Comp2121 Project
Preliminary Design Document

Daniel Murphy – dtsm460
Samuel Swiss – sswi317
Sahil Kapoor – skap473

Valid Move in Quoridor

Standard move: The player may move to any of the four horizontally or vertically adjacent squares to the current player position, so long as the square is not occupied by another player’s pawn and the path to the square is not blocked by a wall.

Jump move: When the player is adjacent to another player’s pawn, the player may jump to the square on the opposite side of the other player’s pawn, so long as there is no wall between the player’s current position and the opponent’s pawn, or between the opponent’s pawn and the destination square. The destination square must be unoccupied.

Alternative jump move: When the player is adjacent to another player’s pawn but is unable to complete a jump move because of a wall between the opponent’s pawn and the destination, the player may jump to one of the other two squares adjacent to the opponent’s pawn, if the squares are empty and there is no wall between the destination and the opponent’s pawn.

Candidate classes

Game
BoardState
PlayerState
AbstractPlayer
HumanPlayer
AIPlayer
Move

Class responsibilities

Game
Represents a game of Quoridor
Creates and manages the players involved in the game
Determines which player is moving next
Keeps track of the board state and the number of walls remaining
Collaborates with BoardState, PlayerState, AbstractPlayer

BoardState
Stores the state of the board: player positions and wall positions
Determines if the board is currently in a winning state
Determines if a given move is valid
Provides a list of all available moves for a given player
Collaborates with Game and Move

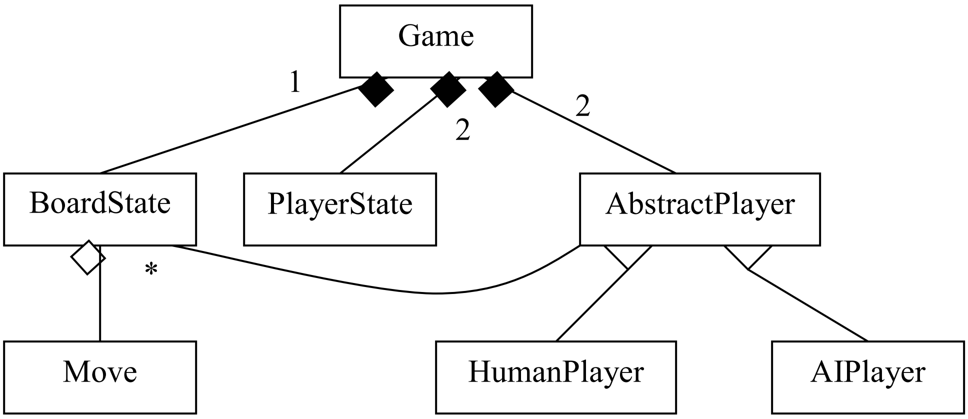
PlayerState
Keeps track of the number of walls a player has left
Collaborates with AbstractPlayer and Game

AbstractPlayer:
Represents one of the players
Subclasses: HumanPlayer, AIPlayer
Generates a new move based on a current board state
Collaborates with Game, BoardState and Move

HumanPlayer:
Represents a human player
Superclass: AbstractPlayer
Provides information about the state of the game to the standard output?
Reads a move from the standard input
Collaborates with Game, BoardState and Move

AIPlayer:
Represents an AI player
Superclass: AbstractPlayer
Generates a move based on the current board state
Collaborates with Game, BoardState and Move

Move:
Represents a player move or a wall placement
Collaborates with BoardState and AbstractPlayer


Game
1
2
2
BoardState
PlayerState
AbstractPlayer
*
Move
HumanPlayer
AIPlayer