Ray Alfano
7/15/12

Instructions for use:
Import the Java project into Eclipse if possible.
If not, the main class is contained in BiologicalSystem.java
The configuration file is named “configFile.properties” and it is located in the project package. It is read into BiologicalSystem.java via a relative path and the parameter values are extracted from there.

The main class that should be run is BiologicalSystemUI.java, and that will display the organism state over the successive invasions. I was able to get the program to end properly, but it may throw some thread errors to console upon doing so. They do not affect the integrity of the data itself.

The configuration file contains within itself the instructions for controlling parameters. The main parameters are the size of the square grid, the number of antibody “sectors” to be distributed throughout the protection area, which is the entire simulation space.

Implementation notes:
Users are able to select a number of possible invasions that happen in sequence (not all at once). Antibodies may be added manually as well. The protection phase of the program is performed programmatically before invader objects are added. Invader objects are structurally similar to the antibodies in almost every way except that I do not allow them to clone. Whereas initially I hoped to allow them to multiply, destroy antibodies, and mutate themselves, time did not permit me to complete error checking for that implementation. Antibodies currently initialize themselves as knowing their self-color range, and assume that all invading objects are enemies.

The RGB values given for each invader and antibody determine the affinity between the two objects. The Hamming distance between the RGB values of invader and antibody determine the effectiveness of antibodies within range at destroying the invader. In my implementation, antibodies within range of an invader clone themselves within a range of RGB values and are represented as controlling grid sectors. This could be viewed as “protected zones” in a military context, such as a base that generates soldiers to fight an invader over time. The number of clones created in the grid sectors represents the total number of selected clones that combat the invading cells.

The local threshold for destroying an invading cell is computed by the contributions of all antibodies within range of the invader that are effective at combating an invading cell. The effectiveness is calculated by the Hamming threshold of 30 minus the Hamming distance between a local antibody’s RGB color and the invader’s RGB color. If the threshold value (30-antibody’s hamming distance to invader) is a positive value, then the antibody is considered effective against the invader, and if the total local contribution of the additive Hamming thresholds is above 10, the invader is destroyed.

My GUI shows the number of clones in an antibody sector and the color value average for the local sector. Invaders do not change in color and are represented by larger circles than the antibody objects. The colors for antibodies change over time as the values converge towards the invader color in order to destroy the invader. Implementing “memory” for the biological system proved elusive, but the effectiveness of antibodies in the protected area does increase over time due to larger numbers of antibodies with a better “spread” of color values to combat invaders.

Please note that in the configuration file there is the possibility that color values too far apart between invaders and antibodies can result in no detection and endless cloning without effective results. I hope to fix this if there is still time before the deadline, but in the time that I have had available my implementation is the best I have been able to put together that gives useful experimental results.

Experiment notes:
My GUI shows the number of clones in an antibody sector and the color value average for the local sector. Invaders do not change in color. Memory is implemented conceptually in this model as the local clones do not decay within the time span of the simulation. However, the desired result of a “protection force” building followed by successive invasions being destroyed faster is implemented. The antibody “sectors” that have previously fought against invaders retain their enhanced abilities to fight against invaders and can deal with later invaders faster. A few hours before I submitted this assignment I figured out how to allow effective antibodies to move towards the invader to maximize the local fighting effectiveness. This is a more biologically-relevant model than the fixed-sector model I originally described.

It is important to note that the function that distributes the

Experimental setup:

All of these experiments are given with the parameter values presented in the config.properties file, which are basically 5 invaders, who appear one after the other in fixed positions.

Initial antibody RGB values, which are then affected by a range distribution:

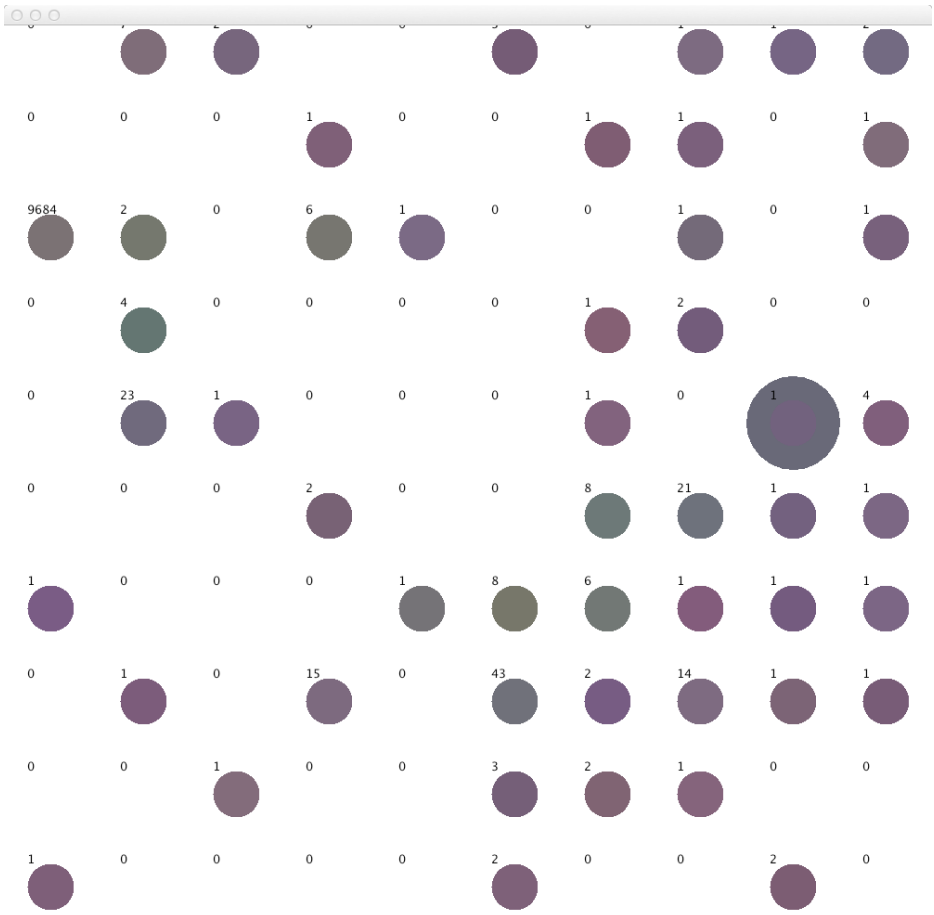
```
ANTIBODY_R=100
ANTIBODY_G=75
ANTIBODY_B=100

INVADER1X=5
INVADER1Y=5
INVADER1R=85
```

```
INVADER1G=100
INVADER1B=75

INVADER2_ACTIVE=enabled
INVADER2X=1
INVADER2Y=2
INVADER2R=75
INVADER2G=100
INVADER2B=75

INVADER3_ACTIVE=enabled
INVADER3X=8
INVADER3Y=4
INVADER3R=80
INVADER3G=80
INVADER3B=95

INVADER4_ACTIVE=enabled
INVADER4X=9
INVADER4Y=7
INVADER4R=95
INVADER4G=80
INVADER4B=80

INVADER5_ACTIVE=enabled
INVADER5X=3
INVADER5Y=4
INVADER5R=100
INVADER5G=80
INVADER5B=90
```

In practice, the invaders may shift around, as may the antibodies for better distribution to prevent a situation where no antibodies combat an invading object.

Experiments Part 1:

Experiment 1.1: First, I present isolated situations wherein a color object is easily defeated by high-affinity objects who do not need to clone many times:

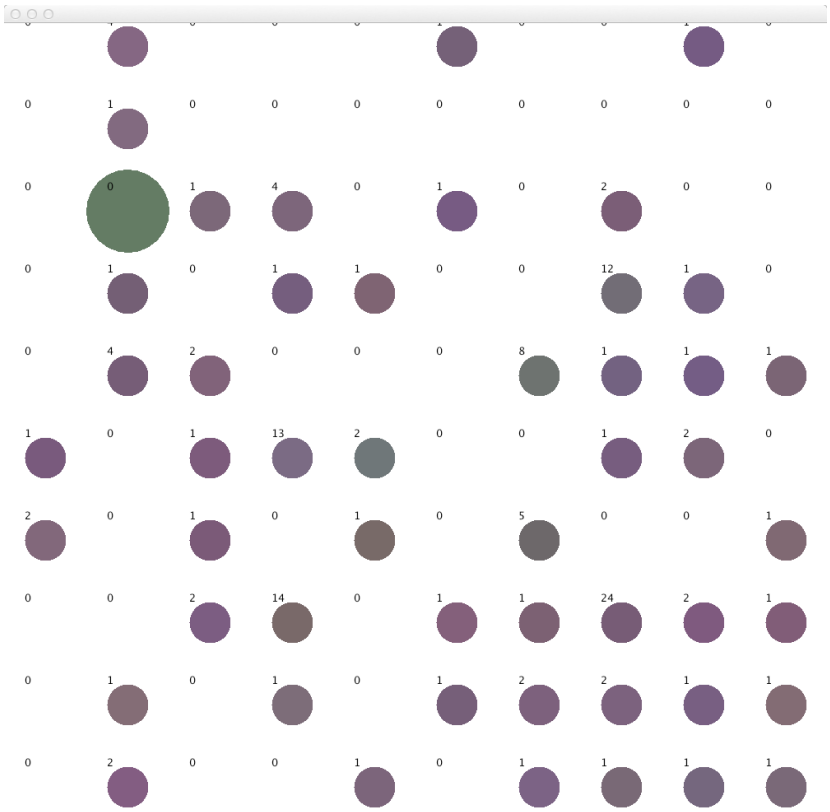


The large circle to the right is a 3rd-wave invader, and no objects in the vicinity have a trained immune response. On the next cloning cycle, the invader is destroyed because of the high-affinity antibodies in its locality.

In this situation, the high-affinity creates a local intensity high enough to defeat the invader without a large amount of cloning. The following invaders met the same fate due to coincidence of RGB distribution among the antibodies, resulting in a situation where the first two invaders took a while to reach an immune response concentration by cloning, and the following three invaders were readily destroyed. This is pictured below:

```
156 cloning steps taken so far.
invader removal triggered. X: 1, Y: 2
9878 cloning steps taken so far.
invader removal triggered. X: 8, Y: 4
9897 cloning steps taken so far.
invader removal triggered. X: 9, Y: 7
9901 cloning steps taken so far.
invader removal triggered. X: 3, Y: 4
9904 cloning steps taken so far.
```

Experiment 1.2:



This invader is dissimilar to the neighboring antibodies. It eventually takes 52 cloning steps to reach a local concentration of force.

Experiment Part 2:

The results of the above parameter sets are given with their results over the course of several runs. The contribution of how evenly distributed the antibodies are becomes important:

Run 1: Relatively evenly distributed antibodies:

```
invader removal triggered. X: 5, Y: 5
```

```
130 cloning steps taken so far.
invader removal triggered. X: 1, Y: 2
166 cloning steps taken so far.
invader removal triggered. X: 8, Y: 4
184 cloning steps taken so far.
invader removal triggered. X: 9, Y: 7
189 cloning steps taken so far.
invader removal triggered. X: 3, Y: 4
235 cloning steps taken so far.
```

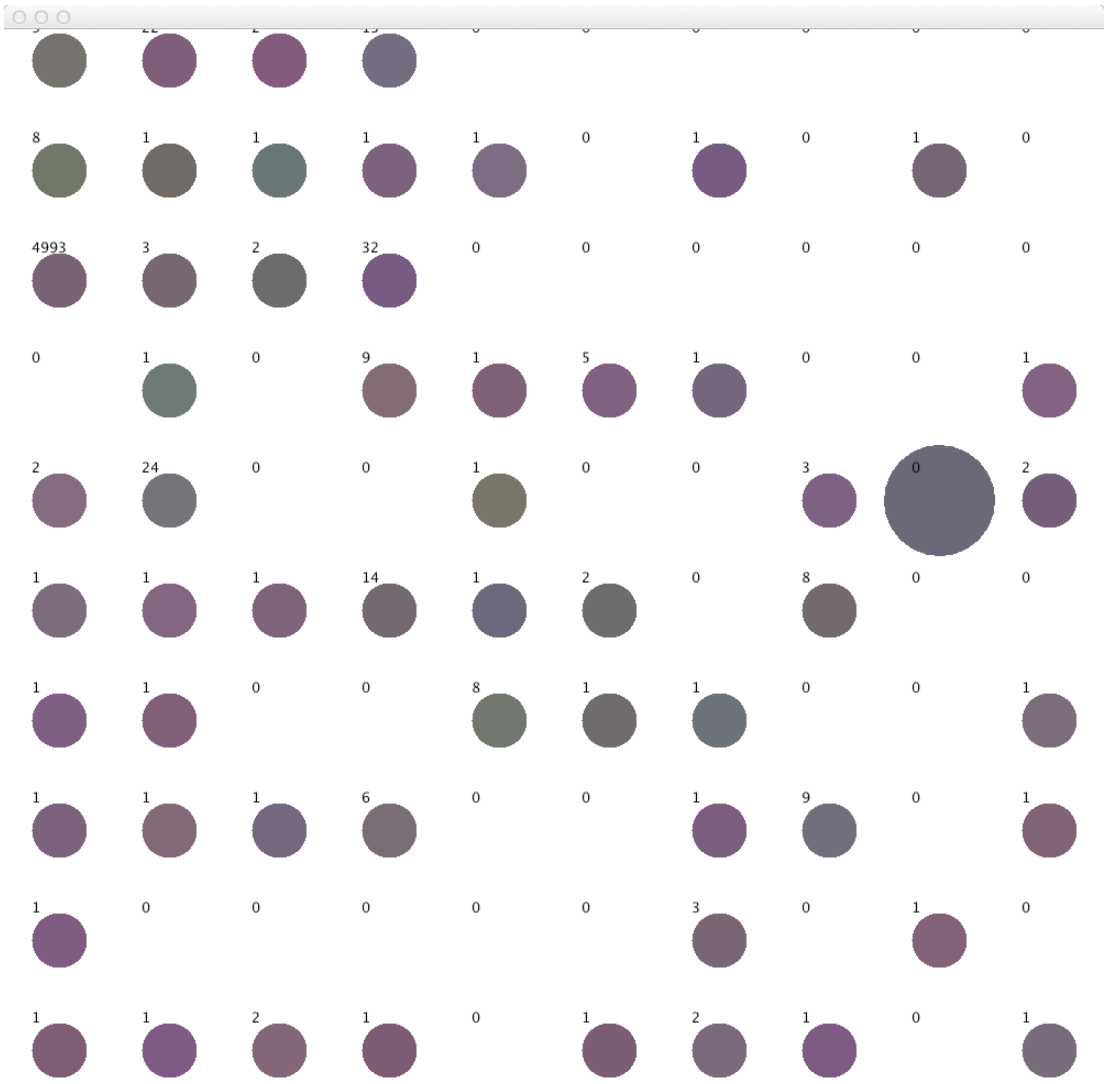
Run 2: Clusters of antibodies away from the center

```
invader removal triggered. X: 5, Y: 5
145 cloning steps taken so far.
invader removal triggered. X: 1, Y: 2
32976 cloning steps taken so far.
invader removal triggered. X: 8, Y: 4
32994 cloning steps taken so far.
invader removal triggered. X: 9, Y: 7
32999 cloning steps taken so far.
invader removal triggered. X: 3, Y: 4
33023 cloning steps taken so far.
```

Run 3:

```
invader removal triggered. X: 5, Y: 5
118 cloning steps taken so far.
invader removal triggered. X: 1, Y: 2
5216 cloning steps taken so far.
invader removal triggered. X: 8, Y: 4
5230 cloning steps taken so far.
invader removal triggered. X: 9, Y: 7
5237 cloning steps taken so far.
invader removal triggered. X: 3, Y: 4
5264 cloning steps taken so far.
```

Interestingly, in this run, the antibody labeled “4993” for 4993 cloning rounds had high enough affinity that it selectively contributed much of the response.



Notes on things I want to improve in my implementation:

After many hours of trying I was unable to find a way to reinitialize the simulation space and redistribute known effective cells throughout the antibody space. Also, the ability to expand the color range of cloning to allow for combatting very different invaders proved elusive as well. This constitutes the best I was able to do in a limited amount of time, but I believe the experimental results are useful. The remaining major problem I could not fully resolve in time was that the overlap of invaders and antibodies does not always trigger an immune response from the overlapped antibody.