//Procedural Modeling

//by Cameron McClure (chmcclur@ucsc.edu)

//Description

The general idea of this project is to procedurally generate several different types of plants on a randomly generated island. Plants will vary in appearance and shape based on biome, water sources and location. If a plant is near a large source of water, and/or on flat ground it will grow larger, have more varied color, and be more complex (more branches, leaves, roots).

//Status

A little behind schedule due to having to study for a make-up midterm we were only told about last week in one of my classes, at least the basic modeling has been set up.

Terrain is stored in memory as a grid, so that when we need to find the properties of a given "square" we can just call up that internal visualization. This has some advanteges and disadvanteges as you'll see below.

We use L-system algorithms to model plants. For example, the basic trees use the following algorithm to generate branches:

variables : 0, 1

constants: [,]

axiom : 0

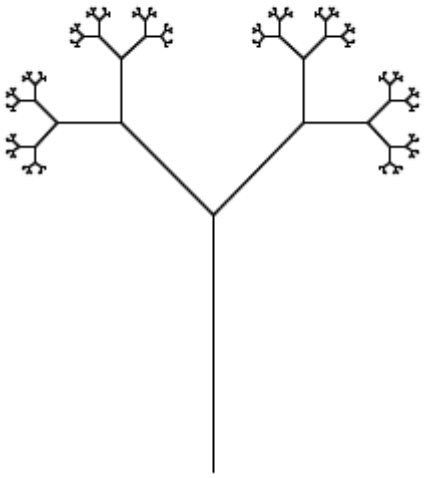
rules : (1 → 11), (0 → 1[0]0)

0: draw a line segment ending in a leaf

1: draw a line segment

[: push position and angle, turn left 45 degrees

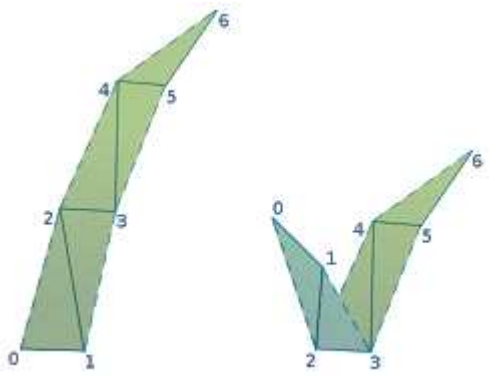
]: pop position and angle, turn right 45 degrees



Basically, every time we read a "[" we save our last angle and position into a stack, and then the next time we encounter a "]" we pop off the stack and begin from there.

The actual rules are a little more complex, but that's the gist of it.

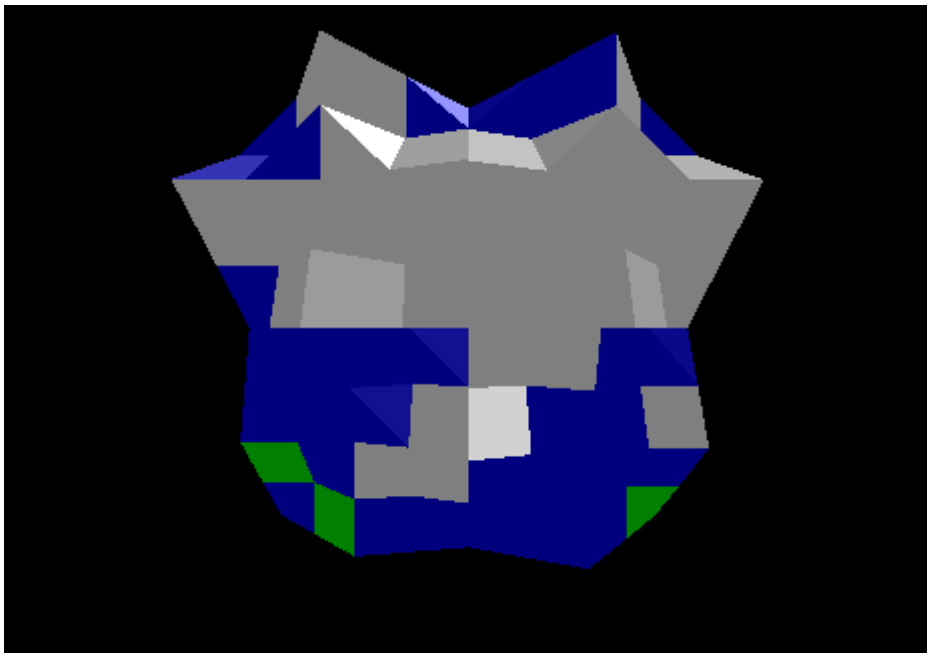
L-system algorithms aren't used for grass though. Instead, all pieces of grass are a simple 2D strip of triangles made up of seven vertices.



Then each grid of the terrain has an "irrigation" value which determines the density of grass.

And when we place down water, the terrain changes accordingly. Water works on a "steepest first" search pattern. If it can not find a lower point, the water spreads out along all the points at the same level, until it can spread no more.

Unfortunately the process does not look very natural at the moment, since I'm utilizing a grid-visualization of the terrain, causing things to look very obviously artificial.



We have some basic camera controls: you can move the camera around and zoom in and out with the mousewheel, we have some simple rendering options (filled versus wireframe), and some shading options (flat versus gouraud). Basically, all the old user-interaction implmented in the previous lab/programs.

//To-Do

Due to being behind, I don't quite have plant modelling working out on non-flat surfaces, and water/biome stuff has to be set up at generation. I also need to work on adding more interaction (allowing the user to place plants, rivers, and biomes) and make the ugly square-ish rendering more natural looking.