**Assignment 4 - Tic Tac Toe**

Alex Parsons, [Dan Hallanger](mailto:halladan@umich.edu)

CIS 296: Java Programming

Professor John Baugh

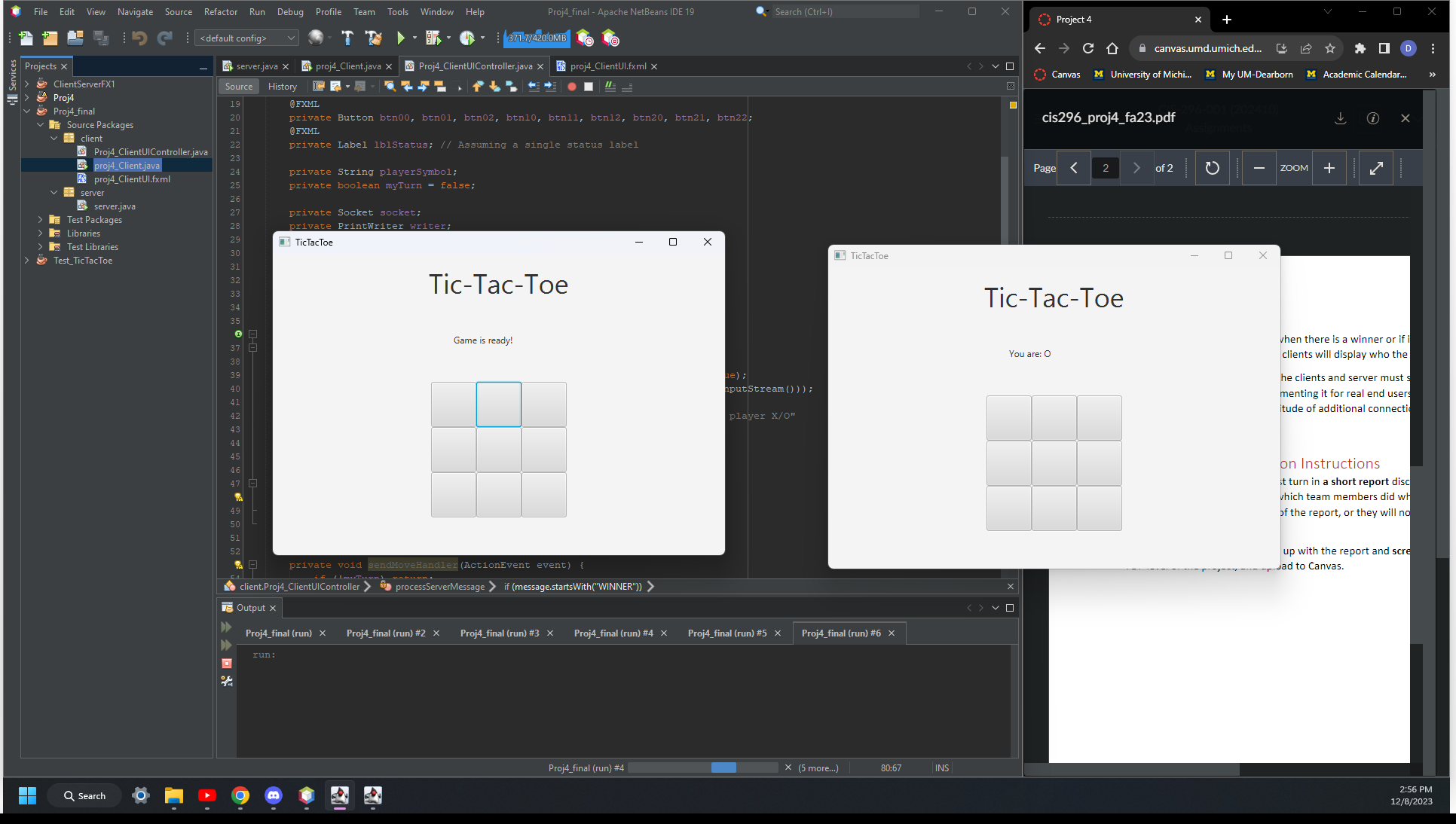
December 10, 2023

**Proj4\_Client.javat**

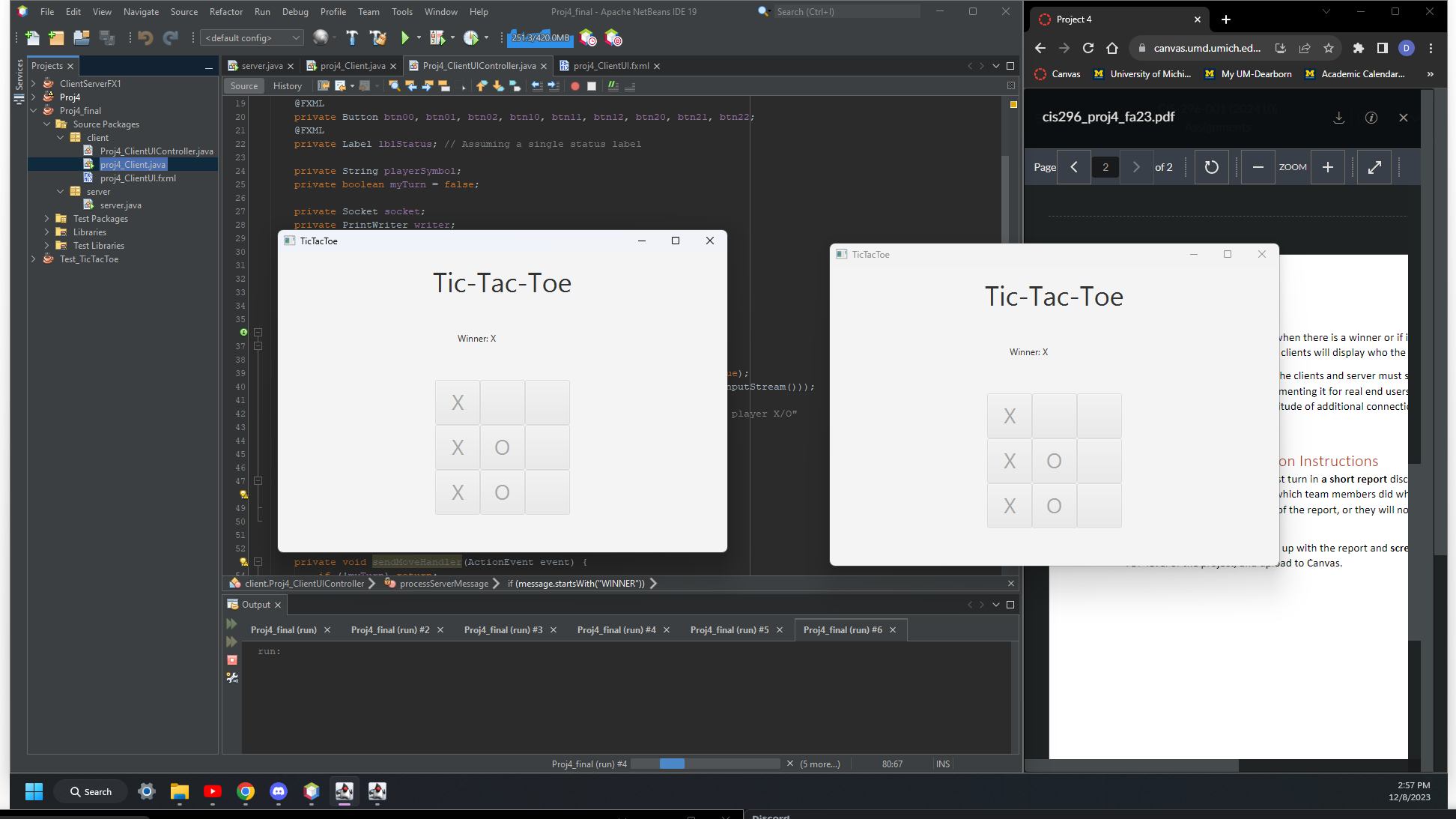
Standard JavaFX launcher for the GUI and scene builder document.

**Proj4\_ClientUI.fxml - Screenshots**

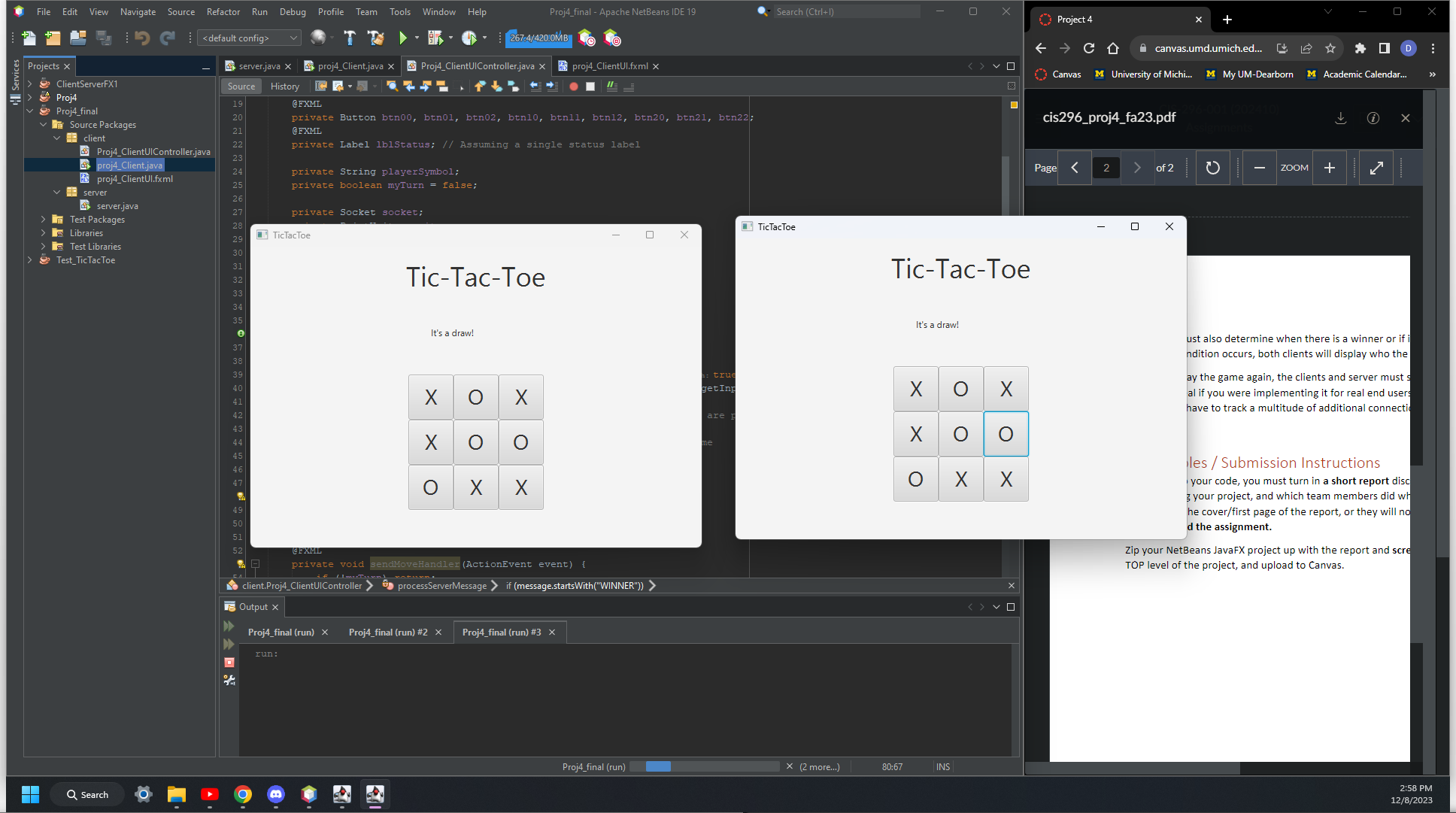
Clients running and connected to each other



Client that is assigned X wins



Draw with both clients



**Proj4\_ClientUIController.java**

sendMoveHandler will detect the button clicked and send it over to the server, needs to send the button ID and the player symbol.

receiveMessages will get a message from the client and process it later. This message will be used to keep track of game status.

processServerMessage takes that message sent by the server and checks if it says winner or draw then updates the label we used. If not it will detect the button changed and update the button with the correct player symbol. Get button switch is used when grabbing the switch id for processing server messages.

DisableAllButtons is called when there is a winner, making it so they can’t keep clicking cause the game is over.

**server.java**

Using the sample for a GUI and command line application, we connected two clients using localhost and a port number. We only allow two clients and update them when the game is ready.

To keep track of the game we initialize a multidimensional array to hold player moves so that way we can check to see if there is a winner in the future.

GameClientHandler is from his chat example. We just modified it to be used for our game.

processMove is used to parse a message from the client and update our gameboard, and then broadcast it to all players. That way they can update the buttons with player moves.

checkForWinner we want to check the gameboard to make sure that there is a winner, we check rows, columns, and diagonals for a winner.

broadcastDraw/Winner/Move is all used to send messages to the client to update the status of the game.

**Team Member Contribution:**

Alex Parsons - worked on game logic implementation, and bug fixes with fxml issues. I did a little with the server sending/receiving messages from clients.

Dan Hallanger - worked on the tie game logic, initial setup of files and some of the server/client implementation