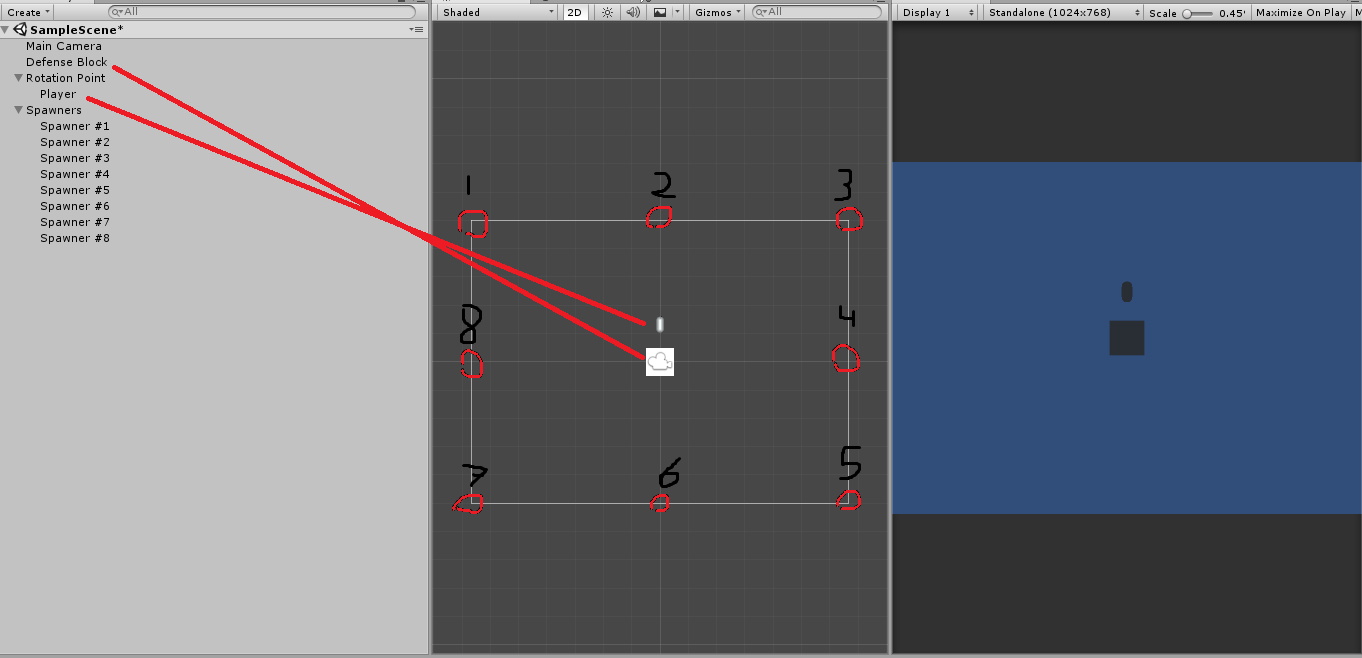
**Step 1 – Creating the scene**

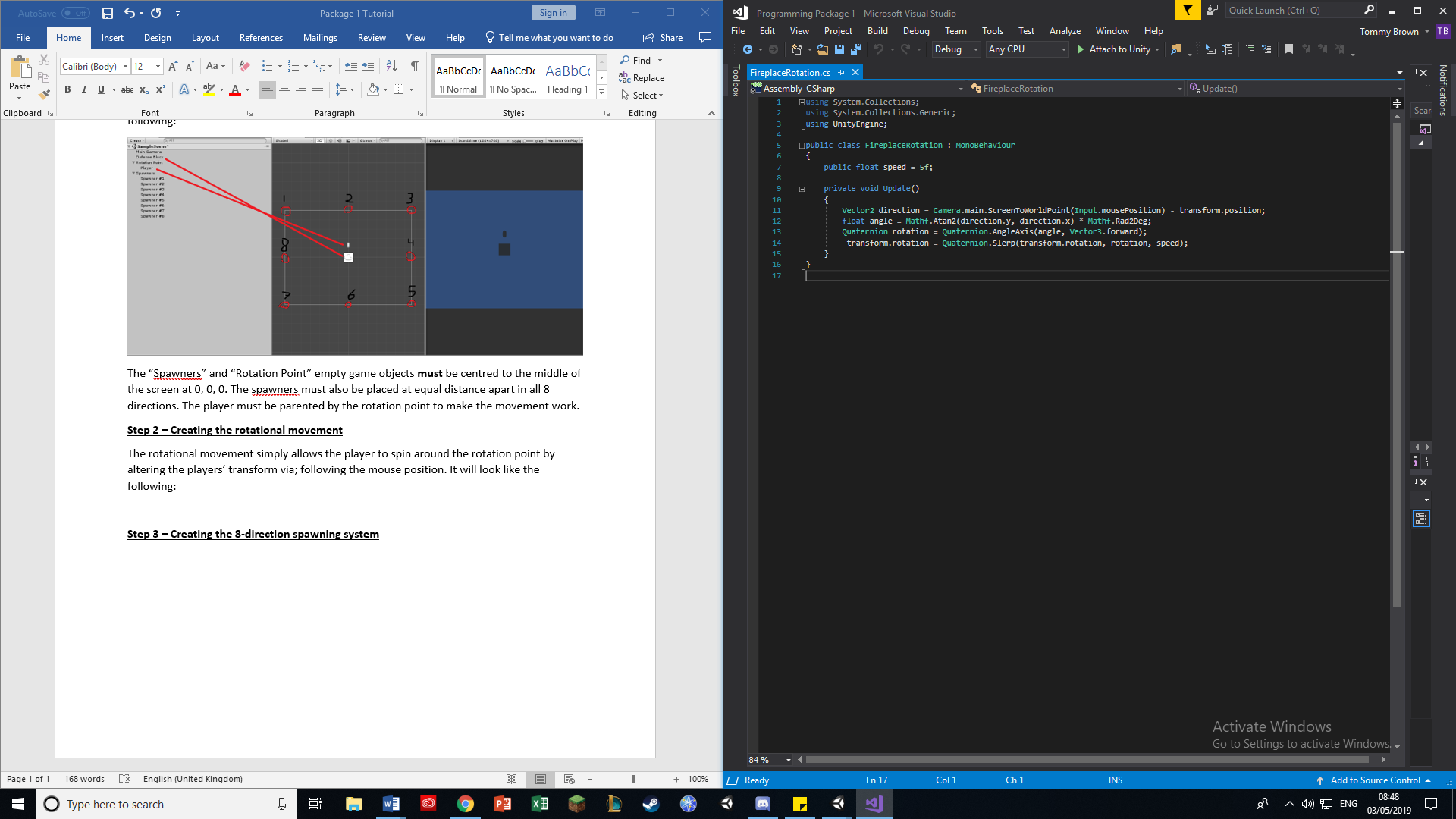
The unity scene is comprised of a box (which will act as the object to defend), a capsule (the player), 1 empty game object (to act as the point of rotation for the player), 1 sphere prefab (to spawn) and 9 additional empty game objects (which will act as the spawn points). The scene will look like the following:



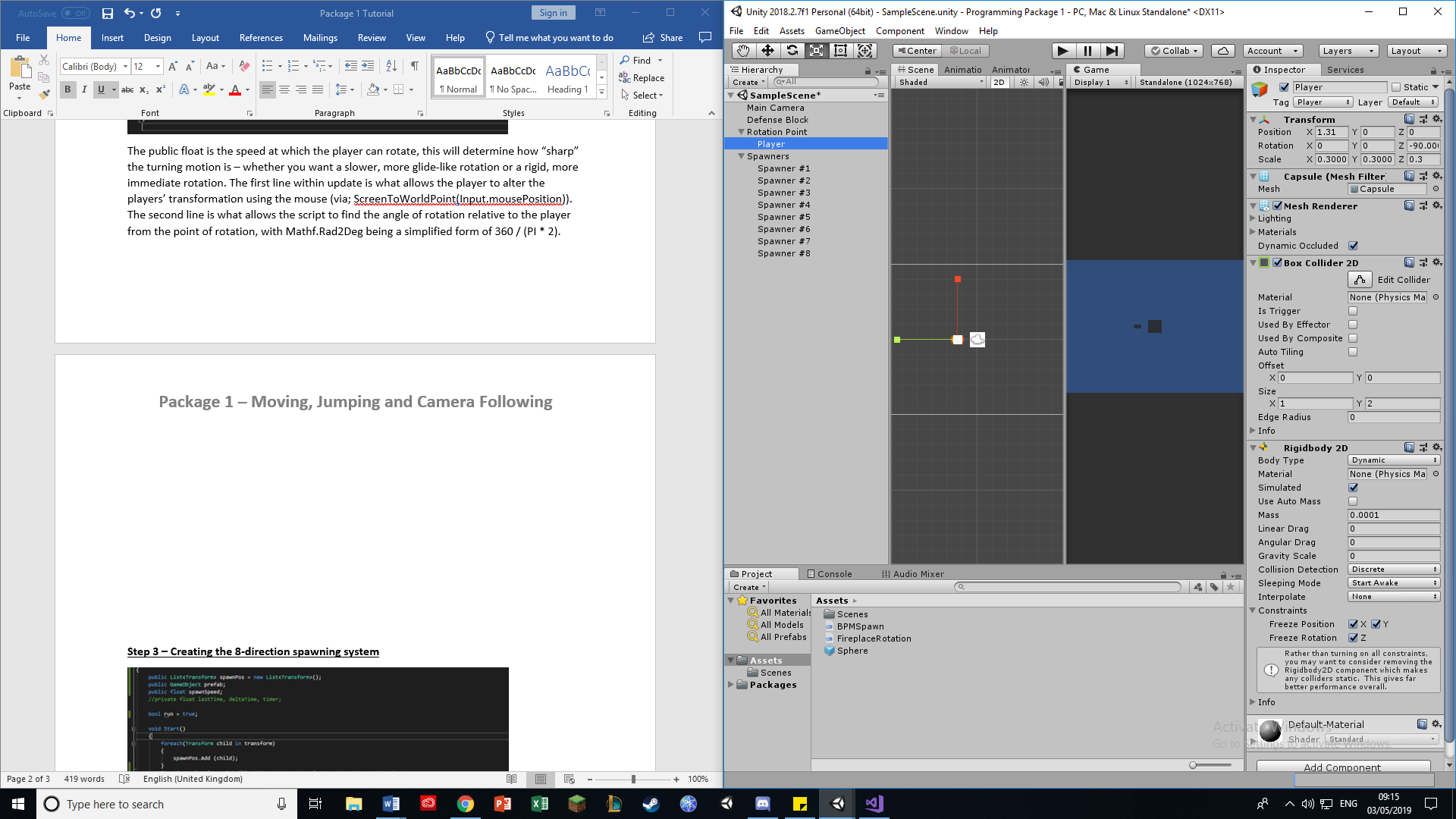
The “Spawners” and “Rotation Point” empty game objects **must** be centred to the middle of the screen at 0, 0, 0. The spawners must also be placed at equal distance apart in all 8 directions. The player must be parented by the rotation point to make the movement work.

**Step 2 – Creating the rotational movement**

The rotational movement simply allows the player to spin around the rotation point by altering the players’ transform via; following the mouse position. It will look like the following:

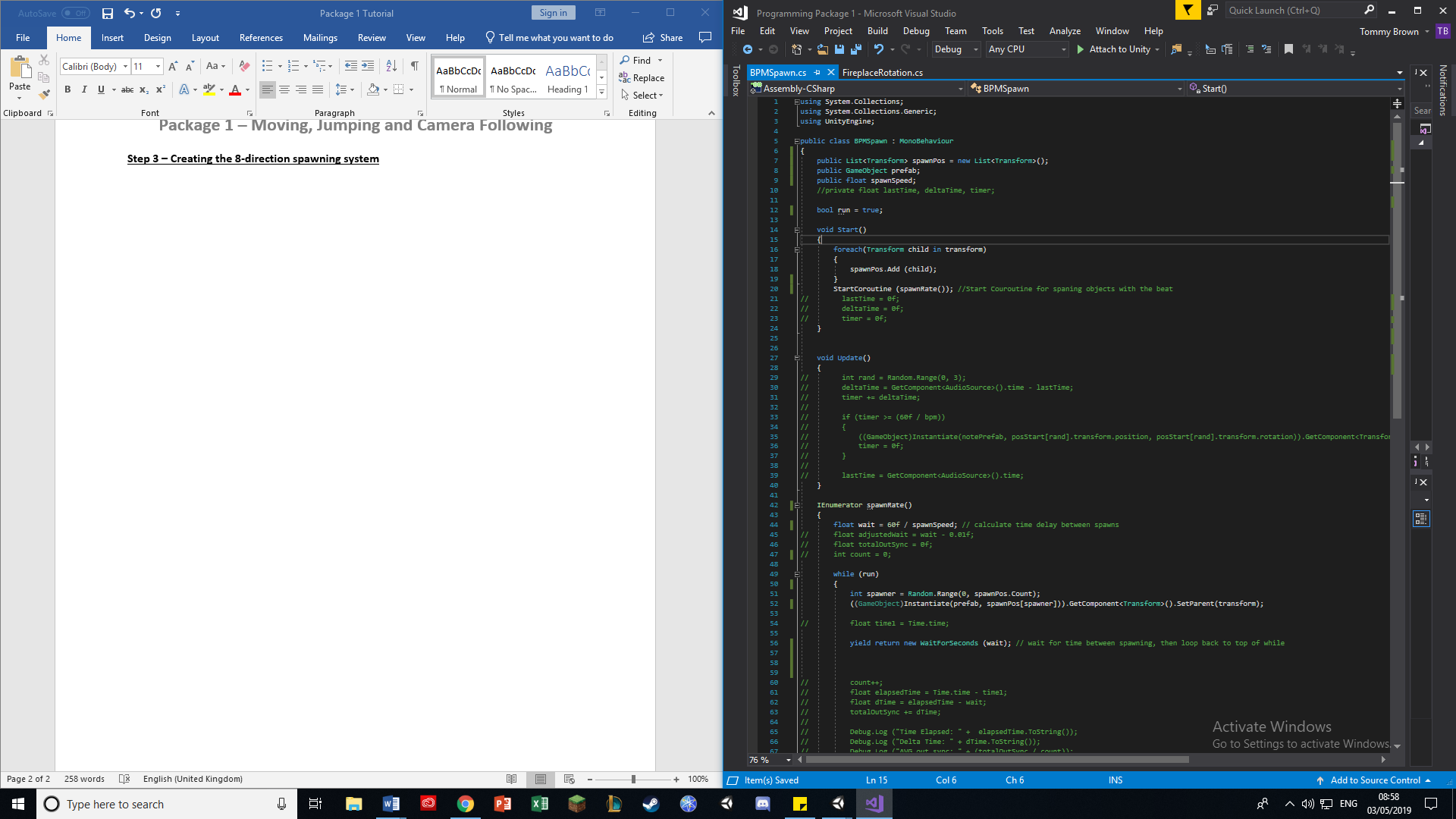


The public float is the speed at which the player can rotate, this will determine how “sharp” the turning motion is – whether you want a slower, more glide-like rotation or a rigid, more immediate rotation. The first line within update is what allows the player to alter the players’ transformation using the mouse (via; ScreenToWorldPoint(Input.mousePosition)). The second line is what allows the script to find the angle of rotation relative to the player from the point of rotation, with Mathf.Rad2Deg being a simplified form of 360 / (PI \* 2).

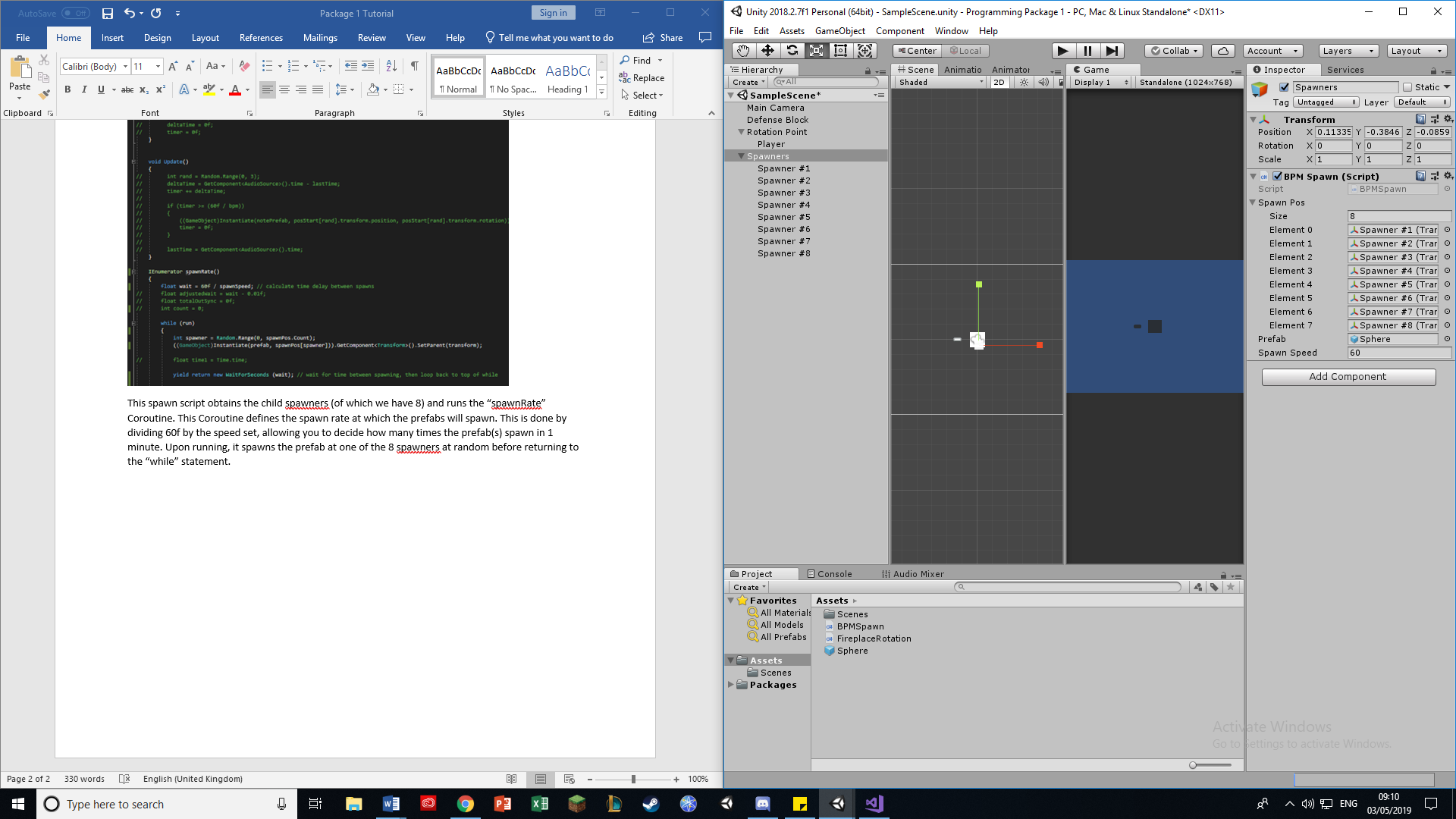


It is also worth noting that the player **must** have a box collider and rigidbody (with the constraints and values set accordingly to the screenshot shown). This is mostly for the sake of the next package, though will bare slight importance to the functionality of this package too.

**Step 3 – Creating the 8-direction spawning system**



This spawn script obtains the child spawners (of which we have 8) and runs the “spawnRate” Coroutine. This Coroutine defines the spawn rate at which the prefabs will spawn. This is done by dividing 60f by the speed set, allowing you to decide how many times the prefab(s) spawn in 1 minute. Upon running, it spawns the prefab at one of the 8 spawners at random before returning to the “while” statement.



As you can see, the spawn position size must be set to 8 (to accommodate the 8 different spawners), of which all 8 spawners **must** be children of the “spawners” empty game object to have the Coroutine take effect. Additionally, you must create and implement a sphere prefab by dragging it into the project screen at the bottom, and then dragging it onto the empty prefab slot. Once you play, you will see sphere prefabs spawning around the player in 8 different locations, though they do not move (yet).