Chris Littlefield Aesthetics and Computation

Project 2: L-System Visualizer in THREE.js

This project was an attempt to port my L-System Visualizer from Project 1 into THREE.js, and with that, I managed to improve upon it significantly. I already did an assignment in THREE.js by making a Lorenz Attractor, so I copied that over to my project for the scene code to same me some time. After that, bringing in the old visualizer and modifying the code for THREE.js was a snap. My Lorenz Attractor already had a free-floating camera, so I left that in so people can zoom in and out on the L-systems they create. I also developed more control over the L-system, there is now an option to change the angle the turtle class rotates at. I also added more control over rotating the angle; “-” and “+” in the set of rules have been replaced with “z” and “Z” respectively. I also created similar rules for rotating on the X and Y axis. The presets from my old visualizer have been updated to reflect those changes, and I am happy they still work. However, as I gave the user more control over the L-System output, the presets that I saved from my old visualizer needed more information stored in them. To that end, I saved the preset in JSON strings. That way, when I need to load a preset, the method for loading first parses the string into a JSON object and assigns the appropriate values to the L-System.

Rules for L-System Visualizer:

F – Moves forward, draws a line.

f – Moves forward, does not draw a line.

[ - Push the current position and rotation to a stack.

] – Pops the current position and rotation from the stack.

x – Rotates the turtle class counter-clockwise in theta on the x-axis.

X – Rotates the turtle class clockwise in theta on the x-axis.

y – Rotates the turtle class counter-clockwise in theta on the y-axis.

Y – Rotates the turtle class clockwise in theta on the y-axis.

z – Rotates the turtle class counter-clockwise in theta on the z-axis.

Z – Rotates the turtle class clockwise in theta on the z-axis.