**Player-Enemy interaction:**

This tutorial will enable the player to kill the enemy by jumping on it and enable the enemy to kill the player, forcing them to respawn at a specified spawn point.

Step 1: creating the spawnpoint

To create the spawnpoint for the player, create and empty object and place it where you want the player to spawn when killed, rename this empty object to “SpawnPoint” and give it a tag with the same name.

Step 2: creating the kill hitbox

We need a hitbox that when touched by the player, will destroy the enemy, to do this, create another empty object and rename it “KillHitbox” with the same name on its tag. Then drag this object over the enemy game object in the hierarchy, this will make it a child of that object, allowing it to stay attached to it and move alongside it which is what we want.

Remember to scale down the collider size for the enemy by going onto the boxcollider2d tab in the inspector and pressing this button Scale the top of the box collider down so that there is room for the killhitbox to sit on top.

Use the same method to scale down and place the killhitbox, so that the top of its collider is in line with the top of the enemy.

Step 3: code for enemy killing the player

This code will make it so that when the player collides with the enemy’s hitbox (anywhere that isnt the killhitbox) it will reset the plays position, creating a respawn mechanic.

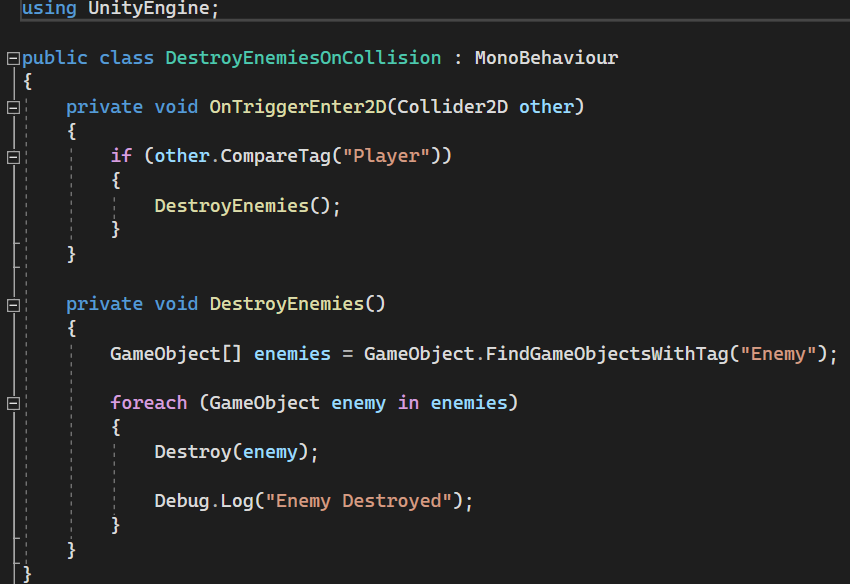
Insert the following code to a new script attached to the enemy called “ResetPlayerPosition”



We use the same OnTriggerEnter2D method as before and use 2 if statements to detect what's colliding with the enemy's collider, if the object colliding is tagged with “Player” it will take that object, find the object tagged with “SpawnPoint”’s location and transform the player to its position. The debug log will then state that the enemy killed the player.

Step 4: Code for player killing enemy

This code will be attached to the “KillHitbox” and will destroy the game object tagged with enemy; it will then display that the enemy has been destroyed in the debug log. The code is as follows.



The is code is like the one used for when the player is killed by the enemy. This one compared the tag of what collided with it, if the object tagged as play collided with it, it will destroy the enemy. The second private void defines what “DestroyEnemies” does, in this case it destroys each instance of an enemy. The debug log then displays “Enemy Destroyed”.