Noel Ball
CSE 13S

plot.sh Design Document

Description:

An automated bash script will run monte_carlo, testing random points to estimate for pi. Those data points are then plotted as graphs through the use of gnuplot. For the first plot it's important that monte_carlo calculates if the random point is within a distance of 1 from the origin. An arcing curve is plotted to demonstrate the distance of 1 from the origin, and all circles plotted within the curve are blue, while the rest outside of the curve will be plotted red. For the second plot the monte_carlo's pi estimation is important. Lines demonstrating different seeds are plotted to show the accuracy of pi over thousands of iterations.

Goal:

Run monte_carlo to generate data, organize data for the different groups of plots, and send a heredoc of plot instructions to gnuplot, in order to generate two separate graphs representing data.

Sudo:

First the tmp files need to be set up for the monte_carlo data outputs, then monte_carlo is run 3,000 times for the first plots data points.

A loop is used to organize the data points into separate files, putting the points inside the circle in one file to be plotted as turquoise, with the points outside the circle in another file to be plotted as pink.

gnuplot parameters are set to output to pdf with axis' and a title as Plot1. Then the boundary curve, turquoise points, and pink points are plotted in succession.

monte_carlo is ready to be reused for the second plot's data, so it is run 2,000 times for each of the three unique seeds, with their own respective tmp files.

A second loop is used to organize the iteration count and the accuracy of pi, through finding the difference between the pi estimation and the real pi, as data into the respective seed files.

gnuplot is set to new parameters with the title as Plot2. Then the three unique seeds are plotted in succession to show the accuracy of monte_carlo's pi estimation over many iterations, from separate starting points.