MARIST

PROJECT 2

Artificial Intelligence
CMSC404
Fall 2014

This is a group assignment. Each group will have two students.

For this assignment you will develop a chess playing program and compete in a chess tournament with other the other groups in the class. A server will be used to organize the game. Each team's program will run on their own computer and communicate with the server to make moves and receive the opponent's move.

The chess games will be timed using compensation time with Fischer delay. The times will be 15 minute games with 5 second increments.

Moves

To denote moves, we will used a modified version of algebraic notation for chess. Most moves will be denoted by five characters.

1. type of piece ∈ {`K`, `Q`, `R`, `B`, `N`, `P`}
2. beginning file ∈ {`a`, `b`, `c`, `d`, `e`, `f`, `g`, `h`}
3. beginning rank ∈ {`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`}
4. ending file ∈ {`a`, `b`, `c`, `d`, `e`, `f`, `g`, `h`}
5. ending rank ∈ {`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`}

For example, a move may be `Pd2d3` or `Nb1c3`. (both opening moves)

One exception is *promotion*. For a promotion the move will have a sixth character.

6. type of piece for promotion ∈ {`Q`, `R`, `B`, `N`}

For example `Pb7b8Q` where the pawn is promoted to a queen.

Other "special" moves can be inferred from the move and the state, such as *castling* and *en passant* captures.

API

To facilitate play the game will be hosted on a webserver that will provide APIs to the playing programs. Each team will be assigned a number and a secret for authentication. For testing, you may use the test team numbers:

Team Number	Secret
1	32c68cae
2	1a77594c

Games can be initiated at the following link:

`http://www.bencarle.com/chess/newgame`

Enter the team numbers of the two teams playing and you will be given a *GameId*.

When both teams are ready, click start game to begin the game.

While it is the opponents turn, your program should poll the API every 5 seconds to check if the opponent has made a move. The polling link is:

`http://www.bencarle.com/chess/poll/`*GameId*`/`*TeamNumber*`/`*TeamSecret*`/`

While the opponent has not yet moved, the response will be:

```
{
  "ready": false,
  "secondsleft": float,
  "lastmovenumber": int,
}
```

Once the opponent has moved, the server will respond with the last move made by the opponent and the time your program has remaining. The format of the response will be:

```
{
  "ready": true,
  "secondsleft": float,
  "lastmovenumber": int,
  "lastmove": string,
}
```

When it is your turn, you can make a move through the following link:

`http://www.bencarle.com/chess/move/`*GameId*`/`*TeamNumber*`/`*TeamSecret*`/`*MoveString*`/`

Games can be monitored at the following page:

`http://www.bencarle.com/chess/display/GameId/`

Language

For this project you may use any programming language that provides the necessary functionality.

Tournament

On Wednesday, December 17, from 3:30 - 6:15, (final exam slot for the course) we will hold a tournament in class to have student groups compete against each other. All students must be in attendance at the tournament to receive credit.

Report

In addition to your source code, each group must submit a report that contains the following:

- A brief description of your strategy (e.g. minimax with alpha-beta pruning)
- A description of your evaluation function
- Instructions on how to compile and run your code.

Grading

Each group must be able to defeat a naive opponent in order to receive credit for the project.

Submission

Submit all source files and your report through iLearn by 3:00 pm on December 17.