

A SuperPixelGrid Companion

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'. Wiggle the mouse. Describe and interpret what you observe.
3. Type 'T' a second time. What happens?
4. Type 'W' followed by 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. Can you find the difference between them?
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?

Instructions for completing the project:

1. There are 3 classes involved, *SuperPixel*, *SuperPixelGrid*, and *CommandHandler*
 - a. Which is the simplest/most straightforward class (POJO)?
 - b. Which of these three classes is responsible for running the program?
 - c. Which class contains stubs for methods you will need to implement?
2. Examine the source code for the *CommandHandler* class.
 - a. How many methods are there? How many have been implemented for you?
 - b. IMPLEMENT all missing methods (except `lifeCommand`). Test as you go.
3. How does the *SuperPixelGrid* communicate with the *CommandHandler* class?
4. How does the *CommandHandler* class communicate with the *SuperPixelGrid* class?

Beyond the project – That's Life (extension):

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 the *SuperPixelGrid* runner class to simulate and display a grid much larger than 32 x 32. This should only require changing one to three lines of code.