

## Future Networking

The current game which is developed by our team does not include game networking (online multiplayer), however the current game has the necessary code to work with for the development of networking in the future. We have multiplayer mode (up to 4 players), but the game must be played in a single system. We will still need to make few changes to existing classes and add new code to implement networking, but we strived to design our application in such a way that we could implement networking to the application without making major changes to the original design or functioning of the game.

For future networking we will use client-server architecture. While playing online multiplayer game every user or player connect to the host server where all the data is stored temporarily or permanently. Currently, we store all the data in the system itself but in networking whenever user makes a move, host server should automatically capture the user action and update it to the client and of all the other players as that are connected to the server. Currently, for the control we start with player one and for each move or pass we go to the next player. We did this by initializing a variable ( $i = 0$ ) and increasing it by 1 every time. We use the logic  $((i\%4) + 1)$  to determine which players turn it is. For the control in online game, server uses an algorithm where it starts the control with player 1 and disable control for other players until player 1 has moved the pawn or passed the control, then server hand over the control to the next player until someone wins the game or the host ends the game. Client application will be responsible for displaying or updating the GUI and receiving the information from host server.

Future networking becomes easier because we have used java to develop the application as Java provides us network support. The player who initiates (host) will have a slightly different interface with few extra options (end game, etc.), but there will not be any major changes necessary for the interface.

The above-mentioned approach is just theoretical but there may arise complications as when we start working on Networking, but the application should not require any major changes to the current code, gameplay and running.