Checkers Game by Alex Gorman and Yi Luan.

My partner and I decided to implement the Checkers game in Java to play over the same network using TCP connection. We used Java 17, and it runs on the Computer Science Department Linux machines. Specifically for our application, we have a Checkers game that you can play between two different people on two different computers in the Spinks Lab, by putting the correct IP address and port number and the users can play checkers and send messages to each other.

For our testing we incrementally built our application. We started small by just trying to implement each menu at the beginning and creating a simple TCP connection. Then when we succeeded in creating the TCP connection, we moved on to the game implementation. We started small by just trying to make simple moves and having our gameboard register these moves and then properly send them to the other player. We just kept incrementally adding small functionality to the game until we could implement every move correctly and properly determine the winner and keep track of score. We also did this for the chatbox, so no matter whose turn, you can keep sending messages.

Overall I would say our project was a success. We fully implemented the checkers game. We learned about using TCP connection in a different language and the obstacles and differences that come with this compared to using C. Our original goal was to have another (3rd or many more users) be able to watch the Checkers game being played, but we ran out of time since the game was harder to implement than we previously imagined.

Alex designed and implemented the entire Checkers game and MVC classes, so the players could obviously play Checkers against each other, and determine when one player won and the other lost and update their respective scores. Creating the game included making the game board, creating the pieces, implementing how they looked, determining correct moves and invalid moves, implementing the three scenarios including single move, single jump and multiple jump in all three cases (not a king, becoming a king and while a king).

Leo designed and implemented the chatting function and dealt with message sending between players by using a TCP socket, and made it able to distinguish if that message is for chatting or gaming. By using a thread, to avoid the potential chat blocking issue that could be caused by game turn. Also makes most of the layout and views besides the checker game core, including, score board, Menu. Also make the application be able to reconnect when one player exits the application or quit the game (back to Menu).