Project - Tic-Tac-Toe game with Java & Prolog

Course/Section: _____

Last Name: Biyyapu_____

First Name: _Venkat_____

NetID (UTD email): vxb220005@utdallas.edu

Do not make any of these problems or your answers to be posted or available in Internet. Do not share with any others. All of the assignment should be done by yourself.

Submission: Submit (upload) softcopy of (1) a word document [this file with your answers and program listing and log of compile-run with test cases], and (2) a zip file of all the files (including all the programs and files that you worked and are needed to compile or run it) via elearning.

Scoresheet

TTT game 100%	#1 5%	#2 coding 60%	#3 log files 10%	#4 5%	#5 10%	#6 10%

No or Poor Documentation (penalty max 50%)	Demo to TA (penalty max 50%)
	Please check with TA for demo schedule & appointment.

** Demo is required (check for the demo schedule from TA to make a reservation) or -50%.
Your documentation here should be well-organized and presented, to follow the flow of your work with ease. Place course information and your name & email (UTD) to the header and page number in footer. Keep 1" margin each side and font-size of 10, single-spaced.

A poor documentation (a word file – this file with your answers) may result in 0 for documentation (-50%).

Note. (If your program run is not manageable) Some problems as your program run may take too long to complete (or aborted out of memory or overflow, etc.). If your program runs over 30+ minutes, or producing over a few hundreds of solutions (or the depth of search tree is too big, etc.), please stop and make a note of it in your submission (here and to TA during Demo) and/or to adjust your search to be a bit manageable. One thing that you can do is to output first few dozens (to show that your program is working or to make the length of the move or depth to be shorter, etc.), in order to make the run manageable. Make a note of this and clearly state it in your documentation for the run and solutions and to TA during the demo. Another option is to make your program smarter, to be manageable (instead of running in brute-force manner).

Warning: All of the assignment should be done by yourself and for this course. Do not make any of the problems and course-materials, or your answers to be posted, do not share or make it available in Internet.

Project - Tic-Tac-Toe game with Java & Prolog
Design and implement ttt game to play N games and to keep the score of player1 and player2.

Task#0.
#1. Create project folder (ttt-netid where netid is your netid).

> Note. Your final project submission to elearning will be: (1) this document and (2) zip file of your project folder containing all your work (including source codes, java files, terminal session log files, etc.). Use protocol("ttt-log.txt") for your prolog program to start your terminal session log/history also to be kept in the log-file provided for each task.

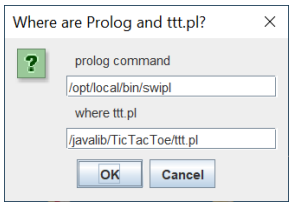
#2. Unzip the sample ttt program files in the project folder (tictactoe.java and ttt.pl).
Note that this java program (using Connect) may generate exception message (but ignored it) if your Java Run-Time Library installed is above Java v8. For this lab, first, create a folder (to unzip and place the programs here) and all the lab work is to be done here. When you run the jar file to start tic tac toe game. It will ask two paths for: (1) swipl and (2) ttt.pl. The first path for (1) swipl is found in the folder where you install swipl prolog. For my case, it is c:\swipl\bin\swipl and the second path for (2) ttt.pl would be c:\prolog\ttt-game\ttt.pl.

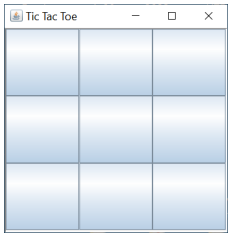
#3. Copy the java program (ttt1.java and its jar file: ttt1.jar and Connect1.java). Update the programs to do the following tasks.

Tasks

#1. Initialize the path names for (a) swipl and (b) ttt.pl so that you do not have to update them in the beginning of the program run. For example,

```
String prolog = "C:/swipl/bin/swipl";
String ttt = "C:/ttt-rkm010300/ttt.pl";
```





#2. Design and implement ttt-board to add and do the following items.
(1) to display your netid (next to Tic Tac Toe) on the top
(2) to add three buttons:
(a) to start (a new ttt game – one game),
(b) to start N games (to complete N games for computer x computer)
(c) to stop/reset (to end a game).
(3) to display who is the player for player1 (X) human and player2 (O) computer
(4) to have an option to select player1 (X) to be human or computer.
You may use the same prolog code to run for player1 in case of computer.
(5) at the end of each game, display who is the winner
(6) to get an integer N (1-10) from user to play N times.
In addition, display the game # (1-N), and the number of win for player1 and for player2.
(7) at the end of N games, it displays who is the overall winner (player1 or player2).
and to display the total winning score of each player

#3. Your java and prolog programs should provide the log of each step into its own log file
(ttt-java-log.txt and ttt-prolog-log.txt) keeping the records to show the current state of the ttt game:
(a) in the beginning – game starts, game 1 of total N=10 games.

(b) at the end of each game – game ends, for game 1 of total N=10 games,
(c) who is winner – player1 (X) wins
(d) total count of wins for each player
(e) for each step (of X or O), for example, display Player1 X takes the cell (1, 2), etc.

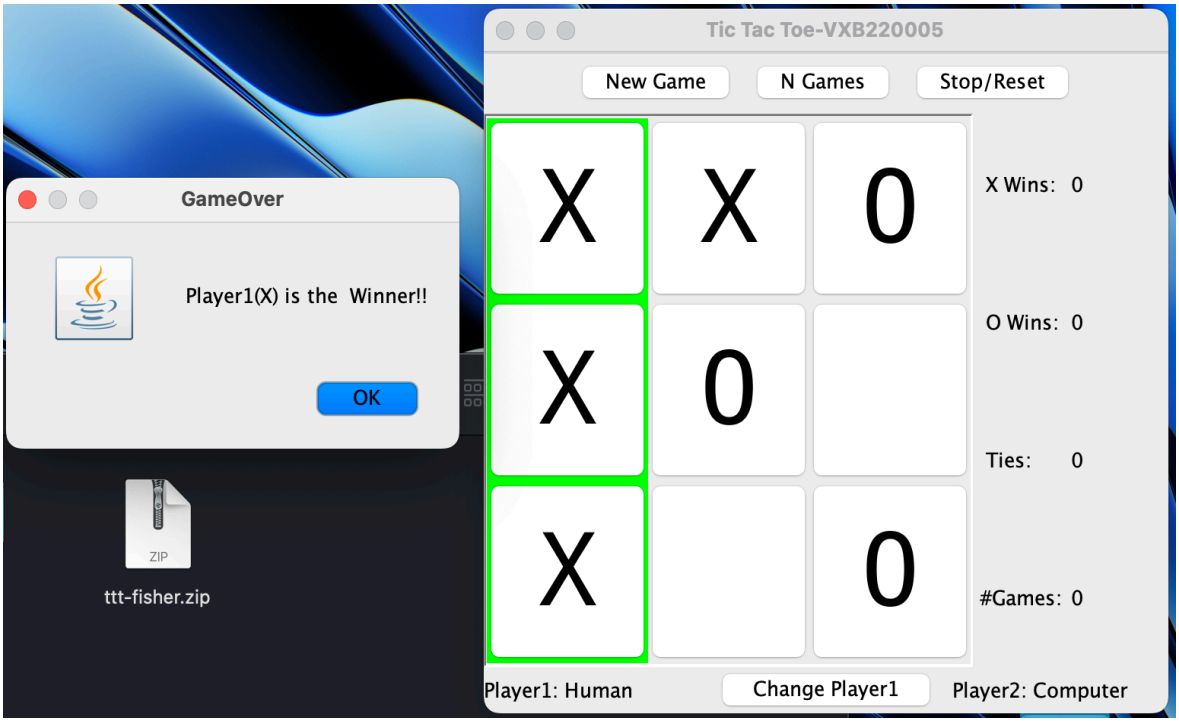
#4. Compile your java program and have a jar file (ttt1.jar) and run of the program.

#5. Your test cases are:
Test case 1. Run human x computer – 1 time
Test case 2. Run human x computer – 3 times
Test case 3. Run computer x computer – 1 time
Test case 4. Run computer x computer – 100 times.

#6. Submit to elearnng:
(1) your word document (this document) with what is done, step by step, each step with subheading, including the listing of the programs of java and prolog, and compile and run of the program) – see below,
and
(2) the zip file folder (TicTacToe2.java and TicTacToe2.jar, all programs, terminal session log/history files of your run).

(3) do the demo of your program & run to TA during office hours. Please wait for TA announcement on demo schedule and guideline.

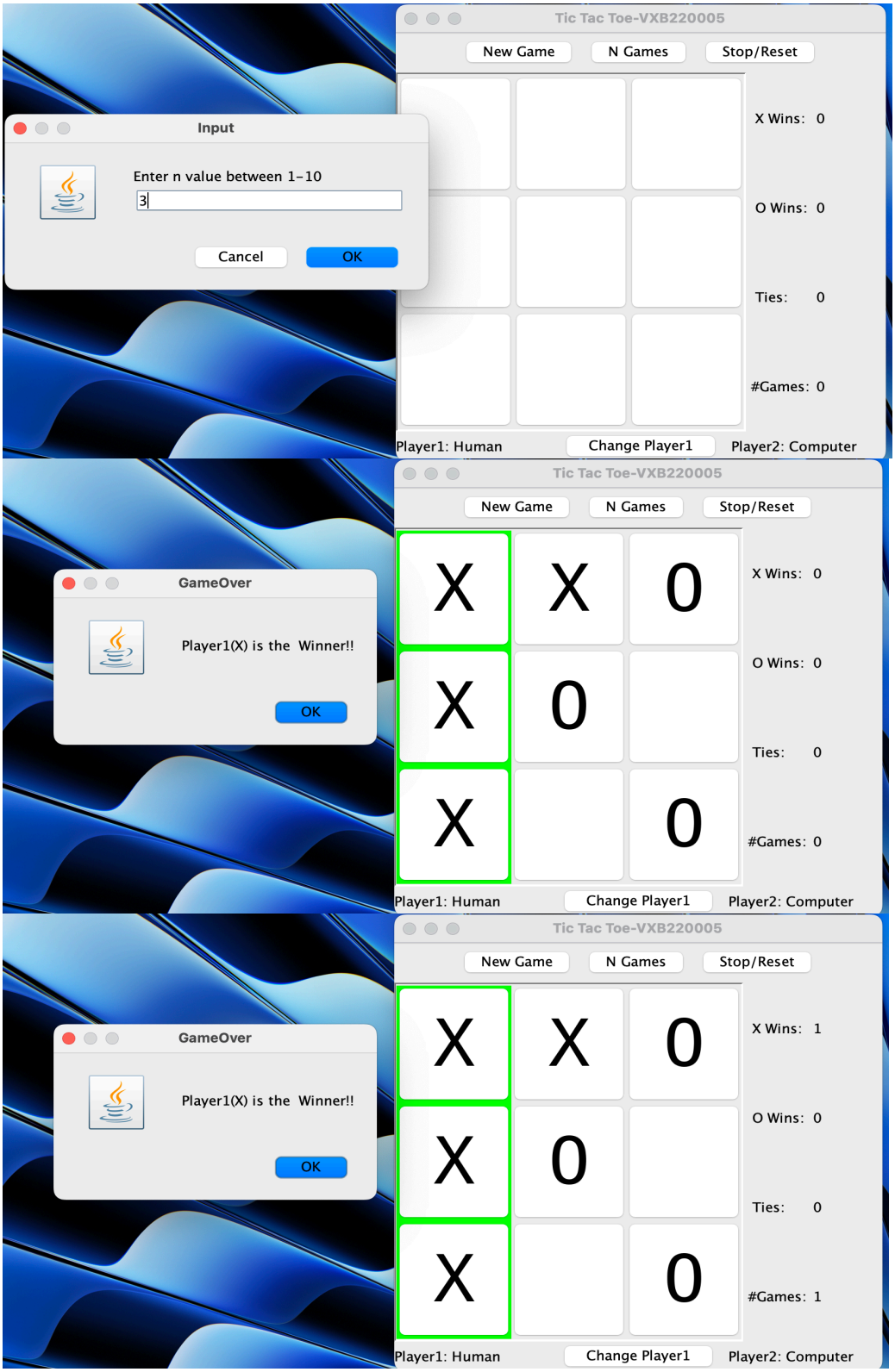
Test case 1. Run human x computer – 1 time

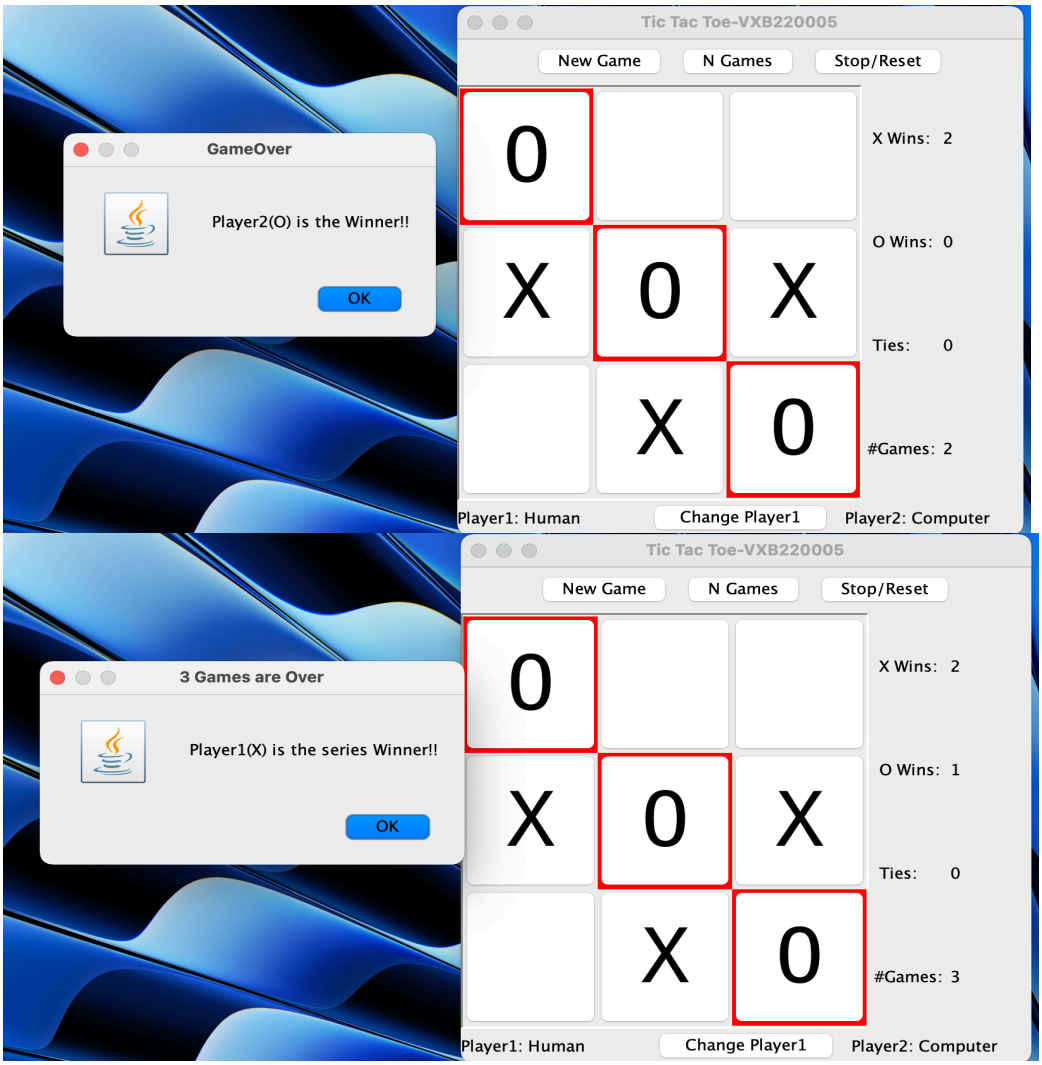


Logs:

```
Nov 03, 2023 10:39:25 PM ttt1 startNewGame
INFO: Game 01 Started
Nov 03, 2023 10:39:26 PM ttt1 actionPerformed
INFO: X's move (2,1).
Nov 03, 2023 10:39:26 PM ttt1 computer_move
INFO: O's move(2,2).
Nov 03, 2023 10:39:26 PM ttt1 actionPerformed
INFO: X's move (1,2).
Nov 03, 2023 10:39:26 PM ttt1 computer_move
INFO: O's move(3,3).
Nov 03, 2023 10:39:27 PM ttt1 actionPerformed
INFO: X's move (1,1).
Nov 03, 2023 10:39:27 PM ttt1 computer_move
INFO: O's move(3,1).
Nov 03, 2023 10:39:27 PM ttt1 actionPerformed
INFO: X's move (1,3).
Nov 03, 2023 10:39:29 PM ttt1 stopResetGame
INFO: Player1(X) wins!!
Nov 03, 2023 10:39:29 PM ttt1 stopResetGame
INFO: Game 1 Ended
```

Test case 2. Run human x computer – 3 times





Logs:

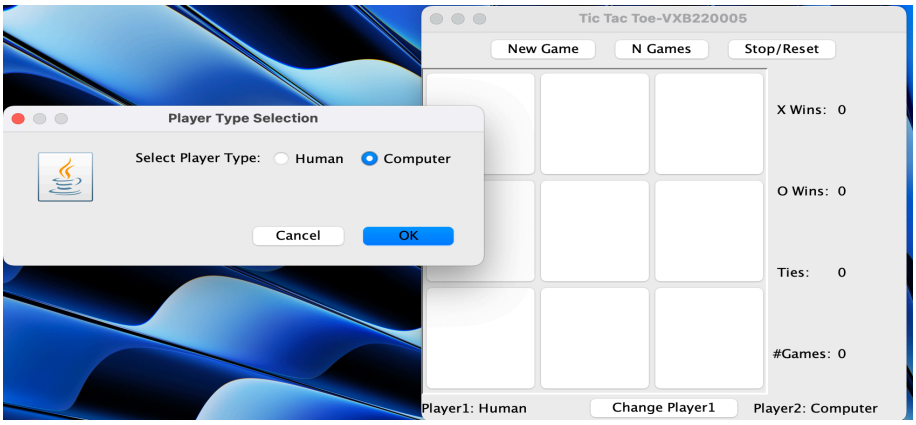
```
Nov 03, 2023 10:46:08 PM ttt1 startNewGame
INFO: Game 01 Started
Nov 03, 2023 10:46:09 PM ttt1 actionPerformed
INFO: X's move (2,1).
Nov 03, 2023 10:46:09 PM ttt1 computer_move
INFO: O's move(2,2).
Nov 03, 2023 10:46:09 PM ttt1 actionPerformed
INFO: X's move (1,2).
Nov 03, 2023 10:46:09 PM ttt1 computer_move
INFO: O's move(3,3).
Nov 03, 2023 10:46:10 PM ttt1 actionPerformed
INFO: X's move (1,1).
Nov 03, 2023 10:46:10 PM ttt1 computer_move
INFO: O's move(3,1).
Nov 03, 2023 10:46:10 PM ttt1 actionPerformed
INFO: X's move (1,3).
Nov 03, 2023 10:46:11 PM ttt1 stopResetGame
INFO: Player1(X) wins!!
```

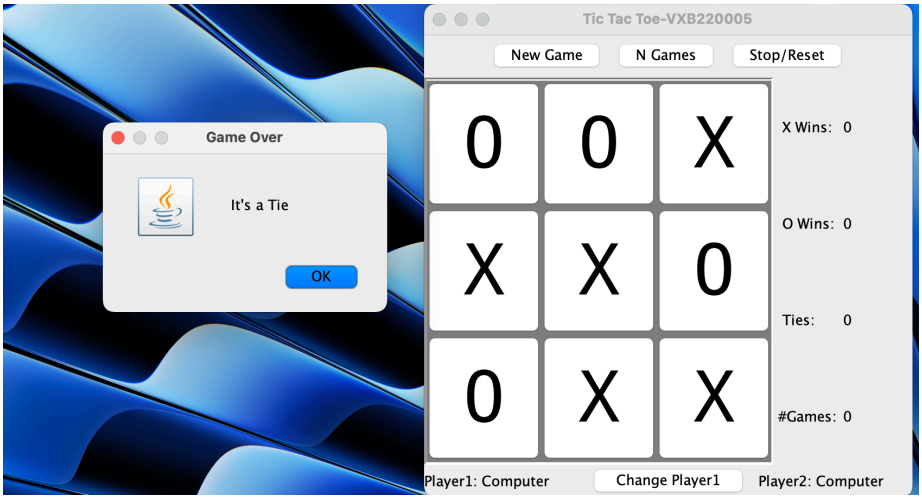
```
Nov 03, 2023 10:46:11 PM ttt1 stopResetGame
INFO: Game 1 Ended
Nov 03, 2023 10:46:11 PM ttt1 startNewGame
INFO: Game 11 Started
Nov 03, 2023 10:46:12 PM ttt1 actionPerformed
INFO: X's move (2,1).
Nov 03, 2023 10:46:12 PM ttt1 computer_move
INFO: O's move(2,2).
Nov 03, 2023 10:46:12 PM ttt1 actionPerformed
INFO: X's move (1,2).
Nov 03, 2023 10:46:12 PM ttt1 computer_move
INFO: O's move(3,3).
Nov 03, 2023 10:46:13 PM ttt1 actionPerformed
INFO: X's move (1,1).
Nov 03, 2023 10:46:13 PM ttt1 computer_move
INFO: O's move(3,1).
Nov 03, 2023 10:46:13 PM ttt1 actionPerformed
INFO: X's move (1,3).
Nov 03, 2023 10:46:14 PM ttt1 stopResetGame
INFO: Player1(X) wins!!
Nov 03, 2023 10:46:14 PM ttt1 stopResetGame
INFO: Game 2 Ended
Nov 03, 2023 10:46:14 PM ttt1 startNewGame
INFO: Game 21 Started
Nov 03, 2023 10:46:16 PM ttt1 actionPerformed
INFO: X's move (1,2).
Nov 03, 2023 10:46:16 PM ttt1 computer_move
INFO: O's move(2,2).
Nov 03, 2023 10:46:21 PM ttt1 actionPerformed
INFO: X's move (2,3).
Nov 03, 2023 10:46:21 PM ttt1 computer_move
INFO: O's move(3,1).
Nov 03, 2023 10:46:23 PM ttt1 actionPerformed
INFO: X's move (3,2).
Nov 03, 2023 10:46:23 PM ttt1 computer_move
INFO: O's move(1,3).
Nov 03, 2023 10:46:26 PM ttt1 stopResetGame
INFO: Player2(O) wins!!
Nov 03, 2023 10:46:26 PM ttt1 stopResetGame
INFO: Game 3 Ended
Nov 03, 2023 10:46:27 PM ttt1$nGamesListener$1 done
INFO: Total X-Wins : 2
Nov 03, 2023 10:46:27 PM ttt1$nGamesListener$1 done
INFO: Total O-Wins : 1
Nov 03, 2023 10:46:27 PM ttt1$nGamesListener$1 done
INFO: Total Ties : 0
Nov 03, 2023 10:46:27 PM ttt1$nGamesListener$1 done
INFO: Overall Winner is Player1(X)
Nov 03, 2023 10:46:30 PM ttt1 startNewGame
INFO: Game 01 Started
Nov 03, 2023 10:46:30 PM ttt1 actionPerformed
```

```
INFO: X's move (2,1).
Nov 03, 2023 10:46:30 PM ttt1 computer_move
INFO: O's move(2,2).
Nov 03, 2023 10:46:31 PM ttt1 actionPerformed
INFO: X's move (1,2).
Nov 03, 2023 10:46:31 PM ttt1 computer_move
INFO: O's move(3,3).
Nov 03, 2023 10:46:31 PM ttt1 actionPerformed
INFO: X's move (1,1).
Nov 03, 2023 10:46:31 PM ttt1 computer_move
INFO: O's move(3,1).
Nov 03, 2023 10:46:31 PM ttt1 actionPerformed
INFO: X's move (1,3).
Nov 03, 2023 10:46:32 PM ttt1 stopResetGame
INFO: Player1(X) wins!!
Nov 03, 2023 10:46:32 PM ttt1 stopResetGame
INFO: Game 1 Ended
Nov 03, 2023 10:46:32 PM ttt1 startNewGame
INFO: Game 11 Started
Nov 03, 2023 10:46:34 PM ttt1 actionPerformed
INFO: X's move (2,1).
Nov 03, 2023 10:46:34 PM ttt1 computer_move
INFO: O's move(2,2).
Nov 03, 2023 10:46:34 PM ttt1 actionPerformed
INFO: X's move (1,2).
Nov 03, 2023 10:46:34 PM ttt1 computer_move
INFO: O's move(3,3).
Nov 03, 2023 10:46:35 PM ttt1 actionPerformed
INFO: X's move (1,1).
Nov 03, 2023 10:46:35 PM ttt1 computer_move
INFO: O's move(3,1).
Nov 03, 2023 10:46:35 PM ttt1 actionPerformed
INFO: X's move (1,3).
Nov 03, 2023 10:46:36 PM ttt1 stopResetGame
INFO: Player1(X) wins!!
Nov 03, 2023 10:46:36 PM ttt1 stopResetGame
INFO: Game 2 Ended
Nov 03, 2023 10:46:36 PM ttt1 startNewGame
INFO: Game 21 Started
Nov 03, 2023 10:46:37 PM ttt1 actionPerformed
INFO: X's move (1,2).
Nov 03, 2023 10:46:37 PM ttt1 computer_move
INFO: O's move(2,2).
Nov 03, 2023 10:46:37 PM ttt1 actionPerformed
INFO: X's move (3,2).
Nov 03, 2023 10:46:37 PM ttt1 computer_move
INFO: O's move(1,1).
Nov 03, 2023 10:46:38 PM ttt1 actionPerformed
INFO: X's move (2,3).
Nov 03, 2023 10:46:38 PM ttt1 computer_move
INFO: O's move(3,3).
```

```
Nov 03, 2023 10:46:40 PM ttt1 stopResetGame
INFO: Player2(O) wins!!
Nov 03, 2023 10:46:40 PM ttt1 stopResetGame
INFO: Game 3 Ended
Nov 03, 2023 10:46:41 PM ttt1$nGamesListener$1 done
INFO: Total X-Wins : 2
Nov 03, 2023 10:46:41 PM ttt1$nGamesListener$1 done
INFO: Total O-Wins : 1
Nov 03, 2023 10:46:41 PM ttt1$nGamesListener$1 done
INFO: Total Ties : 0
Nov 03, 2023 10:46:41 PM ttt1$nGamesListener$1 done
INFO: Overall Winner is Player1(X)
```

Test case 3. Run computer x computer – 1 time





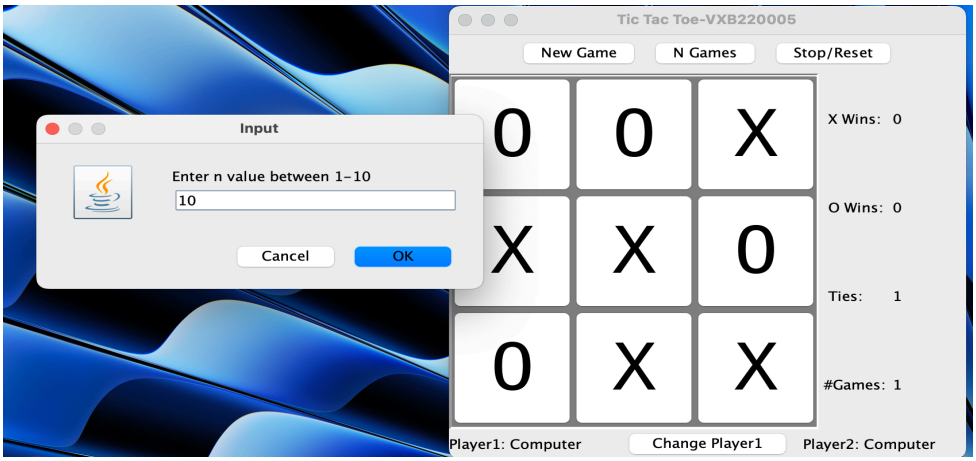
Logs:

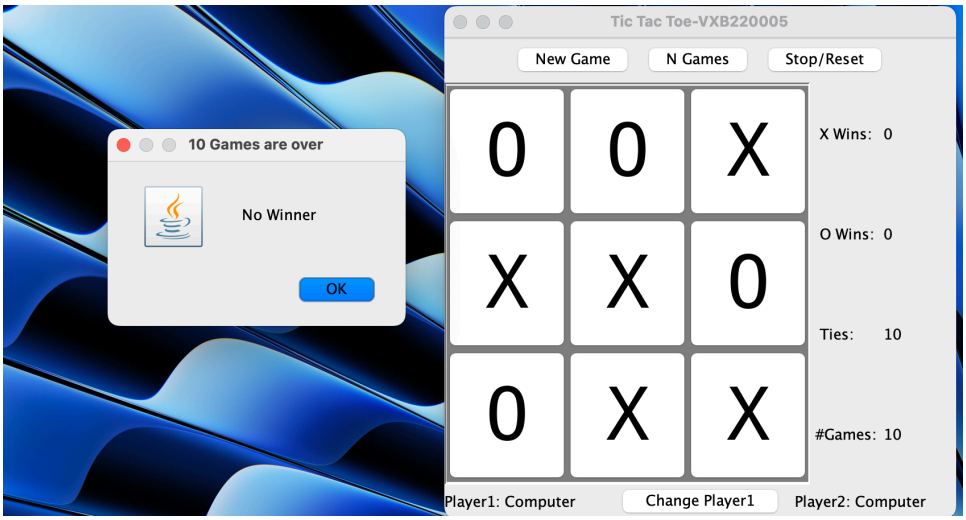
```
Nov 03, 2023 11:02:28 PM ttt1 startNewGame
INFO: Game 01 Started
Nov 03, 2023 11:02:28 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:02:28 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:02:28 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:02:29 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:02:29 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:02:29 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:02:30 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:02:30 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:02:30 PM ttt1 computer_move
```

```
INFO: X placed at (3,2).
Nov 03, 2023 11:02:31 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:02:31 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:02:31 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:02:32 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:02:41 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:02:41 PM ttt1 stopResetGame
INFO: Game 1 Ended
```

Test case 4. Run computer x computer – 10 times.

(You may skip most of the output except half page for the beginning and half page for the end of the program run. Your zip file should contain the full log file of this run for TA to check as needed).





Logs:

```
Nov 03, 2023 11:20:25 PM ttt1 startNewGame
INFO: Game 01 Started
Nov 03, 2023 11:20:25 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:20:26 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:20:26 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:20:26 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:20:26 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:20:26 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:20:26 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:20:26 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:20:26 PM ttt1 computer_move
```

```
INFO: X placed at (3,2).
Nov 03, 2023 11:20:27 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:20:27 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:20:27 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:20:27 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:20:29 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:20:29 PM ttt1 stopResetGame
INFO: Game 1 Ended
Nov 03, 2023 11:20:29 PM ttt1 startNewGame
INFO: Game 11 Started
Nov 03, 2023 11:20:29 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:20:29 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:20:29 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:20:29 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:20:29 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:20:29 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:20:30 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:20:30 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:20:30 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:20:30 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:20:30 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:20:30 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:20:31 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:20:31 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:20:31 PM ttt1 stopResetGame
INFO: Game 2 Ended
Nov 03, 2023 11:20:31 PM ttt1 startNewGame
INFO: Game 21 Started
Nov 03, 2023 11:20:31 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:20:32 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:20:32 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:20:32 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:20:32 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:20:32 PM ttt1 computer_move
INFO: X placed at (1,3).
```

```
Nov 03, 2023 11:20:32 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:20:32 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:20:32 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:20:33 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:20:33 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:20:33 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:20:33 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:20:34 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:20:34 PM ttt1 stopResetGame
INFO: Game 3 Ended
Nov 03, 2023 11:20:34 PM ttt1 startNewGame
INFO: Game 31 Started
Nov 03, 2023 11:20:34 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:20:34 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:20:34 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:20:34 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:20:34 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:20:34 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:20:35 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:20:35 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:20:35 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:20:35 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:20:35 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:20:35 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:20:35 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:20:36 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:20:36 PM ttt1 stopResetGame
INFO: Game 4 Ended
Nov 03, 2023 11:20:36 PM ttt1 startNewGame
INFO: Game 41 Started
Nov 03, 2023 11:20:36 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:20:37 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:20:37 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:20:37 PM ttt1 computer_x_move
```

```
INFO: X placed at (3,1).
Nov 03, 2023 11:20:37 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:20:37 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:20:37 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:20:37 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:20:37 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:20:38 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:20:38 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:20:38 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:20:38 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:20:39 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:20:39 PM ttt1 stopResetGame
INFO: Game 5 Ended
Nov 03, 2023 11:20:39 PM ttt1 startNewGame
INFO: Game 51 Started
Nov 03, 2023 11:20:39 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:20:39 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:20:39 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:20:40 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:20:40 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:20:40 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:20:55 PM ttt1 startNewGame
INFO: Game 51 Started
Nov 03, 2023 11:20:55 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:20:55 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:20:55 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:20:56 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:20:56 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:20:56 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:20:56 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:20:56 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:20:56 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:20:56 PM ttt1 computer_x_move
INFO: X placed at (2,3).
```

```
Nov 03, 2023 11:20:57 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:20:57 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:20:57 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:20:58 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:20:58 PM ttt1 stopResetGame
INFO: Game 6 Ended
Nov 03, 2023 11:20:58 PM ttt1 startNewGame
INFO: Game 61 Started
Nov 03, 2023 11:20:58 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:20:58 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:20:58 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:20:59 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:20:59 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:20:59 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:21:00 PM ttt1 startNewGame
INFO: Game 61 Started
Nov 03, 2023 11:21:01 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:21:01 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:21:01 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:21:01 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:21:01 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:21:01 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:21:01 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:21:02 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:21:02 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:21:02 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:21:02 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:21:02 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:21:02 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:21:04 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:21:04 PM ttt1 stopResetGame
INFO: Game 7 Ended
Nov 03, 2023 11:21:04 PM ttt1 startNewGame
INFO: Game 71 Started
Nov 03, 2023 11:21:04 PM ttt1 computer_x_move
```

```
INFO: X placed at (2,2).
Nov 03, 2023 11:21:04 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:21:04 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:21:04 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:21:05 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:21:05 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:21:05 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:21:05 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:21:05 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:21:05 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:21:05 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:21:05 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:21:06 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:21:07 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:21:07 PM ttt1 stopResetGame
INFO: Game 8 Ended
Nov 03, 2023 11:21:07 PM ttt1 startNewGame
INFO: Game 81 Started
Nov 03, 2023 11:21:07 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:21:07 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:21:07 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:21:08 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:21:08 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:21:08 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:21:08 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:21:08 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:21:08 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:21:08 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:21:09 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:21:09 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:21:09 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:21:10 PM ttt1 stopResetGame
INFO: It's a Tie!!
```

```
Nov 03, 2023 11:21:10 PM ttt1 stopResetGame
INFO: Game 9 Ended
Nov 03, 2023 11:21:10 PM ttt1 startNewGame
INFO: Game 91 Started
Nov 03, 2023 11:21:10 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:21:10 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:21:10 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:21:11 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:21:11 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:21:11 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:21:12 PM ttt1 startNewGame
INFO: Game 91 Started
Nov 03, 2023 11:21:12 PM ttt1 computer_x_move
INFO: X placed at (2,2).
Nov 03, 2023 11:21:13 PM ttt1 computer_move
INFO: O placed at 1,1
Nov 03, 2023 11:21:13 PM ttt1 computer_move
INFO: X placed at (1,1).
Nov 03, 2023 11:21:13 PM ttt1 computer_x_move
INFO: X placed at (3,1).
Nov 03, 2023 11:21:13 PM ttt1 computer_move
INFO: O placed at 1,3
Nov 03, 2023 11:21:13 PM ttt1 computer_move
INFO: X placed at (1,3).
Nov 03, 2023 11:21:13 PM ttt1 computer_x_move
INFO: X placed at (1,2).
Nov 03, 2023 11:21:13 PM ttt1 computer_move
INFO: O placed at 3,2
Nov 03, 2023 11:21:13 PM ttt1 computer_move
INFO: X placed at (3,2).
Nov 03, 2023 11:21:14 PM ttt1 computer_x_move
INFO: X placed at (2,3).
Nov 03, 2023 11:21:14 PM ttt1 computer_move
INFO: O placed at 2,1
Nov 03, 2023 11:21:14 PM ttt1 computer_move
INFO: X placed at (2,1).
Nov 03, 2023 11:21:14 PM ttt1 computer_x_move
INFO: X placed at (3,3).
Nov 03, 2023 11:21:16 PM ttt1 stopResetGame
INFO: It's a Tie!!
Nov 03, 2023 11:21:16 PM ttt1 stopResetGame
INFO: Game 10 Ended
Nov 03, 2023 11:21:51 PM ttt1$nGamesListener$1 done
INFO: Total X-Wins : 0
Nov 03, 2023 11:21:51 PM ttt1$nGamesListener$1 done
INFO: Total O-Wins : 0
Nov 03, 2023 11:21:51 PM ttt1$nGamesListener$1 done
INFO: Total Ties : 10
Nov 03, 2023 11:21:51 PM ttt1$nGamesListener$1 done
INFO: Overall Winner is No One
```

Listing of ttt1.java program

```
import java.awt.* ;
import java.awt.event.* ;
import java.io.* ;
import java.net.* ;
import java.util.concurrent.atomic.AtomicInteger;
import javax.swing.* ;
import javax.swing.border.* ;
import java.util.logging.FileHandler;
import java.util.logging.Logger;
import java.util.logging.SimpleFormatter;

public class ttt1 extends JFrame implements ActionListener {
  JButton b11,b21,b31,
          b12,b22,b32,
          b13,b23,b33,
          newGame, nGames, stopReset,changePlayer;
  JRadioButton humanRadioButton,computerRadioButton;
  FileHandler fh;
  String winner="";
  Boolean game_progress= false;
  boolean myturn ;
  BufferedReader br,br2 ;
  BufferedWriter bw,bw2 ;
  Thread connection,connection2 ;
  Process prologProcess,prologProcess2 ;
  String prolog ;
  String ttt,ttt_x;
  String player1="Human";
  JLabel p1Label, p2Label,xWinCounter,oWinCounter,tieCounter,totGamesCounter;

  Logger logger;
  AtomicInteger gameCount;

  Integer xWins=0,oWins=0,ties=0,totGamesPlayed=0;

  /**
   *  Create a tic tac toe game,
   *  prolog is the prolog command (e.g. "/opt/local/bin/swipl").
   *  ttt is the locator for ttt.pl (e.g. "/javalib/TicTacToe/ttt.pl").
   */
  public ttt1(String prolog, String ttt) {
    this.prolog = prolog ;
    this.ttt = ttt ;
    this.ttt_x =ttt.replace(".pl","_x.pl");
    logger = Logger.getLogger(ttt1.class.getName());

    try {
      fh = new FileHandler("ttt-java-log.txt");
      fh.setFormatter(new SimpleFormatter());
      logger.addHandler(fh);
```

```
} catch (Exception e) {
  e.printStackTrace();
}
b11 = new JButton("") ;
b21 = new JButton("") ;
b31 = new JButton("") ;
b12 = new JButton("") ;
b22 = new JButton("") ;
b32 = new JButton("") ;
b13 = new JButton("") ;
b23 = new JButton("") ;
b33 = new JButton("") ;
b11.setActionCommand("(1,1).") ;
b21.setActionCommand("(2,1).") ;
b31.setActionCommand("(3,1).") ;
b12.setActionCommand("(1,2).") ;
b22.setActionCommand("(2,2).") ;
b32.setActionCommand("(3,2).") ;
b13.setActionCommand("(1,3).") ;
b23.setActionCommand("(2,3).") ;
b33.setActionCommand("(3,3).") ;
Font f = new Font("monospaced",Font.PLAIN,64) ;
JButton[] buttons = { b11, b21, b31, b12, b22, b32, b13, b23, b33 };
for (JButton button : buttons) {
  button.setFont(f);
  button.addActionListener(this);
}

newGame = new JButton("New Game");
nGames=  new JButton("N Games");
stopReset= new JButton("Stop/Reset");
newGame.addActionListener(new ActionListener() {
  @Override
  public void actionPerformed(ActionEvent e) {
    startNewGame();
  }

});
nGames.addActionListener(new nGamesListener());

stopReset.addActionListener(new ActionListener() {
  @Override
  public void actionPerformed(ActionEvent e) {
    stopResetGame();
  }
});

JPanel subPanel = new JPanel();
subPanel.add(newGame);
subPanel.add(nGames);
subPanel.add(stopReset);
this.getContentPane().add(subPanel,BorderLayout.PAGE_START);
JPanel panel = new JPanel() ;
panel.setLayout(new GridLayout(3,3)) ;
```

```
    for (JButton button : buttons) {
      panel.add(button);
    }

    this.setTitle("Tic Tac Toe-VXB220005") ;
    Border panelborder = BorderFactory.createLoweredBevelBorder() ;
    panel.setBorder(panelborder) ;
    this.getContentPane().add(panel) ;

    JPanel gameRecordDisplay =  new JPanel();
    gameRecordDisplay.setLayout(new GridLayout(4, 1));

    JLabel xWinLabel = new JLabel("  X Wins:  ");
    JLabel oWinLabel = new JLabel("  O Wins:  ");
    JLabel tieLabel = new JLabel("  Ties:   ");
    JLabel totGamesLabel = new JLabel(" #Games: ");

    xWinCounter = new JLabel(Integer.toString(xWins));
    oWinCounter= new JLabel(Integer.toString(oWins));
    tieCounter = new JLabel(Integer.toString(ties));
    totGamesCounter = new JLabel(Integer.toString(totGamesPlayed));

    gameRecordDisplay.add(xWinLabel);
    gameRecordDisplay.add(xWinCounter);

    gameRecordDisplay.add(oWinLabel);
    gameRecordDisplay.add(oWinCounter);

    gameRecordDisplay.add(tieLabel);
    gameRecordDisplay.add(tieCounter);

    gameRecordDisplay.add(totGamesLabel);
    gameRecordDisplay.add(totGamesCounter);

    this.getContentPane().add(gameRecordDisplay,BorderLayout.EAST);

    changePlayer = new JButton("Change Player1");
    changePlayer.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        humanRadioButton = new JRadioButton("Human");
        computerRadioButton = new JRadioButton("Computer");
        ButtonGroup group = new ButtonGroup();
        group.add(humanRadioButton);
        group.add(computerRadioButton);
        JPanel panel = new JPanel();
        panel.add(new JLabel("Select Player Type:"));
        panel.add(humanRadioButton);
        panel.add(computerRadioButton);
        int input = JOptionPane.showConfirmDialog(null, panel, "Player Type Selection",
JOptionPane.OK_CANCEL_OPTION);
        if (input == JOptionPane.OK_OPTION) {
          if (humanRadioButton.isSelected()) {
            player1 = "Human";
```

```
            } else if (computerRadioButton.isSelected()) {
              player1 = "Computer";
            }
            p1Label.setText("Player1: " + player1);
          }
        }
      });

      JPanel playerDisplay =  new JPanel();
      playerDisplay.setLayout(new GridLayout(1, 3));

      p1Label = new JLabel("Player1: "+player1);
      p2Label = new JLabel("\t\tPlayer2: Computer");

      playerDisplay.add(p1Label);
      playerDisplay.add(changePlayer);
      playerDisplay.add(p2Label);
      playerDisplay.setSize(300,500);
      this.getContentPane().add(playerDisplay,BorderLayout.PAGE_END);
      this.setSize(450,450);
      this.setLocation(900,300) ;
//      this.myturn = true ;

      Connector1 connector = new Connector1(54321) ;
      connector.start() ;

      Socket sock ;
      try {
        sock = new Socket("127.0.0.1",54321) ;
        br = new BufferedReader(new InputStreamReader(sock.getInputStream())) ;
        bw = new BufferedWriter(new OutputStreamWriter(sock.getOutputStream())) ;
      } catch(Exception x) { x.printStackTrace() ; }

      connection = new Thread() {
        public void run() {
          while(true) {
            try{
              String s = br.readLine() ;
              computer_move(s) ;
            } catch(Exception xx) { xx.printStackTrace() ; }
          }
        }
      } ;
      connection.start() ;

      Connector1 connector2 = new Connector1(54322) ;
      connector2.start() ;

      Socket sock2 ;
      try {
        sock2 = new Socket("127.0.0.1",54322) ;
        br2 = new BufferedReader(new InputStreamReader(sock2.getInputStream())) ;
        bw2 = new BufferedWriter(new OutputStreamWriter(sock2.getOutputStream())) ;
      } catch(Exception x) { x.printStackTrace() ; }
```

```
    System.out.println(br2.toString());

    connection2 = new Thread() {
      public void run() {
        while(true) {
          try{
            String s = br2.readLine() ;
            System.out.println(s);
            computer_x_move(s) ;
          } catch(Exception xx) { xx.printStackTrace() ; }
        }
      }
    } ;
    connection2.start() ;

    Thread shows = new Thread() {
      public void run() {
        setVisible(true) ;
      }
    } ;
    EventQueue.invokeLater(shows);

//      Start the prolog player

//    try {
//      prologProcess =
//        Runtime.getRuntime().exec(prolog + " -f " + ttt) ;
//    } catch(Exception xx) {System.out.println(xx) ; }
//    game_progress = true;

    // On closing, kill the prolog process first and then exit
    this.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent w) {
        destroy_prolog();
        System.exit(0) ;
      }
    }) ;

  }

  public static void main(String[] args) {
    String ttt = "/Users/venkatbiyyapu/Desktop/AI/Projects/ttt-fisher/ttt.pl" ;
    String prolog="/opt/homebrew/bin/swipl";
    boolean noargs = true ;
    try {
      prolog = args[0] ;
      ttt = args[1] ;
      noargs = false ;
    }
    catch (Exception xx) {
      System.out.println("usage: java TicTactoe  <where prolog>  <where ttt>") ;
    }
    if (noargs) {
```

```
        Object[] message = new Object[4] ;
        message[0] = new Label("  prolog command") ;
        message[1] = new JTextField(prolog) ;
        message[2] = new Label("  where ttt.pl ") ;
        message[3] = new JTextField(ttt) ;
        try {
          int I = JOptionPane.showConfirmDialog(null,message,"Where are Prolog and ttt.pl?
",JOptionPane.OK_CANCEL_OPTION) ;
          if (I == 2 | I == 1) System.exit(0) ;
          System.out.println(I) ;
          new ttt1(((JTextField)message[1]).getText().trim(),((JTextField)message[3]).getText().trim()) ;
        } catch(Exception yy) {}
      }
      else
        new ttt1(prolog,ttt) ;
    }
    synchronized void destroy_prolog(){
      if (prologProcess != null) prologProcess.destroy() ;
      if (prologProcess2 != null) prologProcess2.destroy() ;
      game_progress = false;
    }
    private void clearGrid() {
      JButton[] buttons = { b11, b21, b31, b12, b22, b32, b13, b23, b33 };
      for (JButton button : buttons) {
        button.setText("");
        button.setBackground(null);
        button.setOpaque(true);
        button.setBorderPainted(true);
      }
    }
    class nGamesListener implements ActionListener {
      @Override
      public void actionPerformed(ActionEvent e) {
        int input = Integer.parseInt(JOptionPane.showInputDialog("Enter n value between 1-10"));
        resetNGames();
        if (input>=1 && input<=10) {
          SwingWorker<Void, Void> worker=new SwingWorker() {
            @Override
            protected Void doInBackground() throws Exception {
              gameCount = new AtomicInteger(input);
              while (gameCount.get() > 0) {
                synchronized (nGamesListener.class) {
                  if (!game_progress) {
                    startNewGame();
                    waitUntilGameFinished();
                    gameCount.decrementAndGet();
                  }
                }
              }
              return null;
            }

            @Override
            protected void done() {
```

```
            String overAllWinner="";
            if(xWins>oWins) {
              overAllWinner="Player1(X)";
              JOptionPane.showMessageDialog(null, "Player1(X) is the series Winner!!", input + " Games are Over",
JOptionPane.INFORMATION_MESSAGE);
            }
            else if(xWins<oWins) {
              overAllWinner="Player2(O)";
              JOptionPane.showMessageDialog(null, "Player2(0) is the series Winner!!", input + " Games are Over",
JOptionPane.INFORMATION_MESSAGE);
            }
            else {
              overAllWinner="No One";
              JOptionPane.showMessageDialog(null, "No Winner", input + " Games are over",
JOptionPane.INFORMATION_MESSAGE);
            }

            logger.info("Total X-Wins : " + xWins);
            logger.info("Total O-Wins : " + oWins);
            logger.info("Total Ties : " + ties);
            logger.info("Overall Winner is " + overAllWinner);
          }
        };

        worker.execute();
      } else {
        JOptionPane.showMessageDialog(null, "Invalid input. Please enter a valid number.", "Error",
JOptionPane.ERROR_MESSAGE);

      }
    }

    private void waitUntilGameFinished() {
      synchronized (nGamesListener.class) {
        while (game_progress) {
          try {
            nGamesListener.class.wait();
            System.out.println("Waiting is done");
          } catch (InterruptedException ex) {
            ex.printStackTrace();
          }
        }
      }
    }
  }

  synchronized private void startNewGame(){
    logger.info("Game "+totGamesPlayed+1+" Started");
    clearGrid();
    if (game_progress)
      destroy_prolog();
    game_progress = true;
    myturn=true;
    try {
```

```
        //br.readLine();
        prologProcess =
              Runtime.getRuntime().exec(prolog + " -f " + ttt);
        Thread.sleep(100);
      } catch (Exception xx) {
        System.out.println(xx);
      }
      try {
        prologProcess2 =
              Runtime.getRuntime().exec(prolog + " -f " + ttt_x) ;
      } catch(Exception xx) {xx.printStackTrace() ; }
    }

synchronized void computer_x_move(String s) { // " x ## y '
    if(!player1.equals("Computer")){
      return;
    }
    String[] c = s.split(",") ;
    String buttonMove = null;
    int x = Integer.parseInt(c[0].trim()),
          y = Integer.parseInt(c[1].trim()) ;
    //System.out.println(x+","+y) ;
    if(player1.equals("Computer")){
      if (x == 1) {
        if (y == 1) {
          b11.setText("X");
          buttonMove = b11.getActionCommand();
        }
        else if (y == 2) {
          b12.setText("X") ;
          buttonMove = b12.getActionCommand();
        }
        else if (y == 3) {
          b13.setText("X") ;
          buttonMove = b13.getActionCommand();
        }
      }
      else if (x == 2) {
        if (y == 1) {
          b21.setText("X") ;
          buttonMove = b21.getActionCommand();
        }
        else if (y == 2) {
          b22.setText("X") ;
          buttonMove = b22.getActionCommand();
        }
        else if (y == 3) {
          b23.setText("X") ;
          buttonMove = b23.getActionCommand();
        }
      }
      else if (x == 3) {
        if (y == 1) {
          b31.setText("X") ;
```

```
            buttonMove = b31.getActionCommand();
          }
          else if (y == 2) {
            b32.setText("X") ;
            buttonMove = b32.getActionCommand();
          }
          else if (y == 3) {
            b33.setText("X") ;
            buttonMove = b33.getActionCommand();
          }
        }
        try {
          logger.info("X placed at "+buttonMove);
          bw.write(buttonMove + "\n") ;
          bw.flush() ;
          Thread.sleep(200);
        } catch(Exception xx) { xx.printStackTrace() ; }
      }
      System.out.println("Computer player: "+s+"\n");
      if (winner()) stopResetGame() ;
      else if (tie()) stopResetGame();
      else  myturn = false ;
    }

      synchronized void computer_move(String s) { // " x ## y '
        logger.info("O placed at "+ s);
        String[] c = s.split(",") ;
        String buttonMove = null;
        int x = Integer.parseInt(c[0].trim()),
              y = Integer.parseInt(c[1].trim()) ;
        //System.out.println(x+","+y) ;
        if (x == 1) {
          if (y == 1) {
            b11.setText("O");
            buttonMove = b11.getActionCommand();
          }
          else if (y == 2) {
            b12.setText("O") ;
            buttonMove = b12.getActionCommand();
          }
          else if (y == 3) {
            b13.setText("O") ;
            buttonMove = b13.getActionCommand();
          }
        }
        else if (x == 2) {
          if (y == 1) {
            b21.setText("O") ;
            buttonMove = b21.getActionCommand();
          }
          else if (y == 2) {
            b22.setText("O") ;
            buttonMove = b22.getActionCommand();
          }
```

```
      else if (y == 3) {
        b23.setText("O") ;
        buttonMove = b23.getActionCommand();
      }
    }
    else if (x == 3) {
      if (y == 1) {
        b31.setText("O") ;
        buttonMove = b31.getActionCommand();
      }
      else if (y == 2) {
        b32.setText("O") ;
        buttonMove = b32.getActionCommand();
      }
      else if (y == 3) {
        b33.setText("O") ;
        buttonMove = b33.getActionCommand();
      }
    }
    if(player1.equals("Computer")){
      try {
        logger.info("X placed at "+buttonMove);
        bw2.write(buttonMove + "\n") ;
        bw2.flush() ;
        Thread.sleep(200);
      } catch(Exception xx) { xx.printStackTrace() ; }
    }
    if (winner()) stopResetGame() ;
    else if (tie()) stopResetGame();
    else  myturn = true ;
  }


  synchronized public void actionPerformed(ActionEvent act) {
    if (!myturn|| player1.equals("Computer")) return ;
    String s = ((JButton)act.getSource()).getText() ;
    if (!s.equals("")) return  ;
    ((JButton)(act.getSource())).setText("X") ;
    logger.info("X's move "+act.getActionCommand());
    try {
      bw.write(act.getActionCommand() + "\n") ;
      bw.flush() ;
    } catch(Exception xx) { System.out.println(xx) ; }

    if(winner())
      stopResetGame();
    else if(tie())
      stopResetGame();
    myturn = false ;
  }

  /**
   *  Do we have a winner?
   */
```

```
  synchronized boolean winner() {
    return  line(b11,b21,b31) ||
            line(b12,b22,b32) ||
            line(b13,b23,b33) ||
            line(b11,b12,b13) ||
            line(b21,b22,b23) ||
            line(b31,b32,b33) ||
            line(b11,b22,b33) ||
            line(b13,b22,b31) ;
  }
  synchronized private boolean tie() {
    JButton[] buttons = { b11, b21, b31, b12, b22, b32, b13, b23, b33 };
    for (JButton button : buttons) {
      if (button.getText().equals(""))
        return false;
    }
    winner="D";
    return true;
  }
  synchronized private void setBgColors(JButton[] buttonsList, Color bg){
    for (JButton button : buttonsList) {
      button.setBackground(bg);
      button.setOpaque(true);
    }
  }

  /**
   *  Are three buttons marked with same player?
   *  If, so color the line and return true.
   */
  synchronized private void stopResetGame() {
    if(game_progress)
      destroy_prolog();
    if(winner.equals("")){
      JOptionPane.showMessageDialog(null, "Game Stopped", "GameOver", JOptionPane.INFORMATION_MESSAGE);
      totGamesPlayed--;
      logger.info("Game has been Stopped!!!");
    }
    else if (winner.equals("X")) {
      JOptionPane.showMessageDialog(null, "Player1(" + winner + ") is the  Winner!!", "GameOver",
JOptionPane.INFORMATION_MESSAGE);
      xWinCounter.setText(Integer.toString(++xWins));
      logger.info("Player1(X) wins!!");
    }
    else if(winner.equals("O")) {
      JOptionPane.showMessageDialog(null, "Player2(" + winner + ") is the Winner!!", "GameOver",
JOptionPane.INFORMATION_MESSAGE);
      oWinCounter.setText(Integer.toString(++oWins));
      logger.info("Player2(O) wins!!");
    }
    else if (winner.equals("D")){
      setBgColors(new JButton[] {b11, b21, b31, b12, b22, b32, b13, b23, b33},Color.gray);
      JOptionPane.showMessageDialog(null, "It's a Tie", "Game Over", JOptionPane.INFORMATION_MESSAGE);
      tieCounter.setText(Integer.toString(++ties));
```

```
      logger.info("It's a Tie!!");
    }
    winner = "";
    totGamesCounter.setText(Integer.toString(++totGamesPlayed));
    logger.info("Game "+totGamesPlayed+" Ended");
    synchronized(nGamesListener.class){
      game_progress = false;
      nGamesListener.class.notifyAll();
    }
  }
  synchronized void resetNGames(){
    game_progress=false;
    xWins=oWins=ties=totGamesPlayed=0;
    xWinCounter.setText(Integer.toString(xWins));
    oWinCounter.setText(Integer.toString(oWins));
    tieCounter.setText(Integer.toString(ties));
    totGamesCounter.setText(Integer.toString(totGamesPlayed));
  }

  boolean line(JButton b, JButton c, JButton d) {
    if (!b.getText().equals("") &&b.getText().equals(c.getText()) &&
          c.getText().equals(d.getText()))  {
      if (b.getText().equals("O")) {
        setBgColors(new JButton[] {b,c,d},Color.red);
        winner=b.getText();
      }
      else {
        setBgColors(new JButton[] {b,c,d},Color.green);
        winner=b.getText();
      }
      return true ;
    } else return false;
  }
}
```

Listing of Connector1.java program

```
import java.io.*;
import java.net.*;
import java.util.* ;
import javax.swing.* ;

/**
 *  A Connector1 is a server that listens for I/O connections
 *  at a port.  Each client gets a Socket and buffered
 *  i/o streams bundled in a Transducer object.
 *  Text read at any input is rebroadcast to other clients.
 *  <pre>
 *  ========================================
 *   Copyright 1999-- John R. Fisher
 *   jrfisher@csupomona.edu
 *  ========================================
 *  </pre>
 *  @author jrfisher@csupomona.edu
```

```
 */

public class Connector1 extends Thread {
  int clientNum ;
  int port ;
  ServerSocket portalSocket ;
  Vector collaborators ; // Object output streams for clients

  public Connector1(int port) {
    this.clientNum = 1 ;
    this.port = port ;
    this.collaborators = new Vector() ;
   }

   public void run() {
     // Catch big exceptions that prevent server from continuing.
      try { // 1
        portalSocket = new ServerSocket(port) ;
       while(true) {
         // Catch smaller exceptions so server itself can continue..
          try { // 2
            Socket soc = portalSocket.accept() ;
           BufferedWriter out =
             new BufferedWriter(new OutputStreamWriter(soc.getOutputStream())) ;
            collaborators.add(out) ;
           System.out.println("Spawning Transducer for " + clientNum) ;
           Transducer b =
             new Transducer(this,
                       new BufferedReader(new InputStreamReader(soc.getInputStream())),
                       out,
                       clientNum) ;
           b.start() ;
            clientNum++ ;
         }
          catch(Exception e2) {
            JOptionPane.showMessageDialog(null,e2.toString(),"Connector1 EXCEPTION
#2",JOptionPane.WARNING_MESSAGE) ;
          }
        }
      }
      catch (Exception e1) {
        // Could not make ServerSocket
        JOptionPane.showMessageDialog(null,e1.toString(),"Connector1 EXCEPTION
#1",JOptionPane.WARNING_MESSAGE) ;
      }
   }

  /**
     *  From the command line ...
     *     java -classpath <path> Connector1 <port#>
     */
   public static void main(String[] args) {
     try {
       int port = Integer.parseInt(args[0]) ;
        Connector1 prtl = new Connector1(port) ;
       prtl.start() ;
       //System.out.println("Starting portal on port " + prtl.portalSocket.getInetAddress() +  port) ;
      }
      catch(Exception e) {
```

```
      System.out.println(e) ;
      System.out.println("usage: java -classpath <> Connector1 <port>") ;
    }
  }
}

/**
 *  A Transducer is an attachment to a particular client.
 *  The Transducer listens to this client and rebroadcasts its
 *  contribution.
 */
class Transducer extends Thread {
  BufferedReader in ;
 BufferedWriter out ;
 int client ;
 Connector1 portal ;

 Transducer(Connector1 p, BufferedReader instream,
            BufferedWriter outstream, int k) {
   this.portal = p ;
   in = instream ;
   out = outstream ;
   client = k ;
  }

  public void run() {
    while(true) {
      try {
        // if can read a good input ...
        String s = in.readLine() ;
         System.out.println(s);
         if (s == null) {    // THIS CLIENT IS GONE ...
           portal.collaborators.remove(out) ;   // remove this client from server
           break ;  // stop running this client
         }
         // try to tell everyone ...
         Iterator it = portal.collaborators.iterator() ;
        while(it.hasNext()) {
          // Avoid bad collaborators if possible or necessary ...
           BufferedWriter bw = null ;
          try {
            bw = (BufferedWriter)(it.next()) ;
            if (bw != out) {  // don't tell self
               bw.write(s+"\r") ; // return for readLine at other end
               bw.flush() ;
            }
          }
          catch(SocketException | ConcurrentModificationException socEx2){}
          catch (Exception e2) {
            JOptionPane.showMessageDialog(null,e2.toString(),"TRANSDUCER EXCEPTION
#2",JOptionPane.WARNING_MESSAGE) ;
          }

        }
      }
      // running exception not otherwise handled ...
       catch(SocketException socEx1){}
       catch(Exception e1) {
         JOptionPane.showMessageDialog(null,e1.toString(),"TRANSDUCER EXCEPTION
```

```
#1",JOptionPane.WARNING_MESSAGE) ;
        }
      }
    }
}
```

Listing of ttt1.pl program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Prolog TicTacToe alpha-beta expert
%% Design to play against human (Java GUI)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  The empty Tac Tac Toe board
%%   Z1 Z2 Z3
%%   Z4 Z5 Z6  ~  [Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9]
%%   Z7 Z8 Z9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
:- dynamic board/1.

init:-
   protocol("ttt-prolog.txt"),
  retractall(board(_)),
   assert(board([_Z1,_Z2,_Z3,_Z4,_Z5,_Z6,_Z7,_Z8,_Z9])).
:- init.

%%%%%
%%  Generate possible marks on a free spot on the board.
%%  Use mark(+,+,-X,-Y) to query/generate possible moves (X,Y).
%%%%%
mark(Player, [X|_],1,1) :- var(X), X=Player.
mark(Player, [_,X|_],2,1) :- var(X), X=Player.
mark(Player, [_,_,X|_],3,1) :- var(X), X=Player.
mark(Player, [_,_,_,X|_],1,2) :- var(X), X=Player.
mark(Player, [_,_,_,_,X|_],2,2) :- var(X), X=Player.
mark(Player, [_,_,_,_,_,X|_],3,2) :- var(X), X=Player.
mark(Player, [_,_,_,_,_,_,X|_],1,3) :- var(X), X=Player.
mark(Player, [_,_,_,_,_,_,_,X|_],2,3) :- var(X), X=Player.
mark(Player, [_,_,_,_,_,_,_,_,X|_],3,3) :- var(X), X=Player.

%%%%%
%%  Move
%%%%%
move(P,(1,1),[X1|R],[P|R]) :- var(X1).
move(P,(2,1),[X1,X2|R],[X1,P|R]) :- var(X2).
move(P,(3,1),[X1,X2,X3|R],[X1,X2,P|R]) :- var(X3).
move(P,(1,2),[X1,X2,X3,X4|R],[X1,X2,X3,P|R]) :- var(X4).
move(P,(2,2),[X1,X2,X3,X4,X5|R],[X1,X2,X3,X4,P|R]) :- var(X5).
move(P,(3,2),[X1,X2,X3,X4,X5,X6|R],[X1,X2,X3,X4,X5,P|R]) :- var(X6).
move(P,(1,3),[X1,X2,X3,X4,X5,X6,X7|R],[X1,X2,X3,X4,X5,X6,P|R]) :- var(X7).
move(P,(2,3),[X1,X2,X3,X4,X5,X6,X7,X8|R],[X1,X2,X3,X4,X5,X6,X7,P|R]) :- var(X8).
move(P,(3,3),[X1,X2,X3,X4,X5,X6,X7,X8,X9|R],[X1,X2,X3,X4,X5,X6,X7,X8,P|R]) :- var(X9).

%%%%%
%%  Record a move: record(+,+,+).
%%%%%
record(Player,X,Y) :-
   retract(board(B)),
  mark(Player,B,X,Y),
   assert(board(B)).

%%%%%
%%  A winning line is ALREADY bound to Player.
```

```
%%  win(+Board,+Player) is true or fail.
%%    e.g., win([P,P,P|_],P).  is NOT correct, because could bind
%%%%%
win([Z1,Z2,Z3|_],P) :- Z1==P, Z2==P, Z3==P.
win([_,_,_,Z1,Z2,Z3|_],P) :-  Z1==P, Z2==P, Z3==P.
win([_,_,_,_,_,_,Z1,Z2,Z3],P) :-  Z1==P, Z2==P, Z3==P.
win([Z1,_,_,Z2,_,_,Z3,_,_],P) :-  Z1==P, Z2==P, Z3==P.
win([_,Z1,_,_,Z2,_,_,Z3,_],P) :-  Z1==P, Z2==P, Z3==P.
win([_,_,Z1,_,_,Z2,_,_,Z3],P) :-  Z1==P, Z2==P, Z3==P.
win([Z1,_,_,_,Z2,_,_,_,Z3],P) :-  Z1==P, Z2==P, Z3==P.
win([_,_,Z1,_,Z2,_,Z3,_,_],P) :-  Z1==P, Z2==P, Z3==P.

%%%%%
%%  A line is open if each position is either free or equals the Player
%%%%%
open([Z1,Z2,Z3|_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,_,_,Z1,Z2,Z3|_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,_,_,_,_,_,Z1,Z2,Z3],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([Z1,_,_,Z2,_,_,Z3,_,_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,Z1,_,_,Z2,_,_,Z3,_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,_,Z1,_,_,Z2,_,_,Z3],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([Z1,_,_,_,Z2,_,_,_,Z3],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,_,Z1,_,Z2,_,Z3,_,_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).

%%%%%
%% Calculate the value of a position, o maximizes, x minimizes.
%%%%%
value(Board,100) :- win(Board,o), !.
value(Board,-100) :- win(Board,x), !.
value(Board,E) :-
findall(o,open(Board,o),MAX),
  length(MAX,Emax),      % # lines open to o
   findall(x,open(Board,x),MIN),
  length(MIN,Emin),      % # lines open to x
   E is Emax - Emin.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% using minimax procedure with alpha-beta cutoff.
% Computer (o) searches for best tic tac toe move,
% Human player is x.
% Adapted from L. Sterling and E. Shapiro, The Art of Prolog, MIT Press, 1986.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

:- assert(lookahead(2)).
:- dynamic spy/0.  % debug calls to alpha_beta
:- assert(spy).    % Comment out stop spy.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
search(Position,Depth,(Move,Value)) :-
  alpha_beta(o,Depth,Position,-100,100,Move,Value).

alpha_beta(Player,0,Position,_Alpha,_Beta,_NoMove,Value) :-
  value(Position,Value),
   spy(Player,Position,Value).

alpha_beta(Player,D,Position,Alpha,Beta,Move,Value) :-
  D > 0,
```

```
  findall((X,Y),mark(Player,Position,X,Y),Moves),
  Alpha1 is -Beta, % max/min
   Beta1 is -Alpha,
   D1 is D-1,
  evaluate_and_choose(Player,Moves,Position,D1,Alpha1,Beta1,nil,(Move,Value)).

evaluate_and_choose(Player,[Move|Moves],Position,D,Alpha,Beta,Record,BestMove) :-
   move(Player,Move,Position,Position1),
  other_player(Player,OtherPlayer),
   alpha_beta(OtherPlayer,D,Position1,Alpha,Beta,_OtherMove,Value),
   Value1 is -Value,
   cutoff(Player,Move,Value1,D,Alpha,Beta,Moves,Position,Record,BestMove).
evaluate_and_choose(_Player,[],_Position,_D,Alpha,_Beta,Move,(Move,Alpha)).

cutoff(_Player,Move,Value,_D,_Alpha,Beta,_Moves,_Position,_Record,(Move,Value)) :-
  Value >= Beta, !.
cutoff(Player,Move,Value,D,Alpha,Beta,Moves,Position,_Record,BestMove) :-
  Alpha < Value, Value < Beta, !,
  evaluate_and_choose(Player,Moves,Position,D,Value,Beta,Move,BestMove).
cutoff(Player,_Move,Value,D,Alpha,Beta,Moves,Position,Record,BestMove) :-
  Value =< Alpha, !,
  evaluate_and_choose(Player,Moves,Position,D,Alpha,Beta,Record,BestMove).

other_player(o,x).
other_player(x,o).

spy(Player,Position,Value) :-
  spy, !,
   write(Player),
   write(' '),
   write(Position),
   write(' '),
   writeln(Value).
spy(_,_,_). % do nothing

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% For testing, use h(+,+) to record human move,
%%% supply coordinates. Then call c (computer plays).
%%% Use s to show board.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h(X,Y) :-
  record(x,X,Y),
  showBoard.

c :-
  board(B),
  alpha_beta(o,2,B,-200,200,(X,Y),_Value),
  record(o,X,Y),
   showBoard.

showBoard :-
  board([Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9]),
  write('    '),mark(Z1),write(' '),mark(Z2),write(' '),mark(Z3),nl,
   write('    '),mark(Z4),write(' '),mark(Z5),write(' '),mark(Z6),nl,
   write('    '),mark(Z7),write(' '),mark(Z8),write(' '),mark(Z9),nl.
s :- showBoard.

mark(X) :-
```

```
  var(X),
   write('#').
mark(X) :-
  \+var(X),
   write(X).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Play tic tac toe with the Java GUI
%%%  using port 54321.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

connect(Port) :-
  tcp_socket(Socket),
  gethostname(Host),  % local host
   tcp_connect(Socket,Host:Port),
   tcp_open_socket(Socket,INs,OUTs),
  assert(connectedReadStream(INs)),
  assert(connectedWriteStream(OUTs)).

:- connect(54321).  % Comment out for testing

ttt :-
  connectedReadStream(IStream),
  read(IStream,(X,Y)),
   record(x,X,Y),
  board(B),
   alpha_beta(o,2,B,-200,200,(U,V),_Value),
  record(o,U,V),
   connectedWriteStream(OStream),
  write(OStream,(U,V)),
  nl(OStream), flush_output(OStream),
   ttt.

:- ttt.            % Comment out for testing
```

Listing of ttt_x.pl program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Prolog TicTacToe alpha-beta expert
%% Design to play against human (Java GUI)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  The empty Tac Tac Toe board
%%   Z1 Z2 Z3
%%   Z4 Z5 Z6   ~   [Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9]
%%   Z7 Z8 Z9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
:- dynamic board/1.

init:-
retractall(board(_)),
  assert(board([_Z1,_Z2,_Z3,_Z4,_Z5,_Z6,_Z7,_Z8,_Z9])).
:- init.

%%%%%
%%  Generate possible marks on a free spot on the board.
```

```
%%  Use mark(+,+,-X,-Y) to query/generate possible moves (X,Y).
%%%%%
mark(Player, [X|_],1,1) :- var(X), X=Player.
mark(Player, [_,X|_],2,1) :- var(X), X=Player.
mark(Player, [_,_,X|_],3,1) :- var(X), X=Player.
mark(Player, [_,_,_,X|_],1,2) :- var(X), X=Player.
mark(Player, [_,_,_,_,X|_],2,2) :- var(X), X=Player.
mark(Player, [_,_,_,_,_,X|_],3,2) :- var(X), X=Player.
mark(Player, [_,_,_,_,_,_,X|_],1,3) :- var(X), X=Player.
mark(Player, [_,_,_,_,_,_,_,X|_],2,3) :- var(X), X=Player.
mark(Player, [_,_,_,_,_,_,_,_,X|_],3,3) :- var(X), X=Player.

%%%%%
%%  Move
%%%%%
move(P,(1,1),[X1|R],[P|R]) :- var(X1).
move(P,(2,1),[X1,X2|R],[X1,P|R]) :- var(X2).
move(P,(3,1),[X1,X2,X3|R],[X1,X2,P|R]) :- var(X3).
move(P,(1,2),[X1,X2,X3,X4|R],[X1,X2,X3,P|R]) :- var(X4).
move(P,(2,2),[X1,X2,X3,X4,X5|R],[X1,X2,X3,X4,P|R]) :- var(X5).
move(P,(3,2),[X1,X2,X3,X4,X5,X6|R],[X1,X2,X3,X4,X5,P|R]) :- var(X6).
move(P,(1,3),[X1,X2,X3,X4,X5,X6,X7|R],[X1,X2,X3,X4,X5,X6,P|R]) :- var(X7).
move(P,(2,3),[X1,X2,X3,X4,X5,X6,X7,X8|R],[X1,X2,X3,X4,X5,X6,X7,P|R]) :- var(X8).
move(P,(3,3),[X1,X2,X3,X4,X5,X6,X7,X8,X9|R],[X1,X2,X3,X4,X5,X6,X7,X8,P|R]) :- var(X9).

%%%%%
%%  Record a move: record(+,+,+).
%%%%%
record(Player,X,Y) :-
  retract(board(B)),
  mark(Player,B,X,Y),
   assert(board(B)).

%%%%%
%%  A winning line is ALREADY bound to Player.
%%  win(+Board,+Player) is true or fail.
%%    e.g., win([P,P,P|_],P).  is NOT correct, because could bind
%%%%%
win([Z1,Z2,Z3|_],P) :- Z1==P, Z2==P, Z3==P.
win([_,_,_,Z1,Z2,Z3|_],P) :-  Z1==P, Z2==P, Z3==P.
win([_,_,_,_,_,_,Z1,Z2,Z3],P) :-  Z1==P, Z2==P, Z3==P.
win([Z1,_,_,Z2,_,_,Z3,_,_],P) :-  Z1==P, Z2==P, Z3==P.
win([_,Z1,_,_,Z2,_,_,Z3,_],P) :-  Z1==P, Z2==P, Z3==P.
win([_,_,Z1,_,_,Z2,_,_,Z3],P) :-  Z1==P, Z2==P, Z3==P.
win([Z1,_,_,_,Z2,_,_,_,Z3],P) :-  Z1==P, Z2==P, Z3==P.
win([_,_,Z1,_,Z2,_,Z3,_,_],P) :-  Z1==P, Z2==P, Z3==P.

%%%%%
%%  A line is open if each position is either free or equals the Player
%%%%%
open([Z1,Z2,Z3|_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,_,_,Z1,Z2,Z3|_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,_,_,_,_,_,Z1,Z2,Z3],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([Z1,_,_,Z2,_,_,Z3,_,_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,Z1,_,_,Z2,_,_,Z3,_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,_,Z1,_,_,Z2,_,_,Z3],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([Z1,_,_,_,Z2,_,_,_,Z3],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
open([_,_,Z1,_,Z2,_,Z3,_,_],Player) :- (var(Z1) | Z1 == Player),(var(Z2) | Z2 == Player), (var(Z3) | Z3 == Player).
```

```
%%%%%
%% Calculate the value of a position, o maximizes, x minimizes.
%%%%%
value(Board,100) :- win(Board,o), !.
value(Board,-100) :- win(Board,x), !.
value(Board,E) :-
  findall(x,open(Board,x),MAX),
  length(MAX,Emax),      % # lines open to x
   findall(o,open(Board,o),MIN),
  length(MIN,Emin),      % # lines open to o
   E is Emax - Emin.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% using minimax procedure with alpha-beta cutoff.
% Computer (o) searches for best tic tac toe move,
% Human player is x.
% Adapted from L. Sterling and E. Shapiro, The Art of Prolog, MIT Press, 1986.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

:- assert(lookahead(2)).
:- dynamic spy/0.  % debug calls to alpha_beta
:- assert(spy).    % Comment out stop spy.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
search(Position,Depth,(Move,Value)) :-
  alpha_beta(o,Depth,Position,-100,100,Move,Value).

alpha_beta(Player,0,Position,_Alpha,_Beta,_NoMove,Value) :-
  value(Position,Value),
   spy(Player,Position,Value).

alpha_beta(Player,D,Position,Alpha,Beta,Move,Value) :-
  D > 0,
  findall((X,Y),mark(Player,Position,X,Y),Moves),
  Alpha1 is -Beta, % max/min
   Beta1 is -Alpha,
   D1 is D-1,
  evaluate_and_choose(Player,Moves,Position,D1,Alpha1,Beta1,nil,(Move,Value)).

evaluate_and_choose(Player,[Move|Moves],Position,D,Alpha,Beta,Record,BestMove) :-
   move(Player,Move,Position,Position1),
  other_player(Player,OtherPlayer),
   alpha_beta(OtherPlayer,D,Position1,Alpha,Beta,_OtherMove,Value),
   Value1 is -Value,
   cutoff(Player,Move,Value1,D,Alpha,Beta,Moves,Position,Record,BestMove).
evaluate_and_choose(_Player,[],_Position,_D,Alpha,_Beta,Move,(Move,Alpha)).

cutoff(_Player,Move,Value,_D,_Alpha,Beta,_Moves,_Position,_Record,(Move,Value)) :-
  Value >= Beta, !.
cutoff(Player,Move,Value,D,Alpha,Beta,Moves,Position,_Record,BestMove) :-
  Alpha < Value, Value < Beta, !,
  evaluate_and_choose(Player,Moves,Position,D,Value,Beta,Move,BestMove).
cutoff(Player,_Move,Value,D,Alpha,Beta,Moves,Position,Record,BestMove) :-
  Value =< Alpha, !,
  evaluate_and_choose(Player,Moves,Position,D,Alpha,Beta,Record,BestMove).

other_player(o,x).
```

```
other_player(x,o).


spy(Player,Position,Value) :-
  spy, !,
   write(Player),
   write(' '),
   write(Position),
   write(' '),
   writeln(Value).
spy(_,_,_). % do nothing

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% For testing, use h(+,+) to record human move,
%%% supply coordinates. Then call c (computer plays).
%%% Use s to show board.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# h(X,Y) :-
#    record(x,X,Y),
#    showBoard.

# c :-
#    board(B),
#    alpha_beta(o,2,B,-200,200,(X,Y),_Value),
#    record(o,X,Y),
#    showBoard.


# showBoard :-
#    board([Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9]),
#    write('    '),mark(Z1),write(' '),mark(Z2),write(' '),mark(Z3),nl,
#    write('    '),mark(Z4),write(' '),mark(Z5),write(' '),mark(Z6),nl,
#    write('    '),mark(Z7),write(' '),mark(Z8),write(' '),mark(Z9),nl.
# s :- showBoard.

# mark(X) :-
#    var(X),
#    write('#').
# mark(X) :-
#    \+var(X),
#    write(X).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Play tic tac toe with the Java GUI
%%%  using port 54322.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

connect(Port) :-
  tcp_socket(Socket),
  gethostname(Host),  % local host
   tcp_connect(Socket,Host:Port),
   tcp_open_socket(Socket,INs,OUTs),
  assert(connectedReadStream(INs)),
  assert(connectedWriteStream(OUTs)).

:- connect(54322).  % Comment out for testing

ttt :-
```

```
  board(B),
   alpha_beta(x,2,B,-200,200,(U,V),_Value),
  record(x,U,V),
   connectedWriteStream(OStream),
  write(OStream,(U,V)),
  nl(OStream), flush_output(OStream),
   connectedReadStream(IStream),
  read(IStream,(X,Y)),
   record(o,X,Y),
  board(B),
   ttt.

:- ttt.            % Comment out for testing
```