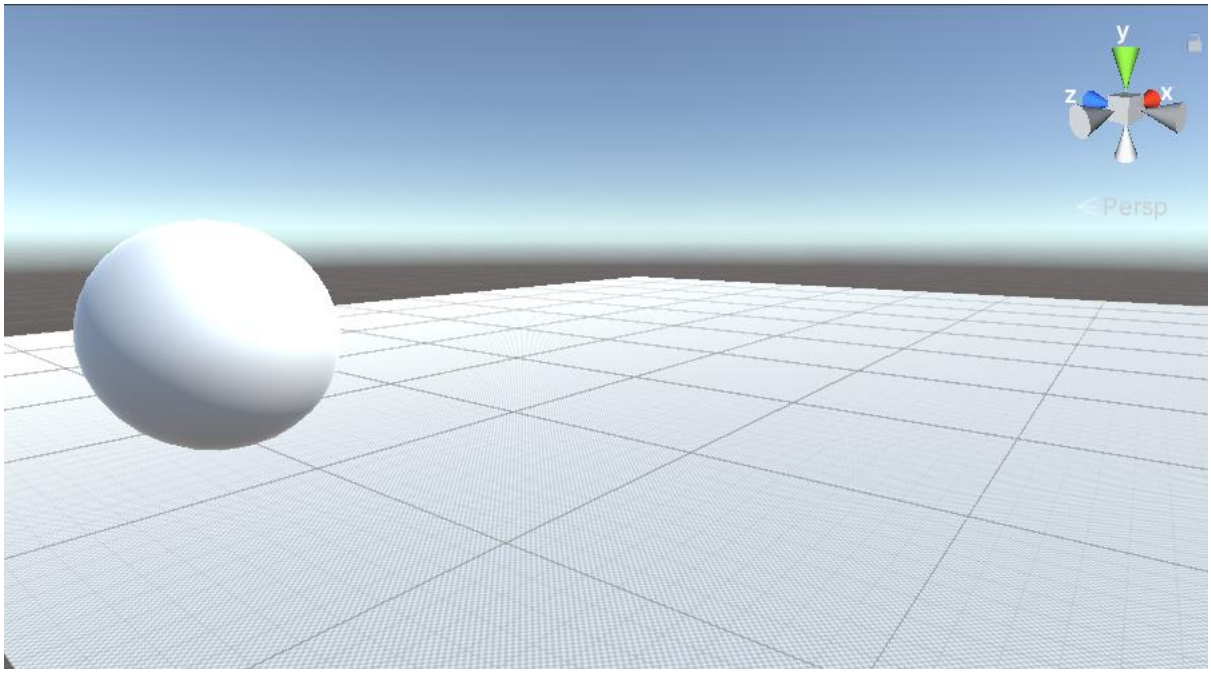
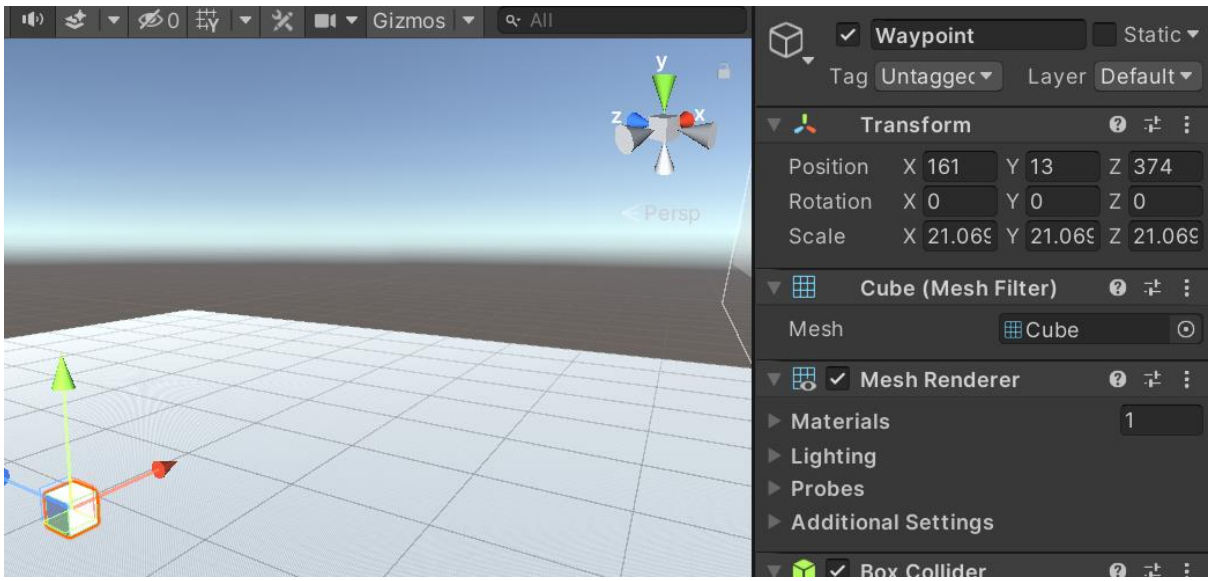
For this tutorial, im going to explaining how to make an easy enemy patroller script.

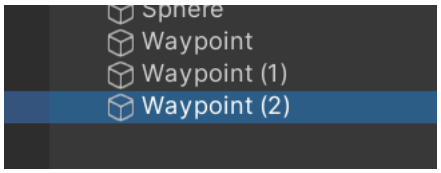
For my enemy I have created a sphere to represent my AI.

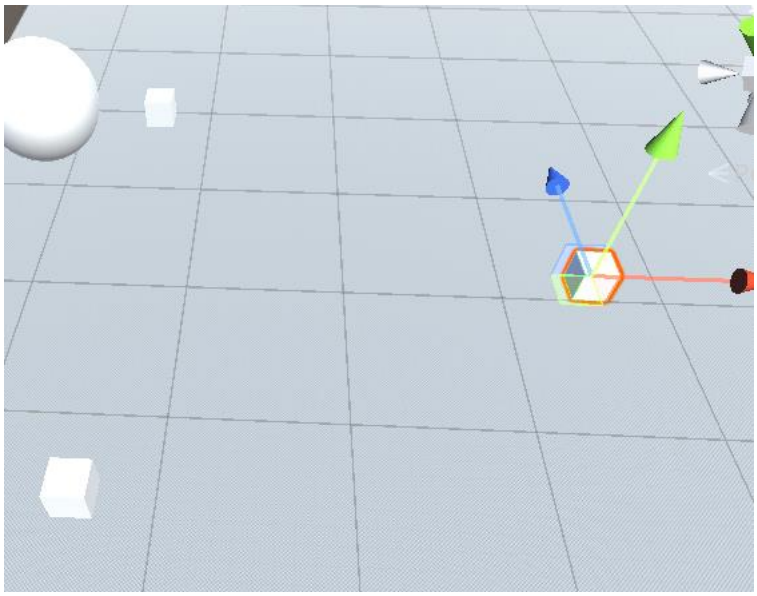


Add a cube to your scene and rename it ‘Waypoint’.

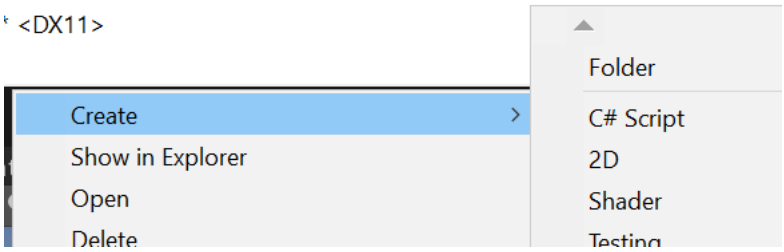


Duplicate your waypoint twice and move them into position. I laid mine out into a simple triangular formation.





Next, we are going to want to star working on our scripts. Right-click in your assets folder, go to create, and select C#Script.



Rename this script ‘Patroller’ and double click to open it.

First, were going to want to start adding our variables. The first one will be our transform variable which will reference the position our waypoints. The second one is going to reference our speed. The third one will be our waypoint index which will reference which waypoint we are on in the array. And lastly we have a float which keeps track of the distance between the AI and the current waypoint.

```
Unity Script | 0 references
public class Patroller : MonoBehaviour
{
    public Transform[] waypoints;
    public float speed;

    private int waypointIndex;
    private float dist;
```

In the start funtion, set the waypoint Index to 0. This is going to start us off at the first waypoint of the array. We also want our transform to be facing towards the waypoint.

```
// Start is called before the first frame update
Unity Message | 0 references
void Start()
{
    waypointIndex = 0;
    transform.LookAt(waypoints[waypointIndex].position);
}
```

Below our update function, create another function called ‘Patrol’. This function is going to control the AI’s movement.

```
1 reference
void Patrol()
{
    transform.Translate(Vector3.forward * speed * Time.deltaTime);
}
```

Underneath that, create another function called (IncreaseIndex). This is going to be used for incrementing our index. If waypoint index has gone out of range, reset back to 0. We then want the AI to be looking at the new waypoint.

```
1 reference
void IncreaseIndex()
{
    waypointIndex++;
    if (waypointIndex >= waypoints.Length)
    {
        waypointIndex = 0;
    }
    transform.LookAt(waypoints[waypointIndex].position);
}
```

In our update function, we want to check the distance between the AI and the waypoint. If the distance is within the range of our waypoint, we want to increase our Index and reference our 'Patrol' function to make sure that the AI is constantly moving.

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

Add the patroller script to the AI and reference your waypoints. Set the speed to about 2 and you should have a working enemy patroller.