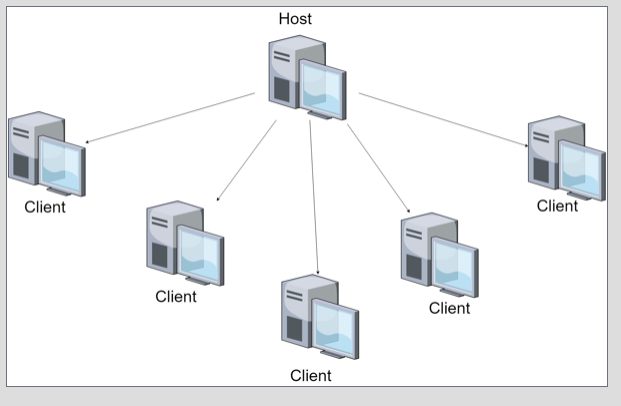
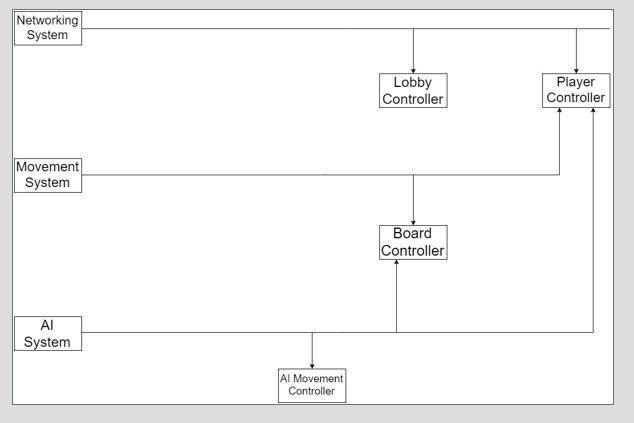
Context

There will be up to 5 users connected to a host. These will be the network resources. There will be other software systems or permanent files.



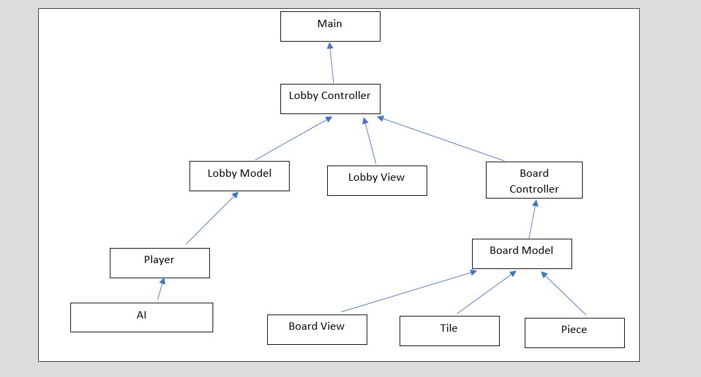
System Architecture

The networking system has a lobby controller which coordinates everything happening in the lobby, as well as a player controller to coordinate player turns. The movement system has a board controller to coordinate where the various pieces are located on the board, as well as a player controller to coordinate the players piece positions. The AI system has an AI movement controller to specifically handle AI movement, a board controller to calculate the best board position to move any given piece to, and a player controller to calculate to calculate how to deal with the pieces of other players.



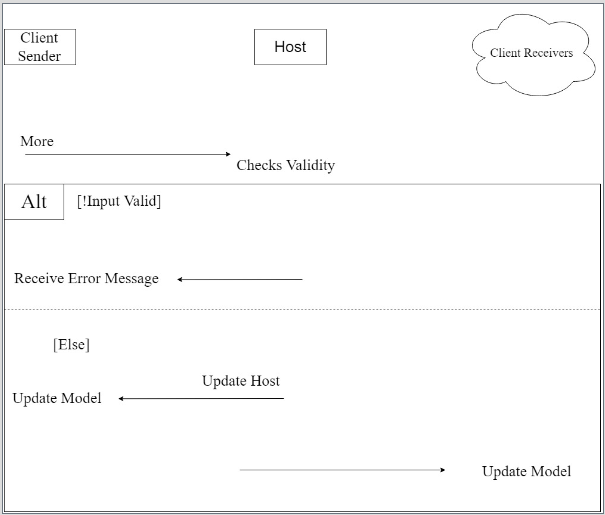
Class Diagram

Used Design Pattern: Singleton Pattern will be used for our Lobby Controller class, because we always want the same instance of the lobby to be used.

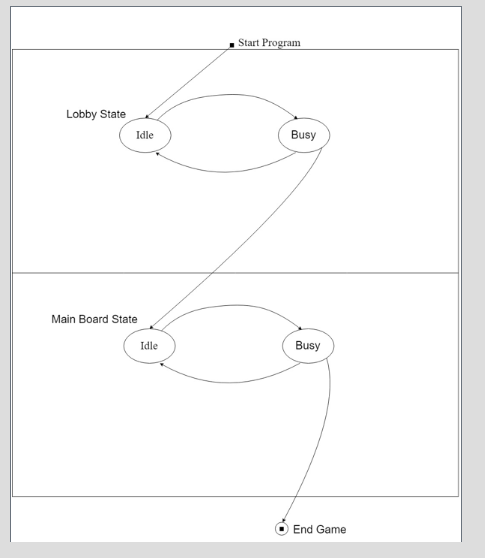


Design Models

Sequence Diagram – Everytime a player attempts to make a move it is sent to the host and checked to see if it is valid. If the move is invalid then the player who attempted the move will receive an error message. If the move is valid the host will update and the update will be sent to the other players.



State Diagram – Shows how the entire program will run from the beginning of the program to end of the game, and the states the program will be in throughout.



Timeline

October End of Week 2:

Rework our prototype to lay it over a picture of a Chinese Checkers board

Add menu screen

October End of Week 3:

Begin research on networking with java games

Begin adding an AI opponent

November End of Week 1:

Begin implementation of networking features

Add player lobby

Fine tune the AI

November End of Week 2:

Make sure the code will compile and run

November 2(End of Project 3):

Wrap-up and begin testing

Add stretch goals(Sound,graphics)