A SuperPixelGrid Companion

Instructions:

1. Consult the project README hosted at http://github.com/APCSJava/SuperPixelGrid
    a. What is the stated purpose of this lab?
    b. In order to compile the classes, what additional libraries will be needed?
    c. How can MacOS users avoid errors that occur when holding keys down?
2. Download or clone the project using the green button.
3. Unzip the downloaded file, if necessary, and move it to your CS folder
4. Open the project using your favorite IDE (JGrasp, for example)

Getting to know SuperPixelGrid:

SuperPixelGrid comes with an executable JAR file. Use it to experiment with the running program and answer the following questions:

1. Describe the appearance of the window, when opened.
2. Place the mouse toward the center of the screen and type a 'T'. Describe and interpret what you observe.
3. Type 'T' a second time. What happens?
4. Type 'W' then the space bar. Repeat several times and describe.
5. Experiment with the 'R', 'G', 'B' and 'C' keys. Which one does not behave like the others? In what significant way is it different?
6. Try out the Left and Right keys. In what ways are they similar? How different?
7. Experiment with the Up and Down keys. Again, does their behavior match exactly?
8. How does clicking the mouse on a superpixel affect the grid?

Exploring the class structure:

1. There are 3 classes involved, *SuperPixel*, *Colorizer*, and *SuperPixelGrid*
    a. Which of these is responsible for running the program?
    b. Which is the simplest/most straightforward class?
    c. Which class will you need to implement/develop?
2. Examine the source code for the *Colorizer* class.
    a. How many methods are there? How many have been implemented for you?
3. How is information passed from SuperPixelGrid to the Colorizer class?
4. How is information passed from Colorizer to the SuperPixelGrid class? Always?

That's Life:

Research mathematician John Conway and his Game of Life a form of cellular automata. What rules determine whether a cell lives or dies from one generation to the next? Implement those rules in response to the 'L' key. Once all is working correctly, modify SuperPixelGrid to simulate and display a grid much larger than 32 x 32.