~	ON Enable Unity Analyti	cs 🕐		

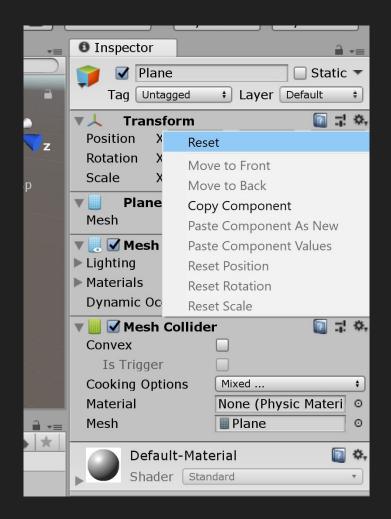
Let's import a plane GameObject

🚭 Unity 2018.2.20	0f1 Personal (64bit) - San	npleScene.unity - Samp	oleProject - PC, Mac & Lin	ux Standalone <dx11></dx11>
File Edit Assets G	ameObject Component	Window Help		
🖑 🕂	Create Empty Create Empty Child	Ctrl+Shift+N Alt+Shift+N		
≒ Hierarchy	3D Object	>	Cube	Store
Create Create C	2D Object Effects	>	Sphere Capsule	
Directiona	Light	>	Cylinder	
	Audio	>	Plane	375
	Video	>	Quad	
	UI	>	TextMeshPro - Text	
	Camera		Ragdoll	
	Center On Children		Terrain	
	Make Parent		Tree	
	Clear Parent		Wind Zone	
	Apply Changes To Pre	fab	3D Text	
	Break Prefab Instance]	$\langle \langle \rangle \rangle$	
	Set as first sibling	Ctrl+=		
	Set as last sibling	Ctrl+-		
	Move To View	Ctrl+Alt+F		
	Align With View	Ctrl+Shift+F		
	Align View to Selected			
Create •	Toggle Active State	Alt+Shift+A		0
Favorites		Assets ⊩		(Q
		Assels 🖻		
Q All Models				
Q All Prefabs				
🔍 All Scripts				

If you click on the Plane object in the Hierarchy on the left, you can view its components on the right.



You can always reset a GameObject's position in the same to 0,0,0 by going to the toggle next to Transform and Reset



GameObjects

Game objects are the items that make up your Unity scenes.

Everything within a scene is considered a game object and these objects are made up of a number of components.

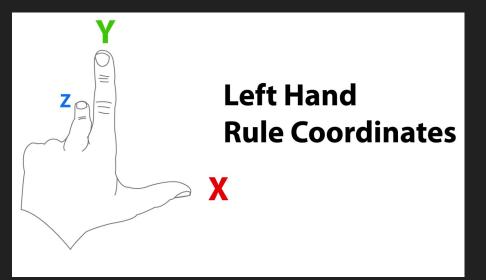
Each has a necessary transform component that contains the position, rotation and scale of the game object.

You can view each game object's components by clicking on it in the Hierarchy.

The Hierarchy is a list of all the game objects in the scene.

GameObject in Unity docs

Unity's 3D coordinate system

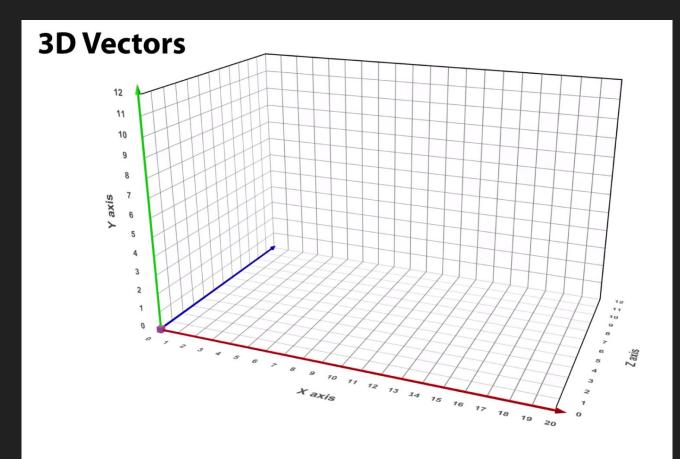


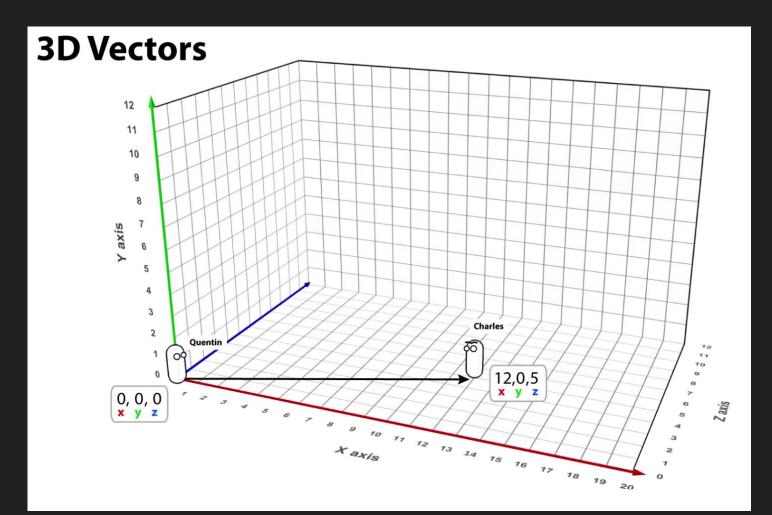
Z axis represents depth

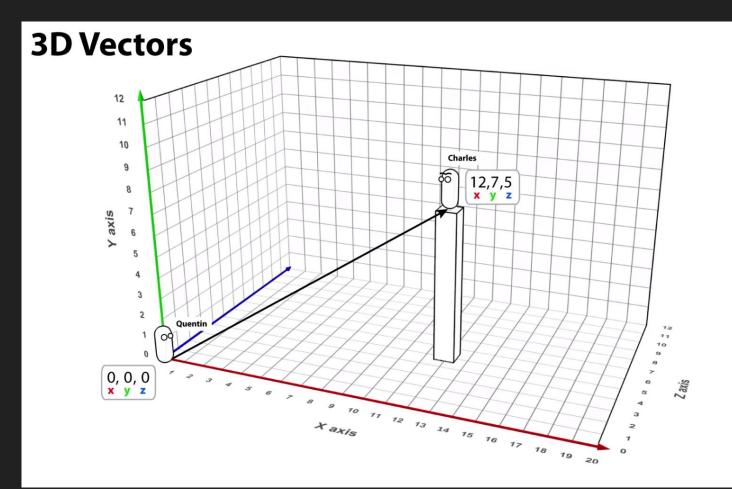
X and Z axes make up the horizontal plane

Y axis points up

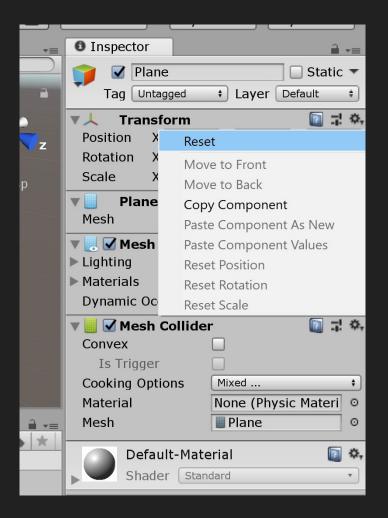
Unity works on a left-handed coordinate system.







Back to resetting - if you reset a GameObject's Transform.position you will set it to 0,0,0 (also called the origin)



Transform

Transform is a class in the UnityEngine that is the position, rotation, and scale of an object.

See this reference for the Transform class in Unity docs

The Transform class has a property called **position**

Transform.position

This is the X,Y,Z position of an object in the scene.

Transform.position in Unity docs

It is represented as a Vector3 object.

You can alter the Transform.position to move a GameObject or get this value to locate the GameObject in the 3D world space.

Vector3

Vector3 is a structure (similar to a class) that represents 3D vectors and points.

It includes the properties x, y, and z.

Vector3 in Unity docs

Vectors

A vector is often defined as a geometric object that has both magnitude (length) and direction.

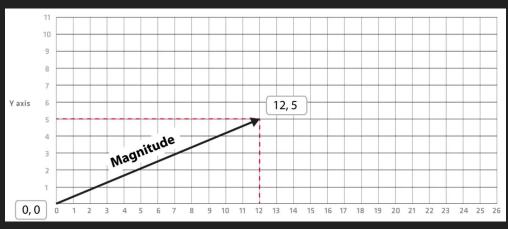
It is represented as an arrow between 2 points with a certain length (and direction, hence arrow).

Here is an example of a 2D Vector.

2D vectors are made of the x and y

components.

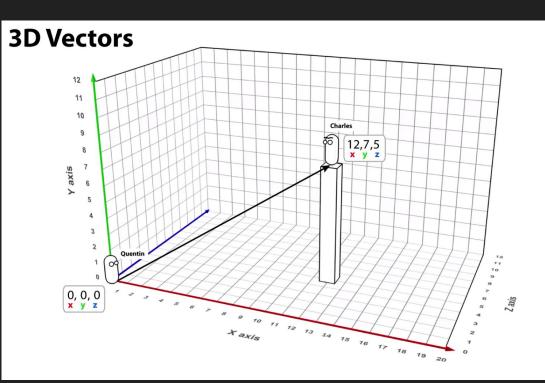
Vector2 objects represent 2D vectors



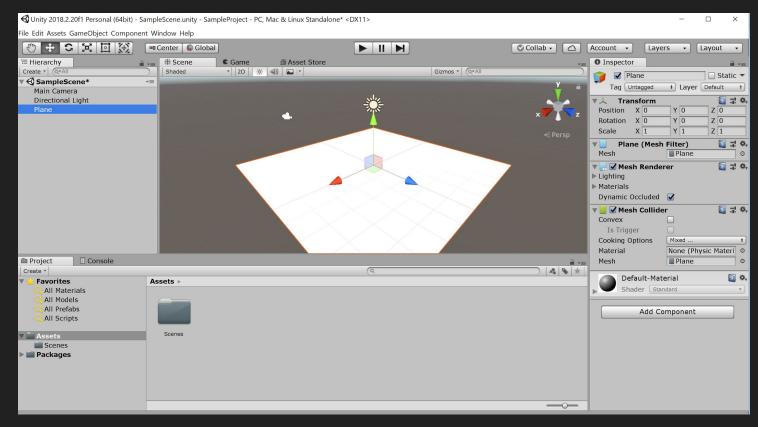
in Unity.

We can also have 3D vectors as we saw before as represented by a Vector3 object in Unity.

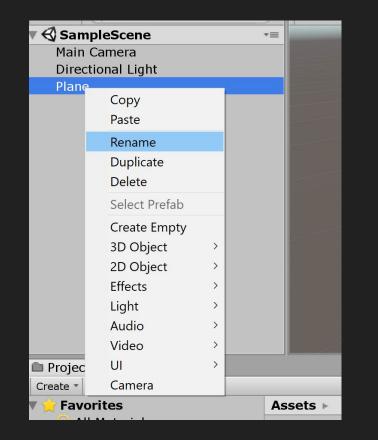
Thus, going back to our Transform. position property, that is a Vector3 object and is made up of the x, y, and z components.



Let's return to our plane that we have created with a Transform.position of (0,0,0)



Let's rename our plane object to 'Ground'





Go to 'Create' under Project and create a new Folder. Name it 'Materials'

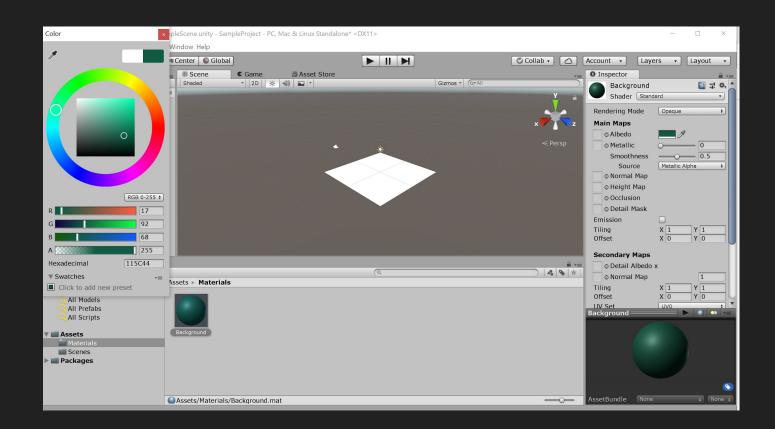


Project	E Console		
Create *			
🔻 🚖 Favorites	🔻 🚖 Favorites		
🔍 All Materia	ls		
🔍 All Models			
🔍 All Prefabs			
🔍 All Scripts			
		Materials	Conner
🔻 💼 Assets		Materials	Scenes
📹 Materials			
💼 Scenes			
🕨 📰 Packages			

Go into the 'Materials' folder and go to 'Create' again and this time create 'Material' and name it 'Background'.



Go to 'Albedo' under Background and click on the color selector and choose a dark green. Then drag and drop the material onto the plane.



Zoom into the scene with your mouse.

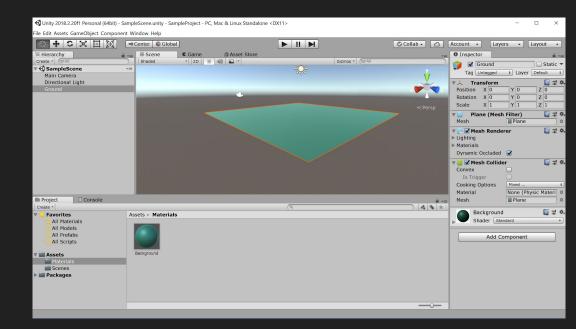
Try adjusting the position, rotation and scale of the plane using the Transform component and the buttons here:

File Edit Assets GameObject Componer

'≡ Hierarchy

Here is some documentation on Scene View navigation

Here are some keyboard Unity shortcuts



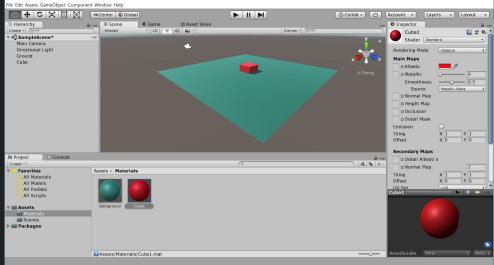
Save your scene and project as you go!

Let's add a 3D Cube to the scene. Once you've added the cube, reset its position so it's at (0,0,0).

Give it a red color by creating a new material called 'Cube1'.

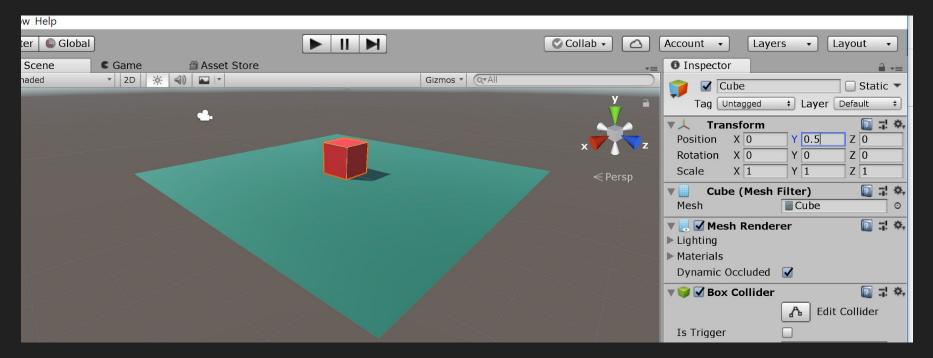
You will see that it sits cut half-way by the ground object.

All Model						
Favorites Q All Mater	ials	ssets ► Materials				
Create *				(0		
Project	Toggle Active State	Alt+Shift+A				
	Align View to Selected	Curralmetr				
	Align With View	Ctrl+Shift+F				
	Set as last sibling Move To View	Ctrl+- Ctrl+Alt+F				
	Set as first sibling Set as last sibling	Ctrl+= Ctrl+-				
		Ctrl+=				
	Break Prefab Instance		3D Text			
	Apply Changes To Prefal		Wind Zone			
	Make Parent Clear Parent		Tree			
	Make Parent		Terrain			
	Center On Children		Ragdoll			
	Camera	· · · · · ·	TextMeshPro - Text			
	Video	· · · · · · · · · · · · · · · · · · ·	Quad	Contraction of the local division of the		
Ground	Audio Video	>	Plane			
Directiona	Light	>	Cylinder			
Main Cam	Effects	>	Capsule			
SampleS	2D Object	>	Sphere			
Create *	3D Object	>	Cube	Store		
	Create Empty Child	Alt+Shift+N				
• + :	Create Empty	Ctrl+Shift+N				



This is because in Unity all primitive objects, cubes, spheres, etc. have a standard size: th are either 1 by 1 by 1 or 1 by 2 by 1 Unity units.

Therefore, we can lift the object up by half a unit in the Y axis to lay it on top of the plane.



Script Components

We are now ready to add a script component to the Cube GameObject.

Scripts can be considered as behavior components in Unity as they control the behavior of the GameObject they are added to.

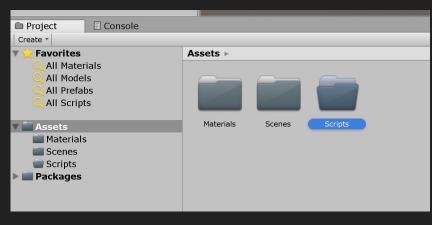
Let's create a Scripts folder in the project panel under Assets.

Scripts can be attached to Objects by

dragging and dropping them to the inspector

panel or by adding a component in the

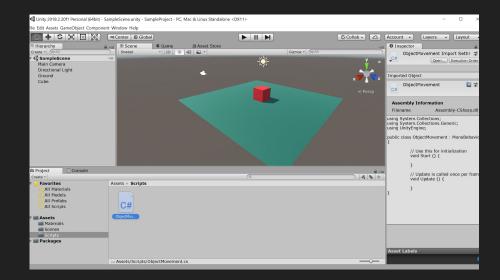
Inspector panel.



Go into your scripts folder and create a C# script.

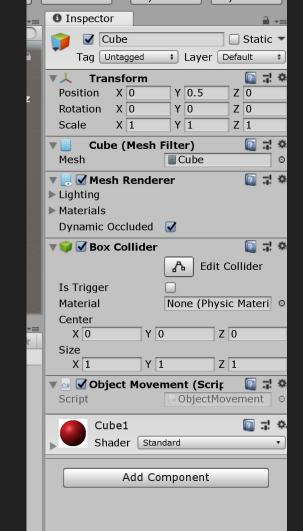
Call it 'ObjectMovement'

File Edit Assets GameObject Component Window Help						
🕙 🕂 C 🖾 🔟 🛞		90	Center 🛛 🌑 Global]		
Create • Q*All	2		# Scene Shaded	€ Game + 2D ※ 4		
GampleScene Main Camera		*≡				
Folder						
C# Script						
Shader	>					
Testing	>					
Playables	>					
Assembly Definition TextMeshPro	>					
Scene			- Jan			
Prefab						
Audio Mixer						
Material						
Lens Flare						
Render Texture			<u>}</u>			
Lightmap Parameters Custom Render Texture						
Sprite Atlas						
Sprites	>	As	sets ⊧ Scripts			
Tile						
Animator Controller						
Animation Animator Override Controller						
Avatar Mask						
Timeline						
Physic Material						
Physics Material 2D						
GUI Skin						
Custom Font			Accote/Corint-			
Legacy	>		Assets/Scripts			
UIElements View						



You will see a preview of the script if you select it but this is not editable.

First, let's add this script to the Cube by dragging and dropping it to the Cube's Inspector. Make sure the Cube object is selected in the Hierarchy!



Let's open the script using the cog wheel next to the script component and select 'Edit script'

*	X 0	Y 0 Z 0					
	Size						
	X 1	Y 1 Z 1					
	🔻 🛃 🗹 Objec	t Movement (Scrip 🛛 🗐 🖬 🌣					
	Script	Reset					
	Cube	Move to Front					
	Shad	Move to Back					
		Remove Component					
		Move Up					
		Move Down					
		Copy Component					
		Paste Component As New					
		Paste Component Values					
		Edit Script					

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
public class ObjectMovement : MonoBehaviour {
```

```
// Use this for initialization
void Start () {
}
// Update is called once per frame
void Update () {
```

}

```
ſ
```

The method Update() is important. It is called before rendering each frame.

We will now add some code to move the cube each frame inside this method.

Let's start with:

transform.Translate(Vector3.up * Time.deltaTime);

transform.Translate will translate the position of the object.

<u>Vector3.up</u> moves the object up. It is shorthand for writing Vector3(0,1,0)

Press Play!

Time.deltaTime allows you to move the object a n units per second rather than per frame (i.e. it slows it down - try the code without it and see)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
public class ObjectMovement : MonoBehaviour {
```

```
// Update is called once per frame
void Update () {
   transform.Translate(Vector3.up * Time.deltaTime);
```

We can also control the speed using a public variable that we can then edit in the Inspector.

You can also just say:

public float moveSpeed;

(without setting the value)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

public class ObjectMovement : MonoBehaviour {

```
public float moveSpeed = 5;
```

```
// Update is called once per frame
void Update () {
    transform.Translate(Vector3.up * movespeed * Time.deltaTime);
```

Now go back to your Inspector and you will see that the variable move speed has appeared.

Try running it again.

Change the value in the inspector and watch the speed of the cube change.

Does it change the value of the variable in the script?

*=	X 0 Y 0 Z 0 Size X 1 Y 1 Z 1 C# Object Movement (Script Script Move Speed 5	
	Cube1 I I I I I I I I I I I I I I I I I I I	

Now let's have the cube move forward along the z axis.

transform.Translate(<u>Vector3.forward</u> * moveSpeed * <u>Time.deltaTime</u>);

You will actually see the cube move away from you because the camera is positioned back along the z axis.

You can have the cube move backward along the z axis with this piece of code:

transform.Translate(-<u>Vector3.forward</u> * moveSpeed * <u>Time.deltaTime</u>);

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

public class ObjectMovement : MonoBehaviour {

```
public float moveSpeed = 5;
```

```
// Update is called once per frame
void Update () {
    transform.Translate(Vector3.forward * movespeed * Time.deltaTime);
```

Object Rotation

Now let's have the cube rotate. transform.Rotate will rotate the object around the X, Y, and Z axes, measured in degrees.

transform.Rotate(<u>Vector3.up</u>, turnSpeed * <u>Time.deltaTime</u>);

The variable turnSpeed is also public (see next slide) and you can control the speed at which the object rotates.

You can rotate in the other direction by negating the turnSpeed like this:

transform.Rotate(<u>Vector3.up</u>, -turnSpeed * <u>Time.deltaTime</u>);

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

public class ObjectMovement : MonoBehaviour {

```
public float moveSpeed = 5;
public float turnSpeed = 50;
```

```
// Update is called once per frame
void Update () {
    transform.Rotate(Vector3.up, turnSpeed* Time.deltaTime);
```

Physics - RigidBody

If we want our GameObjects to behave like real world objects such as coming back to earth with gravity or colliding with other objects, this requires physics.

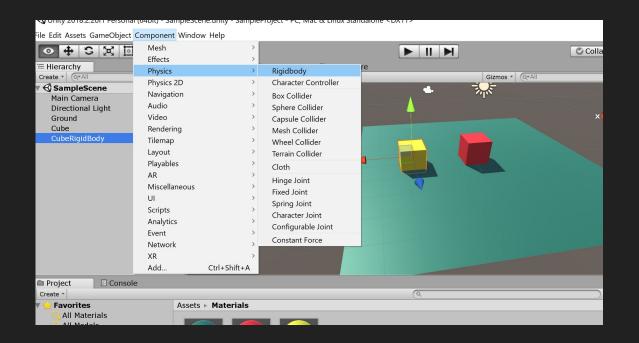
To use physics the GameObject needs a Rigid Body component attached.

Let's create a new Cube called 'CubeRigidBody' to see the differences.

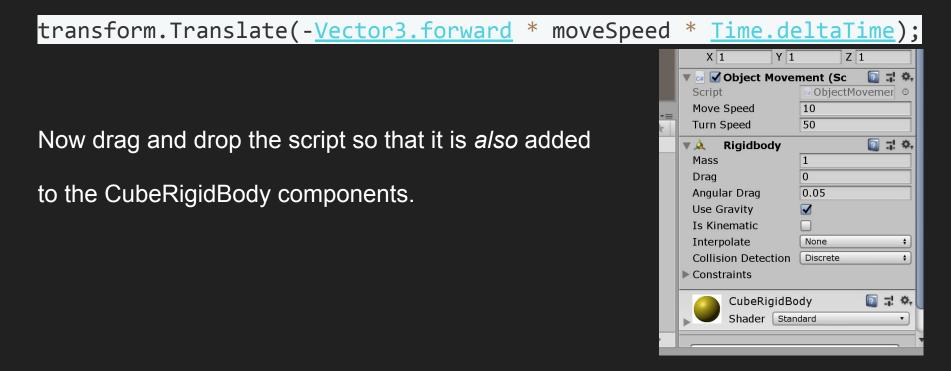
Create your new Cube and give it a yellow color. Put it near your first cube.

With your CubeRigidBody selected, go to 'Component' -> Physics -> Rigidbody.

You will see this new component in the Inspector with 'Use Gravity' checked.



Now, let's edit our script back to the object moving backward along the z axis:



Set the speed of both cubes to 2.

Now Play!

Watch what happens!

Due to the laws of the physics, when the rigidbody cube falls of the plane it drops over the edge!

Now change the code so that the cubes move upward and see what happens! Change the speeds and play it again.

To Explore On Your Own

1. You can learn about the RigidBody class and how to apply force to the RigidBody (RigidBody.AddForce) in the Roll A Ball tutorial, specifically on <u>'Moving</u> the Player'

Try applying the code in that tutorial to move a GameObject. (It's a great tutorial!)

2. Have a look at moving the camera position <u>here 'Moving the Camera'</u>. This will also give you another way of moving an object. (Another great tutorial!)

Please familiarize yourself with Unity a bit as the Lab 2 won't be identical to what we did, but this prep does introduce you to the concepts.