# Networking

Multiplayer online (MMO) are games that allow multiple players to participate in a virtual game world simultaneously. The basic structure of an MMO involves a central server that hosts the game world and allows multiple players to connect to it and interact with each other in real-time.

To facilitate multiplayer online we'd need to Implement a network protocol to enable players to connect to each other and exchange game data, modify the game logic to handle multiple players. This includes handling turn-taking, player actions, and game state synchronization between players. Adding networking code to our game to handle communication between players. This includes code for creating and accepting connections, sending, and receiving data, and handling errors. The user interface will be needed to be updated to show the state of the game for all players, including their current score, which player is currently taking their turn, and any other relevant information.

Additionally, to enable multiplayer functionality, we will need to modify our game to store game data on a server that all players can access. This server can act as a central repository for game state information, allowing players to connect and synchronize their game data in real-time. For this we will need to modify our game's architecture to support player-server communication. This may involve creating a custom network protocol or using an existing one like TCP or UDP to transmit game data between players and the server.

In a multiplayer game, you generally want to send only the information necessary for each player to make decisions and see the current state of the game. To accomplish this, we will use a technique called state synchronization. In this approach, each player maintains a local copy of the game state and periodically synchronizes it with the server or other players as needed. The server acts as the source of truth for the game state, and players receive updates only when the state changes. For example, if a player moves their piece on the board, we will only need to send the new position of the piece and not the entire board state. This minimizes the amount of data that needs to be sent and ensures that players have an accurate and up-to-date view of the game state.

Controlling whose turn, it is in a multiplayer game can be a complex task, so using a turn management method like Centralized turn management is generally easy to execute for turn management in a multiplayer game, especially since the game has complex rules or interactions between players. The server manages the turn order and notifies players when it is their turn. Players send requests to the server to perform actions during their turn, and the server updates the game state and notifies other players of the changes. This approach ensures that turn-taking is synchronized between players and that each player has a clear indication of when it is their turn.