Requirements: Explain in one page or less how our project accounts for or can be adapted for use for networking the game. Page can be text only, or text and diagrams.

Our project does not account for networking; however, it is quite plausible to adapt it for network use in the future.

There are ways to create a server based on a personal computer using Java, most easily by creating a class to represent the server with the main method acting as the constructor. This constructor method would create a server socket using the host computer's IP and a port and set a number of threads for up to 4 potential players. Additionally, the host computer would act as a receiver monitoring each player's thread for updates to the game and joining them before distributing them to all players.

The other computers would connect to the computer-as-server using the IP and port information given to each player wishing to connect. This information could be given out manually to players wishing to play.

The way we would transfer our game data between computers would be by using a modified version of our SaveGame class. We would save the current status of the game after each player's turn creating a SavedState. This could be sent to the connected computers on the network as a string in the same format that the SavedState class currently saves the game state. The networked computers could then use a modified version of our LoadGame class to upload the string containing the most recent saved state.

By loading a new version of the game after each player's turn, all users would have the most recent state of the game when their turn comes around. We would not attempt to show the screen as "live" as in displaying each motion of other players since there's a possibility that sending live updates would cause a significant performance decrease. The update after each piece placement would also allow players to plan their move without being observed by their opponents.