Assignment 2: Attacking cells!

This assignment is based on funny cells. Please look up the following webpage for details: http://www.cse.iitd.ac.in/~aseth/assg/funnycells/funnycells.html In this assignment, you are expected to program *infected cells* that search the map to find and infect susceptible cells. There are two parts in this assignment.

Assignment 2(a)

Objective

The simulation will be started with an arbitrary number of infecting cells launched in random locations. Your objective is to find and infect susceptible cells on the map. Remember that a single infected cell cannot infect a susceptible cell; you need a quorum of at least two infected cells to trigger a successful infection. When infected, the susceptible cell will become yet another infected cell and participate in the infection process. The infected cells must find and infect *all* susceptible cells on the map.

Implementation

Name your class ***AttackingCellImpl***. As before, a stub AttackingCellImpl.java is provided, in which you have to fill in your implementation.

Molecules

As explained on the webpage, infections are triggered by producing the following molecules: *gggc-atat-atga-cccg*. And you have already learned how to detect susceptible cells, and communicate among your own cells.

Execution

Start the server as before:

```
java -cp "lib/java-getopt-1.0.13.jar;build" assg.funnycells.server.Coordinator –c funnycells.conf
–l0 –x0 –w0
```

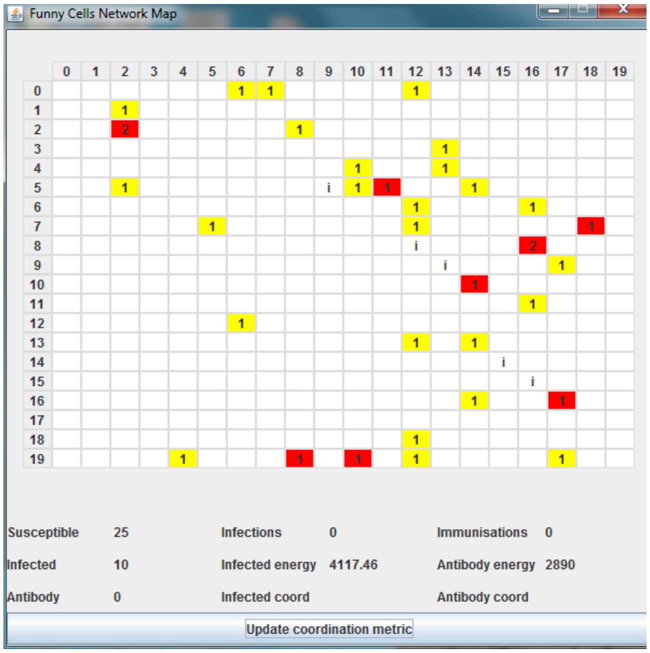
Note the new –x option; this will come in handy in the second part of the assignment.

The –w0 option turns walls off, effectively converting the map into toroid topology.

For launching cells, we have written two scripts for you:

```
perl randomlaunch.pl [map width] [map height] [energy cells] [infected cells] [antibody cells]
[susceptible cells]
```

Set the number of energy cells and antibody cells to 0 for this assignment. This script will launch susceptible cells in random locations on the map, something as follows:

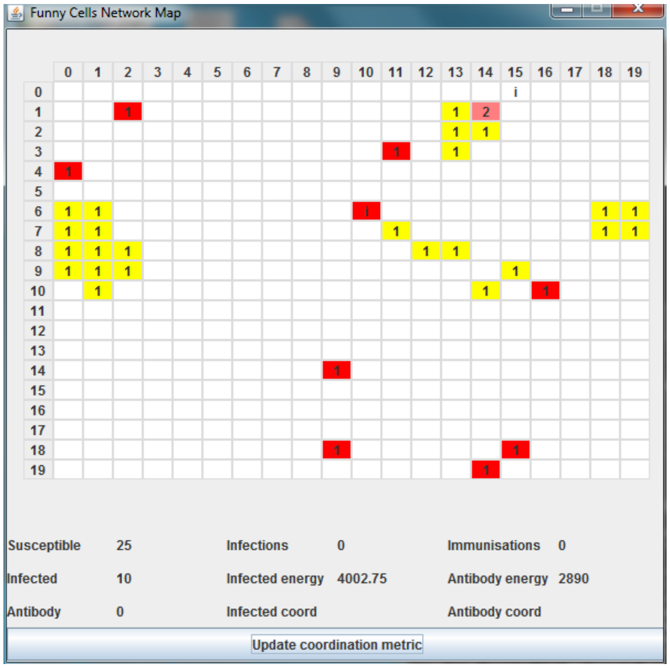


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0							1	1					1							
1			1																	
2			2						1											
3														1						
4											1			1						
5			1							i	1	1			1					
6													1				1			
7						1							1						1	
8													i				2			
9														i				1		
10															1					
11																	1			
12							1													
13													1		1					
14																i				
15																	i			
16															1			1		
17																				
18													1							
19					1				1		1		1					1		

Susceptible	25	Infections	0	Immunisations	0
Infected	10	Infected energy	4117.46	Antibody energy	2890
Antibody	0	Infected coord		Antibody coord	

And another script to launch susceptible cells in different clusters:

perl clusterlaunch.pl [map width] [map height] [energy cells] [infected cells] [antibody cells] [susceptible cells] [joining probability]



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0																i				
1			1											1	2					
2														1	1					
3												1		1						
4	1																			
5																				
6	1	1									i								1	1
7	1	1										1							1	1
8	1	1	1										1	1						
9	1	1	1													1				
10		1													1		1			
11																				
12																				
13																				
14										1										
15																				
16																				
17																				
18										1						1				
19															1					

Susceptible	25	Infections	0	Immunisations	0
Infected	10	Infected energy	4002.75	Antibody energy	2890
Antibody	0	Infected coord		Antibody coord	

Here, joining probability is a parameter between 0 and 1 that new susceptible cells may join an existing cluster, or start their own cluster. Thus, a value close to 1 will result in a few large clusters, and a value close to 0 should result in many small clusters. Think why you may need different search strategies in clustered Vs a random scenario!

Report

Your report should contain a description of your approach (algorithm) to quorum formation, search of susceptible cells, and infection of the cells.

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.

2. Your submission *must* contain 2 things: source code ***AttackingCellImpl*** and your report 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 ***AttackingCellImpl*** and your report.

Assignment 3(b)

Objective

The objective is to do a statistical study of time taken to infect all susceptible cells on the map under different launch scenarios. This part of the assignment is experimental. You are required to test your infected cell under four scenarios:

1. Susceptible cells are launched randomly on the map and die upon being infected. This will tell you the amount of time taken to search the entire map and infect the cells.

2. Susceptible cells are launched randomly on the map and change into infected cells upon being infected. This will of course take lesser time than in the previous scenario, but how much lesser is the question! Do the times decrease linearly, or exponentially, or are these transition points somewhere?!

3. Susceptible cells are launched in clusters and die upon being infected. When compared with the first scenario, this will help you understand the impact of clustering on search.

4. Susceptible cells are launched in clusters and change into infected cells upon being infected. Similar to the earlier scenarios, you should contrast this with the search time in a clustered layout and search + infection in a random layout.

For each of these scenarios you are required to plot a graph showing the *Initial number of Susceptible Cells* (x-axis) vs. *Time taken to infect all susceptible cells* (y-axis). Note that each point on the graph is actually an entire simulation run! And for each point, you have to take multiple readings (at least 10) to be able to show error bars for statistical accuracy!

Execution

This is similar to the first part, except that you will always launch your server with the –x1 option:

```
java -cp "lib/java-getopt-1.0.13.jar;build" assg.funnycells.server.Coordinator –c funnycells.conf
–l0 –x1 –w0
```

This will cause the server to terminate when all susceptible cells have been infected.

With this configuration, you will have to write your own scripts to run the simulation multiple times with different parameter values. The scripts should make use of the server logs to track the time taken to infect all cells. The server logs the following values:

- NEWCELL: cell-id (x, y) + type = t: [time in milliseconds since 1970]

  for each new cell introduced in the simulation. Your timer should start from the first susceptible cell created in the simulation.

- SERVEREXITING: [time in milliseconds since 1970]

  when the simulation ends, after all susceptible cells have been infected. The difference between the server-exit time and the first new cell launch time gives the duration of the simulation in milliseconds.

Now, to get results for the all four scenarios listed above with or without clustering, you will need to change your configuration files. Pay attention to the *newinfectedcell* value. If left blank [*newinfectedcell* =] then susceptible cells will die out upon infection, and if given the same value as infectedcell [*newinfectedcell = assg.funnycells.cells.AttackingCellImpl*], the susceptible cells will convert into infected cells and start infecting. So, you need to do the following:

1. Scenario 1: Use *newinfectecell =* , and run *randomlaunch.pl*

2. Scenario 2: Use *newinfectedcell = assg.funnycells.cells.AttackingCellImpl*, and run *randomlaunch.pl*

3. Scenario 3: Use *newinfectedcell =*, and run *clusterlaunch.pl*

4. Scenario 4: Use *newinfectedcell = assg.funnycells.cells.AttackingCellImpl*, and run cluster*launch.pl*

Report

Your report should contain details of your experiments, scripts, your observations including graphs, and intuitive explanation of your findings. The report will carry a considerable weightage for this part of the assignment.

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.

2. The main submission for this part of the assignment is the report. You may make changes to your infected cell for assignment 3(b) to optimise its behaviour for different scenarios. If so, then resubmit your new implementations as well. We would suggest that you only develop a single implementation that dynamically changes its behavior, because you will need it for the next assignment anyway.