Coding Assignment Tutorial #3

Move Objects from A to B (Waypoints)

In this Code I'll be making the object move from A to B as obstacles

1. First create an object and named it "Obstacles", this object could be a cube or any other shape, so I decide to use a cube instead. Then add a script in the right hand side of the screen in the "Inspector". The script should be called "Waypoints".
2. In the "Waypoint" script enter these codes bellow.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Waypoints : MonoBehaviour {

    public GameObject[] waypoints;
    int current = 0;
    float rotSpeed;
    public float speed;
    float WPradius = 1;

    // Update is called once per frame
    void Update () {
        if (Vector3.Distance(waypoints[current].transform.position, transform.position) < W
Pradius) {
            current++;
            if (current >= waypoints.Length)
            {
                current = 0;
            }
        }
        transform.position = Vector3.MoveTowards (transform.position, waypoints [current].t
ransform.position, Time.deltaTime * speed);
    }
}
```

- "Public GameObject[] waypoints" are arrays of game objects which will be our way points in the game world. "int current = 0" is used for which way point in the array we are going and headed to. "float rotSpeed" is used for the rotation for the object we are applying to. It is not necessary in this project as we are using a cube not an actual NPC. "public float speed" is used for how fast the object is moving, in this case between point A to B.
- "float WPradius = 1" if the object is in within the 1 radius, it means that the object has reach the waypoint so now it's able to head to the next one. This is used because the object sometime misses the waypoint as we use a "empty game object" as our waypoint.
- If (Vector3.Distance(waypoints[current].transform.position, transform.position) < WPradius) Means that if the distant between the object and the waypoint is less than the WPraidus. We add 1 to the current value because it means we can be able go to the next waypoint. Which is done at "current++; if (current >= waypoints.Length) {Current = 0};". "If current >= waypoints.Length" means that it resets to zero if there are no more current waypoint.
- The last bit of code which consist of "transform.position" basically moves the object to the current waypoint position in the world.

3. The third step of the tutorial is for us to make an empty game object which is located in the hierarchy. Created how many empty objects depending on how many way points you it to be. You can name the first way point as “left” and the second waypoint “right”.
4. In the “Obstacles” objet you will see this code located bellow. The size is the amount of waypoints you are going to make, in this case I did two which is for the left and right. Drag in the empty object that you named and order it around. After doing that you have to move the named empty object in the game world, meaning that you set the way points where you see fit and should conclude the waypoint tutorial.