Networking Design

Programming in java helps us provide easy network support. So, for our game networking, we would choose to implement using client-server architecture which is the standard as it is easier to execute. However, it requires low network/data bandwidth. So we would use socket which would allow multiple players to connect to a host server and that would allow data transmission over the network. So Based on our code design we would have to implement a host server through which the data is managed and it will be responsible for connection facilitation and the game setup. While the client-server will be responsible for displaying the GUI and will be used by the player to connect the host and start the game. The game state will be working and updated via networking routes between computers over the server to send updates and played turns; there will be no need to send all information to all players as long as each player receives the information they need that should be good to sustain efficiency and response of the server.

However, the design for Ricochet that we have created and the online networking version that we wish to create, there are a few differences. As for parts of the game, it would function in the same way. One of the differences would be defining a variable or an alert which informs the player(s) from the clients-servers end on who's turn it is while also disabling other players (s) ability to take turn or use any functionality of the game unless it's the player turn. It will be client-servers responsibility to inform the host server when to enable other player turns.