Spike script optional :

using UnityEngine;

public class EnemyMovement : MonoBehaviour

{

    // Waypoints for path movement

    public Transform[] waypoints;

    public float speed = 2f;

    private int waypointIndex = 0;

    // Variables for oscillating movement in different directions

    public float verticalAmplitude = 1f;    // Up and down amplitude

    public float verticalSpeed = 2f;        // Up and down speed

    public float horizontalAmplitude = 1f;  // Forward and backward amplitude (Z-axis)

    public float horizontalSpeed = 2f;      // Forward and backward speed

    public float lateralAmplitude = 1f;     // Side to side amplitude (X-axis)

    public float lateralSpeed = 2f;         // Side to side speed

    private Vector3 startPosition;

    void Start()

    {

        // Store the initial position

        startPosition = transform.position;

    }

    void Update()

    {

        // Call path-following logic

        MoveAlongPath();

        // Call movement oscillation logic

        MoveInAllDirections();

    }

    void MoveAlongPath()

    {

        if (waypoints.Length == 0) return;

        // Move towards the next waypoint

        transform.position = Vector3.MoveTowards(transform.position, waypoints[waypointIndex].position, speed \* Time.deltaTime);

        // If reached the waypoint, move to the next one

        if (Vector3.Distance(transform.position, waypoints[waypointIndex].position) < 0.1f)

        {

            waypointIndex = (waypointIndex + 1) % waypoints.Length;  // Loop through waypoints

        }

    }

    void MoveInAllDirections()

    {

        // Calculate oscillation on all axes

        float verticalOffset = Mathf.Sin(Time.time \* verticalSpeed) \* verticalAmplitude;

        float forwardOffset = Mathf.Sin(Time.time \* horizontalSpeed) \* horizontalAmplitude;

        float lateralOffset = Mathf.Sin(Time.time \* lateralSpeed) \* lateralAmplitude;

        // Apply movement in all directions (X, Y, Z)

        transform.position = new Vector3(

            startPosition.x + lateralOffset,

            startPosition.y + verticalOffset,

            startPosition.z + forwardOffset

        );

    }

}