

## Daily Log

### Monday October 7

Resized chess images from last week to 1280x960 and 960x720. Added differentiating file letter for piece collisions to PGN writer. Added castling. Found bugs: piece jumping, some pawn captures, queenside castling. Set PGN writer to output to .pgn file rather than print to terminal. Fixed pawn capture bug.

### Tuesday October 8

Fixed castling bug (which arose when black and white castled to opposite sides) by correcting faulty checking logic. Fixed piece jumping (same way as in reader). Found black pawn captures were failing—fixed. Patched bug in PGN writer with incorrect pawn file letter. Added en passant.

### Thursday October 10

Added pawn promotion notation. Added checks of three types: by pieces, by pawns, and by promoted pawns. Took tougher images of chessboards for Kevin Chung: high-glare, low-angle, and rotated. Scaled all to both 1280x960 and 960x720. Downloaded PGN of a WCC 2018 game and found my PGN *reader* has an error with b-pawn moves. Fixed. Also found pawn captures from doubled-pawn columns fail: changed logic to fix.

## Timeline

Date	Goal	Met
Sept 30	Fix final bugs in PGN reader, create PGN writer, use reader to check writer	PGN reader done, PGN writer done except end-of-games
Oct 7	Finish PGN writer	All done except end-of-games
Oct 14	Add end-of-game moves to PGN writer, modify partner's board segmentation script to tag images	Not started, partner has basic board segmentation script
Oct 21	Gather and label images from chess videos with board-seg script	Not started
Oct 28	Research and implement image augmentation on segmented images	Not started

## Reflection

This week went well for me. I have a fully working PGN reader, until I find a case that doesn't work again. I also have a fully working PGN writer except for the end-of-game moves: checkmates, which are hard, resignations, which are impossible to detect from the board, and draws, which can be found if stalemate but not if agreed between the players. Rather than implement checkmate and stalemate logic, I'm planning to have the PGN writer prompt the user for a manual end-of-game entry. Although this makes the system not fully automatic, it resolves the cases of agreed draws and resignations, and inputting one move is easy compared to inputting 30-60 moves.

Sample output:

```
# PGN writer's output, given 2D arrays generated from PGN reader
['e4', 'e5', 'Nf3', 'Nc6', 'Bb5', 'a6', ...
'Re6', 'Kd5', 'Rb6', 'Kc5', 'Re6', '{game end, manual input required}']

# actual game PGN, in list form
['e4', 'e5', 'Nf3', 'Nc6', 'Bb5', 'a6', ...
'Re6', 'Kd5', 'Rb6', 'Kc5', 'Re6', '1/2-1/2']

#differences
{game end, manual input required} 1/2-1/2
```

For next week, I'm planning to write a script that makes my partner's board segmentation script more user-friendly, since I'll be using it heavily. Currently, Kevin Chung has it so after you input an image with a command-line arg, a pop-up appears and allows you to select the four corners of the board, then the board is segmented and individual square images are saved to a folder. I'll modify it to take an input directory of boards, and segment each board-image. After segmenting, it will show me the individual square images, and accept keyboard input to tag the image filenames with what piece and what color is shown. E.g. "wk" on my keyboard will translate to a filename like "image\_01\_white\_king". This will streamline the image labelling process immensely, and allow me to begin work on a piece-recognition CNN.