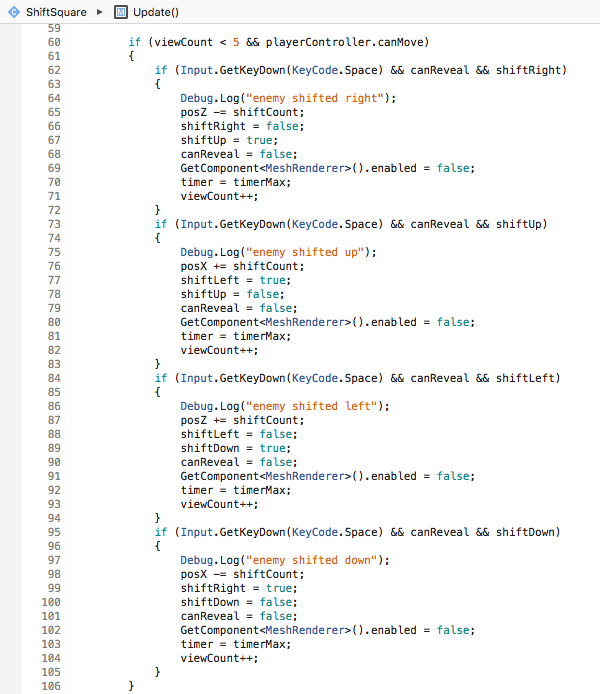
**Component 4 – Movement patterns and game assembly**

In this component tutorial, I’ll be utilising the enemyShift script to create an enemy that is capable of moving in a square.

To do this, copy the contents of the EnemyShift script and paste this in to a new script called ShiftSquare.

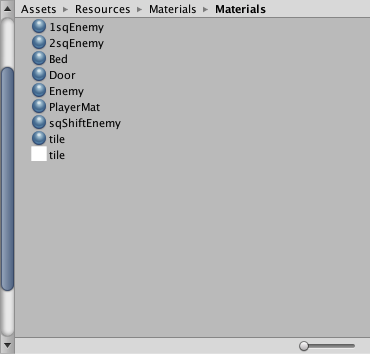
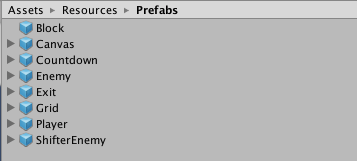
Since we have the basis from our enemyShift script, all we’ll have to do is change the order in which our shift bools are set. Since I want to move the enemy in anti-clockwise square, I’ll start with the shiftRight bool and set shiftUp to true – then I’ll set shiftLeft, shiftDown and shiftRight in order, making sure to set the previous direction to false.

****

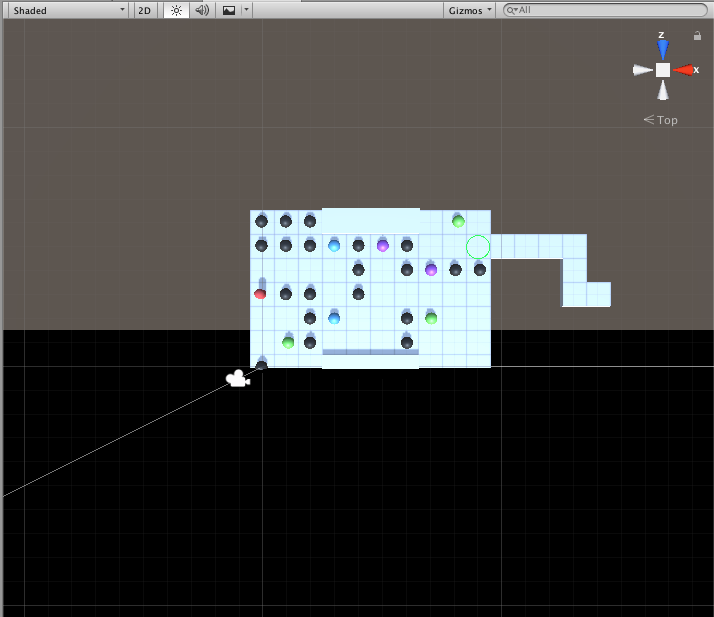
In turn, this creates an anti-clockwise moving enemy.

Final game assembly

Now that I’ve set up my standard vanishing and moving enemies, I can implement them in to my game. To ensure that I my enemies are distinguishable visually and as game objects, I created different coloured materials for each enemy type and made them in to prefabs (as well as other objects to make level creation quicker).



I then assembled my enemies on to my grid with the player, set out the level boundaries using my player controller script and placed standard enemies and movement enemies on to my grid. To indicate the exit for the level, I added a green circle using the exit script.



Blue enemies always move one square right first before moving one square left

Purple enemies always move two squares right before moving two squares left

Green enemies always move one square anti-clockwise.

