Course: Project AI - Symbolic artificial intelligence
Professor: Dr.-Ing. Stefan Fricke
Semester: SOSE23
Date: 2023/07/26

Group Members: Who the fuck cares?
Repository: https://github.com/PraxTube/chess-ai/

Project Report

Python Chess AI

Group C - The Plebs

This is a dummy abstract.
Hello there.

Contents

1 Introduction

Creating a chess AI from scratch is quite a challenging undertaking. Not only will you need to write a whole chess backend, you will also need to implement the AI features. One of the major issues here is to write a chess backend without any bugs and to test your AI properly to make sure the features you add actually make it play better.

Our group chose to use Python for the whole project given that that is what we were most familiar with. The obvious trade-off here is of course that it's easy to prototype but painfully slow and very error prone. I personally would have liked to try to use Rust, though in hindsight we would have probably abandoned the project if we had used Rust simply because the chess engine alone was so much work. On the other side I have acquired some Rust experience now and if I were to write the chess engine (or something of a similar level) I would probably go with Rust.

Regardless of our programming language, for version control we obviously used git and to share our code base we used github. The overall workflow here was pretty smooth.

This was pretty much the first *real* AI project we took on. Our goal was to at least get a basic AI done, that was what we wanted to to reach at least. Our final AI is actually fairly advanced for what we seeked to accomplish. It's obviously not the strongest (given that we are using python that isn't too surprising). However we are very pleased with the end result and with what we created and learned during this project.

In the following chapters we will go into more detail into what, why and how we created our AI. We will also reflect on all these things.

2 Development

Before we are going into the details of the development process let's first look at the high level overview of what we created (note that you can click on the Mst name to see whole documentation of it).

Mst1 - Dummy AI:

- Chess Backend
- Dummy AI (minimax)
- Basic Evaluation (Material only)

Mst2 - Basic AI:

- Improve Backend
- Alpha-Beta Tree Search
- Improved Evaluation (PeSTO)
- Better time management

Mst3 - Advanced AI:

- Restructured Chess Backend
- Speed up Evaluation (through numpy)
- Improve move ordering
- Include King of the Hill in evaluation
- Restructure internal debug info

Mst4 Optimized AI:

- Improve evaluation
- Use Monte Carlo Tree Search
- Implement PVS/negamax
- Add Nullsearch

2.1 Mst1 Dummy AI

3 Results

4 Issues faced

5 Lessons learned

6 Summary

This is a dummy summary.

Hello world.

Literatur

Test