

```
FlatDonut.cs | 0 references | nonpareil, Less than 5 minutes ago | 1 author, 1 change
Assembly-CSharp | 0 references | nonpareil, Less than 5 minutes ago | 1 author, 1 change
FlatDonut
[Range(3, 255)]
public int N = 8;
[Range(1, 5)]
public float Radius = 1.0f;
[Range(2, 10)]
public float OuterRadius = 2.0f;
private float TAU = 2 * Mathf.PI;
private void GenerateMesh()
{
    Mesh mesh = new Mesh();
    List<Vector3> verts = new List<Vector3>();
    for (int i = 0; i < N; i++)
    {
        float theta = TAU * i / N; // angle of current iteration
        Debug.Log("Angle: " + theta + ", which in deg is: " + 360f * theta / TAU);
        Vector3 v = new Vector3(Mathf.Cos(theta), Mathf.Sin(theta), 0);
        verts.Add(v * Radius);
        verts.Add(v * OuterRadius);
    }
    mesh.SetVertices(verts);

    List<int> tri_indices = new List<int>();
    for (int i = 0; i < N - 1; i++)
    {
        int InnerFirst = 2 * i;
        int OuterFirst = InnerFirst + 1;
        int InnerSecond = OuterFirst + 1;
        int OuterSecond = InnerSecond + 1;
        //first tri
        tri_indices.Add(InnerFirst);
        tri_indices.Add(OuterFirst);
        tri_indices.Add(OuterSecond);
        // second tri
        tri_indices.Add(InnerFirst);
        tri_indices.Add(OuterSecond);
        tri_indices.Add(InnerSecond);
    }
    int InFirst = 2 * (N - 1);
    int OutFirst = InFirst + 1;
    int InSecond = 0;
    int OutSecond = 1;

    tri_indices.Add(InFirst);
    tri_indices.Add(OutFirst);
    tri_indices.Add(OutSecond);
    tri_indices.Add(InFirst);
    tri_indices.Add(OutSecond);
    tri_indices.Add(InSecond);

    mesh.SetTriangles(tri_indices, 0);
    mesh.RecalculateNormals();

    GetComponent<MeshFilter>().sharedMesh = mesh;
}

// Unity Message | 0 references | nonpareil, Less than 5 minutes ago | 1 author, 1 change
void Start()
```

