Lisa Qing
Hari Kaushik

A5: “Jungkook” Baroque Chess Player Report

Design Retrospective:

With the huge popularity increase surrounding korean boy band, BTS, we wanted to name our agent after one of the members. Therefore, the agent has the personality of a celebrity who has a concert to attend and needs to finish the game quickly. The agent also will occasionally respond with information about their new album as a way to promote or comment on the status of the game. The way we designed the agent was by first implementing the move functions of each of the pieces. We implemented a king, a pincer, and a queen moving function. This is because all the pieces besides the pincer and king move the exact same way. Then depending on the piece, given the row and column, it will capture its respective ways. For all of the move functions, we chose to return a list of [(start, end), new state] to keep track of all the possible moves and states that a given piece could make. The abilities of the imitator was implemented last. The minimax functions were then made - first without alpha-beta pruning then with. With the basic static evaluation function, since many states have the same static evals, we had to sort the successors based on their start square then by its end square for tie breakers. Since the basic static evaluation returned the same result for many states, the game play was very mediocre. On the other hand, with our personally created new static evaluation function, the likelihood for tiebreakers is much smaller, allowing the agent to choose states that were strategically better. The static evaluation that was created evaluates each piece based on how favorable their position is. This includes proximity to making a capture and making sure the king was protected. Lastly, when our agent, is asked to make a move, it does IDDFS that goes to a max depth which is dependent on how many pieces (more plys if there are less pieces) and calls parametrized minimax to find the best possible move in the given time span.

Partnership Retrospective:

For this project, we began by divvying up the work; however, towards the end when putting everything together got more complicated, we did pair programming. After understanding the game, we each did half of the move functions for the pieces. Once all of the move functions were generating the successors list that we wanted to for a given state. Then we needed to use minimax to find the best successor for a given state. At this point, we ran into a lot of bugs and spent a lot of time debugging. During this process, we found it easiest to work on the project together in person to find and fix the bugs. Since it was the both of us working on the part together, we were able to run through the logic and catch any potential flaws that we would not have caught ourselves. By doing this, we were able to effectively bounce ideas off of each other instantaneously and test out many new iterations and found what worked and what did not.

Partnership retrospective - Hari:

I thought that we did a great job of adapting our development strategies as we faced challenges while working on this assignment. We started out using Git effectively to collaborate

Partnership retrospective - Lisa:

Personally, I thought we did a good job considering the scope of the assignment. Unlike all the previous assignments, this one require substantial amount of knowledge regarding adversarial search, recursion, baroque chess game play, and etc. Since I was out of the state for the first few days when the assignment was released, we had a rather late start. However, we were able to use Git effectively in the end to collaborate on code. Then after coming back, we were able to dive more deeply into the code and work on it together.