Networking was something we initially planned in consideration of. It is almost certain that if our program was expanded to a networked version, we would separate out the host from the clients to greatly ease numerous difficulties with keeping information synchronous.

Other than for hosting and connecting, configuring ports and so forth, which would require a whole 0^{th} step with its own UI, we believe our UI is robust enough to remain visually identical, though in a networked version there would be logic for telling the host when a client changes a setting. During setup we would support anyone making settings alterations, except for names, which would be each player's choice. No Kick/Bans or other such feature is assumed to exist after the initial connection phase, which will allow the UI to remain intact in most respects.

Currently, we make use of an ordered list of Token objects (pretty much what defines our Stack class) for each tile, and transmitting that to and from players should be trivial. In a networked game, we could simply send an updated list of contents for any tiles changed by a players actions, as we use a coordinate system for tracking stacks as well. This is sufficient information, and is much cheaper data-wise than resending the entire contents of the board after every move, and would still be completely synchronized.

We have the Turn class, which is told by the Move class to change the turn over to that of the next player after each and every move. This would likely change so that only the Host is authoritative about whose turn it is, and every other player's program queries the Host instead.

For controlling starting and ending a game, we have an open issue about having CPU Players take over for players even in a non-networked version. In all likelihood, we would modify our program to simply explicitly tell the Host when it's exiting that it is doing so, and have the Host replace the player with a CPU Player, allowing the others to continue. Starting was mostly discussed at the beginning of this document. For both starting and ending in the context of victory and defeat, the Host would act as an authority on both, and inform its clients, though we only consider someone defeated when the game is over, or all their pieces are captured.