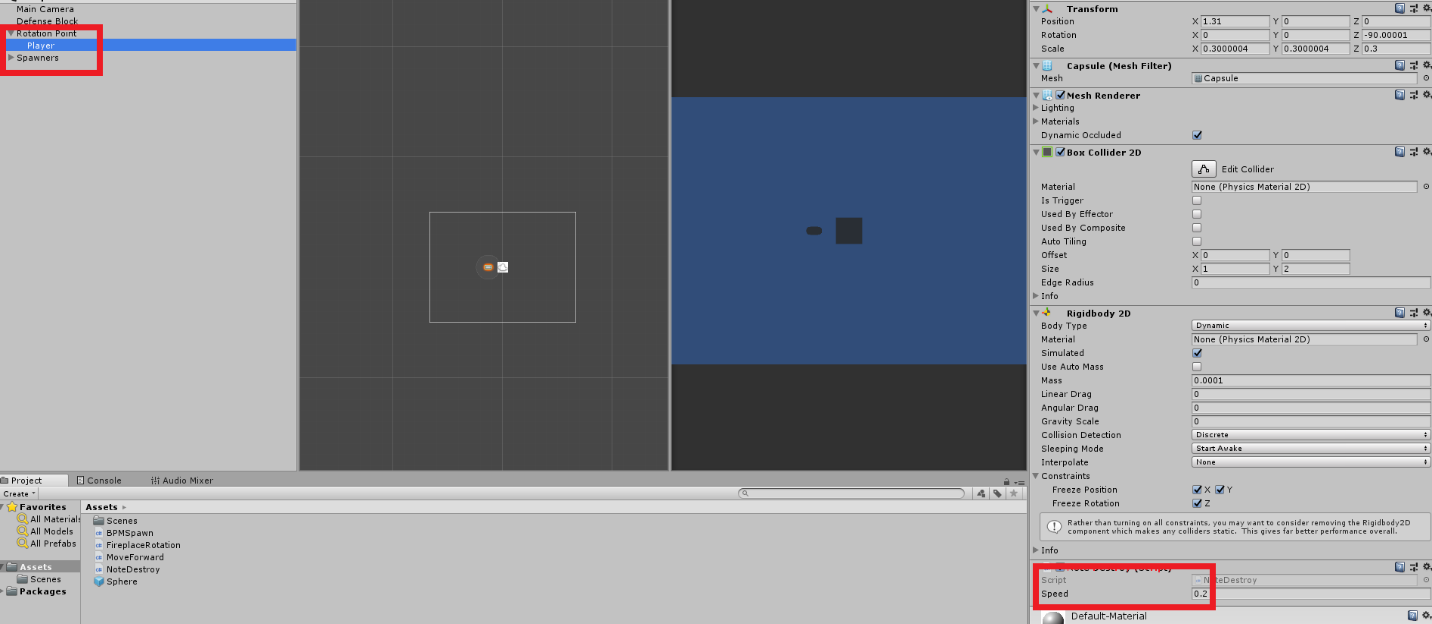
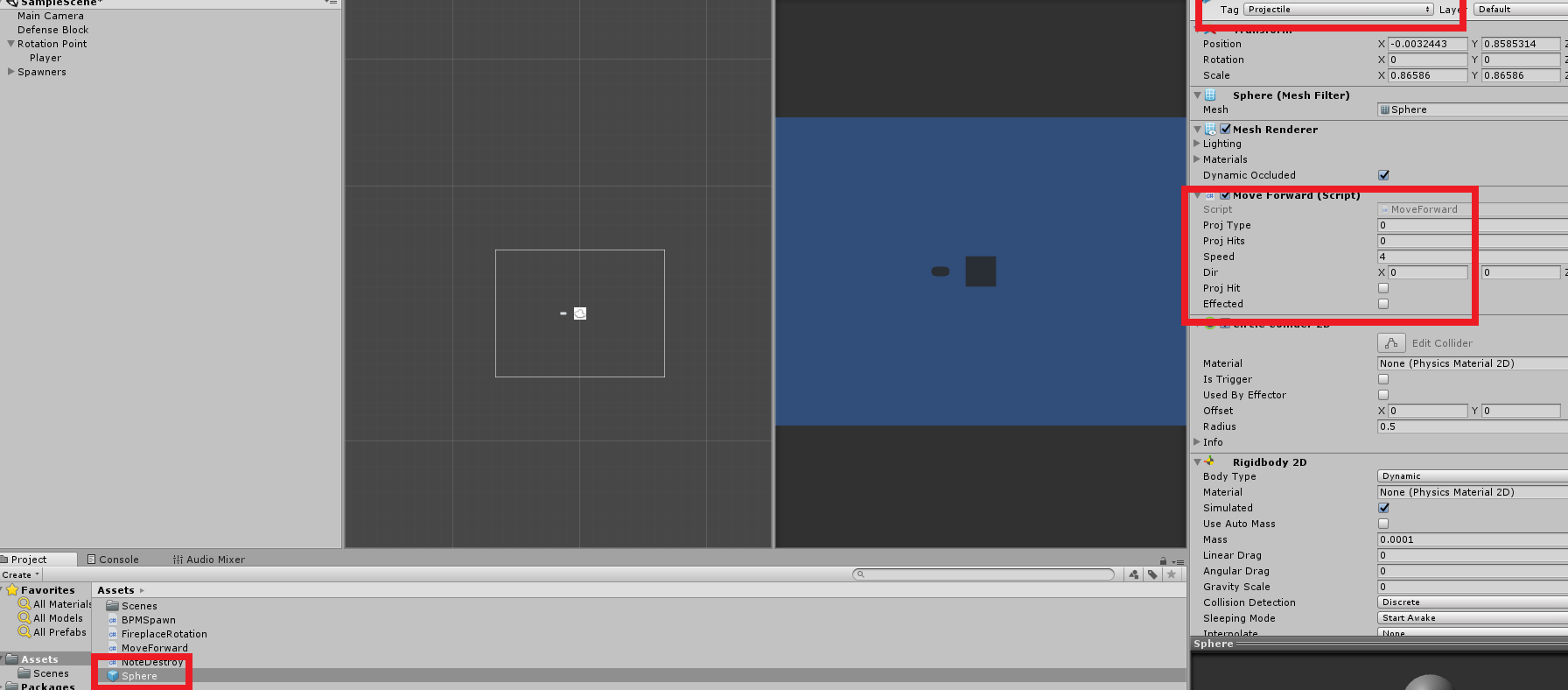
**Step 1 – Creating the scene**

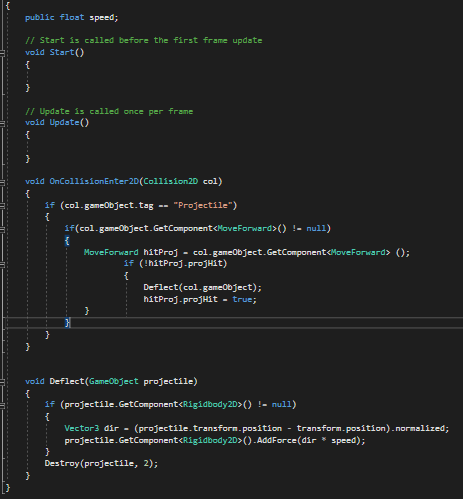
The scene will be a continuation from the last package tutorial. The key differences are that the player will have the “Projectile hit” script attached to it, and the projectile (prefab) will have the “Projectile movement” attached to it. Additionally, you will need to create a “Projectile” tag in order for the player to detect what exactly it has hit upon colliding with the projectile, allowing the script to work. The scene should look as follows:





**Step 2 – Prefab hit script**

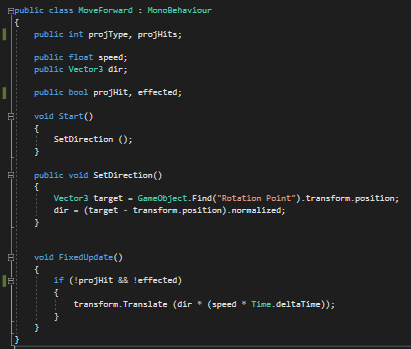
The prefab hit script will be used in order to determine whether or not the prefab has been successfully hit by the player, of which it will then call the Prefab movement script in order to fly in the opposite direction of the player. The script looks like the following:



This script is relatively simple in nature, as all it is doing is detecting whether or not the external object (named “col”) in the OnCollisionEnter2D void has the tag “Projectile”. If it does – and the object **does** have the MoveForward script attached to it, it will run a new void named “Deflect” and set two bool statements to “true” on the MoveForward script.

The deflect void does exactly what it says it will; it will attempt to deflect the object in the opposite direction of the player by adding force to its rigidbody. Additionally, the projectile is set to destroy itself after a float duration of “2”. However, this is not enough, as I also want it to be able to determine what is the “opposite” direction of the player, which is where the prefab movement script comes in.

**Step 3 – Prefab movement**



This script cannot work without the Prefab hit script. All this script does is reference the position of the Rotation Point (the zeroed-out point of which the player is parented to) and moves the projectile in the opposite direction relative to that point.