Alisha Adam & Rohit Talreja
CS 106A – Summer 2016

Section Handout #7 – Iterators, GUIs

Based on handouts by Keith Schwarz, Marty Stepp, and Mehran Sahami

Iterators

1. Remove Even Length (with iterators). Write a method named **`removeEvenLength`** that accepts an iterator over strings as a parameter, and removes all strings of even length. For example, suppose we have an `ArrayList` called `list` containing `["hi", "there", "how", "is", "it", "going"]`. After calling `removeEvenLength(list.iterator())`, `list` should contain `[“there”, “how”, “going”]`.

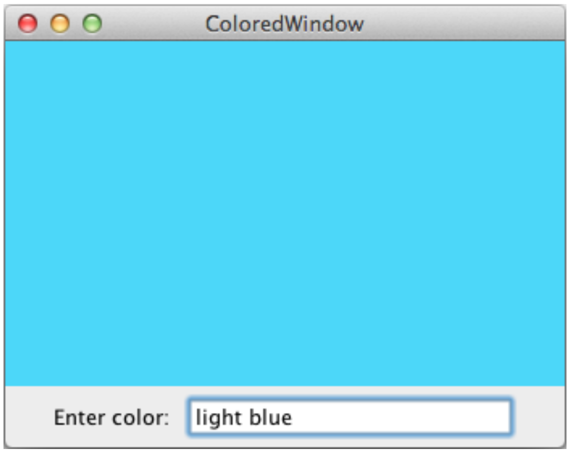
GUIs

2. Window Coloring. Write a GraphicsProgram that allows the user to set the background color of a window by typing the name of a color into a text field and pressing enter. You should look up the red/green/blue values of a particular color from a text file named colors.txt, which contains the name of a color on one line and the values of the red/green/blue components for that color on the next line. *(The data in this file comes from an experiment done by the web comic* xkcd *where people were shown pictures of colors and asked to name them).*

Recall that to create a Color out of its red, green, and blue components, you can use the Color constructor by writing new Color(r, g, b). Your program should only need to read the file once (i.e. you should not re-read the file each time the user types in a color). If the color entered is not found, your program should do nothing.

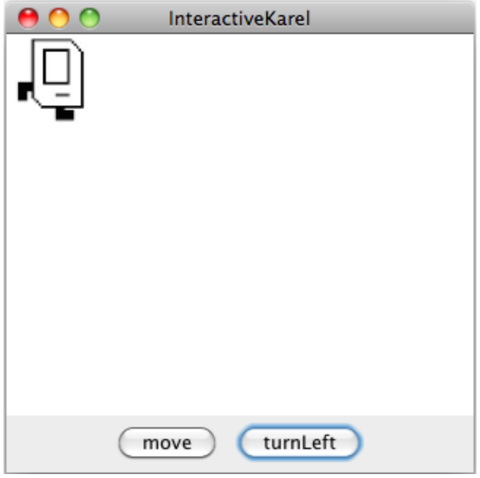
Example colors.txt:

```
pastel blue
72 100 175
baby blue
182 226 245
purple
130 64 234
blue
75 49 234
olive green
111 145 122
```



Enter color: light blue

3. DIY Karel. Write a `GraphicsProgram` that imitates some aspects of how Karel the Robot behaves. As shown in the diagram, you should create a **move** button and **turnLeft** button. Pressing move causes Karel to move one length in the direction it is facing, and pressing turnLeft causes Karel to rotate 90 degrees to the left. If moving Karel would cause it to go off the screen, just keep Karel where it is. You can assume that you are given four images, one for each of the four directions that Karel can face (these images are called `KarelEast.png`, `KarelWest.png`, `KarelNorth.png`, and `KarelSouth.png`,). You can assume that each of these images is 64 pixels in width and height.


InteractiveKarel
move
turnLeft