How you can follow with Unity 5.x and Unity 2017

If you are choosing to follow the course with Unity 5 or Unity 2017 instead of 4 then you will need to be aware of a few changes. The following is a list of the most common differences that students encounter. Don’t worry if you don’t understand them all yet but **bookmark this document** and come back if you find something doesn’t work like the lectures.

## GetComponent<Rigidbody2D>() instead of rigidbody2D

rigidbody2D is no longer a magical variable on every component. You now need to access it the same way as any other component.

For Example:

ball.GetComponent<Rigidbody2D>().velocity = new Vector2(0, 2f);

## Import Settings Texture Type

When importing pictures that are going to be used as sprites, select “Default” or “Sprite (2D and UI)” instead of “Texture” as Texture Type in the Import Settings.

## Changing the color of the particle system

This one is a little bit more technical but here is how it’s done in Unity 5.3:

ParticleSystem main = smokePuff.GetComponent<ParticleSystem>();  
main.startColor = gameObject.GetComponent<SpriteRenderer>().color;

How it’s done in Unity 5.5+:

ParticleSystem.MainModule main = smokePuff.GetComponent<ParticleSystem>().main;  
main.startColor = gameObject.GetComponent<SpriteRenderer>().color;

## SceneManagement instead of Application

Where previously we would use Application.LoadLevel to load a new level. In Unity 5 we now need to use SceneManager.LoadScene. Some of the names and usages are different so please take a look at the [SceneManager documentation](https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.html).

For example:

// At the top of your script  
using UnityEngine.SceneManagement;

// Load next scene from build settings  
SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

// Load new scene by name  
SceneManager.LoadScene("SceneName");

## OnLevelWasLoaded() in Unity 5.x

OnLevelWasLoaded() is deprecated as of Unity 5. Even though it might still work, we recommend to use the [SceneManager](https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.html). If you do not know about [delegates](http://www.unitygeek.com/delegates-events-unity/) yet, do not worry about that. Simply use the code below:

// at the top of your script  
using UnityEngine.SceneManagement;

private void OnEnable() {  
 SceneManager.sceneLoaded += OnSceneLoaded; // subscribe  
}

private void OnDisable() {  
 SceneManager.sceneLoaded -= OnSceneLoaded; //unsubscribe  
}

// This is the new OnLevelWasLoaded method. You may name it as you want  
// Make sure to subscribe/unsubscribe the correct method name (see above)  
private void OnSceneLoaded(Scene scene, LoadSceneMode loadSceneMode) {  
 // your code  
}

## Rotated enemy ships in Laser Defender

In case the enemy ships appear rotated, disable Apply Root Motion in the Animator Component of your enemy prefab.

## Smoothing loops and transitions between animation states

In case you are encountering issues with your animations, for example with a looping rotation of your axe (Glitch Garden section), select the keyframes in the Animation window, click your right mouse button and select ‘Auto’. If that does not work, try ‘Free Smooth’.



For further information on this topic, please read about [Editing Curves](https://docs.unity3d.com/Manual/EditingCurves.html) in Unity’s docs.

## 2D/3D sound

Set Spatial Blend to 0 in your AudioSource component to make the sound 2D.