Journal Report 5

9/30/19-10/4/19
Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Monday September 30

Added queenside castling and pawn promotion to PGN reader. Took more photos of chessboards for Kevin Chung. Added coded return value for endgame to `make_move()`. Downloaded another chess game to test.

Tuesday October 1

Played game against self for pawn test cases: specifically the case with simultaneous capture and pawn promotion. Handled en passant. Added simultaneous capture and pawn promotion to reader (e.g. `dxc8=Q`). Differentiated between ties and wins in PGN writer. Wrote empty methods in preparation for PGN writer.

Thursday October 3

Implemented most moves for PGN writer, save for cases where two pieces could move to same square (which need a differentiating letter), castling, and en passant. Discovered Python passes 2D arrays by reference; added deepcopy to `make_move()` to address this. Cleaned code, removing debug comments and adding explanations.

Timeline

Date	Goal	Met
Sept 23	Finish PGN reader by adding handlers for other pieces, captures, and potentially castling and en passant	Done
Sept 30	Fix final bugs in PGN reader, create PGN writer, use reader to check writer	PGN reader done, PGN writer started
Oct 7	Finish PGN writer	Most piece moves/captures implemented, special cases (en passant, piece collisions, castling) not
Oct 14	Modify partner's board segmentation script to help me automatically tag images	My end not started; partner has basic board segmentation script
Oct 21	Begin gathering + labelling images for CNN with board square labelling script	Not started

Reflection

En passant turned out to be easier to handle than I thought it would be. Since it only occurs after a pawn's initial two-step move, I simply had to check pawn captures from rank 5 to 6 and rank 4 to 3 for if they had a pawn one rank back, which I could hardcode. Pawn promotion turned out to be a challenge, since there are edge cases like a pawn promoting and capturing in the same move:

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

dxc8=Q
```

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

The PGN writer was also more difficult than I anticipated, since when two pieces could end in the same square, a file letter for the correct start piece is added, like `Nge2`. Also, I ran into an issue with how Python passed 2D arrays: I assumed it would copy when passing, but it passes 2D arrays by reference. I added a list comprehension to copy the board array at the start of `make_move()`.

Looking forward, I'll wrap up the PGN writer next week, then further automate the board segmentation script Kevin Chung wrote for me. He has a script that, when given an image of a chessboard and where the four corners of the board are, will output the individual squares of the board as jpegs. I'm planning to streamline this to allow me to press a key and label the each jpeg with the piece on the square, or empty (e.g. `P=pawn` or `E=empty`). After that, I'll gather chessboard images and begin labelling them with this script for the piece recognition CNN.