COMP2396B Object-oriented programming and Java
Assignment 4: A two-player Tic-Tac-Toe Game
Due Date: 13th May 2022 23:59

This assignment tests your understanding on GUI, Java Socket Programming and the application of multi-threading.

In this assignment, you are going to implement a two-player Tic-Tac-Toe Game. The game would only start when the Tic-Tac-Toe Server runs and two players are connected to the server. When the game starts, players need to input their player's name first (i.e., the player cannot make any move if he/she does not enter his/her player's name) (Fig. 1). Players are not allowed to re-input their names again once they have submitted their names (i.e., the textboxes and the submit buttons should be disabled). In addition, their names should be displayed in the frame's title (i.e., change from Tic Tac Toe to Tic Tac Toe-Player: (player's name)) and the message title (located below the menu bar) should change from "Enter your player name…" to "WELCOME (player's name)" (Fig. 2).
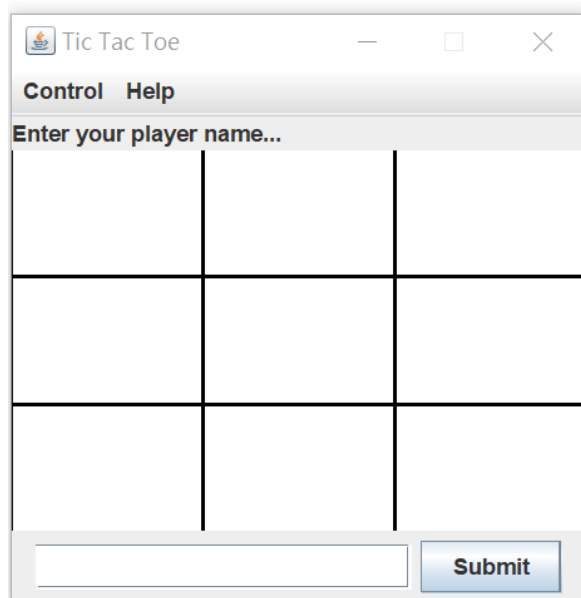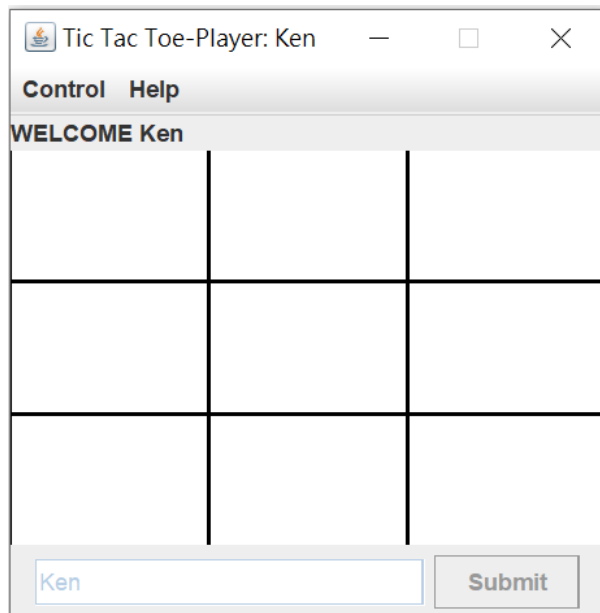
==Initial setting:==

Fig. 1

Fig. 2

After entering the player's name, the game would always be started by player 1 (the player's mark with a "x")'s first move (i.e., player 2 (the player's mark with a "o") cannot make his/her first move until player 1 makes his/her first move). If player 1's move is valid, his/her move would be marked as a "x" on the 3 x 3 board (The mark "x" on the 3 x 3 board should be displayed on both player 1's and player 2's board). The message title in the player 1's board would be changed to "Valid move, wait for your opponent." while the message title in the player 2's board would be changed to "Your opponent has moved, now is your turn.". Besides, player 1 is not allowed to make the next move until his/her opponent moves (i.e., if player 1 makes the next move before his/her opponent moves, it would be considered as an invalid move and nothing would be displayed on the board). The turn would be switched to the opponent (this time player 2) and if player 2's move is valid, it would be mark as a "o" on the 3 x 3 board on both players. The message displayed in the message title of the player 2's board would be changed to "Valid move, wait for your opponent." while the message title of the player 1's board would be changed to "Your opponent has moved, now is your turn." Similarly, the turn would be switched to player 1 again and player 2 cannot make his/her next move until player 1 makes the move (Fig. 3 & 4).
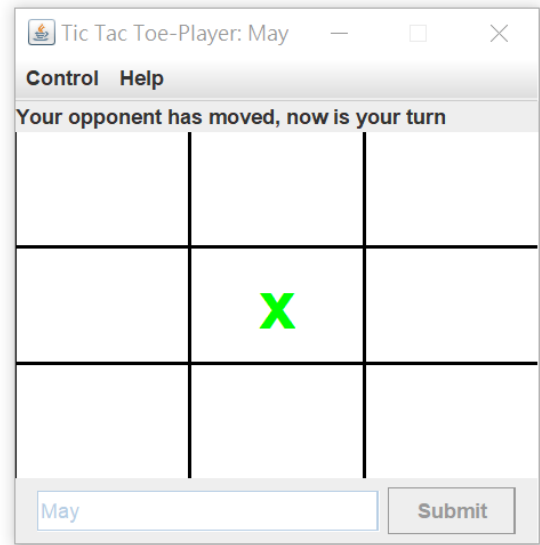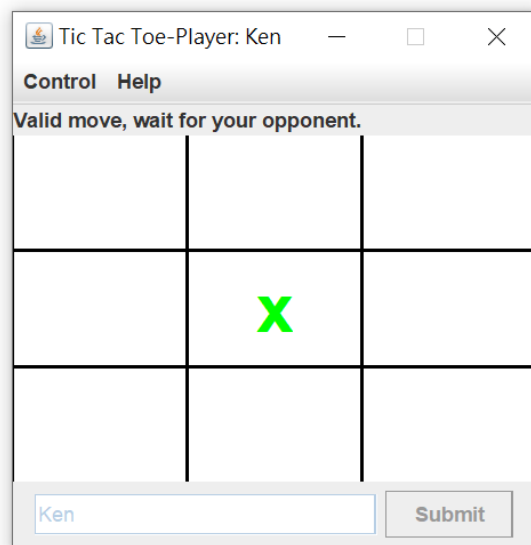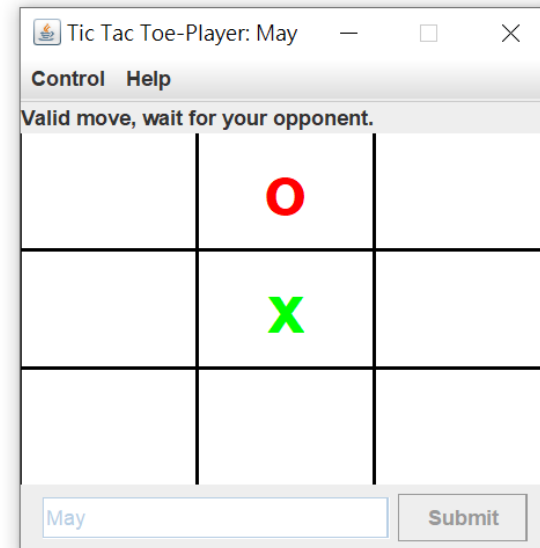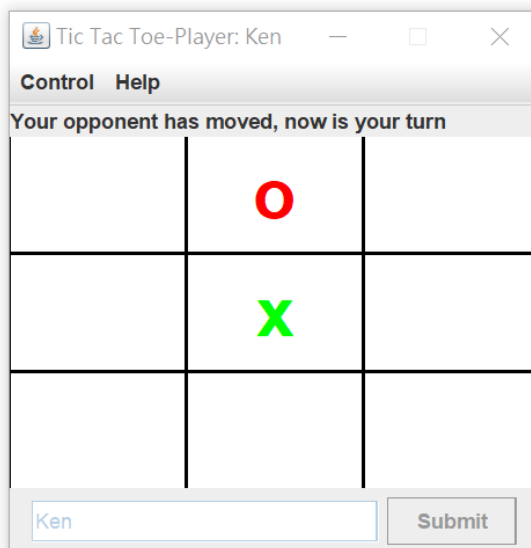
Fig. 3

Fig. 4

Criteria for a valid move:
- The move is not occupied by any mark.
- The move is made in the player's turn.
- The move is made within the 3 x 3 board.

The game would continue and switch among the opposite player until it reaches either one of the following conditions:

- Player 1 wins.
- Player 2 wins.
- Draw.

The winning condition is that when there is any row, column or diagonal that is filled with the same mark (e.g., player 1 would win if there is any row, column or diagonal that is filled with "x", player 2 would win if there is any row, column or diagonal that is filled with "o"). The game will draw if no players satisfy the winning condition after all the board location is filled with mark. Once it reaches either one of the above conditions, Message Dialog would be displayed on both players (the content of the Message Dialog would be different, based on the condition reached) and both players cannot make further move. The following screen captures show the Message Dialog displayed on the two players when either one of the player wins (Fig. 5 & 6) or the game is draw (Fig. 7).
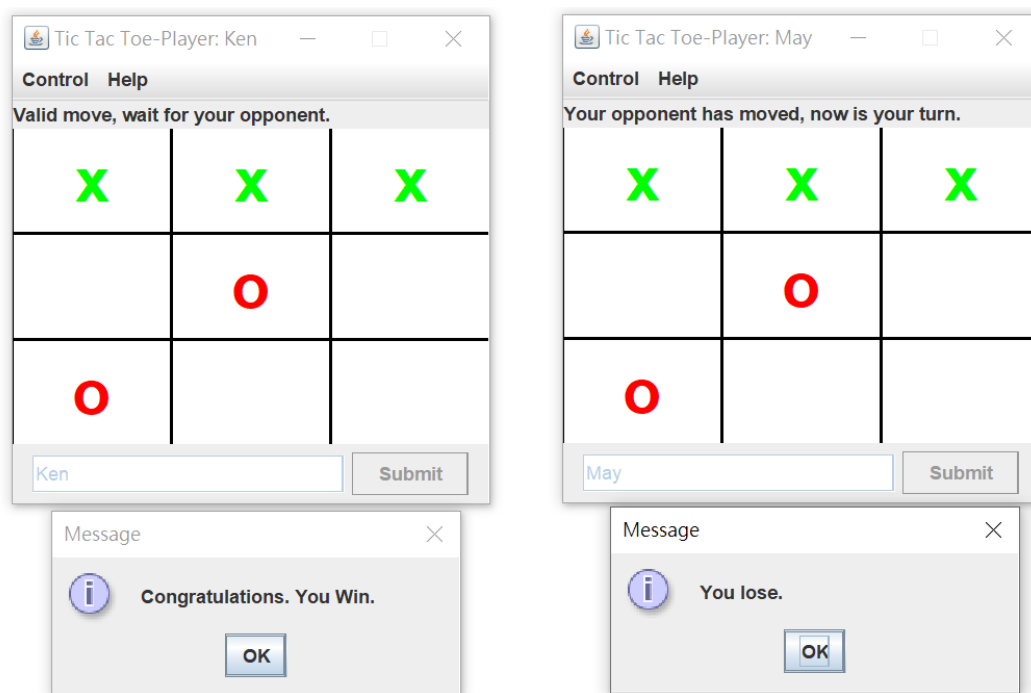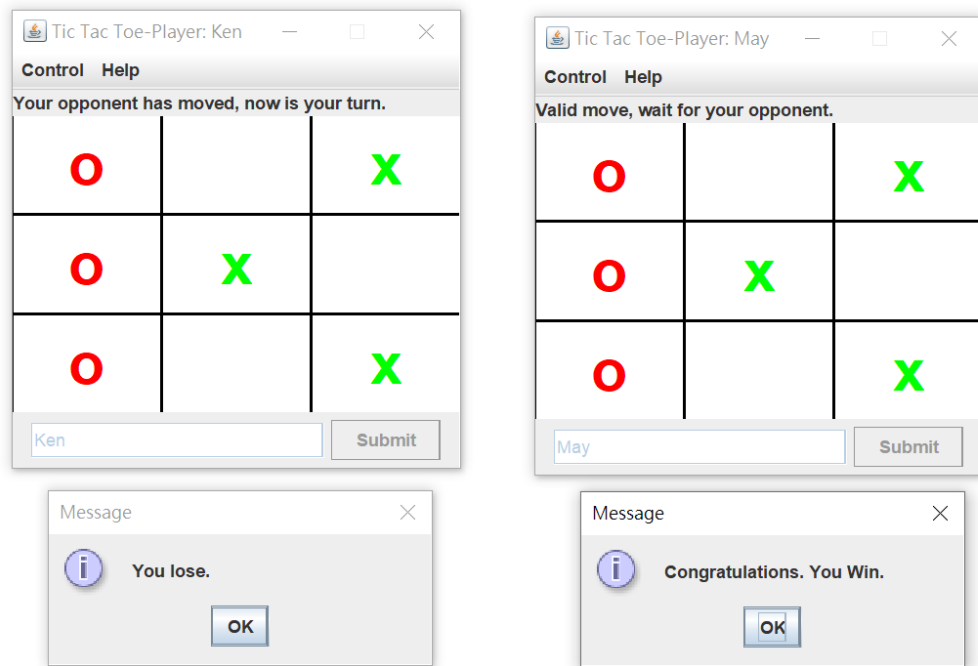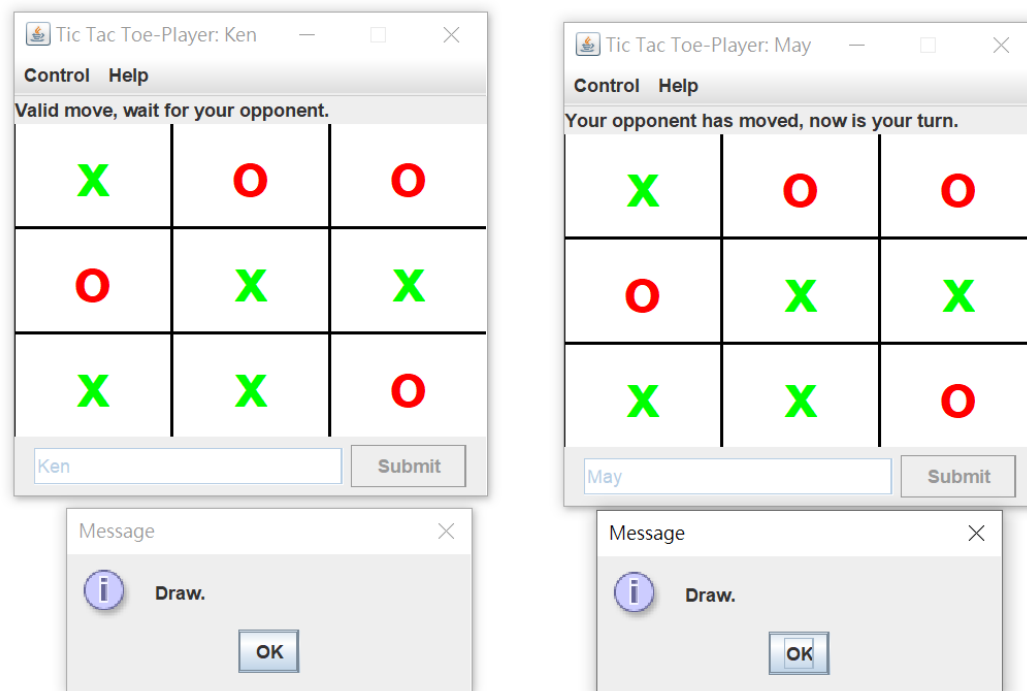

Fig. 5

Fig. 6

Fig. 7

In addition to the above, you also need to handle the case when one of the players left during the game process. In this case, the game would terminate immediately (i.e., the remaining player cannot make further move), a Message Dialog "Game Ends. One of the players left." would be displayed on the remaining player, as shown below in Fig. 8.

Fig. 8

Besides, as you can see in the screen captures, there is a JMenuBar which consists of 2 JMenu, named Control and Help (located above the message title). In the JMenu of Control, it consists of a JMenuItem, named Exit while in the JMenu of Help, it consists of a JMenuItem, named Instruction (Fig. 9 & 10). When the player clicks "Exit", he/she would exit from the game and the game would be terminated immediately. When the player clicks "Instruction", a Dialog Frame consists of some game information would be displayed (Fig. 11).
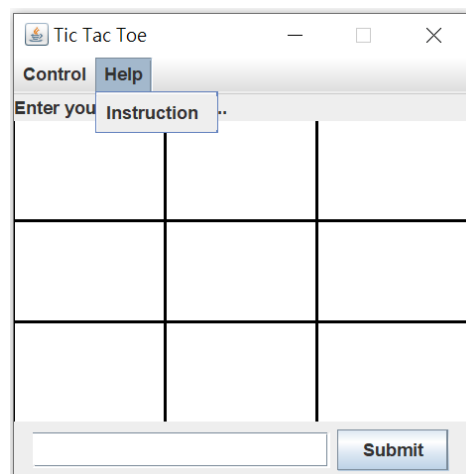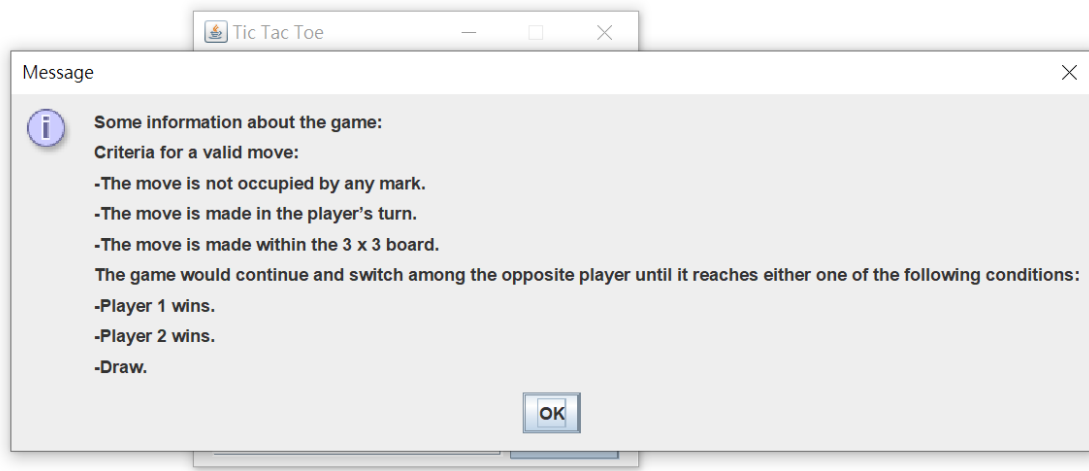

Fig. 9

Fig. 10

Fig. 11

1. In this assignment, you can assume both the server program and the client programs run on local host (i.e., with IP address 127.0.0.1).

2. You can have your own design, but you **must** include the GUI components as shown in the above screen captures and all the functionalities described in this document should be implemented. To ensure your program has implemented all necessary functions, please refer to the marking scheme below as for your references.

3. **This assignment will be marked by features (Your code would not be investigated).** You are required to write JavaDoc for all non-private classes and non-private class member functions. Programs without JavaDoc will lead to mark deduction. However, you don't need to generate JavaDoc htmls. Just write comment blocks in your source program.

4. **You are required to include a readme.txt file which clearly describes how to start and run your program. If you do not include a readme.txt file to describe how to start and run your program, your assignment will NOT be marked. If following the instructions written in the readme.txt file still lead to failure to start and run your program, your assignment will also NOT be marked.**

5. Please submit all files in a single compressed file (in .zip or .7z) to Moodle. Late submission is **NOT** allowed. Do **NOT** submit .class files.

Marking Scheme:

Correct implementation of GUI components: - 1 JMenuBar which consists of 2 JMenu which each JMenu consists of its corresponding JMenuItem (5 marks) - 1 message title (2 marks) - 1 3 x 3 tic-tac-toe board (5 marks) - 1 textbox for player's entering his/her name (2 marks) - 1 submit button for submitting the player's name (2 marks)	Total 16 marks
Correct functionality of the game: - Implementation of restricting players to make their move before they submit their names (5 marks) - Implementation of restricting players to enter and submit their names more than ONCE (5 marks) - Implementation of updating the frame title after players submit their names (5 marks) - Implementation of correct message title after players submit their names and make a valid move (5 marks each, total 10 marks) - Implementation of correct switching between players after a player makes a valid move (5 marks) - Implementation of the game is started by Player 1's move (5 marks) - Implementation of display player's mark on the board (for both players) when one of the players makes a	Total 74 marks

valid move (5 marks) - Implementation of NOT display player's mark on the board (for both players) if the player makes an invalid move (5 marks) - Implementation of the 3 conditions: Player 1 wins, Player 2 wins and Draw (5 marks each, total 15 marks) - Implementation of the case when one of the players left during the game process (5 marks) - Implementation of the functionality of Help (5 marks) - Implementation of the functionality of Exit (4 marks)	
JavaDoc	Total 10 marks