Homework 2: A Classy Karel

Due Date: Tuesday, February 12, 2008.

There is a possible 45 points for this homework.

Problem 1. (10 pts.) Teach (i.e. program) Karel to again play fetch. Program Karel, starting from the origin (1,1) facing east, to jump over hurdles, pick up the beeper just beyond the hurdles, return to the origin and place the beeper there (i.e. at you feet). In this robust version of fetch you do not know how many hurdles there are, nor the configuration of each hurdle. Hurdles can now be of arbitrary height and/or width.

To solve this problem you must create one or more new robot classes (e.g. a BetterUsedRobot class and a FetchOverHurdle class).

Problem 2. (10 pts.) Program a robot named Karel to escape from a maze that contains no islands. The exit of the maze is marked by placing a beeper on the first corner that is outside the maze, next to the right wall. This task can be accomplished by commanding Karel to move through the maze, with the invariant that its right side is always next to a wall. Figure 1 shows one example of a legal maze.

Figure 1: Sample Maze World: starting Configuration
Problem 3. (10 pts.) In Lesson 22 of the Rurple tutorial you are introduced to the “Its Raining” problem. Create a program that has the robot solve this problem; that is have your robot, which is instantiated with sufficient beepers in its bag, “close” all the open windows by placing a beeper in front of the window. You may assume that each window is only 1 wall width wide and that there are no corner windows. Unlike the room in the tutorial, the room may be an irregular shape - not necessarily a rectangle.

To solve this problem you must create one or more new robot classes. e.g. A BetterUsedRobot class, a SimpleWindowCloser class for rectangular rooms, and a WindowCloser class that builds on the SimpleWindowCloser class to work on irregular shaped rooms.

Problem 4. (15 pts.) Program a robot name Karel to arrange vertical piles of beepers into ascending (or descending) order. Each avenue, starting at the origin, will contain a vertical pile of one or more beepers. The first empty avenue will mark the end of the piles that need to be sorted. Figure illustrates one of the many possible initial situations and corresponding final situation.

![Figure 2: Initial and Ending Configurations](image)

Problem 5. (10 pts. Extra Credit) Create a “smarter” Karel; one that is capable of doing arithmetic. Using beepers to indicate magnitude and a positional numbering system, program Karel to do addition. Figure shows the before and after configurations for adding together 2794 and 5132. This is a hard problem so create a simple robot class to add one column of two numbers (without carry) and then make successively more complicated classes: multiple columns, carry out of the leftmost column, etc.

IMPORTANT: For each problem you solve, create one or more classes which inherit from your previously created “utility” robot class BetterUsedRobot. It is highly recommended that
for each problem you create a class hierarchy that solve successively more complicated versions of the task. In addition to correctness (and documentation), you will be graded on your class decomposition/design.

To submit your programs you need to:

1. print out a hardcopy to physically hand in on the due date.
2. email the code to mikeyg@cs.xu.edu for testing.

There will be a 5 point penalty for not doing both submission tasks.