CS 374, Winter 2010: Assignment 1

Bouncing Cells

This is the first assignment where you will start building some funny cells. To know more about funnycells, please take a look: http://www.cse.iitd.ac.in/~aseth/assg/funnycells/funnycells.html

As part of this assignment, you are expected to program a *bouncing cell* which bounces off the boundaries of the map.

Expected behaviour of your cell

When launched at a random location on an arbitrary sized map, your cell should perform the following movement operations from its original location:

1. Move upwards until it hits a wall
2. Move downwards until it hits a wall
3. Move back up to its original location
4. Move left until it hits a wall
5. Move right until it hits a wall
6. Move back to its original location

Thus, the objective is to traverse to all four walls of the map top, bottom, left, right, in that order and come to a stop at the original location.

Implementation

You are given an empty *BouncingCellImpl* class which looks like this:

```
public class BouncingCellImpl implements CellularProcesses {

    public BouncingCellImpl(Integer type, Integer energy, String cellId,
        RateLimBufferedReader in, RateLimPrintWriter out) { }

    public void startCell() { }

}
```

The source file (***BouncingCellImpl.java***) is placed in the funnycells.0.1/src/assg/funnycells/cells folder. On running build.sh, the source code will be compiled and the .class file placed in the funnycells.0.1/build/assg/funnycells/cells.

Execution

You first have to launch the server as follows:

java -cp "lib/java-getopt-1.0.13.jar;build" assg.funnycells.server.Coordinator –c funnycells.conf -l [log level]

> -c [config file]: The configuration file, default being funnycells.conf present in the current directory
> -l [log-level]: 0 (no logging), 1 (info), or 2 (debug)

Your cell can then be launched through the following command, which automatically instantiates your cell by calling its constructor, and starts it through the startCell( ) method.

java -cp "lib/java-getopt-1.0.13.jar;build" assg.funnycells.cells.FunnyCell –c funnycells.conf -x [x-coord] -y [y-coord] –t1 –icell00 -l [log level]

> -c [config file]: The configuration file, default being funnycells.conf present in the current directory
> -x [x-coord]: The x coordinate at which to create the cell. The top left corner is (0, 0), and the size of the map is defined in funnycells.conf
> -y [y-coord]: The y coordinate at which to create the cell
> -t [type]: This can be 0 (susceptible cell), 1 (infectious cell), 2 (antibody cell)
> -i [cell-id]: Any string for a cell-id. Each cell has to have a unique id
> -l [log-level]: 0 (no logging), 1 (info), or 2 (debug)

The FunnyCell class actually creates a socket for communicating with the server and passes the input (*in)* and output (*out)* streams to your constructor, which are of type RateLim. To read and write, you can then use the in.readline() and out.println() statements.

For example, to send a MOVECELL message to server (message formats are given on the website), you can use the following set of statements:

```
out.println(assg.funnycells.server.Cell.MOVECELL);
out.println("incx=1");
out.println("incy=1");
out.println("");
```

Evaluation

To evaluate your assignment, your cell will be launched in a map of arbitrary size and shape between 10x10 and 50x50. The original location (coordinates) of the cell will be arbitrary and unknown to the cell. The cell must traverse to the boundaries as described above in all the runtime scenarios to be judged successful.

Submission Guidelines

You are expected to adhere to the following submission guidelines:

1. The assignment must be submitted on the moodle online course management system: http://jaijaivanti.cse.iitd.ernet.in/moodle/course/view.php?id=73

2. Your submission *must* contain 2 files: source code ***BouncingCellImpl.java*** and a report containing a brief description of your approach to solving the problem in pdf format.

3. Your submission must be a .tar.gz file named as your entry number. The archive must contain one folder with the name as your entry number. This folder must further contain your source file ***BouncingCellImpl.java*** and your report.