Journal Report 4
9/23/19-9/27/19
Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Monday September 23

Added a lookup table for determining the relative positions of moves for every piece, meaning every piece can move correctly. Demonstrated this by stepping through the Sicilian Defense.

Tuesday September 24

Added capturing logic to move method (except for pawn captures). Created method to remove comments from formatted .pgn files, and extract the raw algebraic notation moves. Attempted to convert a real game (Fischer v. Spassky, 1992, game 29) from pgn to board states and found I hadn't implemented castling yet.

Thursday September 26

Added kingside castling as a valid move. Fixed pgn parsing error with newline characters. Fixed bug where piece jumping logic allowed non-knights to jump other pieces by adjusting lookup table. Added pawn captures and fixed pawn capture error where moves like `cxb5`would fail. Demonstrated functionality to Mr. White as a live code demo.

Timeline

Date	Goal	Met
Sept 16	Figure out why phone-to-laptop video feed has latency; create a script to convert digital board states to moves in PGN notation	Still unsure, hoping router fixes issue; made FEN converter and started PGN reader
Sept 23	Finish PGN reader by adding handlers for other pieces, captures, and potentially castling and en passant	Pieces and kingside castling done, queenside castling and en passant not
Sept 30	Fix final bugs in PGN reader, create PGN writer, use reader to check writer	Not started
Nov 7	Gather video of chess match with iP-Camera, router, and tripod; research image augmentation for CNN	Not started
Nov 14	Begin gathering images for CNN with partner's board segmentation script	Not started

Reflection

Save for en passant moves (which I'll have to make a special case for) and queenside castling (which should be nearly identical to kingside) I made a fully functional pgn reader this week. This took more effort than I wanted it to for something tangential to the main goal of creating a pgn *writer*, but I gained familiarity with how algebraic notation works, and having to hardcode 2D arrays of board to create a pgn writer would be painful. With the reader, creating a PGN writer will be very simple, since I'll be able to just check the differences between 2D arrays and output wherever the final position is, instead of considering both the start and end.

Here's an example of a significant bug I fixed. After white castles and black moves Be7, the board looks like this:

```
black
Be7

     a   b   c   d   e   f   g   h
   ---------------------------------
 8 | r | - | b | q | k | - | - | r | 8
   ---------------------------------
 7 | - | p | p | p | b | p | p | p | 7
   ---------------------------------
 6 | p | - | n | - | - | n | - | - | 6
   ---------------------------------
 5 | - | - | - | - | p | - | - | - | 5
   ---------------------------------
 4 | B | - | - | - | P | - | - | - | 4
   ---------------------------------
 3 | - | - | - | - | - | N | - | - | 3
```

```
   ---------------------------------
 2 | P | P | P | P | - | P | P | P | 2
   ---------------------------------
 1 | R | N | B | Q | - | R | K | - | 1
   ---------------------------------
     a   b   c   d   e   f   g   h
```

The next move is Re1, which my script failed to handle on Tues, because it thought both rooks could move to e1. Without a differentiating letter (e.g. Rfe1 or Rae1), it couldn't pick which rook to move. Obviously, the rook in the f file should move to e1, but my move method at the time allowed pieces to jump over others. I fixed this by changing my lookup table to store relative positions from the end position in ordered lists rather than sets, and iterating through in order, breaking the loop when another piece was in the way. After doing this, my reader read Re1 correctly, producing this correct board state:

```
white
Re1

     a   b   c   d   e   f   g   h
   ---------------------------------
 8 | r | - | b | q | k | - | - | r | 8
   ---------------------------------
 7 | - | p | p | p | b | p | p | p | 7
   ---------------------------------
 6 | p | - | n | - | - | n | - | - | 6
   ---------------------------------
 5 | - | - | - | - | p | - | - | - | 5
   ---------------------------------
 4 | B | - | - | - | P | - | - | - | 4
   ---------------------------------
 3 | - | - | - | - | - | N | - | - | 3
   ---------------------------------
 2 | P | P | P | P | - | P | P | P | 2
   ---------------------------------
 1 | R | N | B | Q | R | - | K | - | 1
   ---------------------------------
     a   b   c   d   e   f   g   h
```

...and all the board states after until the move `1/2-1/2` indicating a tie came up. I'll handle that, along with `1-0` and `0-1` next week.