



BOND UNIVERSITY

Bachelor of Interactive Multimedia and Design



Beginner Game Dev Need for Speed Make a Game in 1 Hour.

FACULTY OF SOCIETY AND DESIGN

Making a Racing Game with Unity

© Penny de Byl Faculty of Society and Design Bond University pdebyl@bond.edu.au

Part

A Crash Course in Unity

o begin, download Unity by visiting *http://www.unity3d.com/*. Unity provides a full version as a 30-day trial with all features unlocked, after which it will revert to the free version. The free version includes the majority of Unity's functionality and the Android or iOS tools. You can use the free tool to create and release games. The main difference between the paid and free versions is that the free version includes a watermarked loading screen. If you can live with that, you're free to develop as many games as you like with it. When the download is complete install Unity and run the software. On opening you will notice the Unity Editor is comprised of several tabbed windows as shown and explained in Figure 1.



Figure 1. The Unity Interface



If you didn't create a new project when Unity started create one now by selecting **File > New Project** from the main menu. The name you give the project will be the same as the folder Unity creates to store the game assets. Unity does not use a single file to save the project as you find in other software such as *Word* or *Photoshop* but rather the top-level folder. Unity also likes to have everything in its project folders in the correct order, so best not to mess with the structure or add anything manually.

KEYS	Moving in Unity
W Move	To move, rotate or scale an object use the W, E and R keys
E Rotate	to change the gizmo. Then drag the axis to put the object
R Resize	in the desired position.
Q + Mouse Pan View	_ To change the view you can use Q to pan the camera
ALT + Mouse Rotate View	(move it vertically or horizontally) or ALT with the right
	mouse button to orientate.



We will now create a simple game scene with a sphere. From the main menu select GameObject > 3D Object > Sphere. A new game object called *Sphere* will be created and you'll see it in the Hierarchy tab and Scene and Game windows as shown in Figure 2. When selected the properties of the *Sphere* are shown in the Inspector tab.



Figure 2. Adding a Sphere to the Scene



If the Sphere appears small in the Scene, place the mouse inside the window and scroll to zoom in. Notice the Sphere in the Game is not affected. The Game is the view of the game from the camera in the scene. The camera is essentially the player's eyes into the game world. You can find the Main Camera as a game object in the Hierarchy. Double-click on it.



To change what you see in the Game the Main Camera needs to be moved. To do this, select the Main Camera in the Hierarchy and press W. A set of green, red and blue axes will appear in the Scene on the Main Camera. To move the camera select the tip one of the axes with the mouse and drag it. Notice how the Game view changes with the camera's view.



To change the camera's orientation, with the Main Camera still selected in the Scene, press E. Green, red and blue orbital handles will appear around the camera in the Scene. Dragging the mouse over any of these will orient the camera.

The final manipulation you can make to an object in the Scene is to scale it. In this case, scaling the camera won't do anything, so instead, click on the Sphere either in the Hierarchy or the Scene. Press R. Again green, red and blue axes will appear. This time the axes will have small cubes on the end. There will be a large yellow cube in the centre. Dragging the large yellow cube will evenly resize the Sphere. Dragging the individual cubes at the end of each axis will resize in the same direction. Try this out and observe how the sphere's size changes in the Game.





Try pressing the play button at the top centre of the editor. This runs your game. At this time nothing will happen as there's nothing actually going on in your scene. Press the play button again to stop playing.

NOTE: Do not make changes to your game in Play mode. Otherwise when you stop playing the changes will be lost.



To add physics, with the Sphere selected, choose Add Component > Physics > Rigidbody. The result will be a Rigidbody component added to the Inspector for the sphere. For the Sphere, having a rigidbody simply means it becomes part of the engine's physics system and now will behave as such.

Press play. The Sphere will fall under the influence of gravity.



Add a cube to the scene with GameObject > 3D Object > Cube. Select the Cube in the Hierarchy and use the W key to set it to move mode and use the mouse to drag the cube beneath the sphere. Add a rigidbody to the cube. Without changing any rigidbody settings, press Play. Both the sphere and the cube will fall at the same rate. If you understand basic Newtonian physics there will be no surprises there.



NEED FOR SPEED WITH UNITY

For the Cube, in the Inspector, untick the rigidbody's Use Gravity tickbox. The cube will still be influenced by physics events just not gravity.

Play. The Cube will remain stationary until the sphere collides with it, causing it to move under the sphere's influence.

Add more Cubes and place them beneath the Sphere. To add an exact replica of the Cube you've already created, with it selected in the Hierarchy press CMD+D (Mac OS) or CTRL+D (Win) to duplicate it. Right-clicking on the object in the Hierarchy will also give you the option to duplicate. You can then move the new Cube into position. Because it was duplicated it will already have a rigidbody. Depending on how you have the cubes positioned, the Sphere will now collide with them and cause them to spin. The Sphere will continue falling under the influence of gravity, but the Cubes will only have the Sphere's momentum on collision to carry them and therefore they will not fall in the same way.

NOTE: If you want to set the Main Camera to look at the game world in the same way you are viewing it in the Scene, select the Main Camera and then GameObject > Align with View.



Another property you can manipulate on a game object is its colour. The colour is defined by a material component. It must be created differently to the physics component. To create a material, go to the Project, select the drop down menu under Create and choose Material.





When the material is first created it will be called *New Material*. Click on its name to change it to *Green*. When it is selected in the Project, the properties of the material will be visible in the Inspector. Click on the Main Color colour box to bring up the colour selection dialogue. Set the colour of the material to green.



To apply the material to one of the game objects in the Scene simply drag and drop the material from the Project onto the object in the Scene. Alternatively you can drag and drop it onto the game object's name in the Hierarchy or with the game object selected in the Hierarchy, drag and drop the material onto the Inspector.

You can apply this material to any object in the Scene. To change the colour or other properties of the material, select it again in Project and change it in the Inspector. All objects with this material applied will change in response. For example, if you changed the colour of the material to orange, all objects that were green will now appear orange.



Code can be attached to any game object to control it's behaviour. Simple JavaScript commands can access and modify any properties. For example, you can change the colour of the object, move it around, change it's animation state and much more.

To create a script, in the Project select Create > Javascript. The file will initially be called NewBehaviourScript. Click on it and rename to **revolve**. Double-click the new file in the Project to open it in an editor.

Add the line of code in bold to the script and then save.

```
#pragma strict
function Start () {
}
function Update ()
{
    this.transform.Rotate(Vector3.up, 5);
}
```



Switch back to Unity.

Drag and drop the script file onto one of the cubes in the Inspector. This will attach the script to that particular game object and cause it to follow the instructions of the code.

Play. Watch the cube spin.

Attach this code to any object in the scene that you want to spin.



NEED FOR SPEED

Making a racing game in an hour

n this tutorial we are going to make a racing game with Unity where several cars racing around a track toward the finish line. One car will be player controlled and the others by simple artificial intelligence (AI).

The starter files can be downloaded from http:// holistic3d.com/small-projects/



Download and unzip the files into the same folder. Locate the NEEDFORSPEED.unity file that you will find inside the Assets folder and double click on it to open with Unity.

mouse button to orientate.



A Reminder about navigating

Moving in Unity

KEYS	To move, rotate or scale an object use the W. E and R keys
\mathcal{W} Move	to change the gizmo. Then drag the axis to put the object
\mathcal{E} Rotate	in the desired position.
\mathcal{R} Resize	- To change the view you can use O to pan the camera
Q+ Mouse Pan View	(move it vertically or horizontally) or ALT with the right

ALT' + Mouse Rotate View

The Project Window

The Project window contains all the assets available to your game. Select the Car object from the Project with the mouse and drag it into the Scene, onto the ground. To do this, ensure you can see the ground in the Scene first by navigating into position.

The Hierarchy Window



(move it vertically or horizontally) or ALT with the right

The Hierarchy window contains the objects that are in your game. Notice how the Car is now in your game. To centre the Scene on the car, doubleclick on Car in the Hierarchy. If ever you can't find an object in your game just double click its name in the hierarchy.





Positioning The Car

Ensure the car is above the ground. To do this, select Car in the Hierarchy. The Car model in the Scene will be highlighted. If it doesn't have green, red and blue arrows on it, press the **W key**. Click and drag these arrows to reposition the Car. You can tell when the car is above the ground as it is entirely visible.



Driving

To play the game thus far and drive the Car, click on the Play button.

The Game window will come to the front and you'll be able to drive the Car with the arrow keys. The Game window is your game running live when in play mode.



Press the Play button again to return to the Scene. NOTE: Ensure Play is OFF before you started editing your Scene again.





Add the Race Track

Locate the RaceTrack object in the Project and drag it into the Scene. Position it in the centre of the ground area. Move the car to sit on the track. Take a drive around the track to test out your car's position by pressing Play.



Add The Competition

Locate the Charger, Lambo and Peugeot models in the Prefabs folder in the Project. Drag and drop one of each onto the track in the Scene and line up for the race.





Defining The Race Route

To make the computer controlled opponent cars look smart with respect to following the track, they need a set of points, called Waypoints to follow. This is a common form of defining navigation for nonplayer characters in games. Drag the WayPoint object onto the track.



Complete the Track Waypoints

Continue placing new waypoints by dragging and dropping the WayPoint object onto the track.

The positions of the waypoints should be such that a car following them will not go off the track if it follows a straight line between them. Each waypoint added will be named WP001, WP002, WP003 etc. Locate the WayPoint object in the Project and drag and drop onto the racetrack just in front of the cars. It automatically numbers as the first waypoint.

The cars will follow them in order. Place them so they make a continuous loop as shown by the stars in the image above. Once they are in place, press play to test out the racing action.





Modifying Car Speed and Physics

For any of the cars you can make them drive differently by changing their physics properties. Select a car (Lambo, Charger or Peugeot) in the Hierarchy and in the Inspector locate its values for Mass, Drag, Speed and Rotation Speed. Modify the values for these to make the car move faster or slower. Note, if you make the car move faster, you'll have to increase its Mass otherwise it will roll if it takes the corners too quickly.

Play around with these values and then press Play to see how they affect each car's behavior.





The Finish Line

To track the car's position in the race and determine when the race is over we need to add a finish line.

Locate the FinishLine object in the Project and drag it onto the track. You won't see anything under the mouse until you release it. At this time a green rectangular prism will appear. This is a space on the map called a trigger.



Whenever a car passes through this box, it will trigger an event to count the number of laps.

Play. Details of your position in the race and the number of laps will display. The default total laps for the race is 2. You will find the game is over once any car has lapped twice.

To change the number of laps select the FinishLine in the Hierarchy and change the Laps value in the Inspector.





Landscaping

TERRAIN EDITS
Raise and Lower
Plateau
Smooth
S Paínt Textures
Add Trees
Add Flowers/Grass

To create a landscape and racing track we will use the sculpting tool. Click on Terrain in the Hierarchy. In the Inspector, a toolbar for sculpting will appear.

In the Scene, zoom out so you can see more of the ground. Select the first button in the sculpt toolbar. Set Brush Size to 100 and Opacity to 28. Don't make it too bumpy or it will be impossible to drive around on. Drag the mouse over the ground in the Scene. It will lift up.

If the track gets buried, hold down the SHIFT key while brushing to rub out the changes.

Using the first 3 tools sculpt the landscape around the track to your liking. If you set the brush size to 5 you'll be able to sculpt inside the track.



Colouring

Colouring the terrain involves painting it on in layers. To begin, select the brush tool from the Terrain toolbar. Below this click on the Edit Texture button and then Add Texture.



In the Edit Terrain Texture box that opens, click on the left Select btton. A large window of different textures will

appear. Find the Grass (Hill) texture and select. Click on the Apply button. As this is the first texture selected the entire terrain will turn green. To add another colour, click on Edit Texture and then Add Texture. Find and add the texture GoodDirt. After you click Apply, make sure the new texture is selected in the Terrain editor. Then simply paint onto the terrain by holding down the left mouse button. Do this with as many colours as you like.

The Sky

Last but not least, lets add some sky. Select from the main menu, Window > Lighting. In the Popup Window, select the Scene Tab and then locate the section for Environment Lighting and Skybox. Click on the small round icon. Select the **Sunny2 Skybox** texture. Note not all these textures are suitable for the sky. Only skyboxes.

🖽 Lighting			
Object	Scene	Lightmaps	. 🔊 🖏
Environment Lig	hting		-
Skybox	Sunny2 Skybox		_ ⊙ [
Sun	None (Light)		0
Ambient Source	Color		;
Ambient GI	Baked		÷
Reflection Source	Skybox		+
Resolution	128		+
Compression	Auto		+
Reflection Intensity		0 1	
Reflection Bounces	0	1	
Precomputed Rea	ltime GI		
Realtime Resolution	1	texels per unit	-
	Auto	Build	-
0 non-directional lightma	ups 0 B		
	No I	Lightmaps	
Preview			

Trees and Other 3D Objects

To finish off your racing scene you can scatter some trees and other 3D objects around your environment. In the Project, locate the Terrain Assets folder. In there you'll find a Trees Ambient-Occlusion folder. This has a lot of trees you can drag into the Scene. For example, select Bamboo and drag it onto the terrain for some bamboo.

The Prefab folder also has other 3D objects such as fuel tanks, tires and oil drums. Note: To add more than one of each object simply drag and drop the



object from the Project into the Scene multiple times. Use the W Key to move the objects around, the E Key to rotate them and the R Key to scale them.



.... And Watch out for the Exploding Barrels!!!

Like to Learn More?

Unity is a free game development engine. It can be downloaded from <u>http://www.unity3d.com</u>.

I've two books that can help you get started with programming Unity to create your own desktop and mobile games. The books were written for students and are used in over 40 universities internationally as a text.

They are aimed at the novice and begin gently leading the reader through numerous simple exercises up to complete games.

Read more about them at:

http://www.holistic3d.com

Unity also has a comprehensive set of video tutorials for beginners at:

http://unity3d.com/learn

Contact Me

For more information or help with Unity or associated game creation matters feel free to contact me via email or on social media.

Penny de Byl Penny de Byl pdebyl@bond.edu.au Facebook: https://www.facebook.com/pages/Holistic3D/272575919455432 YouTube: http://www.youtube.com/c/holistic3d

More about Bond's Bachelor of Interactive Media and Design

HTTP://WWW.TINYURL.COM/BONDBIMD

