**Project 6 Update**

**People Involved:** Nienfu Hsieh, Drew Dugan

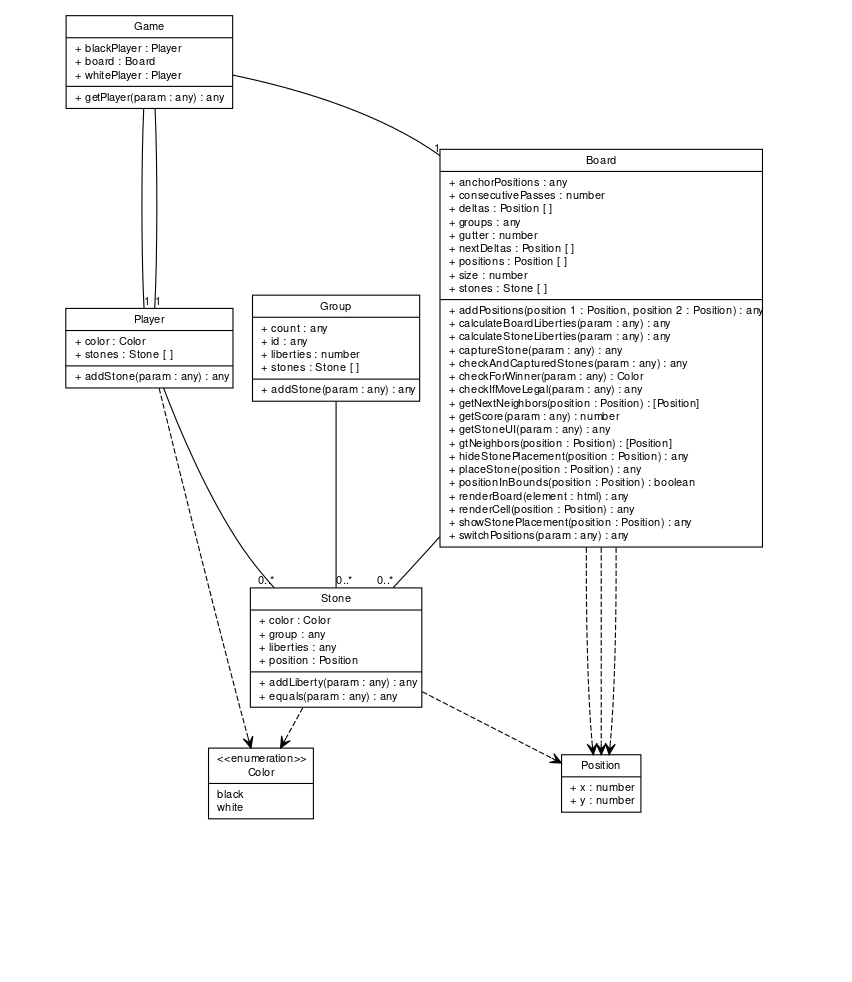
Work Done:

* Render a grid that works as the Board
* Show a ghost stone when hovered over a position if that move is legal
* Detect illegal moves
  + Can’t place a stone in already occupied position
* Place a stone on the board
* Calculate liberties of a stone
* Maintain history of board to check for positional superko
* Alternate between players after a move is made, starting with the black player
* Capture (individual) stones with 0(zero) liberties

**Changes or Issues:**

* Stone:
  + Added property *liberties* that is an array of **Position** objects
  + Added property *group* that stores a reference to the **Group** object it belongs to
  + Added *equals* method that checks if two Stone objects are equal.
  + Added *addLiberty* method to add a liberty to a stone
* Player:
  + *placeStone* method now returns the first stone that is removed from the set
  + *addStone* method adds a **Stone** object to the set
* Group:
  + Added a static *count* property
  + Added *id* property
  + *stones* is now an array
  + *liberties* is now an array
  + Removed *calculateLiberties* method
* Game:
  + Added *getPlayer* method to return a Player object that pertains to a certain color
* Board:
  + Added a *groups* property that contains **Group** objects
  + Added *anchorPositions* property
  + Added *switchTurns* method that switches the *turn* property to the other **player** after a player makes their move
  + *calculateStoneLiberties, calculateBoardLiberties, checkAndCapturedStones, captureStone, checkIfMoveLegal, getStoneUI* methods added

**UML Class Diagram:**



Plan for next iteration:

1. Integrate Groups into the game
2. Improve liberty calculation algorithm
3. Improve illegal move calculation algorithm
4. Scoring algorithm