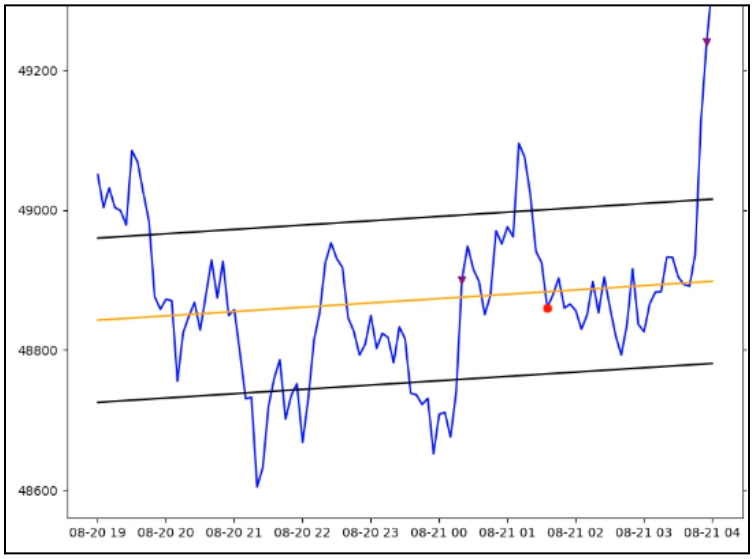
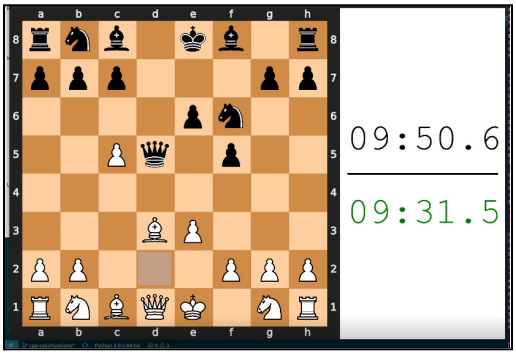
COMPUTER SCIENCE PORTFOLIO

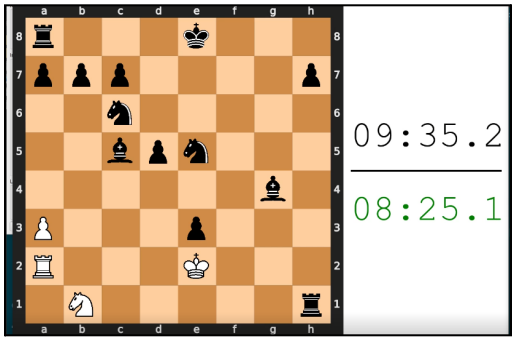
I have a portfolio of my three favorite programming projects that I've completed in Python and C++. The portfolio includes source code and video demos of each project, but I've also summarized the projects here.



1 - Using Python, I programmed a custom library to perform technical analysis of price action using custom data types and indicators. I used important Python tools like Numpy, Pandas, and Matplotlib to bring the project to life. My library finds symbols on the stock market and uses custom indicators to determine their projected future performance. A custom backtesting environment is also provided to make it easy to write automated strategies to test on price data. The picture to the right is taken from an animated backtest using a linear regression strategy. The yellow line is the trendline and the black lines are one and a half standard deviations away.

2 - I used machine learning to recognize player models in the video game Counter Strike: Global Offensive. I programmed this model in Python, leveraging Tensorflow for machine learning. Specifically, I ended up creating a custom dataset of in-game pictures to train my model on. I then leveraged some of the models from Tensorflow's object detection api. These models are trained to recognize generic objects, but I was able to use transfer learning to train them to recognize in-game players that looked like the ones in the dataset I created myself. These two pictures show the model in action, recognizing both types of player models in real time. The model runs at about ~40fps using my GTX 1080 and 9th gen Intel i5.





3 - Finally, I've also implemented a chess AI using a type of minimax algorithm. To inform the minimax, I developed heuristics to score each board the AI came across. The minimax algorithm works by creating a search tree to find the move that will lead to a favorable position some number of moves in the future (usually 6 or so moves). This strategy allows it to easily beat me (a beginner at Chess). I have been working on this project off and on for a couple of years, most recently I have added extensive performance improvements using a dynamic programming strategy called a transposition table. This table saves board positions using a custom hashing function so they don't have to be evaluated twice. The codebase is written in both Python and C++. The frontend GUI is written in Python for usability and the backend minimax AI is written in C++ for performance. The C++ backend actually ended up being quite large, I had to code legal move generation and move pushing and pulling all from scratch.

You can access the Github for the chess AI code here:
https://github.com/piperdaniel1/chess-ai

Link to download full portfolio (~750 MB) on Google drive: Portfolio - Google Drive
All source code is available in the download as well as demo videos for each of the projects.