

Matthew Hellie

CIS 296 Project 4: Tic-Tac-Toe

Design Process

I initially started by using the framework from the 'MultipleClients' example provided on Canvas under Files>Networking. I then got rid of all the JavaFX GUI code from the server to makes things easier. After that I built it up in small bits throughout a few days. I made sure the server does most of the logic. Here's a brief explanation of how some important parts work:

Connecting to the server:

The server must be launched first and it then waits for 2 clients to be simultaneously connected. Once there's 2 players, the game starts and no more clients can connect to the server. It doesn't let player 1 do any moves until player 2 joins.

Keeping track of turns:

The server informs each client what their ID is based on the order in which they connect. If the ID is 1, then that player starts first and the 'isTurn' Boolean is set to true at the start. After that, each time an empty square is clicked, I used `isTurn = !isTurn;` which makes isTurn the opposite of whatever it was previously. If a player clicks a square when its not their turn, a message is displayed only to them.

How the client knows what to display:

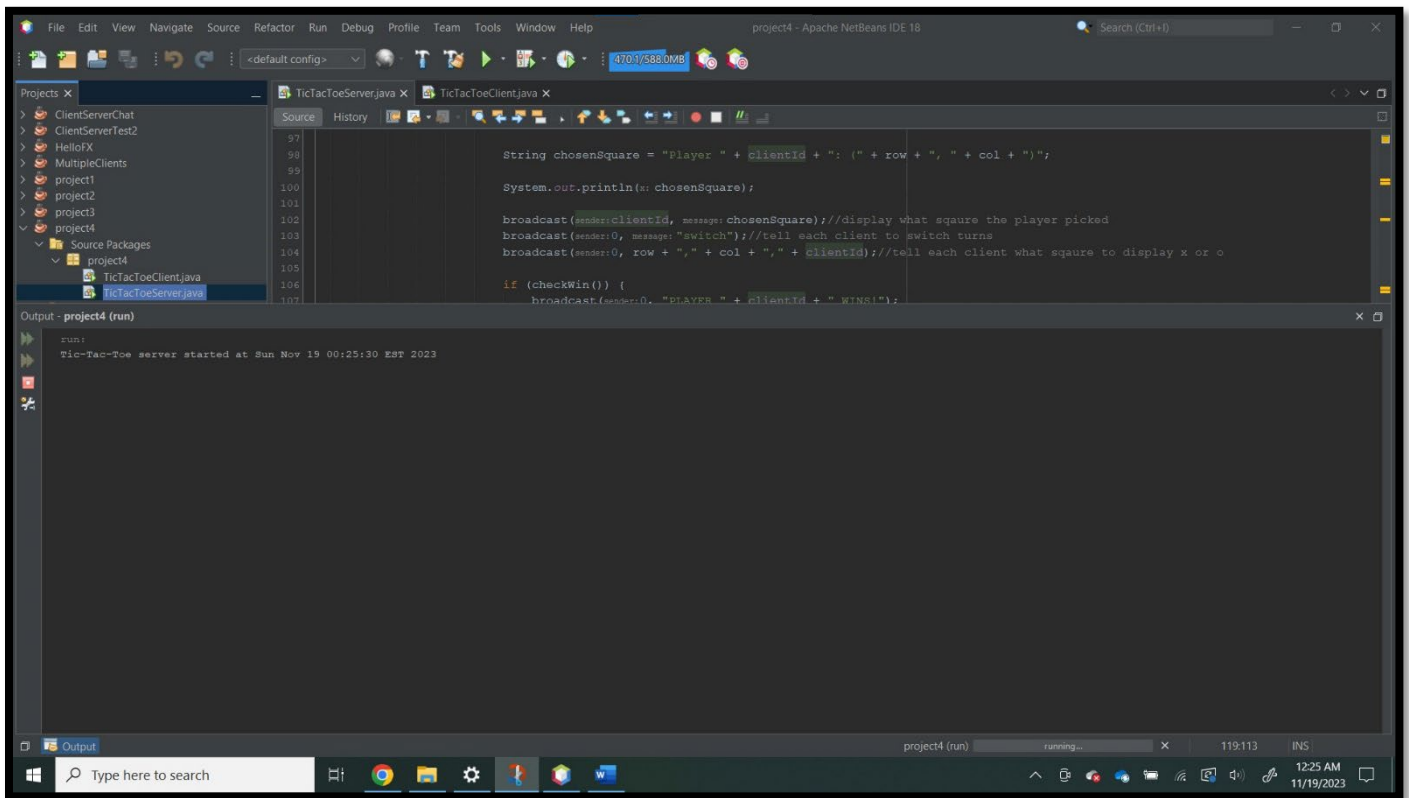
If an empty square is chosen, the client broadcasts to all players the row, column, and ClientID of the player who chose it. The client uses the row and column to know which square to change, and the clientID to know to display an 'X' or 'O'. The server also doesn't let players choose an already filled square and sends a message only to the player who clicked it. Also, when a player makes a valid move, the server sends a message only to the other player of what square they chose.

Detecting a win or tie:

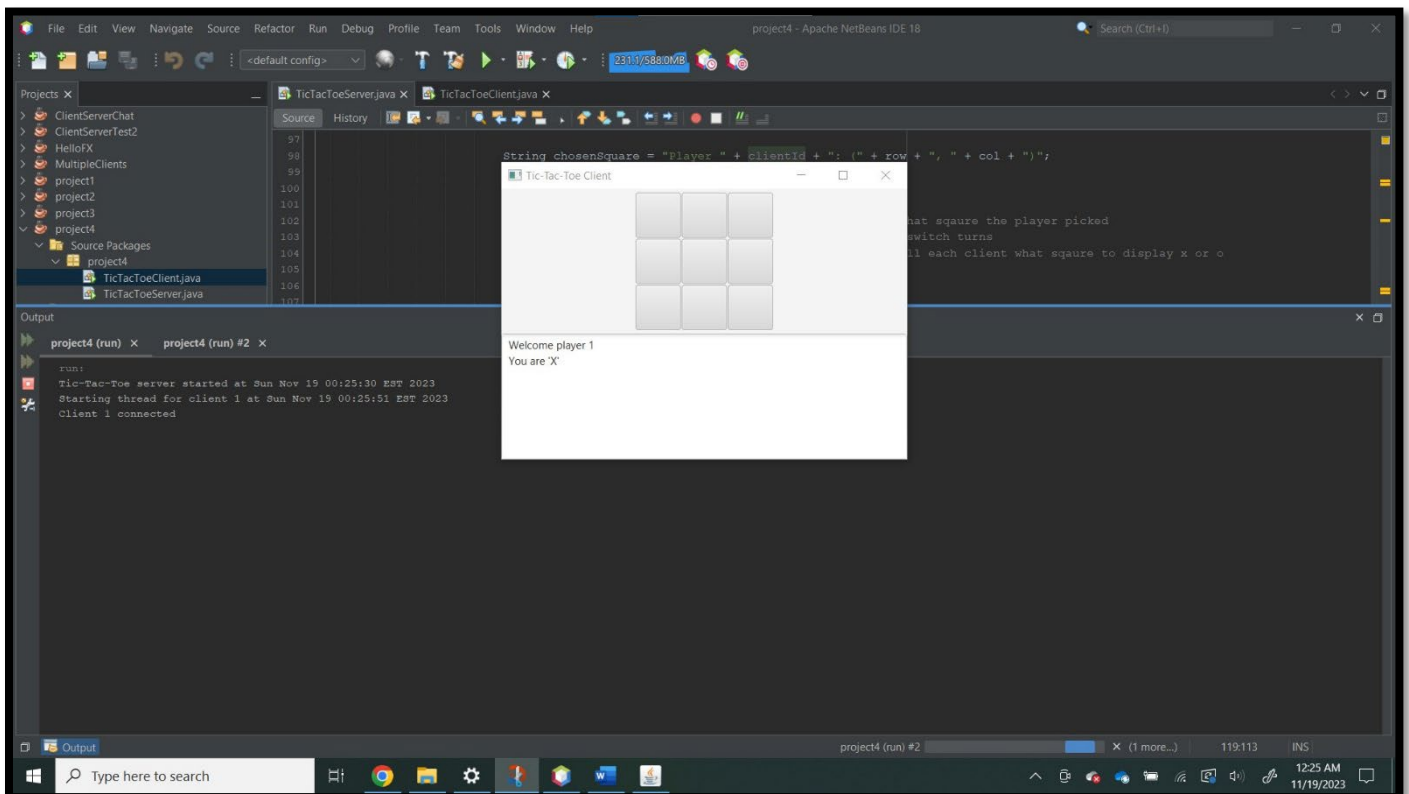
A 3x3 array of integers is initialized to all 0's at the start. Every time a valid move is made by a player, that square value is set to the clients ID. There are then 2 functions for a win or a tie that return true or false if one is detected. Once the game is over, the server broadcasts a message to the clients which then causes the buttons to be disabled. The server and sockets are closed the players must re-launch to play again.

Screenshots

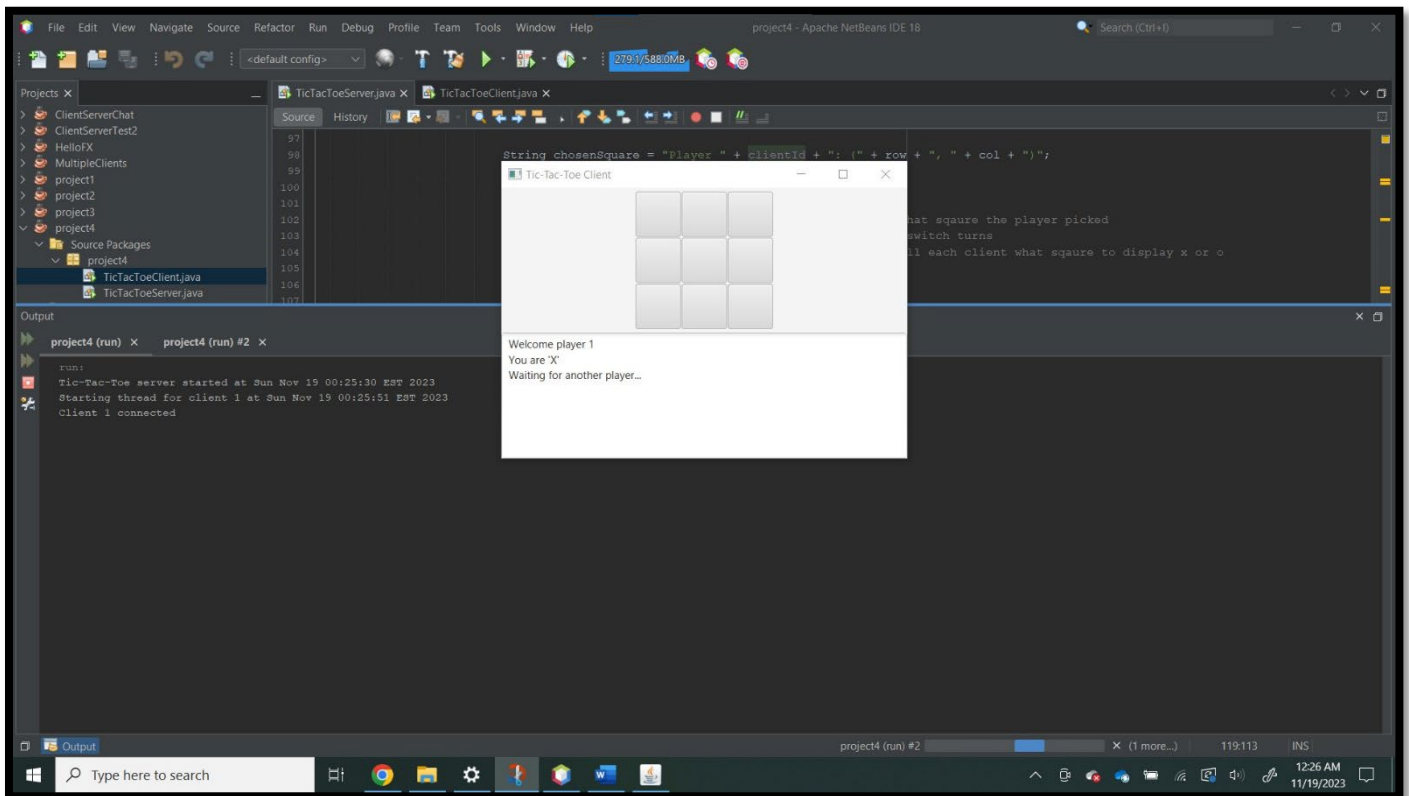
Server starts:



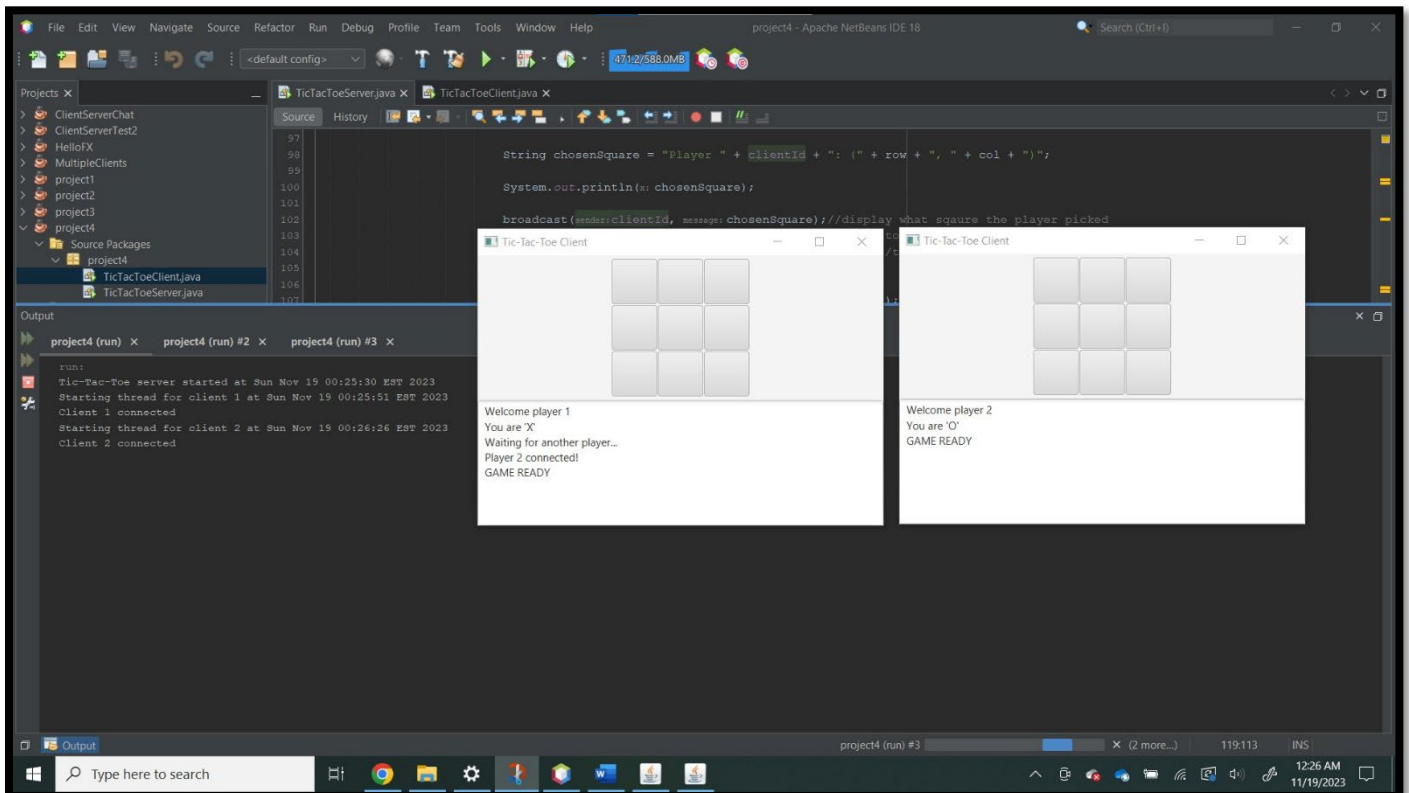
Player 1 connects:



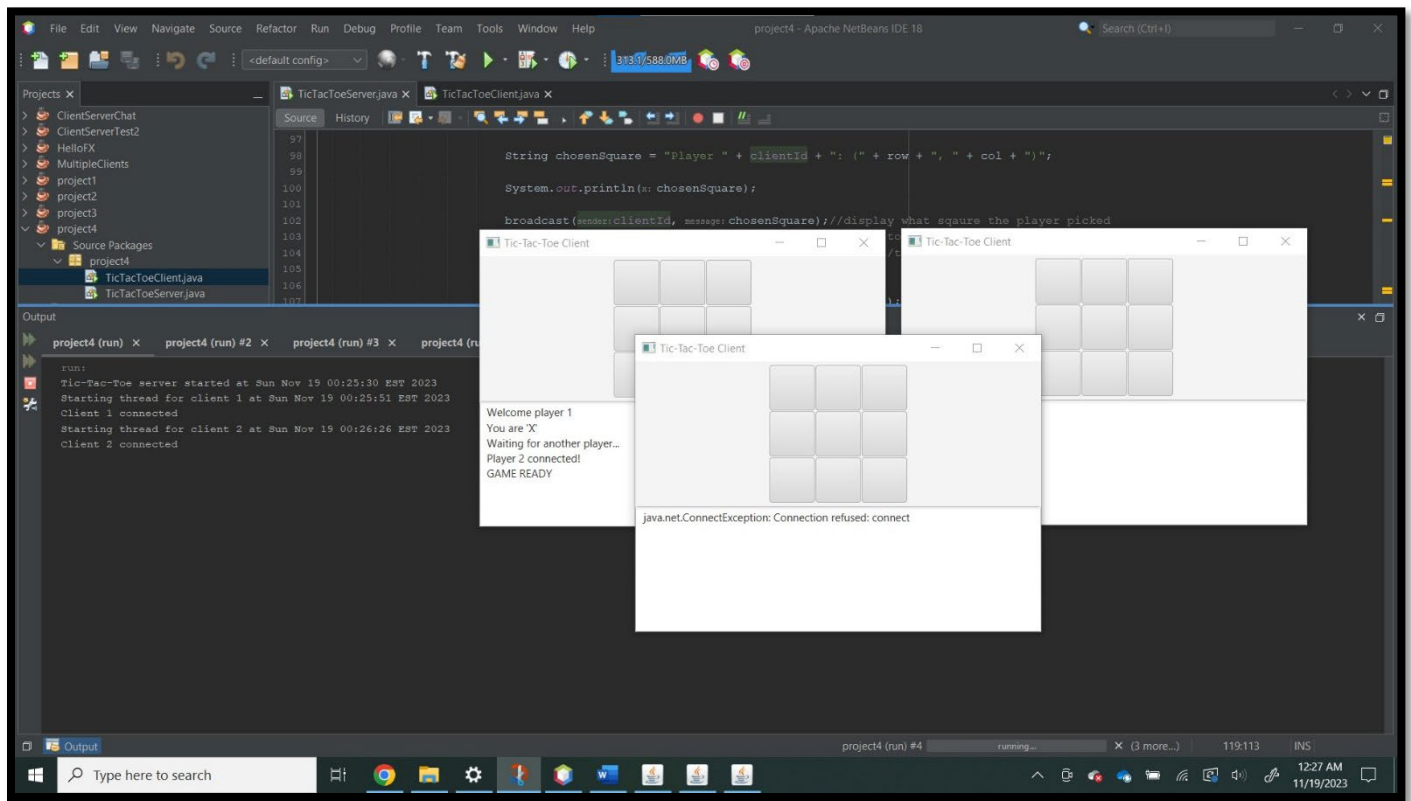
Player 1 tries to click a button before player 2 joins:



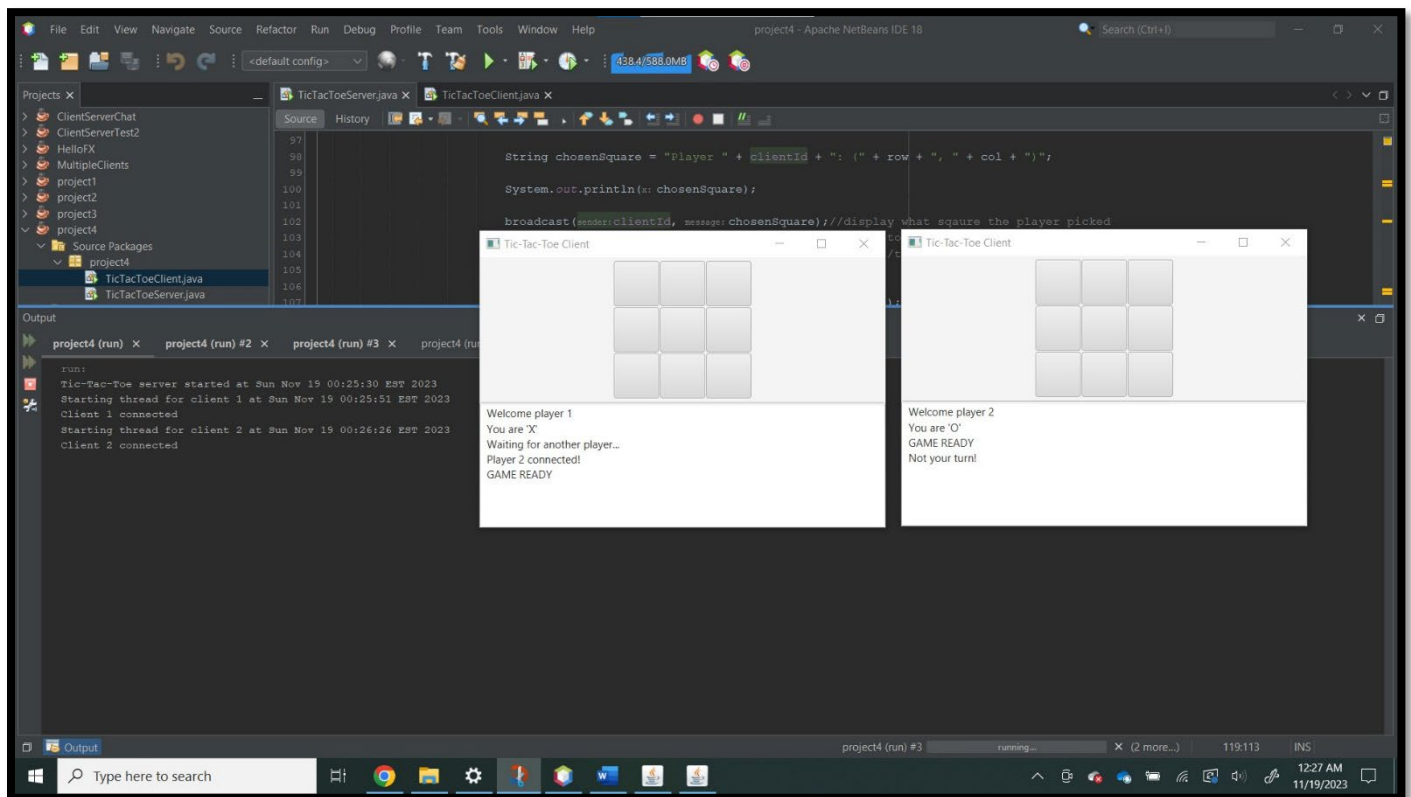
Player 2 joins:



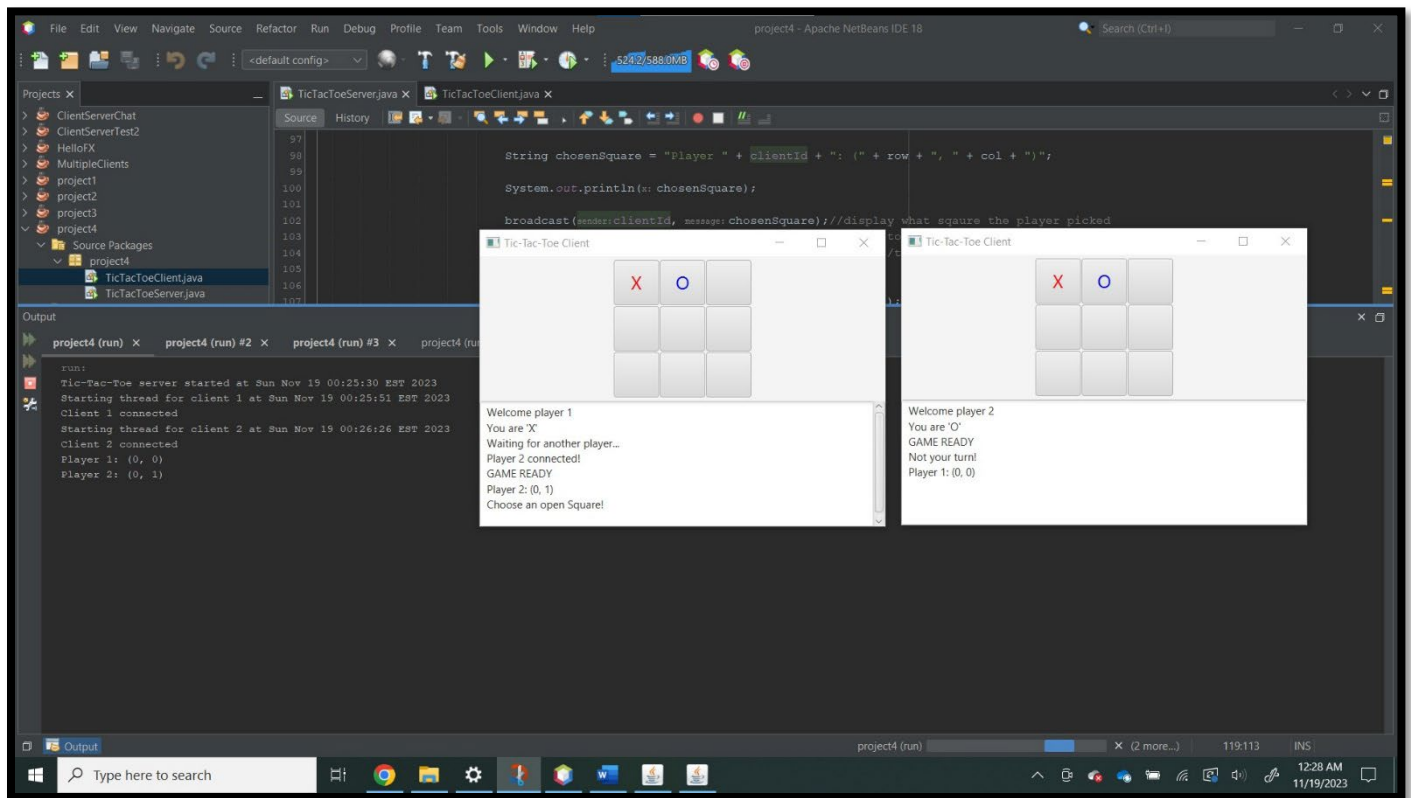
3rd client can't join:



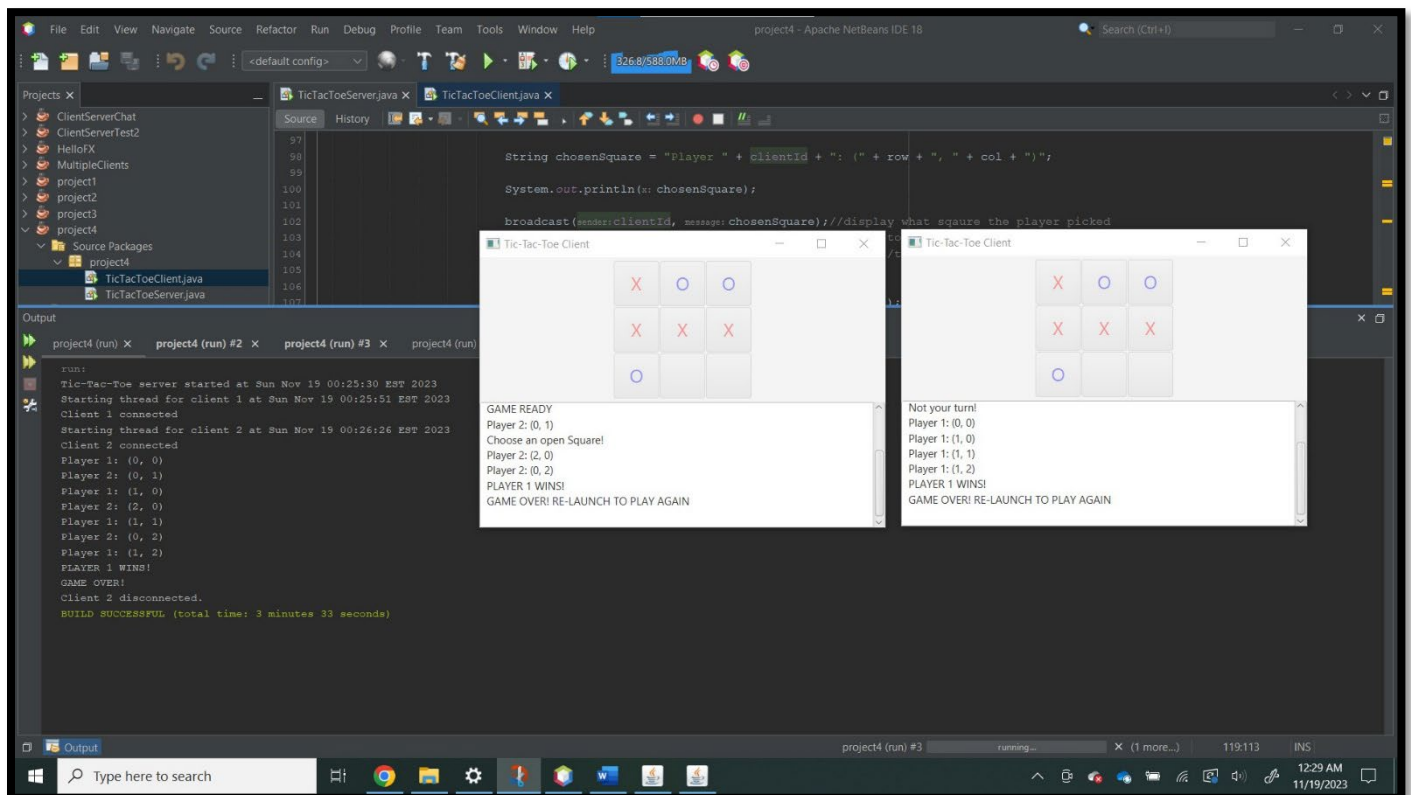
Player 2 tries to play when its not their turn:



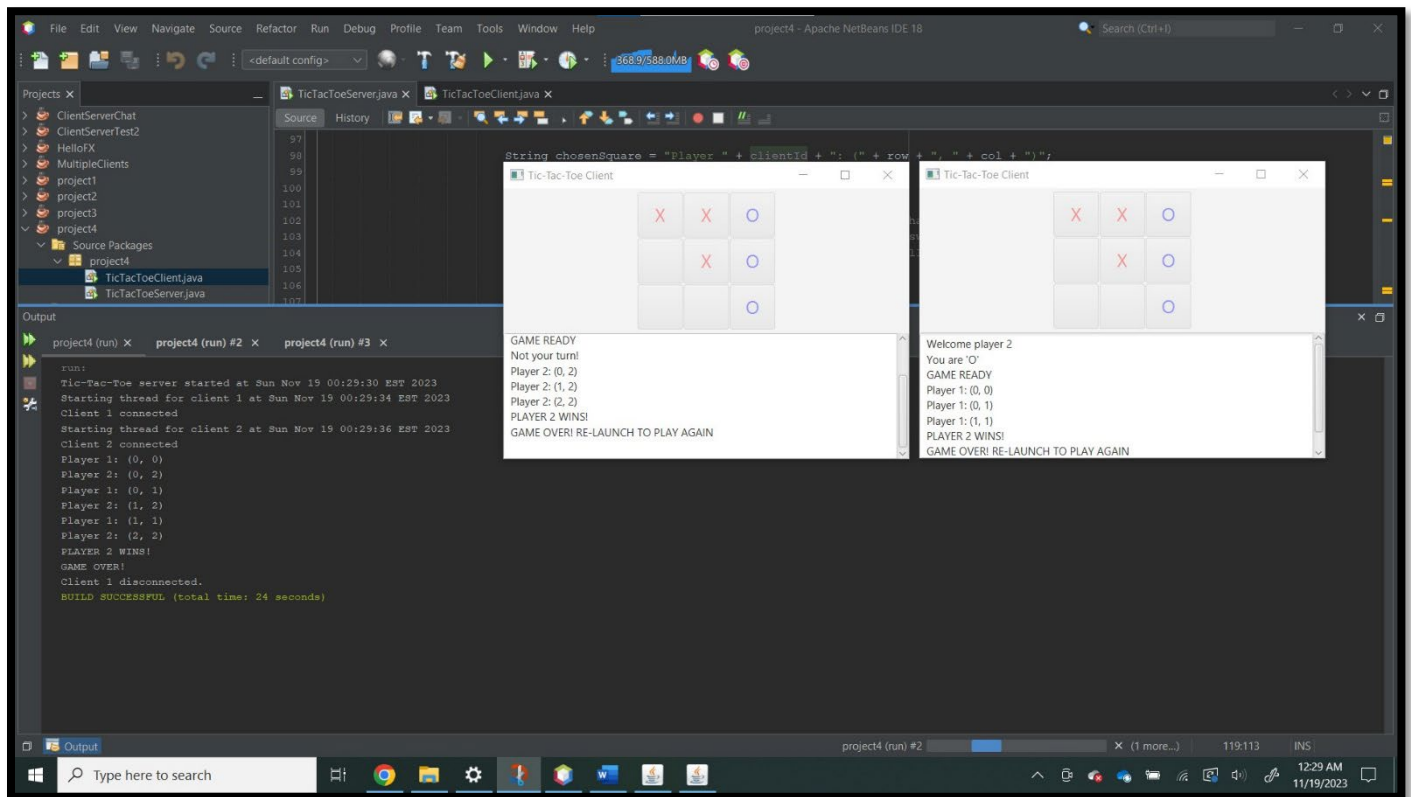
Player 1 clicks on an already filled square:



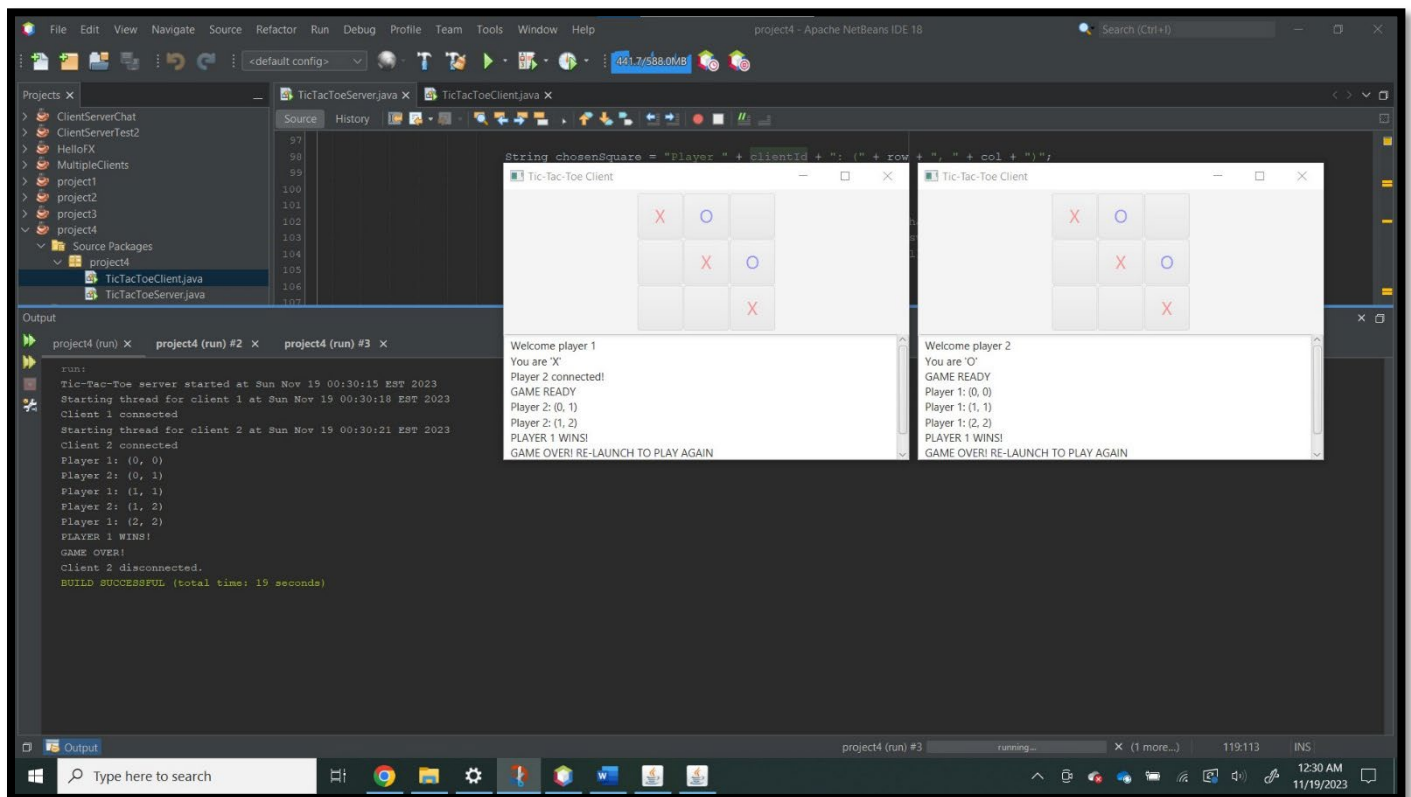
Horizontal win:



Vertical win:



Diagonal win:



Draw:

