CookieByte Testing Document

**Invite Class:**

* Invite Test –
* Test that an invite can be successfully created.
* AcceptInvite Test –
* Checks if the user can accept multiple invites and that a GameContoller is then directly added to the user’s list of CurrentGames.
* DeclineInvite Test –
* Checks if the User declines an invite that invite is removed from their list and no GameController is created.
* Equals Test –
* Test if two invites are equal based on the GameID attached to them.

**SetBoard Class:**

* SetBoard Test-
* Checks the starting positions of all offensive and defensive pieces and then some random areas where there are not supposed to be any pieces. Makes sure that given a board it resets all pieces to their default locations.

**Piece Class:**

* Piece Test-
* Makes Sure a Piece can me properly constructed.

**Board Class:**

* AddPieceToBoard Test –
  + Tested that the board class can add a piece of any type that belongs to any user to a specified location on the board.
* RemovePiece Test –
  + Tested that game could remove a designated Piece from the board given a certain location.
* MovePiece Test –
  + Tested if given locaction to go to and a location to move from the piece is deleted from the from location and is place at the to location.

**User Class:**

* AddOneInvites Test –
  + Tests that if an invite is constructed with two users that the user that it is sent to actually receives it in their invite list.
* AddMultipleInvites Test -
  + Tests that more than one invite can added under a user.

* RemoveMultipleInvites Test –
  + Tests if more than one invite can be removed from a user’s list.
* AddPastGame Test –
* Tested that a GameContoller can be added under the User’s PastGame list properly which stores all games the user was previously engaged in.
* AddCurrentGame Test-
* Tested that a GameContoller can be added under the User’s CurrentGame list properly which stores all games the User is currently engaged in.
* RemoveCurrentGame Test-
* Tests that a GameContoller can be removed from the User’s CurrentGame list which will be used whenever a Game is completed or if a user quits.
* AddLoss Test-
* Tests if the amount of Losses in a User has incremented right.
* AddWin Test –
* Tests if the amount of Wins a in User has incremented right.

**GameController Class:**

* kingWinConditions Test –
  + Creates fake king and game to place in different areas of the default board. If the king is in one of the four corners of the board it asserts true otherwise it asserts false.
  + Checks to make sure other attributes of the game have been properly set when the King winning conditions are meet.
* attackWinConditions Test-
  + Creates a king that is surrounded by 4 opponent pieces and checks if the correlating win conditions and attributes of the game are set.
  + Changes one piece to be on the King's side and expects win condition to fail and it also checks if the king is just surrounded by empty space.
  + The King can also be captured of cornered by the throne.
* isMoveValid Test –
  + Checks if a given move only moves the piece in a straight line horizontally and vertically. Pieces should be able to move in both a negative and positive direction.
  + Checks if a piece tries to move out of range of the board or if the there is another piece in the way. Also checks if the User tries moving when it’s not their turn.
  + If a player tries to move a piece that is nonexistent of a piece that does not belong to them. Also, the King can only be moved to the Throne space which is in the middle of the board.
  + Checks if the move does not run into of collides with other pieces and that the path is clear in all directions.
  + Checks if the piece moving to a corner is not a king or not and only gives kings the ability to occupy that position.
* CapturePiece Test-
  + Checks all 8 conditions where a piece can get captured by just one piece if it’s sandwiched next to one of the four corners.
  + Checks that if the piece is on the edge it can’t be captured and checks all out of bound errors.
  + Makes sure that if the king is not in a corner but is on the edge it cannot be captured.
  + If a Player tries to capture a piece that belongs to them or a piece that does not exist.
  + Makes sure that all pieces except the King can be captured by being sandwiched horizontally or vertically given the corresponding location of the sandwiched piece.
* movePiece Test-
  + Given a to and from location this tests a valid move made by user and checks to make sure that attributes and state of the game/board are properly set.
  + Checks that all Winning, Losing, and Capturing Conditions are taken in consideration on each turn. Depending on the pass and fail of each the state of the board/game is set accordingly.
  + Checks to make sure after every move that is made that the current Turn is switched to the opposite player. The test checks movement based on who’s turn it is.
* Quit Test –
* Test that Player 1 or Player 2 can quit on their turn and status of the game is changed along with the history of each player is updated properly.

**Acceptance testing:**

In order to verify that the game worked correctly, team Cookie Byte regularly and carefully tested all aspects of play and the system. We made sure that invites sent to the intended user only, and that the recipient could both accept or decline an invitation to play. We checked to make sure that a game was created when an invite was accepted. We made sure saved games correctly appeared on users’ home screens even after logging out. We verified that new users could be created, but not with the same user ID as another current registered user. Signing in only works if you have the correct password, and this was carefully verified. As far as gameplay goes, we tested the following moves: regular moves, captures, captures on edge of board, attempting to capture your own piece, moving into a capture scenario (should not be captured on your own turn), captures next to corners and center, the king reaching a corner, the king being captured on all four sides, the king being surrounded on three sides with the center square on the fourth side, attempting to go when it’s not your turn, attempting to move the piece of another player, making a “bad” move, and quitting the game (technically not a move). Due to our thorough and frequent testing, we have a very stable game.