Focus Networking

For our purposes we are going to implement the client-server network architecture. Essentially where a server is a dedicated host and is authoritative about everything gameplay related (game rules, input processing, etc.), this could be the very first player that joins, or each individual can choose to either host or create a game for others (clients) to join. So, everyone will initially start with the same interface, and depending on whether they want to be a server host or a client who wants to join a game it would look different. The game of Focus is going to act as an interface to detect input and a handler is going to make sure all clients connected to the server receive information that their action/move has taken place and not only the client that makes the move.

We currently store the current game state using variables and arrays (player array, turn variable, offboard class with arrays of captured/reserve pieces). The information could be easily to another computer via packets in an online game scenario. The client receives present board state from server and generates it, the client also receives input from player and sends it to server for processing. It is unnecessary to send all information to all computers as this could lead to possible bugs/errors in gameplay. Depending on the player, they will get specific information that is specific to that players gameplay.

Our turn variable currently controls who's turn it is, starting with player 1 and so on in a loop till game is over. On a network of computer, the host can be player 1 and depending on the order that the players join the host they will player 2, player 3, and so on, the turn will be passed numerically. We can use error handling to make sure the game does not start with less than 4 players (as we already have). The game can start once all 4 players have joined and are ready, and the game progresses normally, and once a player has dominated most pieces, the game ends and that player wins.

A shared memory can be implemented for proper information sharing. Shared memory is memory that can be accessed by multiple programs with aims to provide communication between them. Can be implemented by creating an area in RAM which other processes can access.

Latency would have been an issue, however, the fact the game is turn-based means that lag and packet loss would not be a big of an issue as it would be in a more dynamic game which would need real-time decision making (like a first person shooting game).