A page (at most) explaining how your design accounts for - or would need to be adjusted for - possible future networking of the game

1. **Starting the game:**
    a. There could be an initial player who took the role of the **host**.
    b. Then all the other players would be **clients** that connect to the server.
    c. The clients would need an IP address and port number of the host to make a connection with them.
    d. A lobby and server browser system would be needed to easily get all the players into the same room.
2. **Updating the board for all clients:** After every turn, the host calls *updateAllClients()* to update the state of the board for all the players connected to the host.
3. **Decide who will take the next turn:** There is a function called *nextTurnBegin()* that is called by the host at the beginning of every turn, and figures out what player is going to take a turn next. If the next player to take a turn is not the host or the AI, then instead of running *playerTurn()* or *aiTurn()*, it would run *networkTurn()*.
4. **Telling a client to take their turn:** The function *networkTurn()* will send a message from the host to the client taking the next turn to make their selection. The host waits for a response from the client.
5. **Updating game after client has taken turn:** Once the response is received, the host runs *updateAllClients()* and *nextTurnBegin()* again.

It would be possible, with some work, and adding a new type of turn for networked players.