

# Distributed Systems, Monsoon 2018

## Distributed Tic Tac Toe

### Assignment 2

**Deadline: 30-08-18**

## Task

The aim of the project is to develop a distributed Tic Tac Toe, in a client-server setup using Java RMI protocol.

You can think of the server as moderator conducting the game. The server will be maintaining the state of the game and also the server will be one providing APIs to inquire or change the state of board.

Clients will be the player, which will inquire or play their chance by remotely calling the server APIs.

## Conduct

### Server-Client Relationship

1. Server will be up always.
2. User will deal only with client. User should be prompted to start the game. As soon as she wants to play, client should connect to server prompting to join the game.
3. At a time, only two clients can be connected to server. The server should send a Busy/Wait message in case third client wants to connect.

### Game Play

1. When first client connects, the server should wait for the second client to connect and send a corresponding message to the first client.
2. As soon as second client connects, it assigns player1/player2 randomly to the two clients and should start the game. Player 1 has 'x' marker and Player 2 has 'o' marker.
3. Whichever player has to make a move will receive the current state of the board from the server and other player will wait for the other player to make a move. Respective messages should be displayed for both.

4. Player 1 is requested to make a move first.
5. As soon as Player 1 makes the move, updates in the board state will be made and Player 2 will receive the updated board state.
6. When the game is over:
  - a. Player who won should get the “Win” message
  - b. Player who lost should get the “Lose” message
  - c. In case of draw, both should get “Draw” message.
  - d. Both should be prompted with “Play another game?” message. If both agrees, fresh game should start else both the clients disconnect and close.

### Input/Output

1. Please follow the input/output guidelines for smooth evaluation.
  2. Server will not output anything.
  3. When client runs, it should prompt the user if she wants to join the game.
  4. On joining, if other player has not joined, respective message should be displayed.
  5. When game starts, Player1/Player2 should be displayed for respective user.
  6. Whichever player has to make a move will have her client print the board state and prompt for an input. Other player will receive a “Wait for your turn” message.
- Board state will be like following. Whatever cell is filled will have respective marker in it.
  - Others will have the cell number printed in it. The user will input cell number to make a move.

1	2	3
4	5	6
7	8	9

x	2	3
4	5	6
7	8	9

x	2	3
4	5	6
7	8	o

7. When game is over, respective Win/Lose/Draw should be displayed and a prompt should be made to the user by client to play another game.
8. User should input 1/0 to play another game.

## Bonus

### 1. Game Parallelization

- a. The server should be able to run multiple games in parallel.
- b. If  $X$  clients connect to server, where  $X$  is even,  $X/2$  games should run in parallel.
- c. The players will play in the order they connect. If  $P1$  and  $P2$  connect, they play a game together. While the game for  $P1$  and  $P2$  is going on, if  $P3$  and  $P4$  connects, their game starts fresh irrespective of the first game.

### 2. Timer

- a. For each move, a 10 seconds timer should be started. If the player is not able to make a move in 10 seconds, she will lose the game and prompt for another game should be displayed.

## Upload Format

1. One folder(named as Assignment2\_<Roll No>) containing only two files, server and client files.
2. Name of the zip file should be Assignment2\_<Roll No>.zip

## Note:

1. There will be VIVA asked for each question.
2. Any form of plagiarism shall result in negative marks.