A valid move in Qoridor:

A valid move in quoridor consists of either moving the player piece (or pawn) to one valid space or placing one wall on a valid border between spaces.

A space that is valid for moving to is any space on the board that is adjacent to the player piece , not diagonal from the player piece, does not have any wall between the player piece and the space and does not have another player piece on the space. If another player piece lies on the space adjacent to the current player's piece the current player can “jump” over the other piece, and go to the space adjacent to the other player in line with both pawn's positions prior to the move. If there is a wall between the other player piece and the adjacent destination space the current player can jump to any other space adjacent to the other player piece that is not sealed off by a wall. If a third player's peace lies in any of the spaces that can be reached with a jump, that move becomes invalid, and cannot be taken.

A wall must be placed on the border adjacent to two sets of two spaces. A wall cannot cross another wall that has already been placed. A wall cannot be placed if the wall makes it so that any player is totally sealed off by walls from their goal at the opposite end of the board consisting of all the spaces along that edge of the board. Some goal spaces can be sealed off with a wall, but never all goal spaces.

Candidate classes and responsibilities:

Game Factory:

- Gets new commands
- Creates a new empty Game given a valid “new game”command
- Gets move validity from move file given a valid “load game” command
- Creates a new game with a number of move commands given a valid set of moves

Game:

- Contains a number of walls on borders.
- Contains a number players with points representing pawns on spaces
- Gets new commands
- Moves players or places walls based on moves commands
- Exports game state to a file when given a valid save command
- Displays current game state to output

Wall:

- Has some point and direction

Command:

- Can Take some commands from standard input and parses them
- Can Take some move from AI and stores it
- Can Take some move from a string at an index and stores it
- Stores command type and data

Validator:

- Given some set of commands returns a boolean representing validity

Player

-Abstract

- Has some Unique identification
- Has some point representing a pawn

AI player

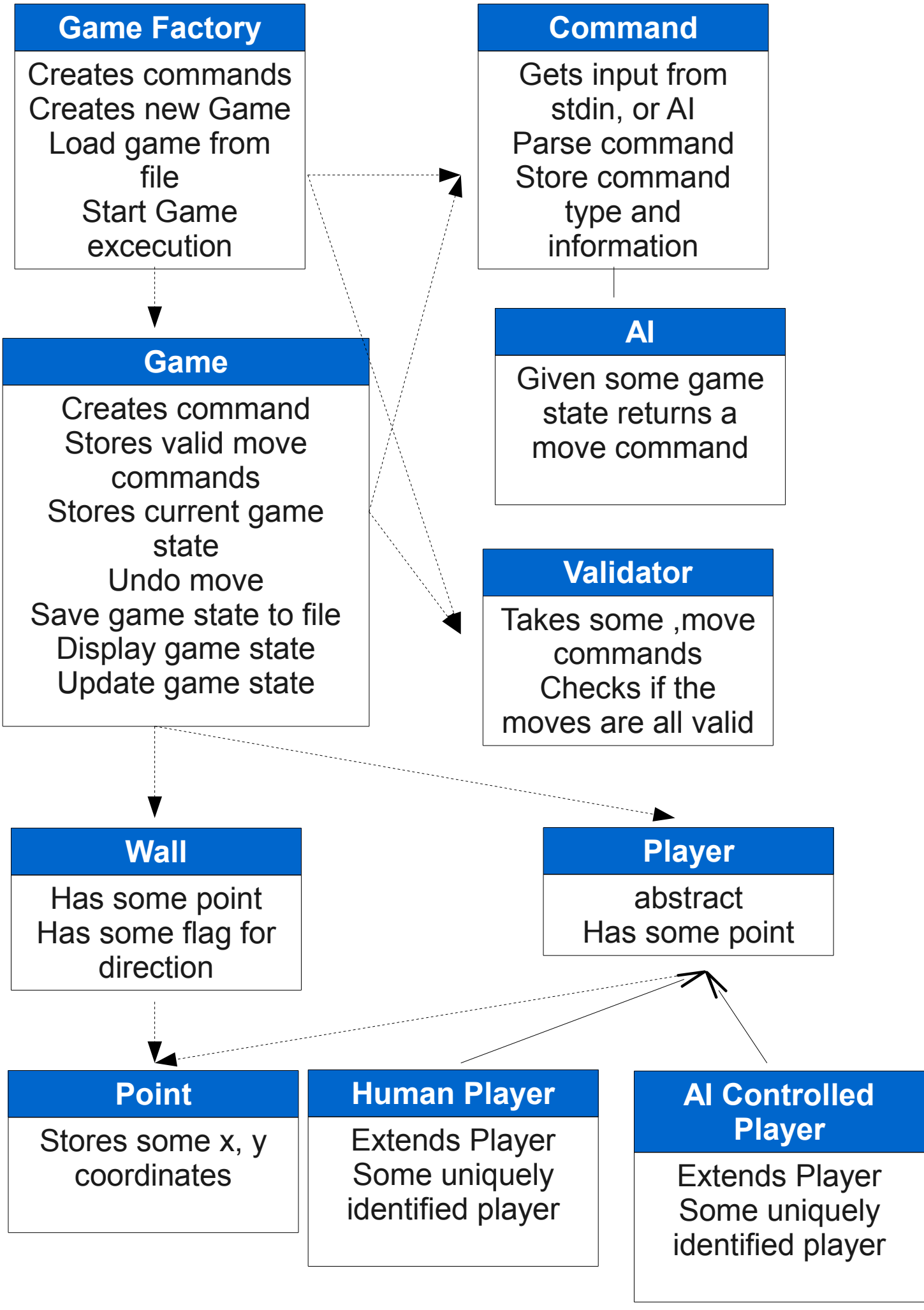
- extends player
- Some ai flag

Human Player

- extends player

AI

- Given some game state returns a good valid move command


Game Factory
Creates commands
Creates new Game
Load game from file
Start Game excecution
Command
Gets input from stdin, or AI
Parse command
Store command type and information
AI
Given some game state returns a move command
Game
Creates command
Stores valid move commands
Stores current game state
Undo move
Save game state to file
Display game state
Update game state
Validator
Takes some ,move commands
Checks if the moves are all valid
Wall
Has some point
Has some flag for direction
Player
abstract
Has some point
Point
Stores some x, y coordinates
Human Player
Extends Player
Some uniquely identified player
AI Controlled Player
Extends Player
Some uniquely identified player