Class	Responsibilities	Class Relationships
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 and AbstractPlayer
BoardState	• Stores the state of the board; the player positions and wall positions • Determines if the board is currently in a winning state • Determines if a move is valid • Provide a list of all available moves for a given player	• Collaborates with Game, Move and Validator
PlayerState	• Keeps track of the number of walls a player has left	• Collaborates with Game and Move
AbstractPlayer	• Represents a template for a player • Generates a new move based on a current board state	• Has subclasses HumanPlayer and AIPlayer • Collaborates with Game, BoardState and Move
HumanPlayer	• Represents a human player • Provides information about the state of the game to standard output • Reads a move from the standard input	• Is a type of AbstractPlayer which is thus its super class • Collaborates with Game, BoardState and Move
AIPlayer	• Represents an Artificial Intelligent player (computer player) • Generates a move based on the current board state	• Is a type of AbstractPlayer which is thus its super class • Collaborates with Game, BoardState and Move
Move	• Represents a player move or a wall placement	• Collaborates with BoardState and AbstractPlayer
Validator	• Checks the validity of a single given move on the board	• Collaborates with BoardState and MoveStringParser
MoveStringParser	• Seperates a string of moves into seperate moves and returns the number of moves remaining	• Collaborates with Validator