

Potential Network Implementation in Group 15's Blockus

In our design for the Blockus game, while network play was not a part of the current scope of the project, we've designed our application in order to simplify the addition of this feature as much as possible. While it will require some alteration to existing classes, as well as additional code to implement, our application is designed in such a way that this feature could be implemented without the need to make great alterations to the original design or functioning of the game.

We strove for low coupling and high versatility in our design in order to make changes and future implementations easier. by the current design of our application, the game consists of alternating turns, and each turn is occupied by either an object of the HumanPlayer class or of the ComputerPlayer class. If a turn is a HumanPlayer turn, then the corresponding player chooses their piece, rotates and places it using mouse clicks and the scroll wheel. On a ComputerPlayer turn, the placement of their piece is determined by their piece selection and placement algorithms. In order to implement network play, we've added but not implemented a third type of player class, the NetworkPlayer.

In a hypothetical implementation of network play, when instantiating a game, the user (who we will call User 1) would have the option to instantiate not just with human and computer opponents, but also network opponents. This would connect them to a server with other users (who we will call users 2 through 4) who have instantiated their game with network opponents. User 1 would be represented in their own game by an object of class HumanPlayer, while Users 2 through 4 would be represented by NetworkPlayer objects. In User 2's application, they would be represented by the HumanPlayer and Users 1, 3, and 4 would be represented by NetworkPlayer objects. And so on for Users 3 and 4. While currently just a clone of the HumanPlayer class that sees no implementation in the program, a future functioning NetworkPlayer object would, on their turn, receive a message from the server indicating the corresponding network opponent's move. This would also mean that the behavior of the HumanPlayer class would need to be altered so that a move made by a player in their own application would be sent to the server and then to the applications of all network opponents.