

README

Description

- Code:
 - Code is mainly spilt in 2 files:
 - Main
 - Threading is done to manage both the background functions and javafx display for the controls
 - Buttons calling the make, seed, and Step functions from assignment4 are created and triggered by an actionEvent
 - Timer.cancel() is used to stop the animation within the javafx thread
 - Critter
 - Critter implements all the code needed to draw objects. It calculates a shape size proportional to the world size given by the Params class. And calls the polygon functions for TRIANGLE, STAR, and DIAMOND defined in the Painter class
 - Made the display GUI a 500x500display and used that to calculate the size of all the blocks in the GUI for all possible Param values
- Graphics:
 - 2 separate windows pop up:
 - One window is the controller GUI. It holds all the buttons that controls the display GUIs animation and display functions
 - The other window is the display GUI. This window consists of a GridPannel holding all critters that are instantiated by the controller GUI.

Features

- All features except the following 2 listed below work correctly:
 - Speed:
 - We could not figure out how to a slider that can change the speed at which the animation ran without it crashing within a few seconds. Therefore, we removed it and kept the animation running smoothly in 1 unchangeable speed.
 - Threading:
 - 2 threads are run in this program. One is the JavaFX application thread and another is a background thread running the logic. We attempted to make it run without eventual crash by using the Task object from `javafx.concurrent.Task<V>`. However, it does not work and program crashes in about 2 minutes of continuous runtime due to overuse of RAM.