CMPUT 275 Project Proposal

Chess

Arun Woosaree & Tamara Bojovic

March 18, 2018

Introduction

The project is an improvement on our CMPUT 274 final project, which was a player vs. player Arduino chess game. We plan to first import our project into python, making use of the object oriented features of python, that we did not use in the old version. For example, each chess piece will inherit from a generic chess piece class, and have their own moves. Instead of the Arduino, we will use pygame as our GUI. We will improve on the algorithms from our original game, using new algorithmic knowledge gained in 275. (particularly check and checkmate tests, which were inefficient). Once a player vs. player implementation is up and running, we will then focus on creating a basic computer opponent. This will involve a move generator, and an evaluator function that determines scores for different board positions. We would use our knowledge of search trees, to implement the computer players moves.

Timeline

1. **March 22**
   Import our old project into python. Create classes for the pieces, set up the board representation and use our old functions within the new classes. Demo: Fully working player vs. player chess game in pygame
2. **March 25**
   Improve the existing algorithms to be more efficient, with our new knowledge of search trees and graphs. Focus on the very inefficient check/checkmate detection function Demo: Player vs. player chess game with improved algorithms
3. **April 2**
   Create a move generator, that generates all the possible moves from a cer-

tain board position. Old functions that checked for valid moves can now utilize this. Computer player will select a move randomly, no optimization or intelligence Demo: Ability to play against a very dumb computer opponent

4. **April 6**
   Create a simple evaluation function, that calculates the best move just based on the pieces on the board. More important pieces will have a higher score. Demo: Computer opponent has some method to their madness, but cannot look ahead to pick the best move

5. **April 10**
   Increase the depth of the search to 3, and use the minimax algorithm to select the best move. Improve the evaluation function by adding more bonuses to scores, such as a protected king, control of the center of the board adding a bonus Demo: A fully working chess game with a basic computer opponent.