Shallow Blue

Project 2

Due date: April 5th

Introduction

• Second in series of three projects
• This project focuses on getting AI opponent
• Subsequent project will develop Networking

• You will work individually for this project!
  - No groups
  - Encouraged to talk about solutions with classmates
  - Can help each other debug code
  - Cutting and pasting or mailing code → too much
Details (1 of 2)

• Make a computer-controlled chess opponent (Shallow Blue) for your chess game
  - Build upon project 1
• The game should support:
  - Legal chess moves (as in Project 1) taken in turn, by Blue.
  - "Smart" moves by Blue
    • Blue should try to win and not just move pieces randomly
  - Levels of difficulty. Blue should have settings, making it an easier or harder opponent
    • You choice: Depth/Time, Evaluation, Random
    • GUI front end allow player to change

Details (2 of 2)

• Advanced:
  - "Think" time: does not have to spend same amount of time per move, but make sure doesn’t think too long
  - Sophisticated Evaluation function: Basic count of material (ie-pieces) good, but a lot of additional enhancements that might be added
  - Opening vs. Mid-Game vs. End-Game: Evaluation of opening moves or end-game state may proceed differently than mid-game state.
  - Personality: Different versions of Blue (selected randomly, is fine) may be more aggressive or defensive or ...
    • Can change with Evaluation
• You will use:
  - MinMax search algorithm
    • with AlphaBeta pruning
  - A board evaluation (lecture notes coming up have details)
• Implement in your choice of language (C, C++, Java, ...)
  - Use a language you are familiar with!
Links

• See project Web page:
  • MinMax tree links
  • Evaluation strategies for Chess
  • Chess links from project 1
  • Note, may have tournament!
    - May the best Blue win
    - Prize!

Submission

• Will use command line (Unix) turnin
• Submit all source files necessary to compile and run program
  - Any Makefiles, etc.
  - All art (as appropriate)
• README
  - Saying how to build, platform, etc.
Grading

- **Basic Game** 65%
  - Plays basic chess game, with legal moves.
- **Pruning** 10%
  - Decision tree AlphaBeta pruning implemented.
- **Controlled Think Time** 10%
  - Means to limit think time, either literally by time or limiting depth search when there are many choices.
- **Sophisticated Evaluation** 5%
  - In addition to pieces, consider king safety, mobility, pawn placement, ...
- **Opening and End-game** 5%
  - Separate consideration for opening moves and end-game state.
- **Style** 5%
  - Consideration of different personalities (ie- aggressive, or defensive or ...)