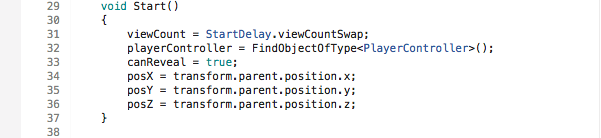
**Component 3 – Moving Enemies**

For this component, I’ll be creating vanishing enemies that can move in one of 4 directions upon the player using the reveal ability.

Copy the contents of the EnemyVanishscript and paste this in to a new script called EnemyShift. Since we’ll need to reference the current positions of the enemies before we move them, we’ll create floats called posX,Y and Z which we’ll later assign in the Start class. We’ll also create a public integer called shiftCount to determine how many spaces the enemy will move. Finally I’ll add some public bools to determine which way the enemy will move – these bools will be called shiftLeft,Right,Up and Down and will be set to false.

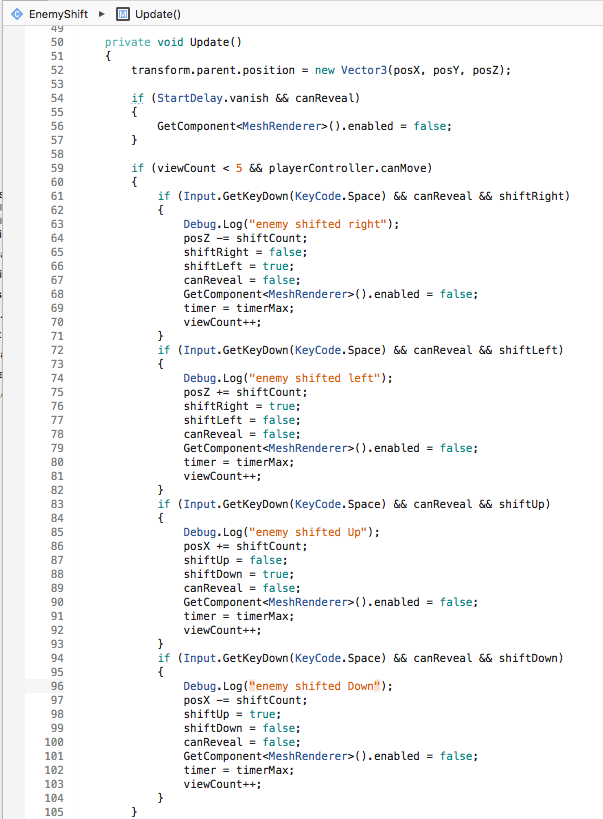


In start, we’ll check the current position of the enemy and assign it to our float variables. To do this we’ll use the code pos(x/y/z) = transform.parent.position.(x/y/z);. Since we want to move both the collider and mesh, w use transform.parent.position instead of transform.position.

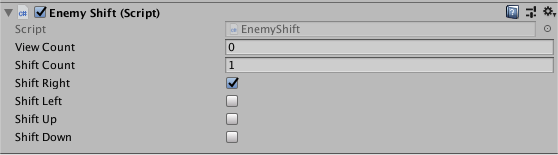


Next, we’ll move our enemies using a system of if statements. To do this, we’ll add our shift variable to our if statement. We’ll start with enemies that move right by typing && shiftRight. Since we’ll want out enemy to move right, we need to decrease its position on the Z axis. By typing posZ -= shiftCount, our enemy’s position on the z axis will be decreased by whatever we set our shiftCount to in the inspector. Since I want my enemy to move back to it’s original position, I no longer want it to shiftRight (which we’ll set to false) but instead shiftLeft (which we’ll set to true).

After you’ve done this if statement, you can then copy and paste it 4 times and adjust which shift bool is detected, which axis we want to increase or decrease (+/-) and which shift bools are set to true or false.



In our inspector, we can now set the shift count to determine how many block the enemy will move and what it’s starting movement pattern will be before it’s reverse. For example, if an enemy is set to shift right, it will be set to shift left after the space bar is pressed (and vice versa).



In our final component tutorial, I’ll be using this script to create an enemy that can move in a square pattern and assembling the final elements of the game.