Alex Walker
Mitchell Biewen
Isaac Garfinkle

Quoridor:

The goal of our project is to program a computer game that resembles the board game Quoridor. In the game, players alternate turns, in which they are able to either move their pawn 1 space across the 9x9 grid-style board or place a wall on the board. The goal of the game is to move your own pawn to the other side of the board before anyone else. Each wall has a length of 2 squares. Players may place walls to block the other players' paths across the board, but they are not allowed to completely trap a player. Every player must have a possible path to the other side of the board at any time.
In the actual board game, up to 4 players are able to play against each other, but for the sake of our project, we will only implement 2 players to start. (Allowing 4 players could be one of our potential extensibility features.) We don't think an AI implementation is feasible for this project in the given amount of time, so our program will require 2 users playing against each other. All of the other rules and gameplay will be identical.

Things we plan to implement:

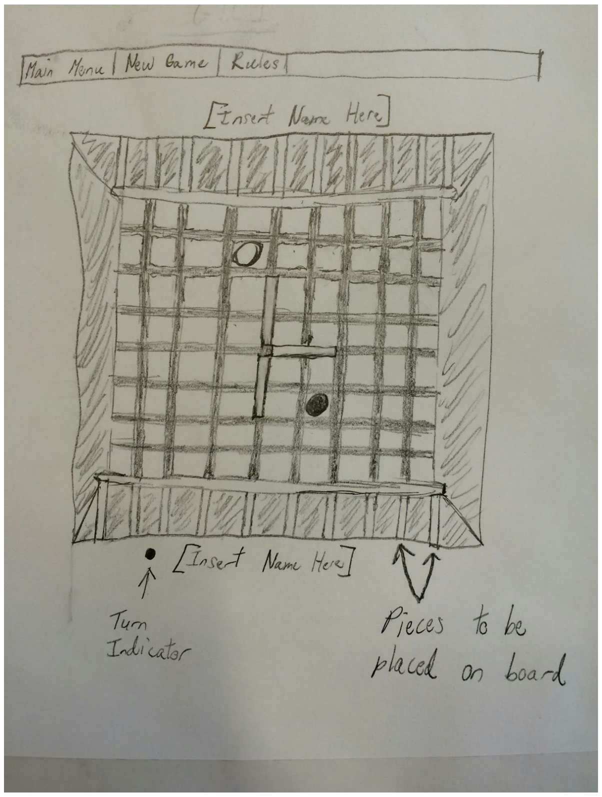
- User interface for viewing the board and making moves. Features include an easily viewable board and the ability to play moves by interacting directly with the board.
- The actual mechanism for playing the game including legal moves, turn taking, etc. Each instance of the game will be an object containing information about the game state. It will also reference player objects for whose turn it is and other player-centric information.
- A help/how-to page with game rules and basic tips. This will detail basic (and possibly more intricate) rules for the game and also simple strategy ideas.
- Main menu screen to initiate game, containing title information and our names. If settings are implemented, this screen will also allow users to set settings for this game.
- (Potentially) a settings page.
- (Potentially) a computer player.

Design patterns to use:

- Model View Controller (compound pattern)
- Observer pattern

Alex Walker
Mitchell Biewen
Isaac Garfinkle

Sketch of GUI:



Individual Roles:

Isaac: Implement rules for game and backend for how game data and mechanics are stored
Alex: Construct GUI and how players will interact with program
Mitchell: Handle communication between GUI and backend