

JAVA PROGRAMMING LANGUAGE - 36B

BERKELEY CITY COLLEGE

SPRING 2019

LAB C

1) Game of Life

Create a GUI application for Conway's game of life.

The game of life application should consist of a grid board of size $N \times N$ cells. A cell can either be empty (dead cell) or occupied by a creature (live cell). The surrounding cells are called the neighbors of this cell.

Each game state is called a generation. The game progresses from one generation to the next according to the following rules:

1. A creature that has two or three neighbors will continue live in the next generation.
2. A creature that has more than 3 neighbors will die of overcrowding. Its cell will be empty in the next generation.
3. A creature that has less than 2 neighbors will die of loneliness.
4. A new creature born in an empty cell that has exactly 3 neighbors.

The initial state of the grids is called the seed of the program. You can choose any initial configuration.

Write a GUI program that consists of an $N \times N$ grid board and displays the state of the grids at each generation. The program should contain the following features:

- 1) Prompt for the size of the board.
- 2) Allow the player to use the mouse to select the initial state of the grid. By clicking on a cell you can change to state of the cell. You can indicate the state of the cell using two different colors, one color of an empty cell and another for a cell occupied by a creature.
- 3) A start and stop button to start and stop the game.
- 4) Display the state of the cells at each generation. You can use a pre-determined time to update the grid between generations. (Use the `Thread.sleep()` method to pause the program for a specified time.)

Note that the changes in the board in each generation are done simultaneously on all the cells, so you may need to update the all the grid locations before you display the grid again. You may need to use two grid arrays for computing the changes

2) Battleships

Write a program for a network version of the Battleships game. The game is played between two players, each on a separate computer. On each computer there is a display which consists of two grids and a communication box. One grid is used to show the arrangement of the location of your ships and the other is used to record the shots by the opponent. You can use a square grid (for example 25 x 25) – and each cell in the grid is identified by row number and a column label (using alphabets). The communication box is used to send and receive messages between the two players. An example arrangement of the grids and the communication box is shown below:

Player

	A	B	C	D	E	Y
1										
2										
3										
4										
5										
..										
..										
..										
..										
25										

Opponent

		A	B	C	D	E	Y
1	1										
2	2										
3	3										
4	4										
5	5										
..	6										
..	7										
..	8										
..	9										
25	10										

Message Box:

Send Message:	Message received:
<i>Type message to be sent here.</i>	<i>Received message is displayed here</i>
<div>Launch</div>	

Before play begins, each player will secretly arrange the ship on the grid for that player. Each ship must occupy the specified number of squares and must be placed on consecutive squares on the grid, arranged either horizontally or vertically. The number of squares for each ship is determined by the type of the ship. The ships cannot overlap (i.e., only one ship can occupy any given square in the grid). The types and numbers of ships allowed are the same for each player. These may vary depending on the rules. You can choose the complement of ships. For example, you can use the following complement of ships.

<u>Type of ship</u>	<u>Size</u>	<u>Number</u>
aircraft carrier	5	1
battleship	4	2
submarine	3	2
destroyer	3	2
patrol boat	2	4

Rules of the game:

After the ships have been positioned, each player will take turns firing a shot to attack enemy ships.

On your turn, you will send a message with a letter and a number indicating a cell on the grid. The opponent will check that space on their lower grid, and reply "miss" if there are no ships on that cell, or "hit" if you guessed a space that contained a ship.

You will mark the shots on the opponents grid (on your computer), with white squares for misses and red squares for hits, to keep track of your guesses.

When one of your ships is hit, put a red square into that ship at the location of the hit. Whenever you any one of your ship is completely destroyed (has every slot filled with red squares), you must announce to your opponent that you have lost your ship.

Victory: The first player to sink all opposing ships wins.

Extra credit.

Write a program that simulates the opponent, so that, instead of an opponent, you can play against the remote computer.