

## Homework 1: Fetch Karel, Fetch

**Due Date:** Thursday, January 31, 2008.

There is a possible 20 points for this homework.

**Problem 1.** (10 pts.) Teach (i.e. program) Karel to play fetch. Program Karel, starting from the origin (1,1) facing east, to jump over three hurdles, pick up the beeper just beyond the hurdles, return to the origin and place the beeper there (i.e. at you feet). You are to employ the top-down approach described in the class. Furthermore, each function that you define should have at least two lines of documentation accompanying it: preconditions and postconditions. Figure 1 illustrates the world that Karel must play fetch in.

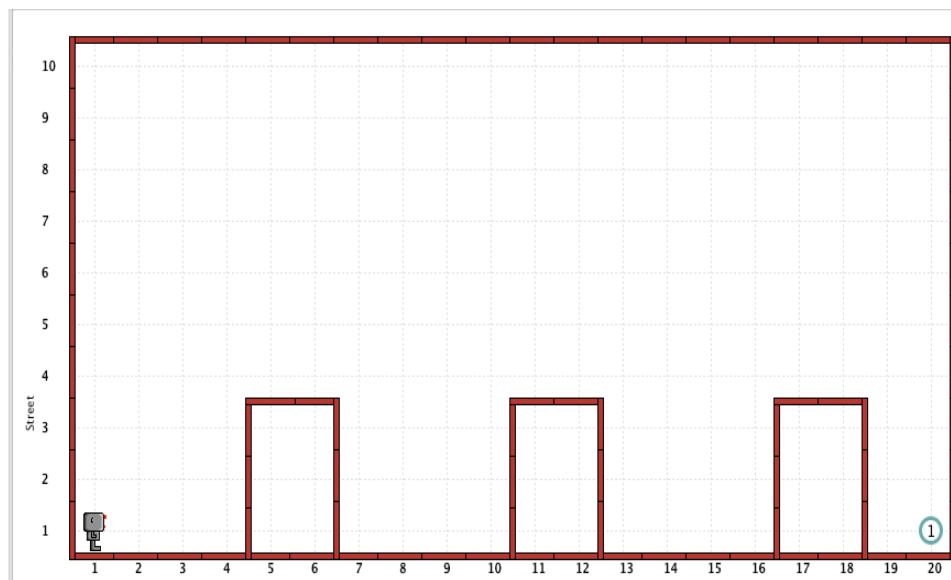


Figure 1: A Fetching World

**Hint:** It always helps to solve a simpler problem first if having trouble with the assigned problem. For example instead of there being three hurdles, create a world with only one hurdle and see if you can get Karel to play fetch in that world. Once successful, you can then expand Karel's playground to first include two hurdles and then eventually three hurdles. Don't move on to the next step until you are successful with the earlier (hopefully easier) step. This is called "iterative improvement" and is a classical strategy in solving problems.

**Problem 2.** (10 pts.) Teach (i.e. program) Karel to “harvest” a baseball diamond field of beepers. See Figure 2. The precondition for this problem is that Karel is at the origin (1,1) facing east. The postcondition is that Karel is at 6th avenue and 9th street, facing north with 16 beepers in its bag.

You to again apply the principle of top-down problem solving: that is use functions to encapsulate each stage of the solution. Its called top-down because one first thinks about and designs the high-level functions that essentially solve the problem (e.g. `harvestField`). One eventually works one’s way down the chain of functions until one gets to the bottom and works on the low-level detailed functions (e.g. `turn_right` or `moveToNextRow`).

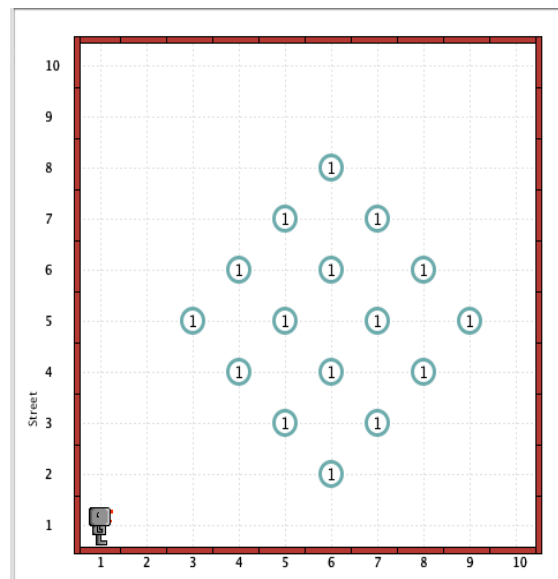


Figure 2: The Field of Dreams

In addition to correctness and documentation, you will be graded on your function 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](mailto:mikeyg@cs.xu.edu) for testing.

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