

## Comp 2005 Group 3 Iteration 4 – Possible Future Networking

In the gaming industry, networking can be a huge and vital part to any game and the measure of its success. Now while our groups implementation of a game of Blokus is not as high end as big games on the market, it still brings out the question as to how we could provide a multiplayer aspect to it. While our Blokus doesn't currently have multiplayer functions or was designed to cater to networking right away, it is possible to transform it into a functioning multiplayer game. The first step that allows us to make this possible is that we are coding in a Java environment. While Java may not be the best to use as a game programming language, it allows for relatively easy network support. If we were tasked with providing networking, we would use sockets to do so. Sockets allow for multiple clients to connect to a host server and transmit data over it. Therefore, at the very least, we would have to implement a host server socket on which the data was controlled and which facilitated that connections and actions of the game, and a client server socket which would be used for the players to connect to said host server to play the game. Other than implementing these sockets, our design wouldn't need too much change. The game is already implemented to be played locally so the groundwork for the game is functioning. Some adjustments that may have to be made however is the starting screen where the number of players is chosen. The number of human players may have to be implemented in the host socket server to allow for the correct number of clients to join the game. Right now, since the game is all managed locally, it doesn't really make a difference considering the number of players. However, when networking is considered, you don't want to start a game with the incorrect number of desired client players. For the most part, the game will work the same across a network as it does locally. We would just use variables giving the clients who's turn it is control of the board while deactivating the buttons for each other client in the game that isn't currently taking a turn. The client's action will then be sent to the host server where the action would be made visible to all clients and the client who's turn it is next, will have their board available. The changes that need to be made within the code or mostly minor fixings that will allow for the proper playing of a turn-based game across a network. Lastly, if a client were to leave a game during the middle of it, it would be smart to adjust our code so that an AI player takes their place instead of ending the game completely.

