Program 1: Online Tic-Tac-Toe Game

Documentation:

OnlineTicTacToe( ):
This constructor is default constructor performing the actions of the counterpart and reading the actions done by its counterpart, and that number into blocked cells, and changing the myTurn as true. It checks if it is myTurn, then chose a number randomly between 0 to 8 and select it is not blocked yet. Sending this information to the first player by using writeObject.

OnlineTicTacToe(InetAddress addr, int port) :
This constructor establishes the TCP connection between the client and server by using accept() and enabling the makeWindow function as “true” and “false” client/server. We will check the address it has to establish TCP connection between client and server, if the address is not localhost, then we will set the timeout to make it non-blocking. Using Object input/output stream it communicates with the counterpart.

OnlineTicTacToe(String hostname):
This constructor launches a remote OnlineTicTacToe through JSCH. it reads username and password from the console, creates a command to be executed at the remote machine. When the player clicks on the grid of windows, actionPerformed is called.It then calls counterpart.

actionPerformed(ActionEvent event):
This is called by AWT whenever any button has been clicked. In this function, I have used synchronized(myTurn) to make this unblocking from multiple threads. If it is myTurn, checking which button was clicked, marking the button, adding this number into blocked cells. I am also checking if “O” has won the game by calling func checkForWin(myMark), then display the windows for the winner. And sending this message to my counterpart by writeObject. Then updating the myTurn as “false” and using “notifyAll” to notify the running thread.

Counterpart.run( ):

I have used synchronized(myTurn) to make this unblocking from multiple threads. If the myTurn is false, then this thread is used by the second player to read the information sent by the first player, and marking the button. It will also check the if the second player i.e. “X” wins after every move. Then updating the myTurn as “true” and using “notifyAll” to notify the running thread.

checkForWin(String mark):

This method checks the winning condition for mark (“O” or “X”), it is a boolean function, so if the mark is vertically, horizontally or diagonally same in the three cells, then it returns “true” else it returns “false”.

Source code:

Zip folder contains source code for:

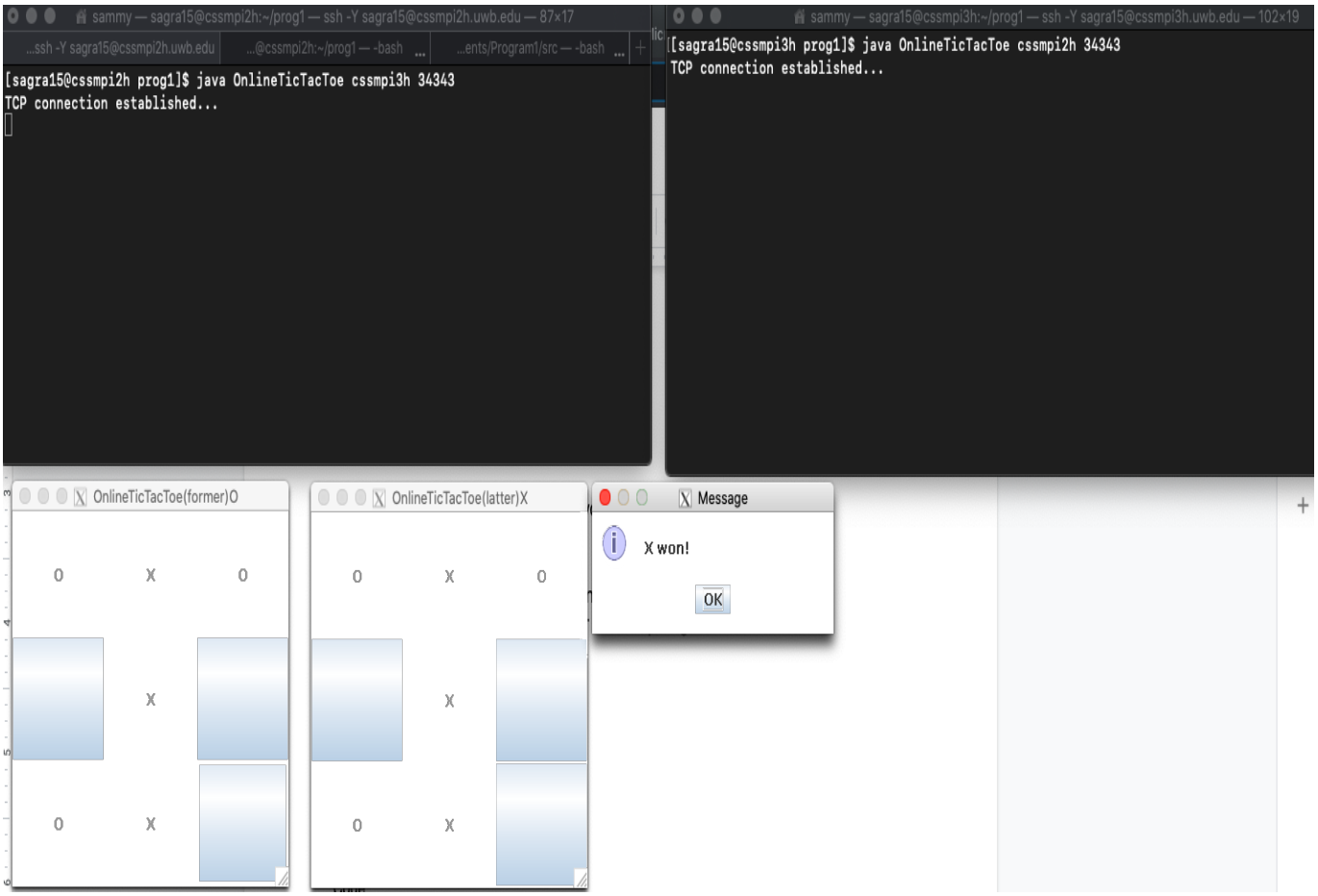
1. Prog1
2. Lab1
3. Lab2
4. Lab3

Attached in Canvas

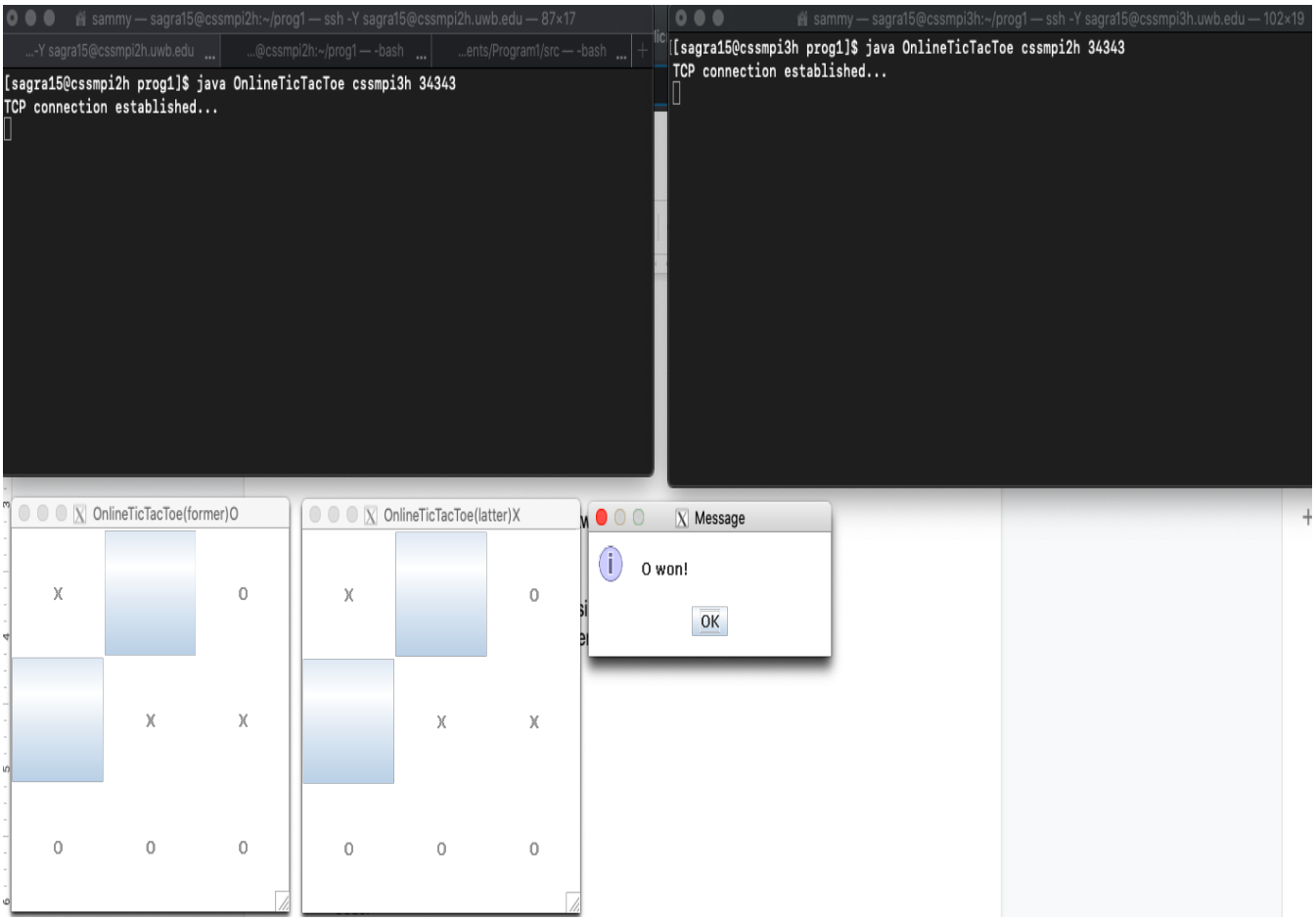
Execution output:

(1) A two-user interactive game over two computing nodes:

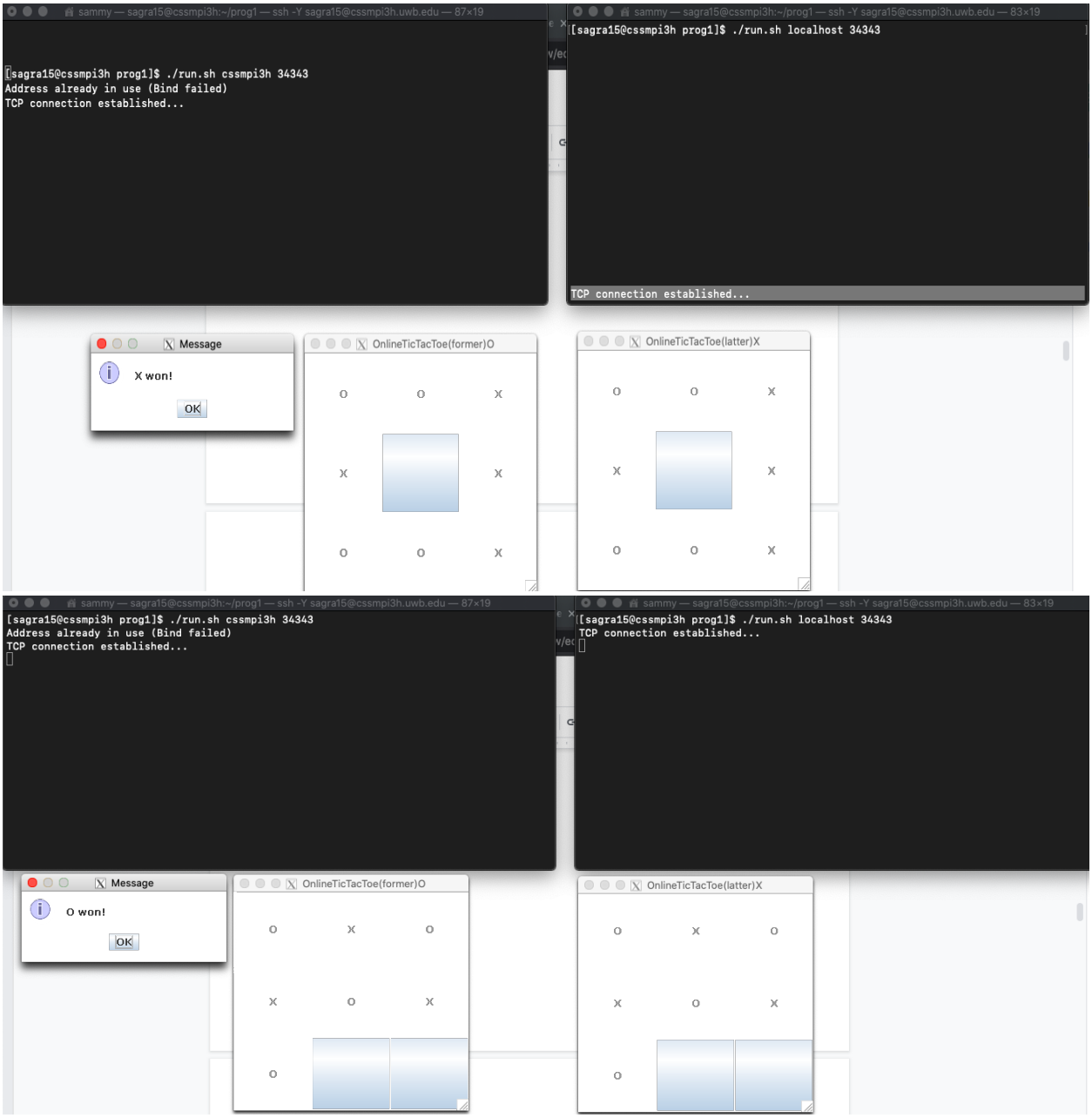
“X” wins:



“O” wins:

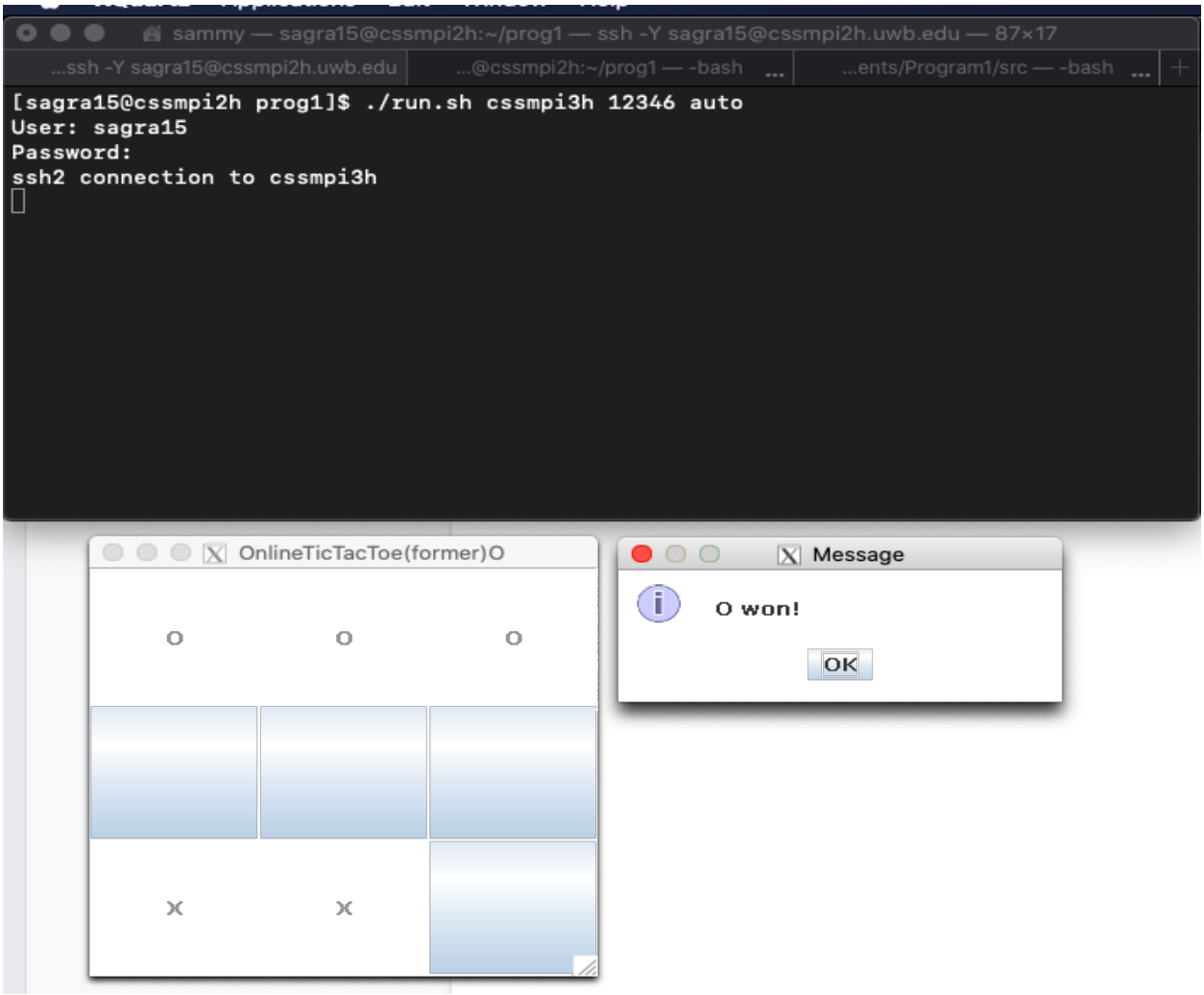

sammy — sagra15@cssmpi2h:~/prog1 — ssh -Y sagra15@cssmpi2h.uwb.edu — 87×17
[sagra15@cssmpi2h prog1]$ java OnlineTicTacToe cssmpi3h 34343
TCP connection established...
sammy — sagra15@cssmpi3h:~/prog1 — ssh -Y sagra15@cssmpi3h.uwb.edu — 102×19
[sagra15@cssmpi3h prog1]$ java OnlineTicTacToe cssmpi2h 34343
TCP connection established...
OnlineTicTacToe(former)O
OnlineTicTacToe(latter)X
Message
O won!
OK

(2) A two-user interactive game on a single computing node:

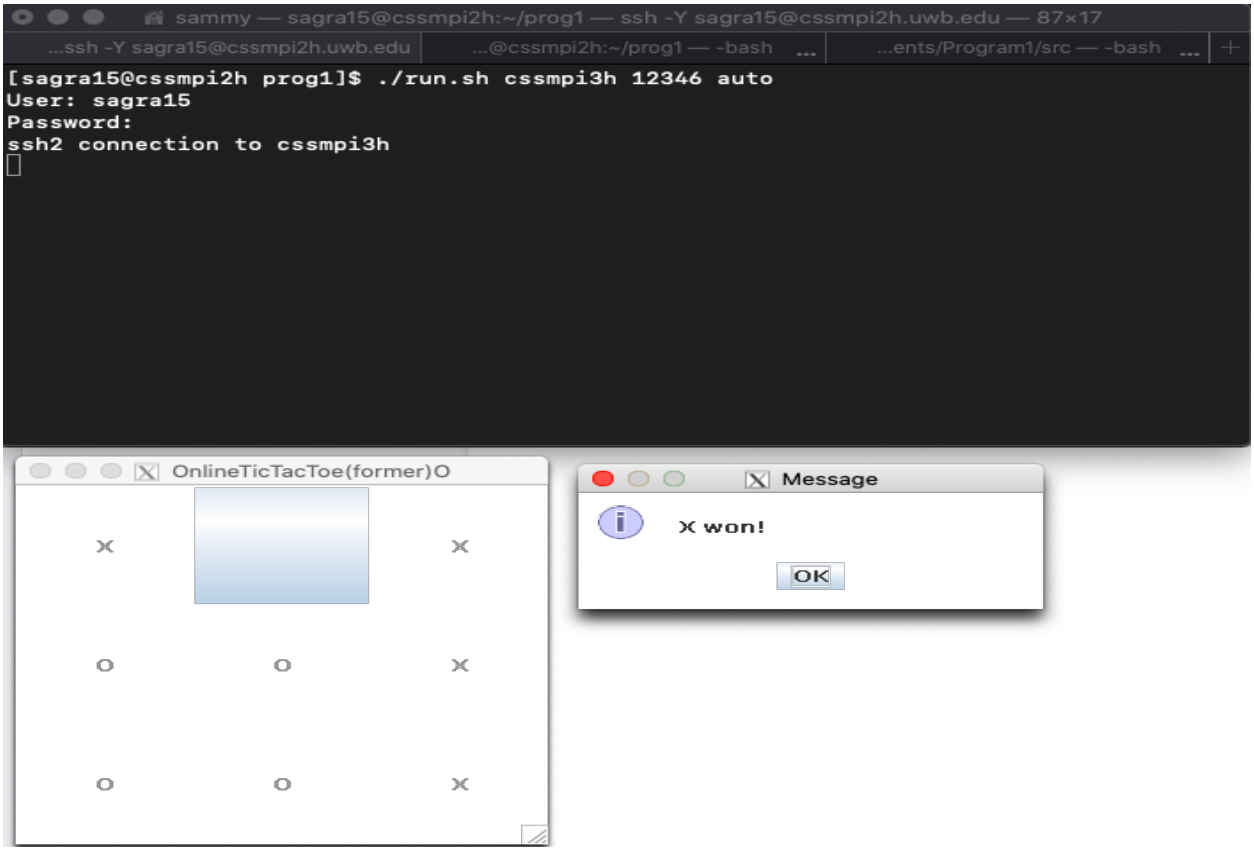


(3) A single-user automated game over two computing nodes:

“O” wins:



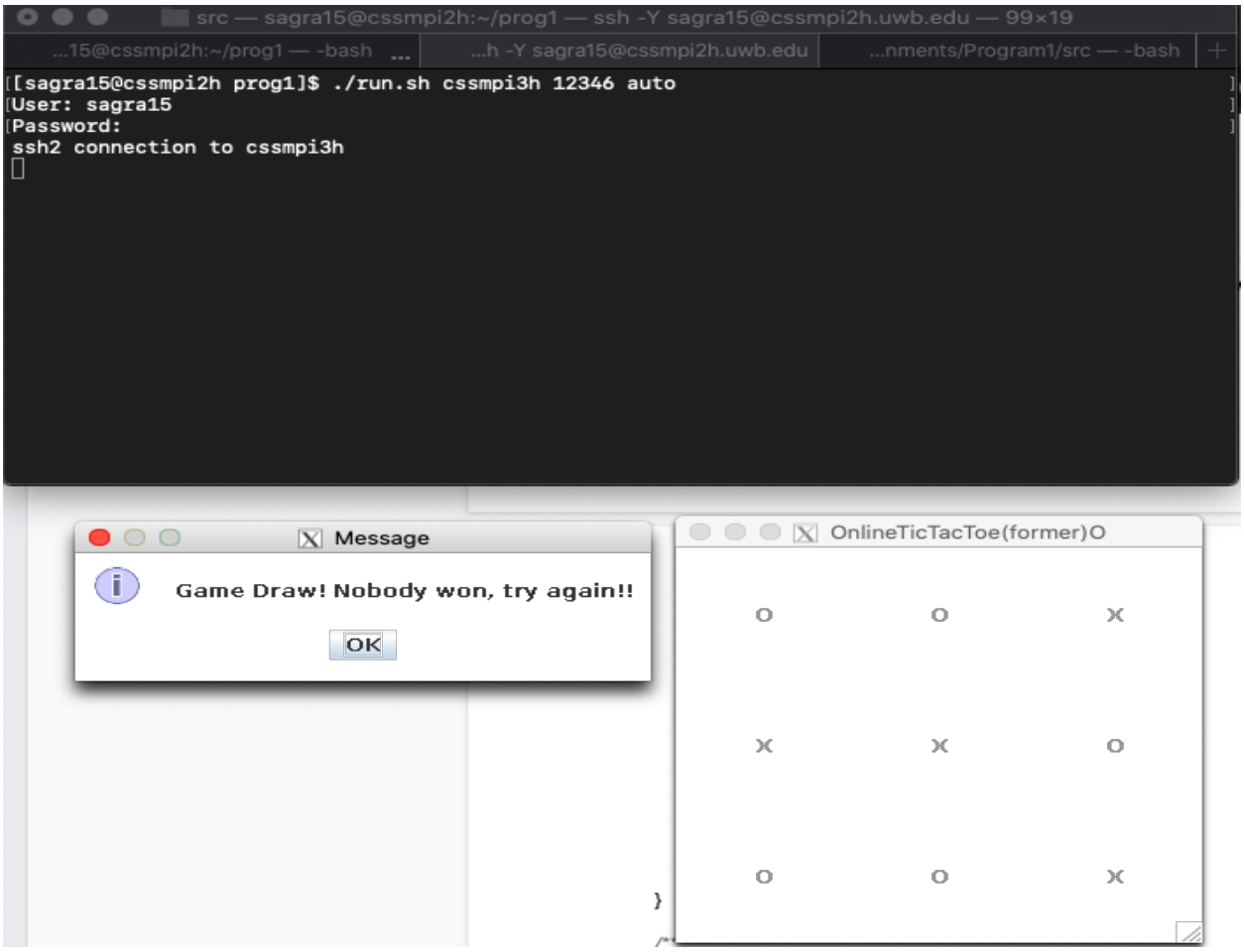
“X” wins:

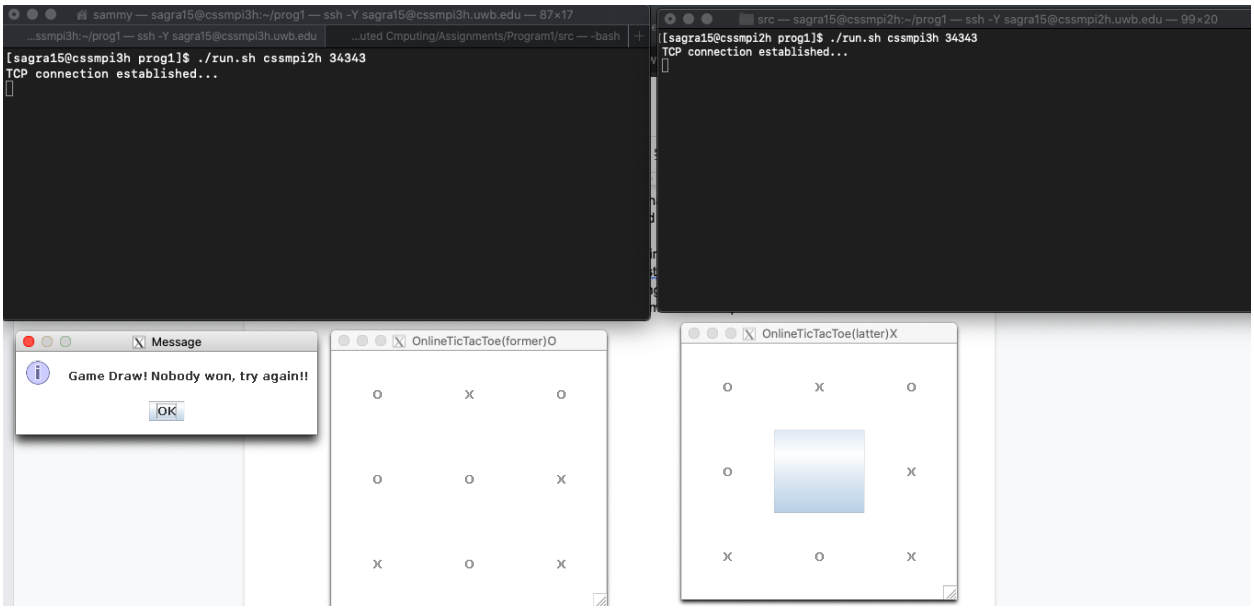


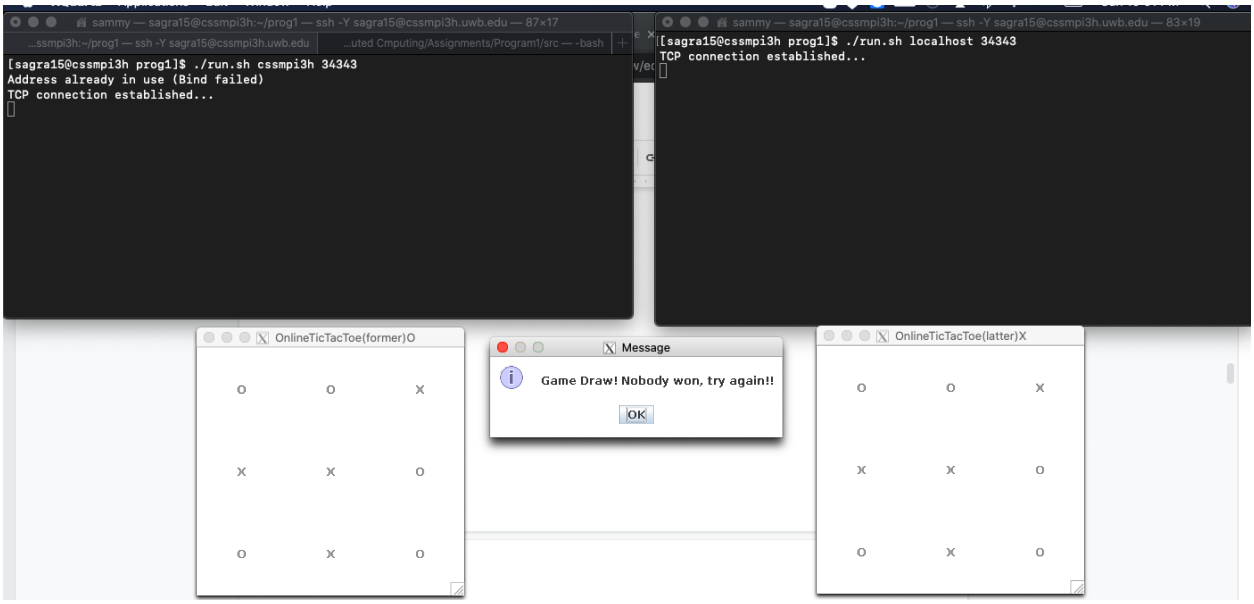
Discussions:

Additional features

1. checkForGameDraw(): I have added a function checkForGameDraw() which do not need any arguments to be passed, and has a return value as “boolean”. This function will check if the game is drawn, means no player has won the game, try again. I have initialized a flag value default as “false”. It will check for the size of the blocked cell on the window for 9, if yes check nobody won, set the flag as “true”, and its a draw.
2. showDraw(): Pops out a small window that indicates a winning message: “**Game Draw! Nobody won, try again!!**”. Here are the screenshots.


src — sagra15@cssmpi2h:~/prog1 — ssh -Y sagra15@cssmpi2h.uwb.edu — 99×19
[sagra15@cssmpi2h prog1]$ ./run.sh cssmpi3h 12346 auto
User: sagra15
Password:
ssh2 connection to cssmpi3h
Message
Game Draw! Nobody won, try again!!
OK
OnlineTicTacToe(former)O
O O X
X X O
O O X


[sagra15@cssmpi3h prog1]$ ./run.sh cssmpi2h 34343
TCP connection established...
[sagra15@cssmpi2h prog1]$ ./run.sh cssmpi3h 34343
TCP connection established...
Message
Game Draw! Nobody won, try again!!
OK
OnlineTicTacToe(former)O
OnlineTicTacToe(latter)X



Limitations

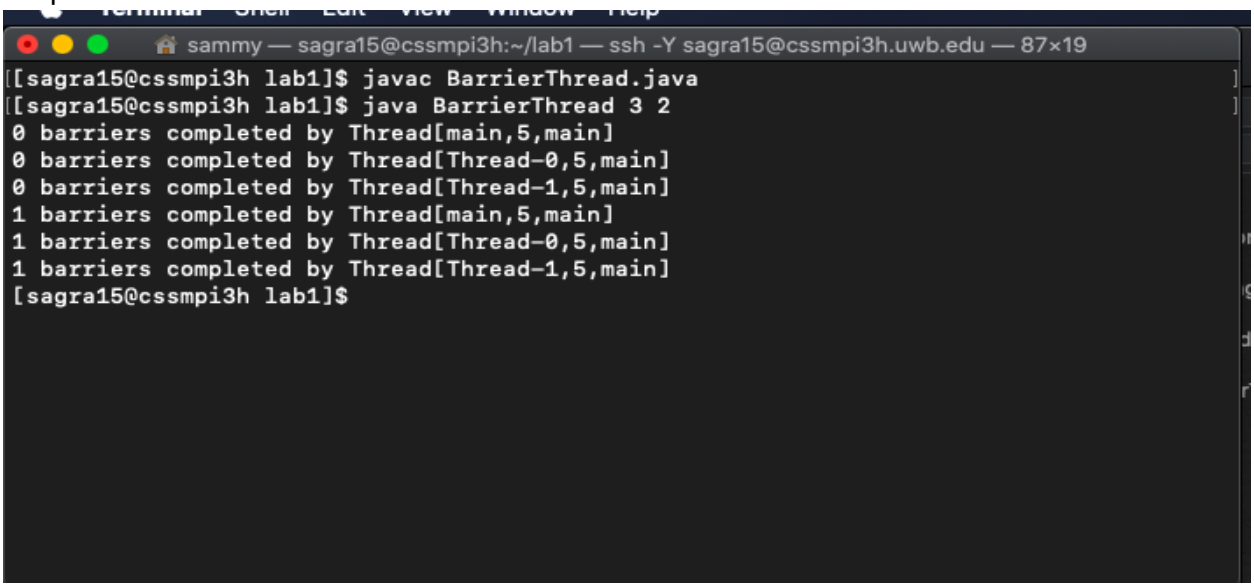
1. When the game gets over, if the message window gets closed after clicking on “ok” button, the OnlineTic TacToe window should get closed automatically.
2. As in the auto player, “O” is always the first player, there should be flexibility in the game to choose the first player and second player.

Possible Improvements

1. After the game is draw, that means no player has won the game, so the message displayed is to try again, then the game should be started again, meaning it can refresh the game and start the game again.
2. We can add a button “Restart” on the game window to start the game again whenever player wants to drop the current game and restart game.
3. We can add a button “Timer” with timer ongoing for the game in the reverse order, and displaying that 2 minutes remaining for the game to be completed.
4. When the auto player is selecting the “button” to be clicked, in the current program it is randomly selected, we can implement a clever approach to select the button by checking two O’s in a row or column.
5. We can add player’s profile picture and use icon to mark the buttons, to make more attractive and personal.

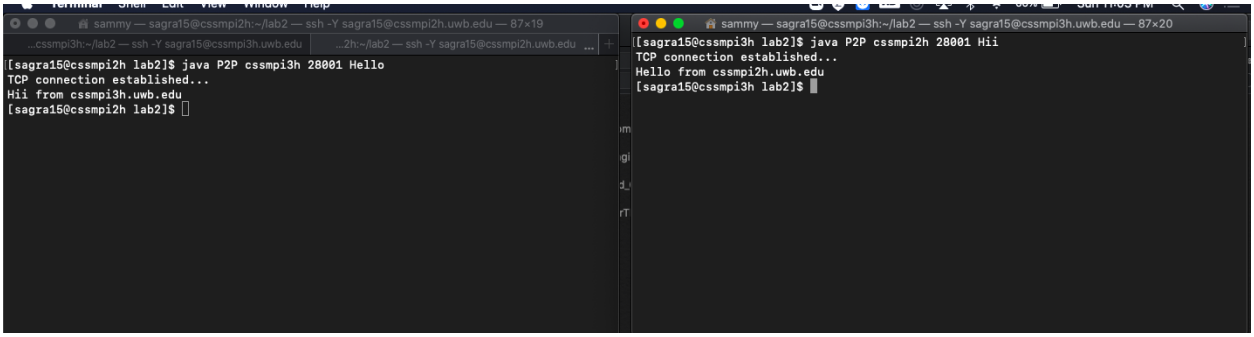
Lab Session 1:

Output:



Lab Session 2:

Output:



Lab Session 3:

Output: