```
using UnityEngine;
using System.Collections;

public class CatmullRomCurveInterpolation : MonoBehaviour {

    const int NumberOfPoints = 8;
    Vector3[] controlPoints;

    const int MinX = -5;
    const int MinY = -5;
    const int MinZ = 0;

    const int MaxX = 5;
    const int MaxY = 5;
    const int MaxZ = 5;

    float u = 0;
    int segNum = 0;

    int PositiveMod(int a, int b)
    {
        return (a % b + b) % b;
    }

    /* Returns a point on a cubic Catmull-Rom/Blended Parabolas curve
     * u is a scalar value from 0 to 1
     * segment_number indicates which 4 points to use for interpolation
     */
    Vector3 ComputePointOnCatmullRomCurve(double u, int segmentNumber)
    {
        int index0 = PositiveMod(segmentNumber - 2, NumberOfPoints);
        int index1 = PositiveMod(segmentNumber - 1, NumberOfPoints);
        int index2 = PositiveMod(segmentNumber, NumberOfPoints);
        int index3 = PositiveMod(segmentNumber + 1, NumberOfPoints);
        float tau = .5f;

        Vector3 c3 = -tau * controlPoints[index0] + (2 - tau) * controlPoints
          [index1] + (tau - 2) * controlPoints[index2] + tau * controlPoints
          [index3];
        Vector3 c2 = 2 * tau * controlPoints[index0] + (tau - 3) *
          controlPoints[index1] + (3 - 2 * tau) * controlPoints[index2] + -tau
          * controlPoints[index3];
        Vector3 c1 = -tau * controlPoints[index0] + tau * controlPoints
          [segmentNumber];
        Vector3 c0 = controlPoints[index1];

        return Mathf.Pow((float)u, 3) * c3 + Mathf.Pow((float)u, 2) * c2 +
          (float)u * c1 + c0; ;
    }
```

```
    void GenerateControlPointGeometry()
    {
        for(int i = 0; i < NumberOfPoints; i++)
        {
            GameObject tempcube = GameObject.CreatePrimitive
              (PrimitiveType.Cube);
            tempcube.transform.localScale -= new Vector3(0.8f,0.8f,0.8f);
            tempcube.transform.position = controlPoints[i];
        }
    }

    // Use this for initialization
    void Start () {

        controlPoints = new Vector3[NumberOfPoints];

        //Set points randomly
        controlPoints[0] = new Vector3(0,0,0);
        for(int i = 1; i < NumberOfPoints; i++)
        {
            controlPoints[i] = new Vector3(Random.Range(MinX,MaxX),Random.Range
              (MinY,MaxY),Random.Range(MinZ,MaxZ));
        }

        GenerateControlPointGeometry();
    }

    // Update is called once per frame
    void Update () {
        u += (float)Time.deltaTime;

        // Check if u reaches the end of current segment
        if (u >= 1.0f)
        {
            u -= 1.0f; // Reset u
            segNum++; // Move to the next segment
            if (segNum >= NumberOfPoints) // Ensure segNum stays within bounds
                segNum = 0;
        }

        Vector3 temp = ComputePointOnCatmullRomCurve(u,segNum);
        transform.position = temp;
    }
}
```