

Explanation of DroneMotionPlanner.cs Code

1 Overview

This script is written in C# for Unity and is used to control a drone moving along a predefined path. The path is represented as a list of waypoints in 3D space, and the drone moves through them sequentially at a specified speed.

2 Code Components

2.1 Path Points

```
public List<Vector3> pathPoints = new List<Vector3>();
```

Purpose: Stores the list of 3D waypoints that the drone will follow. Each point is represented by a `Vector3`.

2.2 Speed

```
public float speed = 5f;
```

Purpose: Specifies the movement speed of the drone (in units per second).

2.3 Current Point Index

```
private int currentPoint = 0;
```

Purpose: Tracks the index of the waypoint the drone is currently moving toward. It starts at 0, indicating the first point in the path.

3 Main Methods

3.1 Update Method

```
void Update()  
{  
    MoveAlongPath();  
}
```

Purpose: Called by Unity every frame to continuously move the drone along the path.

3.2 MoveAlongPath Method

```
private void MoveAlongPath()
{
    if (currentPoint >= pathPoints.Count)
        return; // Reached the end of the path

    Vector3 target = pathPoints[currentPoint];
    Vector3 direction = target - transform.position;
    float step = speed * Time.deltaTime;

    if (direction.magnitude < step)
    {
        transform.position = target;
        currentPoint++;
    }
    else
    {
        transform.Translate(direction.normalized * step, Space.World);
    }
}
```

Purpose: Moves the drone step-by-step toward the next waypoint in the path.

Steps:

1. Check if the path is complete:

```
if (currentPoint >= pathPoints.Count)
    return;
```

Stops the function if the drone has reached the last waypoint.

2. Set the target point:

```
Vector3 target = pathPoints[currentPoint];
```

Identifies the next waypoint to move toward.

3. Calculate the direction:

```
Vector3 direction = target - transform.position;
```

Computes the vector from the drone's current position to the target.

4. Calculate the step size:

```
float step = speed * Time.deltaTime;
```

Determines the distance the drone should move this frame.

5. Check if the drone is close enough to the target:

```
if (direction.magnitude < step)
{
    transform.position = target;
    currentPoint++;
}
```

If the drone is within the step size of the target, move it directly to the target and update the index to the next waypoint.

6. Move toward the target:

```
else
{
    transform.Translate(direction.normalized * step, Space.
        World);
}
```

Otherwise, move the drone closer to the target using a normalized direction vector scaled by the step size.

3.3 UpdatePath Method

```
public void UpdatePath(List<Vector3> newPath)
{
    pathPoints = newPath;
    currentPoint = 0;
}
```

Purpose: Updates the drone's path with a new list of waypoints.

Parameters:

- **newPath:** A new `List<Vector3>` representing the updated waypoints.

Behavior:

- Replaces the existing `pathPoints` with `newPath`.
- Resets `currentPoint` to 0 so the drone starts from the beginning of the new path.

4 How It Works

1. Attach this script to a drone object in Unity.
2. Define the waypoints (`pathPoints`) either in the Unity Inspector or by calling `UpdatePath`.
3. When the game runs, the drone moves from its current position to the first waypoint, then sequentially to the others.
4. The movement stops when the drone reaches the last waypoint.