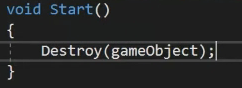
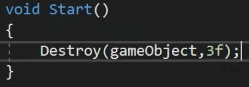
# **Unity C# Cheat sheet – By: Patricia & Tatum**

**Destroy Function:**



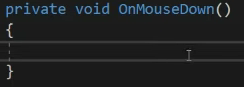
This deletes the object upon starting the program. | This is written in the Start.

**Destroy after given amount of time:**



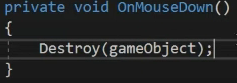
This deletes the object after given amount of time. | This is written in the Start.

**Detecting mouse clicks:**



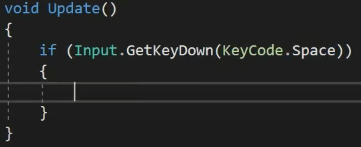
You can input a task to execute upon clicking the object with the mouse. Example down below.

**Destroy object upon mouse click:**



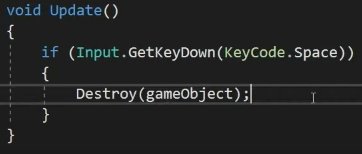
This destroys the object upon clicking it with the mouse.

**Take keyboard input:**



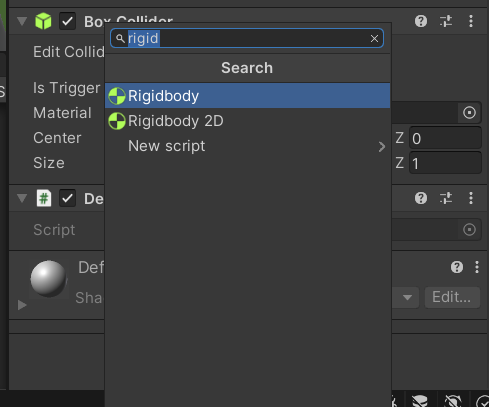
This is an if-statement that checks for keyboard input every frame. GetKeyDown = check if a key is being pressed. KeyCode = what key it is. | Needs to be written in the Update.

**Destroy object on keypress:**



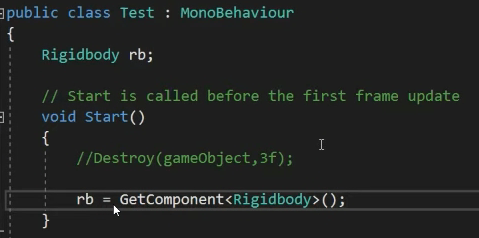
This will destroy the object when pressing space. | Needs to be written in the Update.

**Add Rigidbody:**

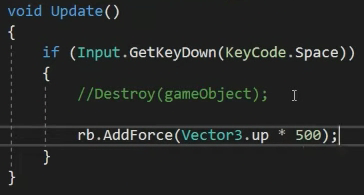


Select the object you want to add a Rigidbody to, then scroll down in the inspector and select “Add Component”. You can then search for the Rigidbody component.

**Refer to Rigidbody as ‘rb’:**

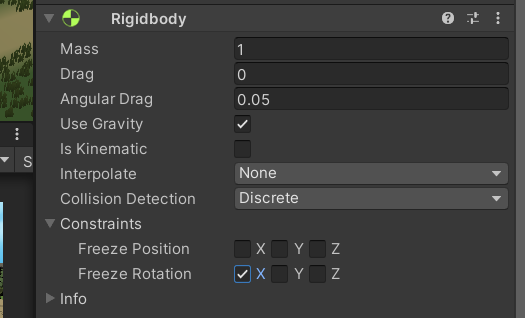


**Add force (upwards) to rb:**



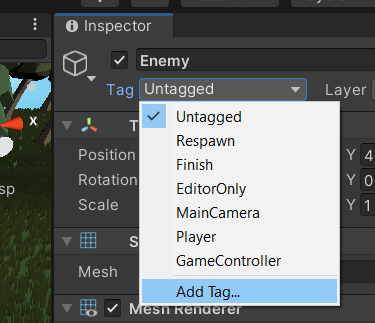
This moves the object upwards upon pressing space. If you’ve added a Rigidbody, it’ll come back down, creating a jumping motion. AddForce = makes the object move in a certain direction. 500 = the amount of force applied to move the object.

**Prevent an object from rotating on the move:**



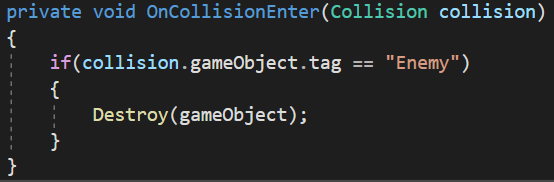
Go into Rigidbody, then check Freeze Rotation to prevent the object from rotating along that axis.

**Add a tag to an object:**



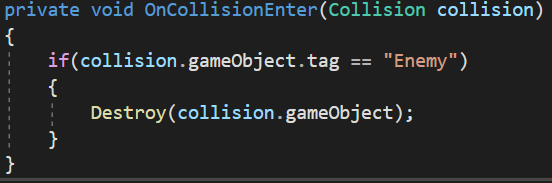
Select the object, then click on Tag in the inspector, Add Tag, then create the tag you want to make. Next up go back to Tag in the inspector and select the one you just created.

**Destroy an object after colliding with an Enemy:**



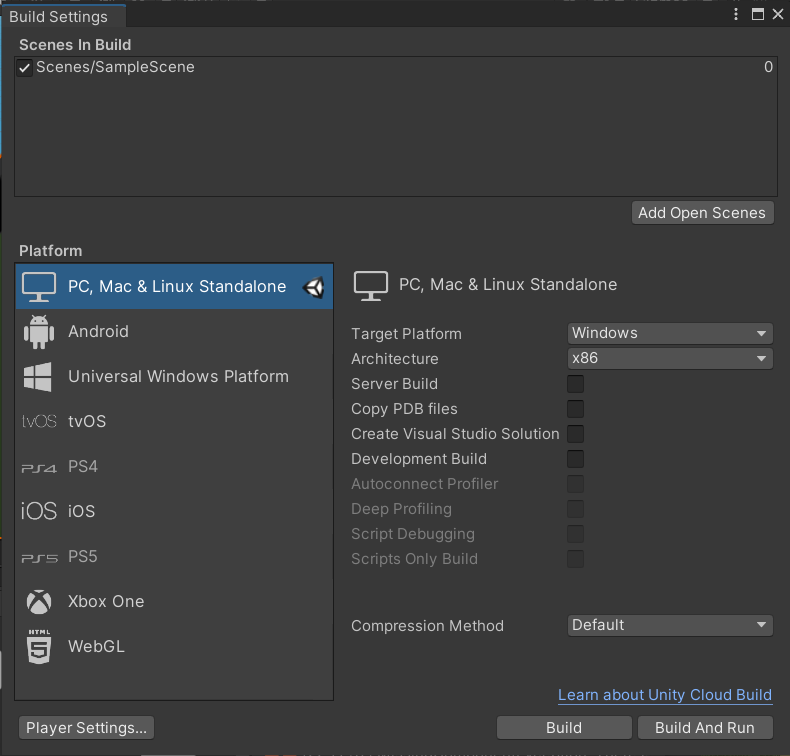
This code checks whether or not the colliding object has the tag “Enemy”. If it does, the gameObject will be destroyed. *(Collision = botsing)*

**Destroy an enemy after colliding with the gameObject:**



This code does the same as the previous one, except this time it destroys the Enemy instead of the gameObject.

**Switch from scene to scene:**

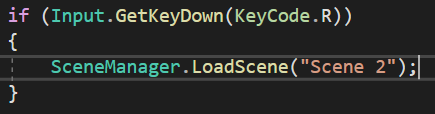


Go to File > Build Settings > Add Open Scenes. Then you close that window without pressing Build, and go to File > New Scene. Save that scene and go back to your original scene, which should be located in your Assets in the Scene folder. Finally, you need to go back to the Build Settings, open your new scene in the background and then press “Add Open Scenes” and close it once again.

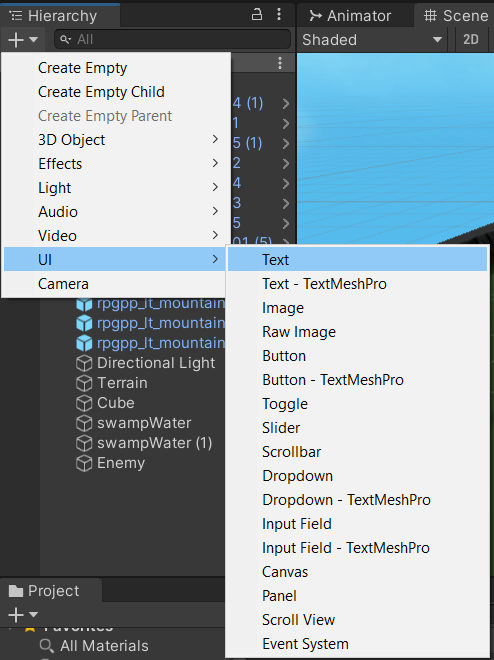
Next up you will go to your script. First, you want to import a new namespace:



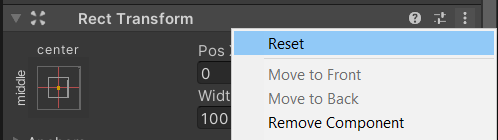
Then you can assign a key that when pressed will switch your program over to the next scene:



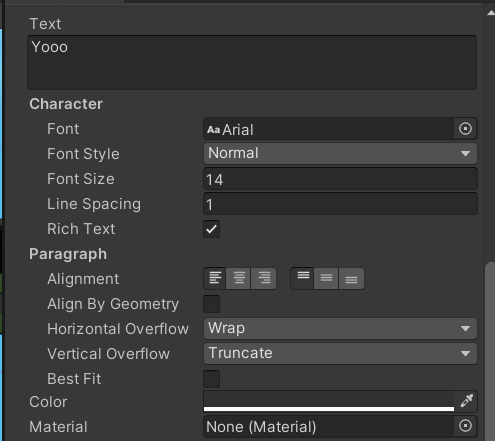
**Display Text on screen:**



First, go to the Hierarchy, click on the plus sign and hover over UI, then select Text.

Once you’ve done that, select your text and press “Reset” in the inspector to reset its position.

Then you can go to 2D mode to see it better. You can change the font size along with other properties in the inspector.

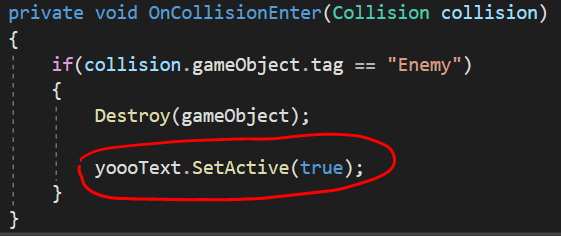


Next up you need to make a gameObject with the name of your text:

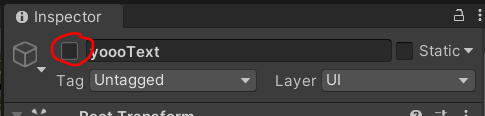


**This is written in the start.**

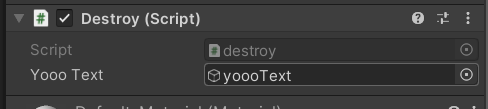
Then, you can display it when the enemy collides with your gameObject as follows:



At the start you need to disable visibility of the text by unchecking this box:



Then you select your gameObject and drag the text into the text variable under Scripts in the inspector:



This should work.

Result:



**WASD Movement Script:**

First, you want to get rid of the Start function. We only need the Update function.