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 piece 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:

Responsibilities

- Contains a number of walls on borders.
- Contains a number of lists of pawns associated with some players
- Returns which player is currently making their move
- Checks for validity when given a player and a move
- Has some players
- Takes moves from the user interface, or the AiPlayer until a game is suspended or finished

Collaborators

- Pawn
- Wall
- Player

Pawn:

Responsibilities

- Has some position corresponding to a space on a quorridor board
- Has some index based on when it was moved

Wall:

Responsibilities

- lies between two points corresponding to intersections between borders
- Has some index based on when it was placed

Display:

Responsibilities

- Display game state
- Display if a move is valid or not based on validity
- Display player information including scores and some player identification

Collaborators

- Game

User Interface:

Responsibilities

- Takes some commands from standard input and parses them
- Creates, saves and loads games accordingly
- Reads moves and gives them to a board
- Has some display to show game information
- plays some game until the game is finised.

Collaborators

- Game
- Display

Player:

Responsibilities

- Has some information identifying a player
- Has some score associated with a player
- Has some boolean for representing ai controlled platers

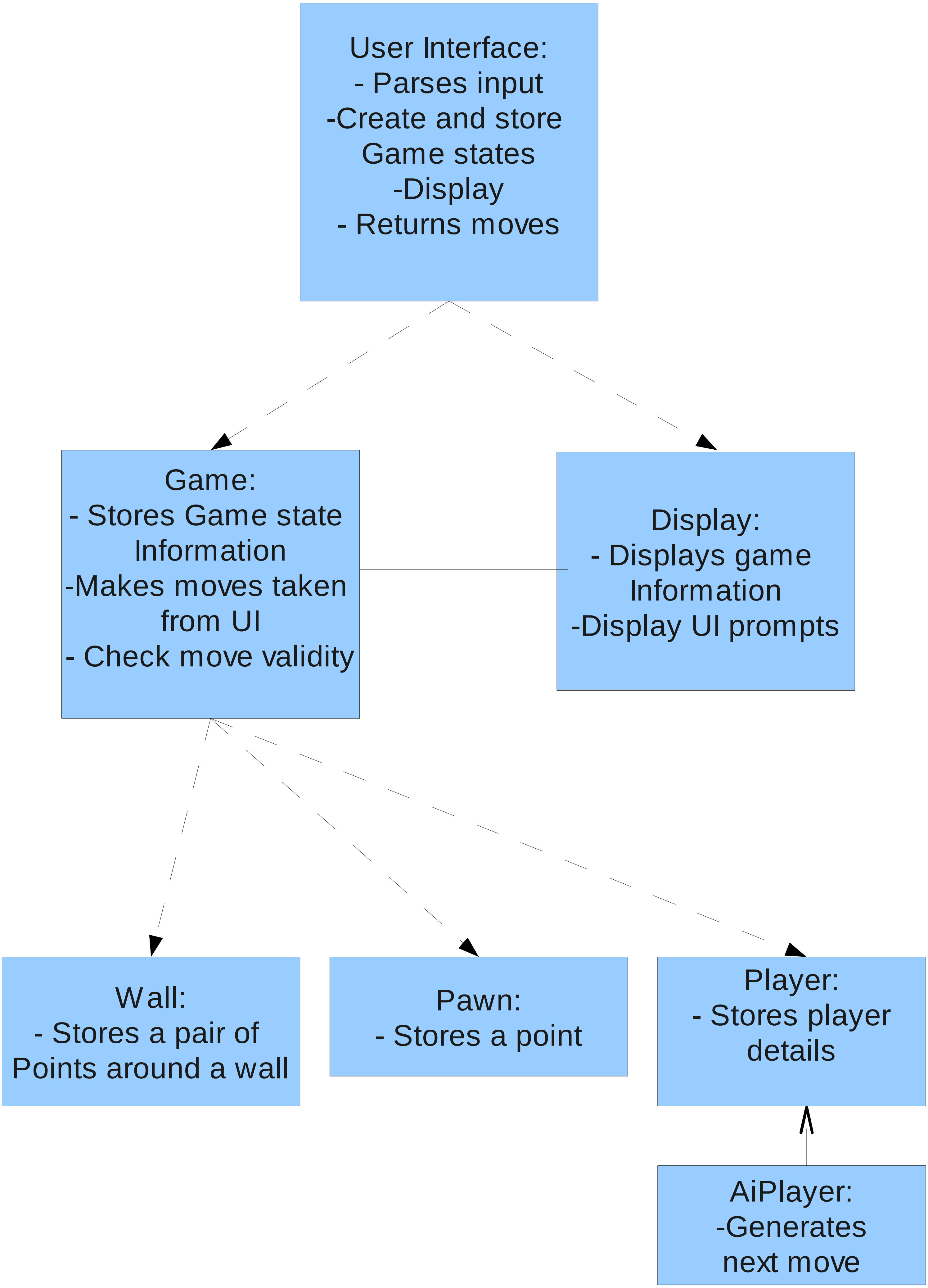
AiPlayer:

Responsibilities

- Returns some valid move given a board state

Collaborators

- Player



User Interface:
- Parses input
-Create and store Game states
-Display
- Returns moves

Game:
- Stores Game state Information
-Makes moves taken from UI
- Check move validity

Display:
- Displays game Information
-Display UI prompts

Wall:
- Stores a pair of Points around a wall

Pawn:
- Stores a point

Player:
- Stores player details

AiPlayer:
-Generates next move