TUTORIAL UPGRADE GUIDES

2D ROGUELIKE

This short guide will help you follow the 2D Roguelike tutorial series using recent Unity versions and the updated 2D Roguelike assets

You can find the video tutorials on our Learn page here:
https://learn.unity.com/project/2d-roguelike-tutorial

You can find the updated Survival Shooter assets here:
https://www.assetstore.unity3d.com/en/?_ga=1.60623208.498822501.1470663031#!/content/29825

To make best use of this guide, be aware of each video's timestamps before proceeding with the tutorial so that you know when to pause and review the Notes.

TIME

06. WRITING THE PLAYER SCRIPT

Once the script in the video is written, we need to add another variable with which we check whether the player character is currently moving.

Below the other variables towards the top of the script add the following line:

```
private bool isMoving;
```

Now, within the Move function, change the following line:

```
if(hit.transform == null)
```

to:

```
if(hit.transform == null && !isMoving)
```

Now, let's set isMoving to true within the SmoothMovement Coroutine. Add the following line above the sqrRemainingDistance declaration:

```
isMoving = true;
```

Within the same SmoothMovement Coroutine, add the following two lines after the `while(sqrRemainingDistance > float.Epsilon)` loop:

```
rb2D.MovePosition(end);
isMoving = false;
```

TIME

09. WRITING THE PLAYER SCRIPT

08:48 *Application.LoadLevel* has been replaced by *SceneManager.LoadScene*. To use SceneManager we need to add the namespace declaration for SceneManagement. Under the namespace declaration

```
using UnityEngine.UI;
```

Add the following line:

```
using UnityEngine.SceneManagement;
```

In writing the *Restart()* function in the tutorial we write:

```
Application.LoadLevel(Application.LoadedLevel);
```

Replace that line in *Restart()* with the following:

```
SceneManager.LoadScene (0);
```

06:39

02. PLAYER CHARACTER

As part of the introduction of the new SceneManager system for loading and unloading scenes, the *OnLevelWasLoaded* function has been deprecated. It has been replaced with the *SceneManager.sceneLoaded* event. In order use the *sceneLoaded* event we must add a delegate to get notifications when a scene has been loaded. For more information on events and delegates please see our lessons on events here and delegates here.

In order to use *SceneManager* we will first add it's namespace declaration. At the top of the **GameManager** script, after the namespace declaration for **UnityEngine.UI** add the following:

```
using UnityEngine.SceneManagement;
```

Instead of adding the *OnLevelWasLoaded* function to the GameManager script, **add the following three functions.**

```
//This is called each time a scene is loaded.
void OnLevelFinishedLoading(Scene scene, LoadSceneMode
mode)
{
     //Add one to our level number.
     level++;
     //Call InitGame to initialize our level.
InitGame();
}

void OnEnable()
{
     //Tell our 'OnLevelFinishedLoading' function to
start listening for a scene change event as soon as
this script is enabled.
SceneManager.sceneLoaded += OnLevelFinishedLoading;
}

void OnDisable()
{
//Tell our 'OnLevelFinishedLoading' function to stop
listening for a scene change event as soon as this
script is disabled.
//Remember to always have an unsubscription for every
delegate you subscribe to!
SceneManager.sceneLoaded -= OnLevelFinishedLoading;
}
```