Journal Report 3
9/16/19-9/20/19
Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Monday September 16

Out sick.

Tuesday September 17

Read up on FEN notation (used to indicate starting states, or the absolute position of a board) and PGN notation (used to indicate moves and captures; the relative position of a board). Also learned about algebraic notation—how PGN formats moves and captures. Wrote functions that convert FEN notation to a 2D array, turn algebraic notation into 2D array coordinates, and display the 2D array board visually. Started writing function to convert from 2D array back to FEN notation.

Checked the phone-to-laptop link, still laggy. Hoping local router fixes the issue. Works fine at home though.

Thursday September 19

Wrote a function to convert 2D board array to what I've termed half-FEN: the board-only segment of a FEN string. Began writing move-making function, which takes moves in PGN notation and returns an updated board. Discovered PGN moves are written as end coordinates only, with implied start coords. Currently, move function can handle forward pawn moves only. Started adding knights.

Timeline

Date	Goal	Met
Sept 9	Clean raw chess footage from online; set up router to avoid using phone hotspot	Chess footage downloaded, cut, and uploaded; no router yet
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	Pawns handled, knights started, capturing unfinished
Sept 30	Create PGN writer (given two board states, output PGN of the move between them)	Not started
Nov 7	Gather video of chess match with iPCamera, router, and tripod; research image augmentation for CNN	Not started

Reflection

I didn't realize how difficult making a PGN notation reader/writer would be, since I forgot that moves are only described by their end squares instead of their start to end pairings. However, I'm proud of the work I've done with FEN notation so far, which notates boards in their absolute states. The 2D array behind the scenes of my FEN converter will be the backbone of my PGN reader and writer, and the display methods and basic IO scripts will help immensely.

My code is in `show_game.py`, which imports methods from `chess_convert.py`. Usage is as follows, so far:

```
python show_game.py
```

which for the moves 1. e4 c5 outputs the following board states:

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

```
 3 | - | - | - | - | - | - | - | - | 3
   -----------------------------
 2 | P | P | P | P | P | P | P | P | 2
   -----------------------------
 1 | R | N | B | Q | K | B | N | R | 1
   -----------------------------
     a   b   c   d   e   f   g   h
```

e4

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

c5

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

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

and fails on 2. Nf3 because it moves the wrong knight:

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

This is the problem I ran into: a PGN reader has to have chess logic built-in to properly translate moves, since by default, there's no notation for the starting square of a piece. The pawns were simple enough since they only move forwards (excluding captures), but the other pieces will be more difficult. I think I'll be able to create handlers for every piece and captures by the end of the week; a reach goal would be handling castling and en passant as well by then.