Procedural Generation of Flowers in Unity

Zoe Rowbotham
18014180

University of the West of England

May 15, 2022

The procedural generation of flowers in Unity for exporting and use in games.

1 Introduction

Flowers are algorithmic in nature (Prusinkiewicz and Lindenmayer, 1990), they can be described with angles and formulas, meaning they can be procedurally generated with mathematics and algorithms. Unity allows for the creation of a mesh in code; this project attempts to create a tool for the procedural generation of realistic looking flowers.

2 Related Work

2.1 Unity Mesh Generation

Generating meshes in Unity requires creating a Mesh object and adding vertices and triangles Unity, 2021. The Mesh class also has options to add UV coordinates to vertices for texturing.

In order to show the mesh in a scene a MeshFilter and MeshRenderer are needed. A MeshFilter holds a mesh reference; the referenced mesh can be changed through code. A MeshRenderer uses the Mesh referenced by the MeshFilter to render the mesh in the scene.

2.2 Cubic Bezier Curves

A bezier curve is a parametric curve. A cubic bezier curve has two anchor points and two control points. The curve begins and ends at the anchor points, while the control points provide directional information.

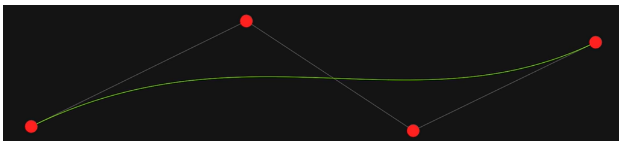

Figure 1: *A Cubic Bezier curve showing all four points (Lague, 2018)*

Lague, 2018 provides a visual explanation of how bezier curves are formed.

The equation for a cubic bezier curve can be found in Equation 1. Where t is a parameter between 0 and 1. The points of the curve are calculated using the equation at different t values.

$$P = P_1(1t)^3 + 3P_2(1t)^2t + 3P_3(1t)t^2 + P_4t^3$$

Where:
$P_1 = AnchorPointOne$
$P_2 = ControlPointOne$
$P_3 = AnchorPointTwo$
$P_4 = ControlPointTwo$
$t = TimeValue$
(1)

2.3 Vogel's Formula

Chapter 4 of The Algorithmic Beauty of Plants (Prusinkiewicz and Lindenmayer, 1990) references Vogel's formula describing the arrangement of seeds in a flower, specifically a sunflower head.

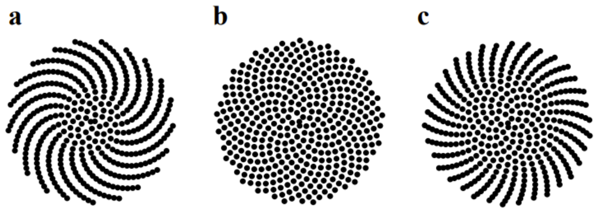

Figure 2: *Vogels Formula with the angles 137.3 (a), 137.5 (b) and 137.6 (c) (Prusinkiewicz and Lindenmayer, 1990)*

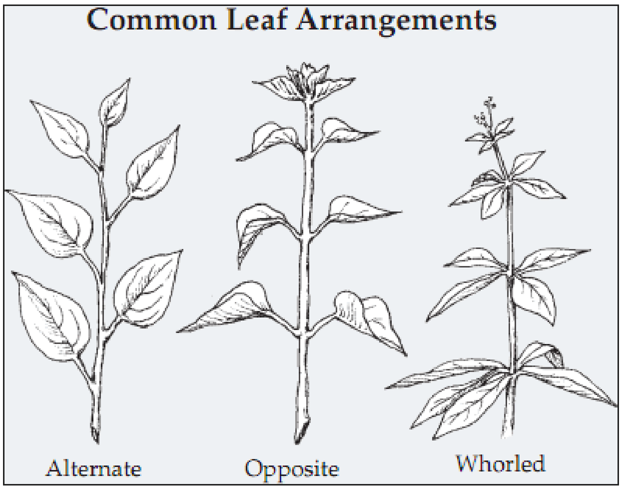

Figure 3: *Leaf arrangement diagram (Plant, 2011)*

2.4 Phyllotaxis

Phyllotaxis is the arrangement of leaves on a plant stem (Plant, 2011). The way leaves are arranged are designed to maximise the amount of sunlight for photosynthesis.

Leaves on a stem can be positioned in three ways: alternate, whorled and opposite as shown in Figure 3.

The golden angle is the smaller angle when a circumfrence is sectioned into the golden ratio and is equal to 137.508 degrees (Wikipedia, 2021). The golden angle is important in phyllotaxis as it is the angle between petals in some flowers.

2.5 Delaunay Triangulation

Delaunay Triangulation (Delaunay, 1934) refers to triangulation of vertices where no circumcircle of any triangle contains any vertex as shown in Figure 8. The Bowyer-Watson algorithm is a useful method for computing the Delaunay Trianglulation of a set of vertices. Bowyer and Watson devised the algorithm independently in two different papers at same time: Bowyer, 1981 and Watson, 1981.

The Bowyer-Watson algorithm uses an initial triangle that encompasses all points, then loops over all points and finds the circumcircles that contain that point. The edges of these triangles are then used to form new triangles with the current point, except for edges that

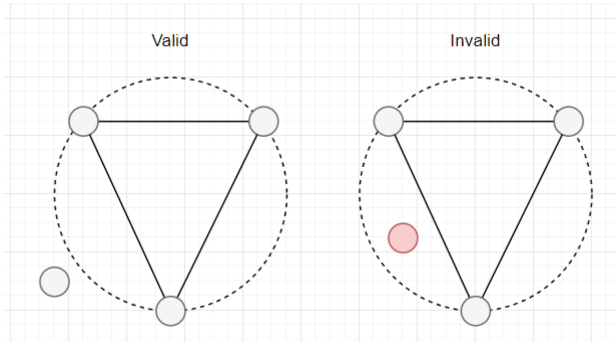

Figure 4: *A valid and invalid Delaunay Triangle.*

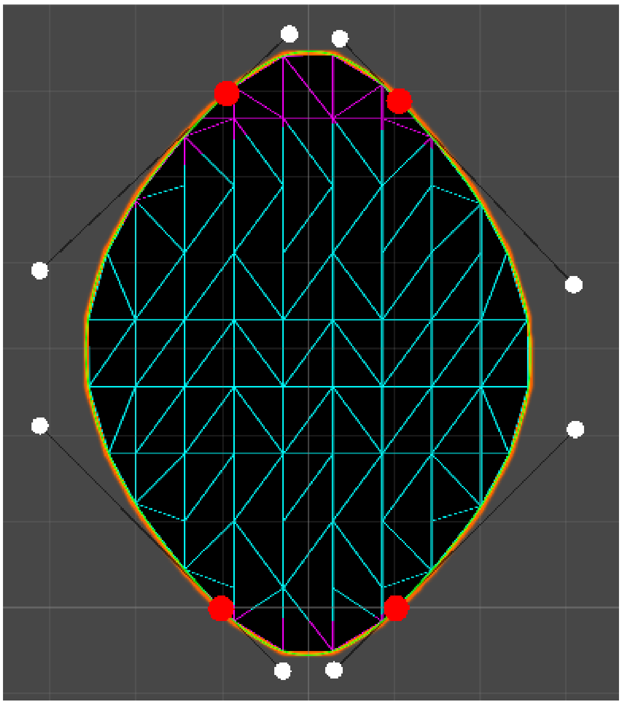

Figure 5: *Shape tool working to create a leaf shape.*

are not unique to a single triangle.

3 Method

3.1 Shape Tool

Generating leaf, petal and seed shapes could potentially be extremely difficult and therefore, a shape tool has been created where the user can define a shape and export the mesh for use in the flower generation tool.

3.1.1 Bezier Path

The shape tool consists of a number of cubic Bezier paths connected together to form a shape, as shown in Figure 5.

There is an option in the the shape tool to automatically set the control points to form smooth connections between the Bezier paths. The direction of an automatic control point is calculated as the

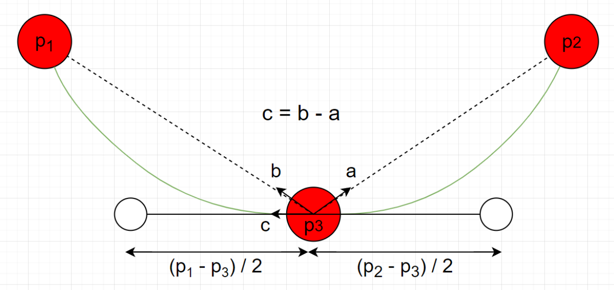

Figure 6: *Automatic control points calculation diagram.*

perpendicular vector to the vector that bisects the anchor point before and after the current anchor point. This is calculated by subtracting the direction vector of the current anchor point to the next anchor point from the direction vector of the current anchor point to the previous anchor point. The distance of an automatic control point from the anchor point is half the distance to the corresponding previous or next anchor point. These calculations are described in Figure 6.

3.1.2 Mesh Generation

The first step of generating the mesh for the shape is generating the vertices, this process starts by calculating evenly spaced points around the edge of the shape. Evenly spaced points are calculated by looping over each segment of the shape and stepping along the segment by a resolution value and marking the points at a certain distance.

From the evenly spaced points, the shape is divided both horizontally and vertically by a specified number of divisions. When dividing horizontally, the evenly spaced points are used to find an intersection between the shapes edge and the X value of the division; when dividing vertically the Y value is used. These points make the edge of the shape, and a grid is constructed inside the shape to generate all the vertices required, as seen in Figure 7. The reason for diving the shape is to be able to manipulate it based on curves in the flower generation tool.

Using Delaunay Triangulation and the Bowyer-Watson algorithm on the set of vertices generates the triangles needed for the shape as shown in Figure 5.

3.1.3 3D Shape

To generate the shape as 3D the grid vertices are duplicated and moved along the z-axis by a depth value. These duplicate grid vertices are connected to the vertices on the edge of the shape and the triangles are duplicated and reversed; this creates the same shape as the 2D Bezier curves but with some depth.

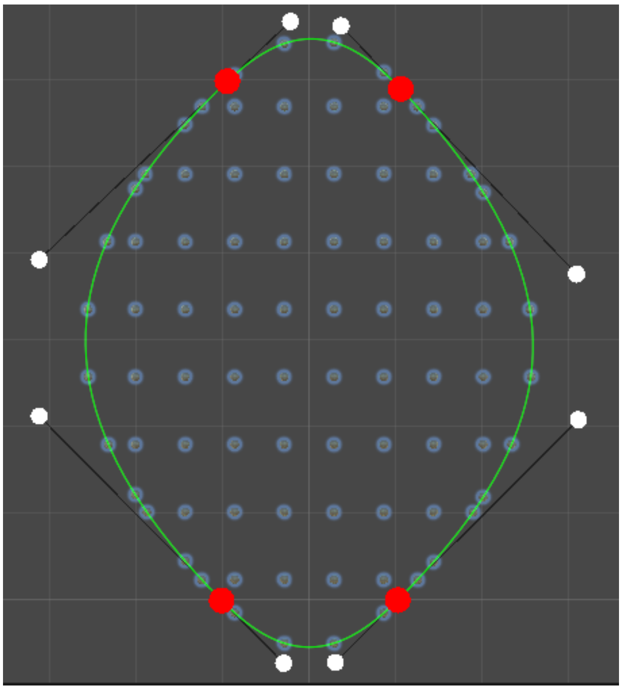

Figure 7: *Vertices (as blue circles) calculate to generate the shape mesh.*

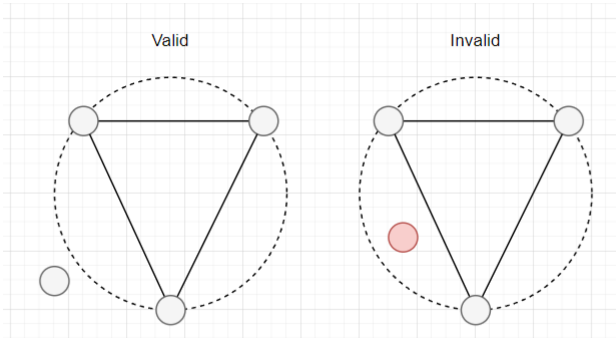

Figure 8: *A valid and invalid Delaunay Triangle.*

3.1.4 UVs

To generate the UVs for the vertices the position of the vertex is scaled by the bounds of the shape to be with 0 and 1. This is calculated by Equation 2 and Equation 3, where x and y refer to the the position of the vertex and the min and max values refer to the bounds of the shape.

$$U = (x - minX)/(maxX - minX) \tag{2}$$

$$V = (y - minY)/(maxY - minY) \tag{3}$$

3.2 Stem

Appendix B describes the parameters that can be manipulated for the stem.

The stem is split into sections as determined by the parameters, each section of the stem is a cylinder.

The resolution of the cylinder (i.e. the number of vertices creating the circle face) is determined by the parameters. A random angle is generated for each section so the stem is not straight. Circles of vertices are generated at the end of each section, beginning the next section.

Triangles are generated as the vertices are generated, using the previous circle of vertices to generate triangles around the circumference of the circle.

3.3 Leaves

Appendix C describes the parameters that can be manipulated for the leaves.

3.3.1 Positioning

The leaves are positioned vertically up the stem dependant on the Number of Leaves parameter, they are positioned with equal spacing along the stem. To get an alternate pattern around the stem the Spiral Angle parameter is used which determines the angle between the previous leaf and the current leaf. The direction to rotate the leaf to look towards is calculated using Quaternion.AngleAxis to manipulate the forward vector by the angle. The vertical angle of the leaf is determined by the Angle parameter, the smaller the angle the closer to the stem the leaf is.

3.3.2 Bending

Due to the divisions created by the shape tool, the leaves and petals can be deformed by a curve, horizontally and vertically. The curve is evaluated from 0 to 1 and each vertex along the axis manipulated based on the curves value at that point.

3.4 Flower Bud

Appendix D describes the parameters that can be manipulated for the flower bud and Appendix E describes the parameters for the seeds of the bud.

3.4.1 Dome Calculation

The flower bud at the top of the stem is a dome shape with a number of deviations specified by a parameter. The position of the vertices in the dome is calculated by the code below.

```
for (int i = Divisions; i >= 1; i--)
{
  x = Radius * (i / Divisions);
  y = -(x * x * (1 / Radius));
  y += StemTopHeight;

  for (int j = 0; j < Deviations; j++)
  {
    Vector3 dir = transform.forward;
    dir *= i * widthPerSection;
    dir *= Quaternion.AngleAxis(
      currentAngle,
      transform.up
    );

    Vector3 position = stemTopCenter + dir;

    float height = transform.up * y;
    height *= heightScaleFactor;
    position += height;

    vertices.Add(position);

    uvs.Add(new Vector2(j / Deviations, 1));
  }
}
```

3.4.2 Seed Generation

Seeds are generated using Fermat's spiral for the X and Z position of the seed, the Y position is calculated using the height of the dome at the X,Z position of the seed.

3.5 Petals

Appendix F describes the parameters that can be manipulated for the petals of the flower.

3.5.1 Positioning

The petals are positioned using an angle parameter which determines the angle between the current petal and the previous petal. The vertical angle between the stem and the petal is determined by inner and outer angle parameter; the first layer on the outside of the flower head uses the outer angle and the last layer on the inside of the flower head uses the inner angle.

3.5.2 Curling

To achieve a bud like effect, the petals at the centre of the flower curl more than the petals at the outside of the flower. The curve parameter for the vertical curve of the petals is scaled depending on the index number of the flower, where 0 is the starting petal on the outside and the last index is the petal closest to the center.

4 Evaluation

The flowers in Figure 9 show the tool in use to create three different types of flowers: sunflower, daisy and tulip. The shape tool is effective in creating the shapes of the different petals, leaves and seeds, giving a unique look to the different flowers. There are some clipping issues with the petals and leaves that can be solved with parameter tweaks but this limits the tool when

Figure 9: *Different types of flowers generated by using the generation tool.*

Figure 10: *Generated flowers in a pre-made field of grass.*

attempting to create certain flowers, like the tulip, where the petals should be very close together.

5 Conclusion

This project aimed to create a tool to generate procedural flowers. The resulting tool allows the user to change parameters to generate different types of flowers as seen in Figure 9. The shape tool also allows for the creation of many different shapes of leaves, petals and seeds. The tool also allows the user to export the flowers as FBX files for use in other software. Figure 10 shows the flowers made by the tool being used in a field.

Bibliography

Bowyer, A. (Jan. 1981). “Computing Dirichlet tessellations*”. In: *The Computer Journal* 24.2, pp. 162–166. ISSN: 0010-4620. DOI: 10.1093/comjnl/24.2.162. eprint: https://academic.oup.com/comjnl/article-pdf/24/2/162/967239/240162.pdf. URL: https://doi.org/10.1093/comjnl/24.2.162.

Delaunay, B (1934). “Sur la sphere vide. A la memoire de Georges Voronoi”. In: *Bulletin de l’Académie des Sciences de l’URSS. Classe des sciences mathématiques et na* 6, pp. 793–800.

Lague, Sebastian (2018). *[Unity] 2D Curve Editor (E01: introduction and concepts)*. URL: https://youtu.be/RF04Fi9OCPc.

Plant, Life of (2011). *Leaf Arrangements*. Last accessed 24 March 2022. URL: http://lifeofplant.blogspot.com/2011/03/leaf-arrangements.html.

Prusinkiewicz, Przemyslaw and Aristid Lindenmayer (1990). *The Algorithmic Beauty of Plants*. Springer-Verlag.

Unity (2021). *Unity: Using the Mesh Class*. URL: https://docs.unity3d.com/Manual/UsingtheMeshClass.html (visited on 12/03/2021).

Watson, D. F. (Jan. 1981). “Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes*”. In: *The Computer Journal* 24.2, pp. 167–172. ISSN: 0010-4620. DOI: 10.1093/comjnl/24.2.167. eprint: https://academic.oup.com/comjnl/article-pdf/24/2/167/967258/240167.pdf. URL: https://doi.org/10.1093/comjnl/24.2.167.

Wikipedia (2021). *Golden Angle*. Last accessed 27 March 2022. URL: https://en.wikipedia.org/wiki/Golden_angle.

Appendix A: Development Video Diaries

Week One: https://youtu.be/5TMKmmlyHfw
Week Two: https://youtu.be/V9gyhJerVB8
Week Three: https://youtu.be/PkWGLhRIYpE
Week Four: https://youtu.be/7qG8d-WnQYg
Week Five: https://youtu.be/UliPYAXiK4E

Appendix B: Stem Parameters

- Height: Total target height in Unity units.
- Stem Top Height: Height of the top of the stem to the bud.
- Number of Stems: Total number of flowers to generate.
- Generation Radius: Radius to generate flowers in.
- Number of Sections: Vertical divisions in the stem.
- Deviations: Number of vertices in the circle that makes the stem.
- Radius: Stem radius.
- Max Angle: The maximum angle a section can go where 0 is directly up.
- Material: Material for the stem mesh.

Appendix C: Leaf Parameters

- Number of Leaves: Number of leaves to generate up the stem.
- Shape: Mesh generated from the shape tool.
- Material: Material to apply to the leaf mesh.
- Size Scalar: Multiplier to scale the shape by.
- Angle: The angle the leaf comes off the stem where 0 is directly up.
- Horizontal Leaf Curve: Curve manipulation of the leaf horizontally.
- Vertical Leaf Curve: Curve manipulation of the leaf vertically.
- Start At Base: Whether the leaves start at the base of the stem or not.
- Spiral Angle: For Alternate leaf arrangement, the angle around the stem horizontally between leaves.
- Num Leaves In Whorl: The number of leaves in a whorl on the stem.

Appendix D: Flower Bud Parameters

- Radius: The radius of the bud.
- Height: The height of the bud.
- Divisions: The divisions across the dome of the bud.
- Material: The material of the bud.

Appendix E: Flower Seed Parameters

- Golden Angle: The angle used in Fermat's spiral.
- Spiral Multiplier: Constant multiplier value for Fermat's spiral.
- Shape: Mesh generated by the shape tool.
- Material: Material to assign to the mesh.
- Size Scalar: Multiplier to scale the shape by.

Appendix F: Petal Parameters

- Layers: The number of layers of petals to generate.
- Angle: The angle between one petal and the next.
- Shape: Mesh generated by the shape tool.
- Size Scalar: Multiplier to scale the shape by.
- Outer Angle: The angle of the petals on the outside of the flower head, where 0 is vertical.
- Inner Angle: The angle of the petals on the inside of the flower head, where 0 is vertical.
- Horizontal Curve: Curve manipulation of the petal horizontally.
- Vertical Curve: Curve manipulation of the petal vertically.
- Min Curling Multiplier: Minimum value for curve scalar.
- Max Curling Multiplier: Maximum value for curve scalar.
- Material: Material to assign to the mesh.