



Generating Plant Models Using Fractals

Makayla Moster

Advisor: Dr. Brittany Morago

Department of Computer Science

University of North Carolina Wilmington



Introduction

The purpose of this directed independent study was to expand on topics covered in CSC 360 and CSC 370. The objectives for this course were to use fractals to generate a plant model, apply real time lighting, and to visualize basic leaf venation.

Methods

STEP ONE: Generate a pattern and interpret pattern characters.

The pattern used for this project was $F[F][-F][+F]$.

- F: Forward
- [: Save position
-]: Return to last saved position
- -: Rotate the branch right
- +: Rotate the branch left

During the interpretation, the position of the points (X, Y, and Z coordinates) are stored in an array.

STEP TWO: Draw tree branches

- Iterate through the array of points
- Draw a line between each consecutive set of points

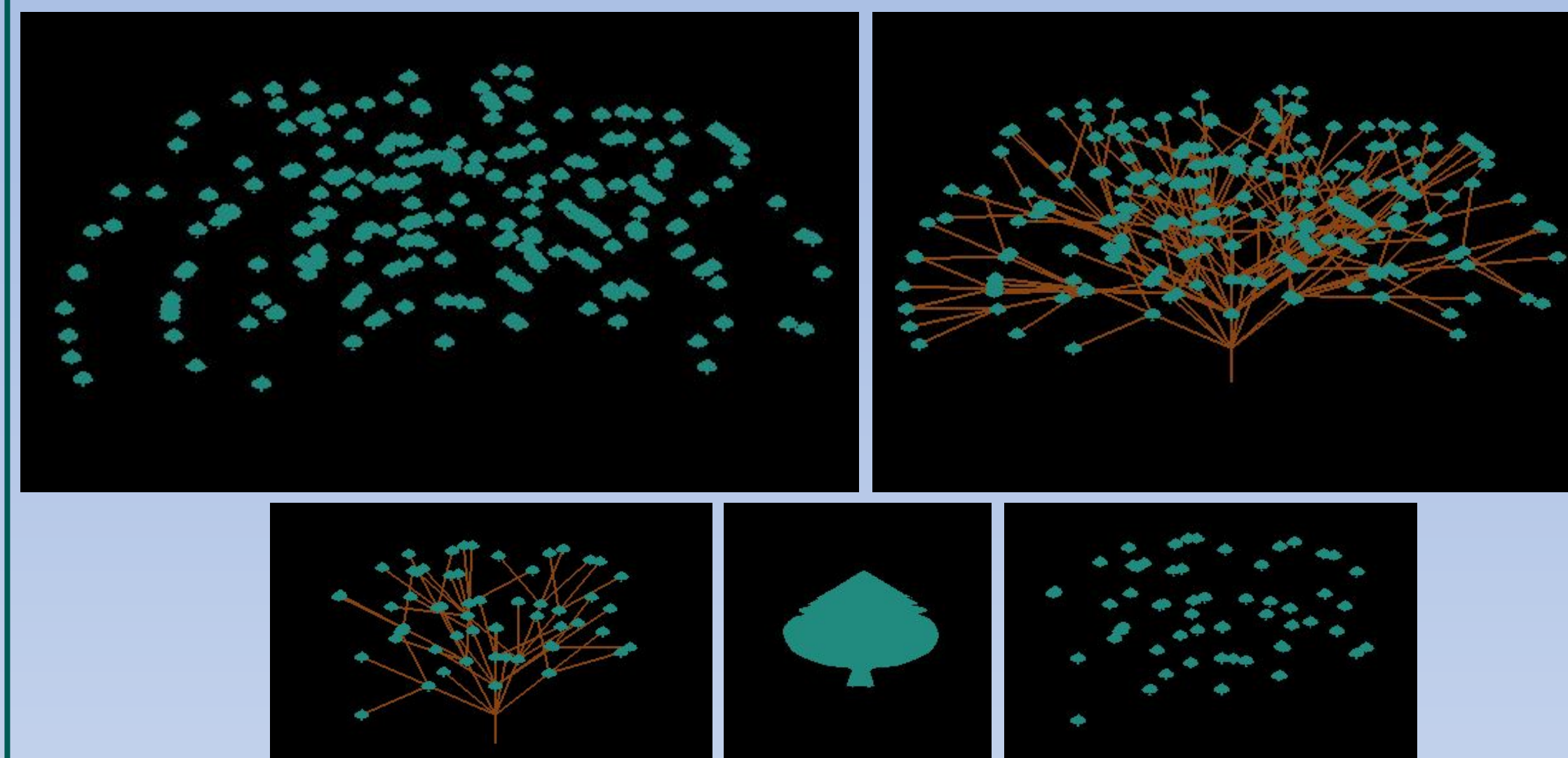


STEP THREE: Create and load a leaf into the scene.

- Create a leaf using the tools in Maya
- To load the leaf, vertices have to be stored in an array
- Normals have to be computed for the faces and vertices
- Store vertices and normals in separate arrays
- For multiple leaves, the vertex coordinates are stored in the same array multiple times.

STEP FOUR: Translate leaf in scene.

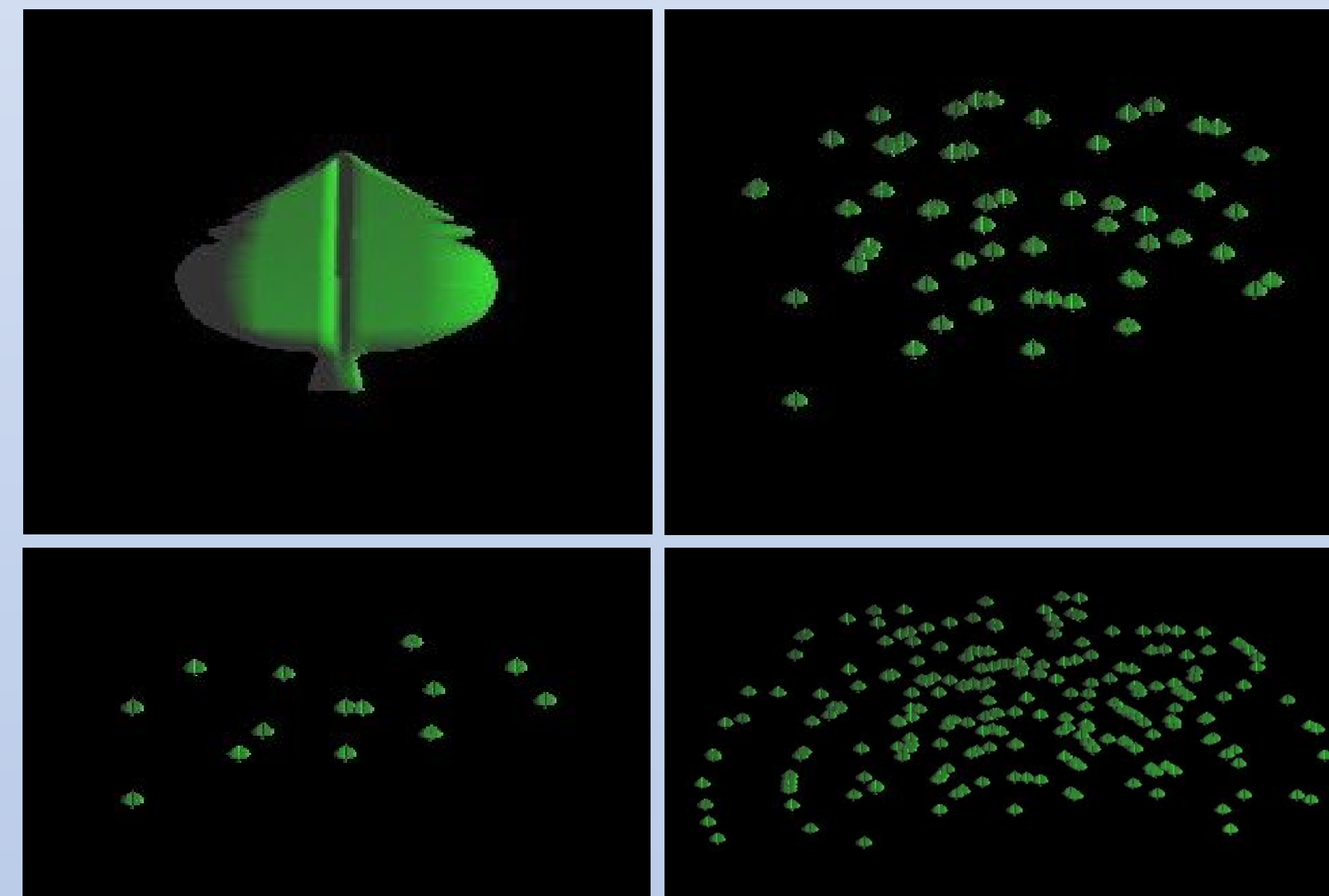
- Create rotation, scale, and translation matrices
- Choose rotation, scaling, and translation factors, and update matrices
- To translate the leaf, multiply each vertex in the array by the previously created matrices



Methods Continued

STEP FIVE: Apply realistic lighting to the leaves

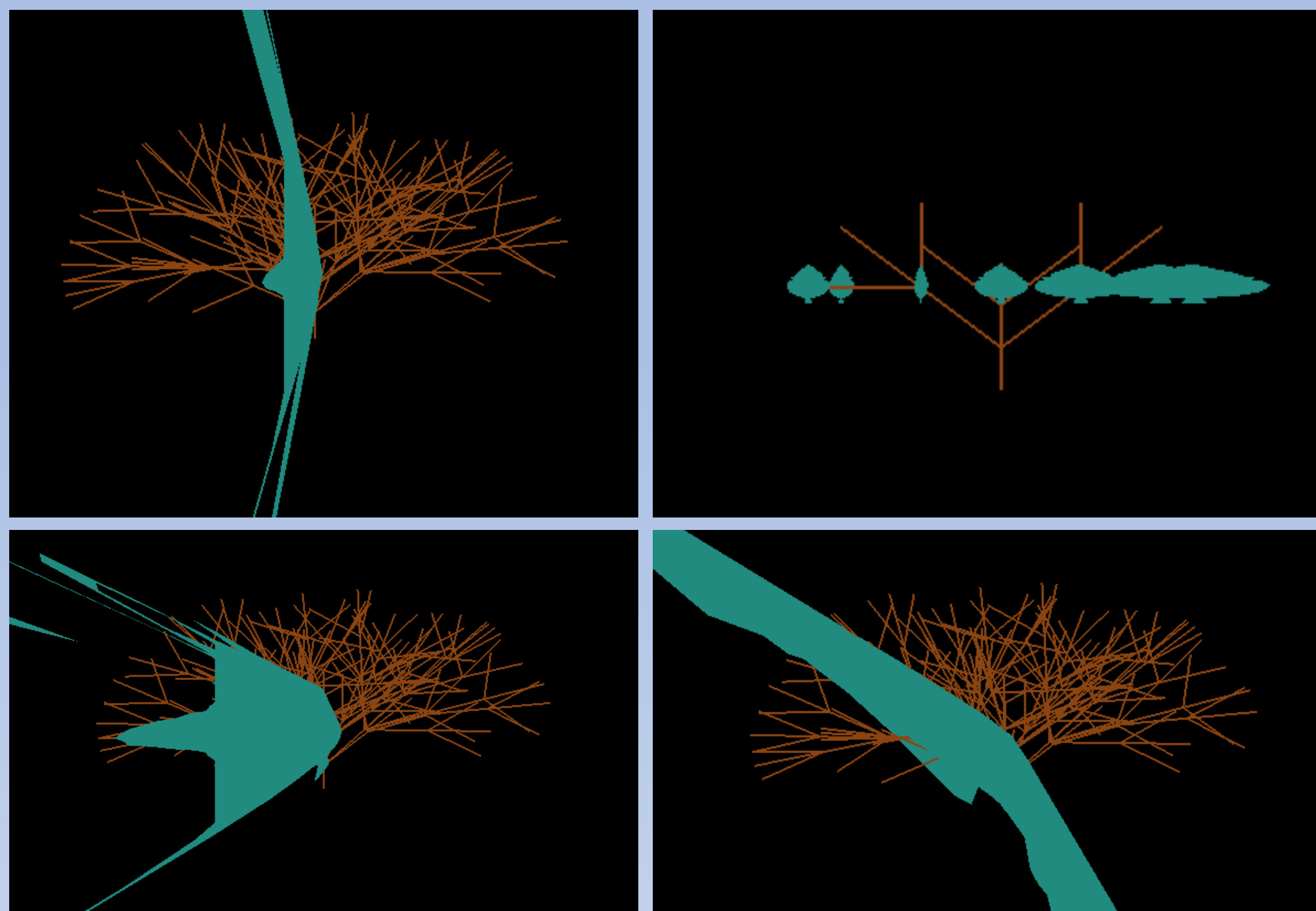
- Update current vertex and fragment shader to use the Phong model
- Calculate the position of the normals and the position of the viewer's eye within the vertex shader to pass into the fragment shader
- Within the fragment shader, the elements for the Phong model are computed which include:
 - Diffuse lighting
 - Specular lighting
 - Ambient lighting
- The lighting components are then added together to create the lighting effect upon the leaves



Difficulties

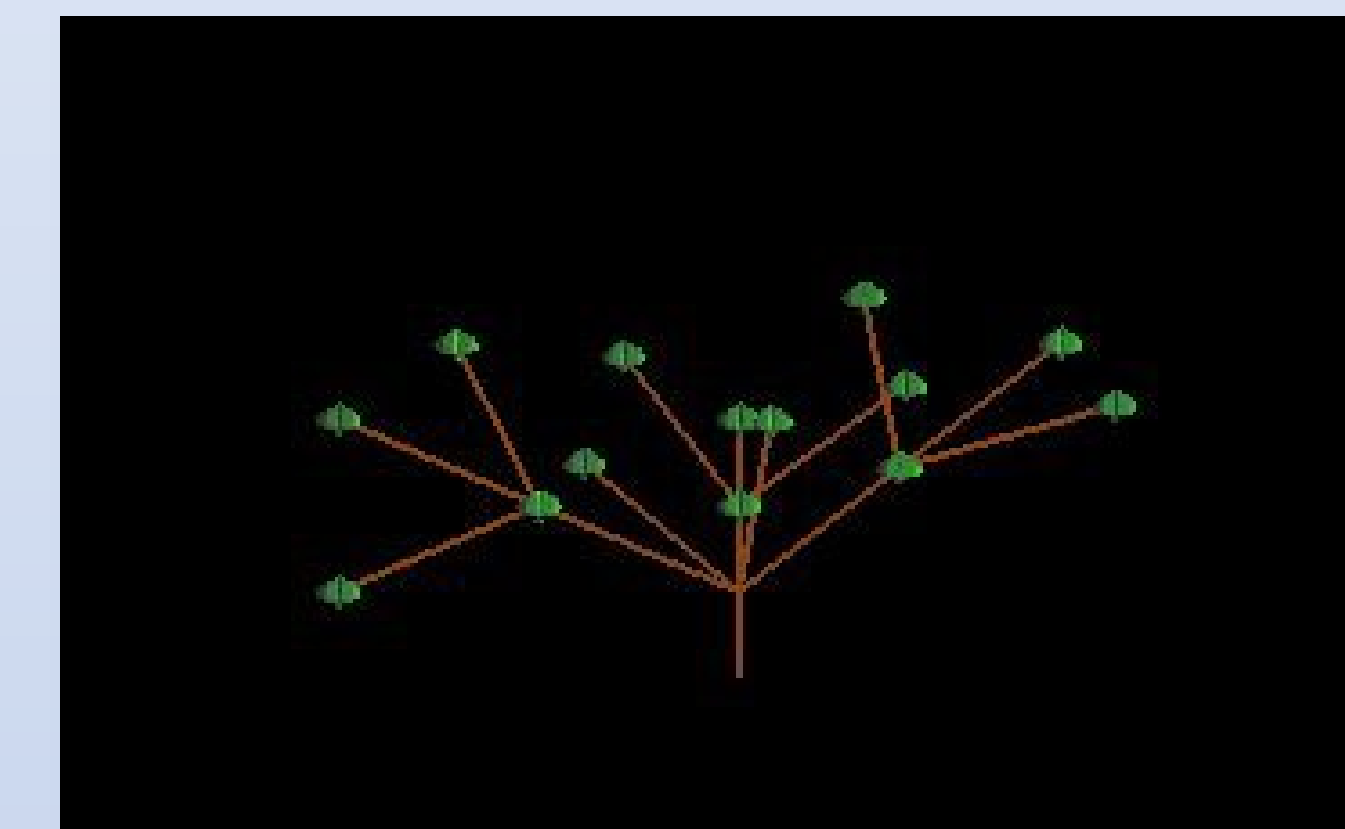
Some difficulties that were encountered are:

- Multiplying the leaf coordinates by translation, rotation, and scaling matrices caused some abstract art to be created.
 - OpenGL uses transposed matrices, the matrices used in the beginning of this project were not transposed
- Adding the correct vertex and fragment shaders to light the scene accurately
- Ensuring that array size was correct to hold coordinates for each leaf

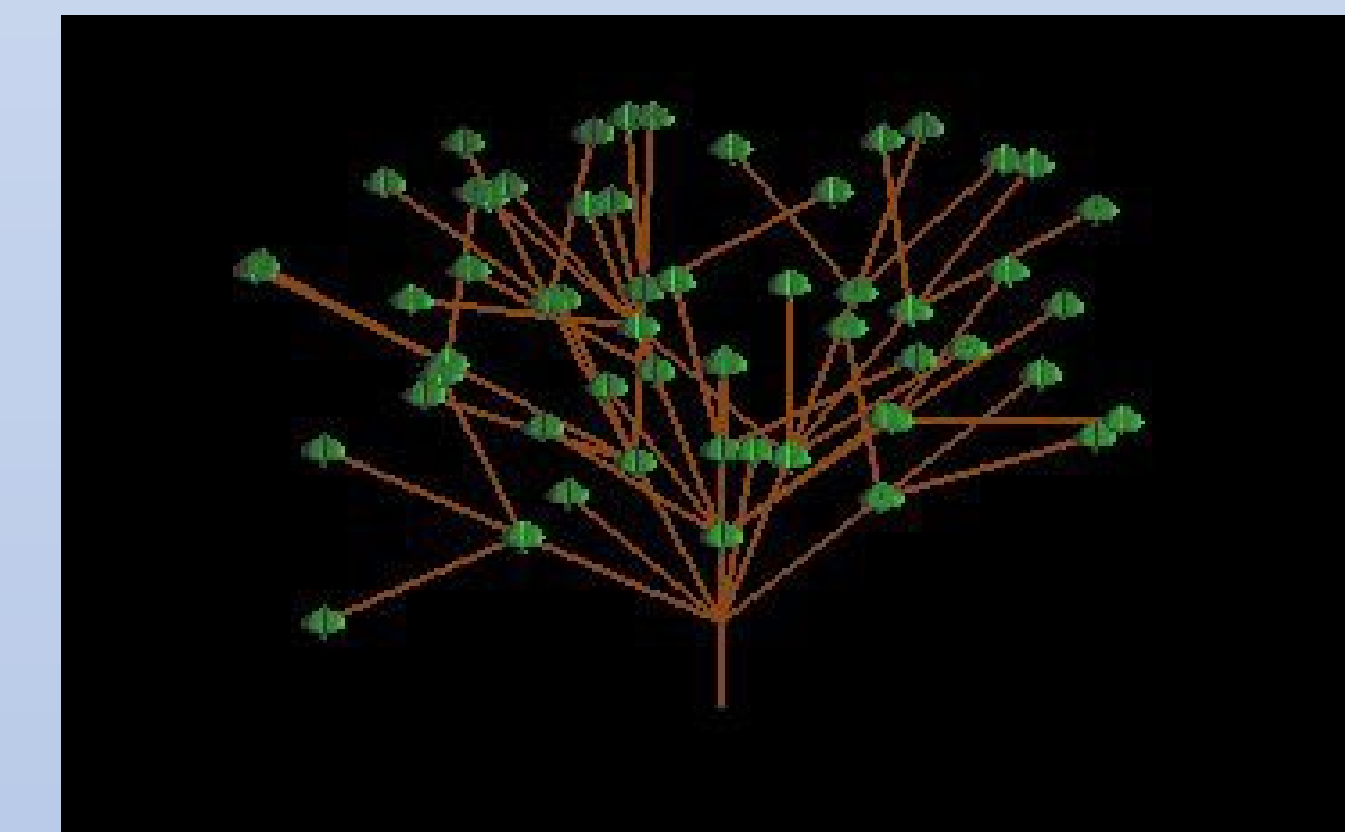


Results

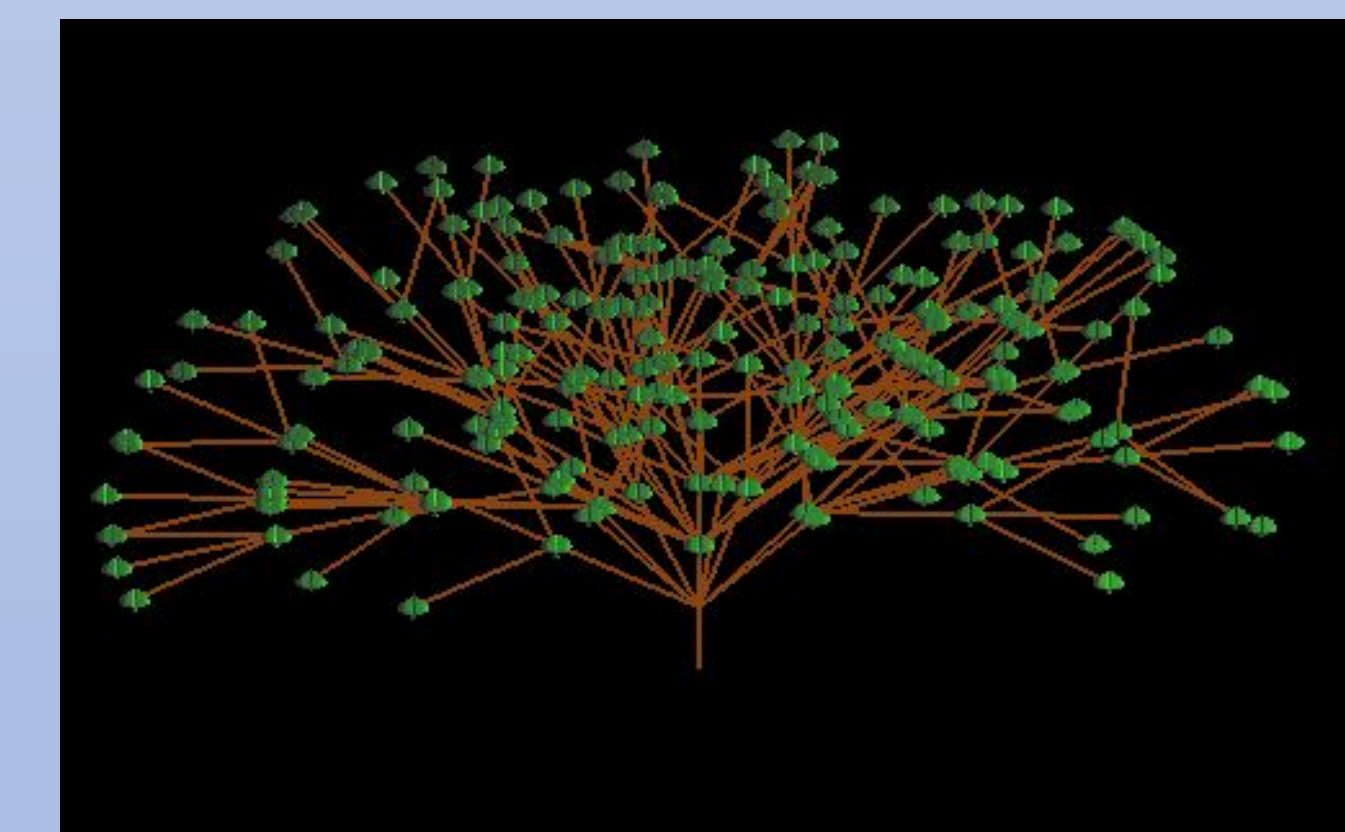
The result of this project is an automatically generated, two dimensional tree that uses a Lindenmayer system to generate a pattern for the tree, based on the number of iterations given. The tree also generates enough leaves for the total number of branches based on the number of iterations. The leaves are then realistically lit using the Phong lighting model.



2nd Iteration tree with lit leaves



3rd Iteration tree with lit leaves



4th Iteration tree with lit leaves

Future Work

In the future, I will continue to improve this project by:

- Making the tree three dimensional
- Researching a geometry shader to make the tree limbs thick or thin depending on their location within the tree
- Adding more leaves to the branches to make the tree look more full
- Making the scene interactive with keys to change the viewing window and to navigate around the scene