Adam Stammer

Radar Simulation

CS 341

Preface: Back when I was using java extensively I developed a graphics engine for java that utilizes the built in awt and swing packages. It’s not a library, just a template with a lot of prewritten code and a framework to build upon. I designed it to help make games but it seemed to still be appropriate in this instance. I really need to go through and comment most of it but it in itself was not fundamental to this assignment. The code most relevant to this project you will find in the following files:

GridMenu.java – this is the meat and potatoes. Here is where all of the fancy looking stuff is done and the detection of the points. This also has the static method used to access the data file. This is probably the code you care about most.

CoordListMenu.java – this is used to list the coordinates, and that’s it. I haven’t implemented a means of removing or adding points yet, but the framework with which to do it is there.

PointObject.java – this is used for the points. It is used to store their data and render them with a visual decay timer.

My program is also lacking a lot of failsafe features, especially regarding file reading. This is something that would certainly need to be fixed. The list window also has a rather low cap on the maximum amount of point that can be displayed. My intention was to add a page system but I never got around to that.

In the case of each of my sweep detection methods, the complexity is O(n). Each one loops through each point and checks to see if the points distance is smaller than a threshold value of the sweep bar. This could be made more efficient simply by not looking at all of the points every time. This could be accomplished by sectionalizing the points, although this would have to be done for each sweep method, which would add programming complexity. Another option would be to organize the list of points based on location and adding a “cursor” of sorts so that any point already “found” wasn’t checked again. Again, this would have to be done slightly differently for each sweep method. A hybridized method would likely be the most efficient, but personally I would focus on the latter because the first method I mentioned has diminishing returns when the amount of points is too low. There are other ways to make it more efficient but those seem to be two of the more effective and straightforward approaches. I definitely used one of the most inefficient methods.

You never heard any more specifics on the design documentation you prefer so below is a picture of the design diagram I used.