Multiplayer online games typically work via a **Client - Server** connection. 2 or more players connect to a centralized server which acts as the governor of the actions the players are trying to accomplish. A player will request to perform an action and the server will do the actual performing of the action, and update accordingly for the other players. Due to the small nature of our game, players will likely connect to each other via a P2P (Peer-to-Peer) connection, where players connect directly to a host's computer. Changes to implementation start in the game setup, which will act as our player lobby.

Many P2P implementations work similarly to a Client-Server relationship, with the host acting as a server for connecting players, as well as a player themself. Therefore all changes made by a connecting player will be sent to the host, and then reflected back to the other players. In the game setup, the same UI can be used, but some things will have to be changed to incorporate multiplayer play. Options such as computer difficulty, start game etc will be available to the host player only, while connecting players are allowed to change their names and color, prompted the color isn't already chosen by another player. But how will a connecting player know which colors have already been chosen, as well as other information that must be passed between players?

Once a player has established a P2P connection with the host, packets containing information can be sent back and forth between the host and other players. During the game, many bits of information must be exchanged whenever a turn is happening, which means such bits must be saved so that they can be sent. Whenever a turn is completed, the aspects of the game that must be saved are the aspects that affect everyone in the game, such as the position of the placed stone (or moved stone in a special combination), the updated leaderboard based on the newly placed stone, and the number of remaining stones for the player who played the turn. A simple approach would be to call some sort of method to collect all of this information before the player's turn ends, and send it to the host. In addition to that, a way must be established to determine which player has control over the program at a certain time.

Our game already has a function for determining who's turn it is, which could be used for passing control between computers. Typically offline a turn is passed between players automatically when a stone is placed, but this would likely have to change when playing with multiple players. The most effective way to implement this would be leaving the turn passing to the host exclusively. This way, once the host receives information from a player describing the turn they performed, the host can send information to all connected players about who's turn it is. Since the host informs every player whose turn it is, we can prevent unwanted actions from other players by blocking inputs on their GUI, as well as ignoring any information sent from them. This way, we can effectively pass control from player to player as the game progresses.