**Enemy Patrol Tutorial**

First of all, make sure you have a Unity 3D empty project open. Once you have that open, create the 3D object ‘cube’ in the hierarchy by right clicking. Name this cube “Platform”.

Set the position of ‘Platform’ to; X= 0, Y= 0.5, Z= 0. Set the scale of ‘Platform’ to; X = 20, Y= 0.5, Z = 20. Right click in the hierarchy and create a 3D object ‘Capsule’. Name this capsule “Enemy”. Set the Y position of the capsule to 1.73, and make sure the X and Z values are both 0.

Once you have set the position and scale of the platform and Enemy, select the camera in the project. Set the position of the camera to; X= 0, Y= 9.7, Z = 18. Change the camera X rotation to 26.

Right click in the hierarchy and create an empty object, set the Y position to 1.75. Rename this object to “Waypoint”. Duplicate this object 3 more times. Rename them in their respective numbers 2, 3 and 4. Change the position of the first waypoint to Z = 5, the second waypoint to Z = -5, the third waypoint to X = 5 and the final waypoint to X = -5.

By right clicking in the ‘asset folder’ in the project, create a script called “Patroller”. Double click the script to open Visual Studio (or the coding software you use).

Once the script is open, create some space above ‘Void Start’. In this space type out the following **bold** code:

*Public class Patroller : MonoBehavior*

{ **public Transform[] waypoints;  
public int speed;**

**private int waypointIndex;  
private float dist;**

These will be used and referenced to throughout the rest of the code. In the curly brackets ( { } ) ‘Void Start’ function, type out this **bold** code:

*void Start()*

*{*

**waypointIndex = 0;  
transform.LookAt(waypoints[waypointIndex].position);**

*}*

This code is for the enemy to look at the different waypoints and determine where the waypoints are. Go past ‘void Update’ and start to type another void a couple of lines after, type this **bold** code:

**void Patrol()  
{**

**transform.Translate(Vector3.forward \* speed \* Time.deltaTime);**

**}**

This code is to determine the speed at which the enemy moves towards the different waypoints. Under the above code, type out the following **bold** code:

**void IncreaseIndex()**

**{**

**waypointIndex++;  
if(waypointIndex >= waypoints.Length)**

**{  
waypointIndex = 0;  
}  
transform.LookAt(waypoints[waypointIndex].position);**

**}**

This code is checking if the waypoints are over 4 and if it is, the enemy will go back to the first waypoint. The last bit of code that needs to be typed out is in the curly brackets for ‘void Update’. Type out the following **bold** code.

*void Update()*

*{*

**dist = Vector3.Distance(transform.position, waypoints[waypointIndex].position);  
if(dist < 1f)  
{  
IncreaseIndex();  
}  
Patrol();**

The code above checks the distance of the waypoints and the patrolling enemy. It also makes sure that our enemy is patrolling and moving so it can know which waypoint to go to next.