CS 2112—Spring 2014

Assignment 6

Graphical User Interface Design

Due: April 24

Draft overview due April 17

In this assignment you will use the JavaFX API to build a graphical visualization of the critter world described in the project specification. This visualization will have a graphical user interface that permits you to advance time in the world simulation and also to inspect and take control of one critter at a time.

0 Changes

- Removed mention of plants. There are no plants. (4/14)
- Added more detailed requirements for displaying the currently selected critter. (4/13)
- The UI for controlling a critter has been converted to a karma problem. (4/13)

1 Requirements

Your program should be able to display the current state of the world, which it will initialize by loading the world files from Assignment 5. The critters will be controlled by programs using the simulation engine you have already built.

The user interface must allow the user to step the world state one time step at a time, or to start the world running continuously. In either case, the graphical display will be updated as the simulation progresses to show the new state of the world.

The total number of time steps taken must be displayed on the user interface, along with the total number of critters alive in the world. As in Assignment 5, the user should be able to create a new world, load a world and to load a specified number of critters.

When loading a critter program file, the user should be able to either specify a number of critters to be randomly placed throughout the world or to select a particular hex in which to place a critter.

The user should be able to set the maximum rate at which the simulation advances. Regardless of how quickly the simulation is progressing, the graphical display should not be updated more often than 30 times per second. Thus, if the simulation is progressing very rapidly, the world state may not be displayed for some time steps.

Another part of the user interface will allow the user to inspect a single critter somewhere in the world. The user can click on the hex containing a critter to make it the currently displayed critter. The user interface will indicate which critter is currently displayed and will also display the state of the controlled critter, corresponding to the 8 initial memory locations, along with the critter program and information about the most recently executed rule. As the simulation progresses, this information will be updated accordingly.

For more details about how the simulation of critters and other parts of the world work, consult the project specification.

1.1 Running your program

Your program must support the following command-line interface:

- `java -jar <your_jar>`
  Start the simulation with a world populated by randomly placed rocks. The program should automatically read the input file `constants.txt` and set the value of the various simulation parameters accordingly.

Note that there are no command line options. All user interaction should be done through the graphical user interface.

2 Evaluation

We will be evaluating your user interface on multiple grounds. The visual appearance and layout will be factors, as will the design of the controls. We are looking for an attractive and functional user interface, but we have not specified its precise appearance and layout, nor exactly how the UI will allow the user to control the system. This is deliberate; we want you to think through the design. It's a good idea to storyboard and also to experiment with more than one UI design. See what works best—don't get locked into design decisions too early in the process.

Take care not to entangle your model with this new interface, as proper separation of the simulation and UI will factor into your grade The model should not depend on the user interface in any way because such a dependency will prevent a distributed implementation of the simulation, affecting your ability to be successful in Assignment 7. We will also be looking for good documentation of your classes and their methods, using the documentation methodology described in class.

GUI code often becomes quite long and you will likely have to make more conscious effort to keep you code clean and readable then in previous assignments. In addition to organizing your classes in packages, give thought to how you will organize external resources (The constants file, plus any other files you add).

3 Programming tasks

You will want to figure out with your partner how to break up the work involved in this assignment. To get you started thinking about this, here are some of the major tasks involved:

- Designing the user interface with all the components needed to control critters. A good way to start this is by "storyboarding" the user interface with sketches that show how the different user interface components will be placed on-screen. Don't forget that not all components need be visible at all time. Consider menus, pop-ups, tabbed panels, etc., when sketching possible designs.

- Implementing a new component or components to display the state of the critter world. This will involves graphically rendering hexes and critters. It should be possible to least distinguish critters of different species (as defined by the program), and to see the size and direction of a critter.
- Connecting the simulation engine from A5 to the display. This means both allowing the display to update as the simulation progresses and allowing the display to stop, start and step the simulation.
- Implement the loading of both critter and world files, including placing critters into an existing world.

4 Restrictions

You may use any standard Java libraries from the Java 7 SDK. Please avoid using features only found in Java 8. If for some reason you want to use a third-party library, this is likely ok, but post to Piazza to receive confirmation before using it. We have provided no skeleton for this assignment, and you need not adhere to skeletons given in earlier assignments.

Further, you may hand-code your GUI, write the UI in JavaFX's XML, or use the GUI builder. You are not restricted on design tools.

5 Overview Draft

As on previous assignments, we are requiring you to submit an early draft of your design overview document. You may not be able to predict what your design and testing strategy will look like in full at that point, but we want to see how far you have gotten. THis is a good time to submit design sketches for the UI. But don't think of this as much as a grade as an opportunity to get feedback early.

6 Submission

You should compress exactly these files into a zip file that you will then submit on CMS:

- *Source code*: You should include all source code required to compile and run the project.
- *Other files*: It is possible to use other files as part of your UI. For example, you might read in image files or other data files that control appearance. Don't forget to include these.
- `overview.txt/html/pdf`: This file should contain your overview document. This should include a discussion of why you choose the design you did and what other designed you

considered. This document is your opportunity to make the case for your design, and is very important.

Do not include any files ending in `.class`, or the `bin` directory.

7 Karma

For good karma, add a user interface that allows the user to control the selected critter and decide what move it performs on each turn.