Final Project: CS 2410

The LORENZ ATTRACTOR is part of a dynamical system and Chaos theory. A dynamical system moves to and stays near a certain point, called an attractor. An attractor could also be a set of points.

The Lorenz attractor is an attractor that arises in a simplified system of equations. These set of equations were developed by Edward Lorenz to model a simplified weather system. The differential equations are as follows:

1. Χ(ρ−Ζ)−Υ

The system of diff equations describe 2-D flow of fluid of uniform depth, in other words the equations model the flow of fluid(usually air) from a hot area to a cold area. Where is proportional to convective intensity, Υ is prop. to temp diff between descending and ascending currents and is prop to diff in vertical temp profile from linearity in the system of equations.

The solution to the differential equations is a dynamical system also known as the Lorenz system. The solution can be plotted with varying initial conditions. It is determined be highly deterministic, unpredictable and irregular. It is part of The Chaos Theory.

It is sensitive dependent on initial conditions.

Conceptual definitions:

-Chaos: aperiodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions.

Jargon words= aperiodic: doesn’t repeat and doesn’t settle down to equilibrium

-Attractor (A):

properties:

1. invariant set: starts in A, stays in A
2. Attracts an open set of initial set of conditions
3. No proper subset of A satisfies 1) and 2)

The solutions to the set of three differential equations are derived using Euler’s method. Euler’s method is a simple and basic solution to ordinary differential equations(ODE). The idea is that a differential curve is broken into smaller segments to approximate values for solutions of ODE.

The parameters to create the Butterfly effect are set integers derived from Lorenz, himself. I enjoyed this project because I was exposed to different methods. I’ve never used openGL but now I have a basic understanding, so now I can practice. Last semester I took Discrete Math and Calculus II and this project did help me remember certain definitions. The idea for this project was based on a Strange Attractors tutorial. I found many articles on the different methodologies from this set of system. Further, I would like to research communicative encryption using Chaos Theories. I would like to encrypt a phone call or images. I value privacy in this day of age, so this would definitely be an exciting prospect senior project. The algorithm for Euler’s method was based on the Youtube video listed below. Visualizing this system helped me grasp a better understanding of the Chaos Theory.

Functions:

Class function: My class defines the parameters for the Lorenz system. I chose the parameters to render the Butterfly effect. Different parameters can be inputted to render other types of attractors. Before fall semester starts, I will add a function to initiate an x-value according to userInput. The euler algorithm can find points approximate to the userInput.

Void render(); = The while loop within this function reads the estimated coordinate solutions from the Euler’s algorithm. The while loop has to between the [ glBegin-glEnd() ] block. Many of the functions within are openGL functions to set up the display.

glColor3fv(XY\_Points[count]);

glVertex3fv(XY\_Points[count++]);

-The functions above:

parameters : A pointer to an array of three elements. The elements are the x, y, and z coordinates of a vertex.

Void reshape(); = this function is necessary and many of the functions within are verbatim to the openGL guide. This function is vital to set up the camera and object(Lorenz Attractor). As the image renders, the glut window modifies itself, accordingly.

Void userKey(); = this is switch-case loop, accepting a char as an argument. This function allows for the attractor to be interactive with user.

Void idle(); = This is an openGL function. This function is also vital to displaying the attractor. Model image would seem stuck or frozen without this function. No actual movement would take place. As new points are plotted, this function draws out the next frame.

Void lorenzAttractor(); = this is where I set-up parameters(using a class) and figure out initial conditions using Euler’s algorithm. Dt acts as the “stepper” according to the algorithm. By changing this from .01 to .001, the segment is decreased, allowing it to be more accurate and smooth.

Int main(); = call functions here

int main parameters: how command line arguments are passed

-argc: means argument count, number of arguments pointed to by argv(# of arg. being pass into the program from command line)

-argv: means argument vector

Sources:

openGL

http://glprogramming.com/red/chapter02.html

<https://nathanselikoff.com/training/tutorial-strange-attractors-in-c-and-opengl>

<http://www2.kuas.edu.tw/prof/cfchang/pdf/3DCG/3_TRANSFORMATION.pdf>

http://glprogramming.com/red/chapter03.html

Inspiration:

<http://glprogramming.com/red/chapter03.html>

Euler’s method:

<https://www.youtube.com/watch?v=RGtCw5E7gBc>