

# Project 5 ReadMe

Ankith Kandikonda, avk393

Donovan McCray, dom342

## Project Structure

The GUI controls and display are all done through CritterWorld.java file. We have one stage to display the user controls and the other stage to display the actual world. We created a static GridPane to be able to update it without having to change the parameters in Critter to update the GUI. Most of the buttons and user interactive GUI fields are static for the same reason. When drawing the grid, I didn't realize the displayWorld parameter was Object Pane so I decided to create a bunch of rectangle objects and draw them on the grid. I used the width and height of the world to align the grid in order to make it even.

We created different tab objects and added each of our user control options to the corresponding tab, which is all done in CritterWorld. We also disabled user inputs/controls while the animation is running. We used AnimationTimer for this which was very easy to implement and runs at the selected frame rate chosen on the animation tab.

Displaying the graphics on the grid was a little trickier. When critters are created (CreateCritter), we draw them in CritterWorld in the addCritter function. They are created as Shape objects and get added to the corresponding coordinates (passed as parameters) on the GUI. Going through worldTimeStep, we loop through the population to find the dead critters and erase them from the gridPane. At the end of worldTimeStep, we redraw all the living critters that have moved since the last time step and draw the baby critters reproduced during that time step.

## Problems

One problem we had was fixing our errors from project 4. For most of our incorrect stats, we were only 1 or 2 off from the expected result. This made debugging very difficult as there was a lot of things going at once to keep track of. We don't know if we were able to find all the issues, but each of us took half of the failed test cases and tried to reproduce the error to figure out where it was coming from. We found most of the errors by sitting together for a few hours and sharing test cases or possible errors we think we found. We were not able to find the solution to all failed test cases because we wanted to start on project 5 so we wouldn't fall behind.

Another problem we had was drawing a star shape. Creating our shapes using the Shape abstract class in java posed some limitations. Simple shapes were easy enough like rectangle and circle, but doing diamonds and other shapes using the Polygon class was harder. First we established how to define points for Polygon, but given the small size of our grids, making a

perfect star was too difficult because there were not enough points in each location to make it look nice, so we instead decided to make it a pentagon shape.