Eric Roberts
CS 106A

Handout #9
January 6, 2016

Karel Contest

Due: 5:00 P.M., Wednesday, January 20

The first assignment is designed to teach you about Karel's capabilities and to reinforce basic programming concepts. Karel is a good medium for teaching such concepts but also has other virtues. In particular, programming Karel can be a lot of fun. This contest gives you the opportunity to explore this aspect of Karel's world. It is entirely optional, but gives you a chance for some extra credit in the course.

Your mission, should you decide to accept it, is to program Karel to solve an interesting and exciting problem of your own choosing. You can program Karel to produce the definitive work of computer art, to illustrate a story, or to tackle a conceptually difficult task. The entries will be judged by the CS 106A staff (see official rules below), and a prize will be awarded in each of two categories:

- *Aesthetic merit.* This prize is awarded to the program that, in the opinion of the judges, has the greatest literary, artistic, or entertainment value.
- *Algorithmic sophistication.* This prize is awarded to the Karel program that solves the most challenging task in the most interesting way.

Prizes

The first prize in each of the categories will be that we will replace whatever individual score most negatively affects your grade with a 100%. Thus, if you are one of the two Karel contest winners, and you bomb an assignment, the midterm, or even the final, we will overlook that misstep and count it as a 100%. By putting in a little extra effort now, you could reduce substantially the amount of pressure later in the course, which can come in handy. Best of luck!

New extensions

The current implementation of Karel includes a few extensions beyond those described in *Karel the Robot Learns Java*. These new extensions are:

1. *Better support for animation.* The old version of Karel made it difficult to write animated programs because there was no good way to control the speed of the display. The new version of the `SuperKarel` class includes a new built-in command

   ```
   pause(milliseconds);
   ```

   that suspends Karel's operation for the specified number of milliseconds. The usual approach to using this statement is to have Karel perform some operation and then pause for a short time (typically on the order of 20 milliseconds or so) to give the display time to catch up. We will use this same approach when we start writing animated programs in Java. For more details, you can look at the section on "Simple

graphical animation" on page 122 of the Java book. If you use **pause**, you will want to run Karel at the maximum speed.

2. *A larger range of colors.* The new **SuperKarel** class allows you to paint squares in a much more expansive range of colors. In the new version, the **paintCorner** method has been extended so that you can call it as

   ```
   paintCorner(red, blue, green);
   ```

   where *red, blue,* and *green* are numbers between 0.0 and 1.0 indicating the intensity of the corresponding color. For example, calling

   ```
   paintCorner(1.0, 0.0, 1.0);
   ```

   is identical to calling

   ```
   paintCorner(MAGENTA);
   ```

   Using more precise values for *red, blue,* and *green* allow you to generate millions of colors. For example, you can create a nice brown color using the command

   ```
   paintCorner(0.35, 0.20, 0.05);
   ```

   The same extension applies to the **cornerColorIs** test condition.

Official rules:

1. Only students registered in CS 106A are eligible to submit contest entries.
2. Only one entry per student will be accepted.
3. All entries must be submitted electronically through the submission system in Eclipse and must be received by 5:00 P.M. on Wednesday, January 20. Late entries will not be accepted.
4. Each submission must consist of a Karel program and one or more worlds for execution. In addition, you may submit a short text explanation, not to exceed 250 words, describing what Karel is doing.
5. Karel programs must limit themselves to the language features described in *Karel the Robot Learns Java* for the **Karel** and **SuperKarel** classes, along with the new extensions described earlier in this handout. You may **not** use other features of Java, even though the Eclipse-based version of Karel accepts them. In particular, you may not declare variables or use the value of the **for** loop index.
6. Contest entries should be sensitive to Stanford's individual and cultural diversity. Programs or narratives that have the effect of perpetuating negative stereotypes will not be eligible for prizes.
7. Contest entries will be evaluated initially by Eric Roberts and Alisha Adam. The best entries will then be evaluated by representatives of the entire course staff, which will choose the winners and runners-up in each category.