



Ramo Estudantil IEEE - UEL



IGOR SHOITI TUTIDA SUGUIEDA(igor.tutidasuguieda@uel.br)
GUSTAVO MARCUCCI ORTIZ(gustavo.marcucci@uel.br)

RELATÓRIO FINAL:
Projeto IA De Xadrez

Londrina
2023

Contato do Ramo: sb.uel@ieee.org
Institute of Electrical and Electronics Engineers – IEEE
Universidade Estadual de Londrina - UEL • Paraná - Brasil



Ramo Estudantil IEEE - UEL



IGOR SHOITI TUTIDA SUGUIEDA
GUSTAVO MARCUCCI ORTIZ

RELATÓRIO FINAL:
Projeto IA De Xadrez

Relatório apresentado ao Ramo Estudantil
IEEE da Universidade Estadual de Londrina.

Londrina
2023

Contato do Ramo: sb.uel@ieee.org
Institute of Electrical and Electronics Engineers – IEEE
Universidade Estadual de Londrina - UEL • Paraná - Brasil



Ramo Estudantil IEEE - UEL



Ortiz, Gustavo Marcucci. Suguieda, Igor Shoiti Tutida. **Relatório Final:** Projeto IA de Xadrez. 2023. 09 folhas. Relatório apresentado ao Ramo Estudantil IEEE da Universidade Estadual de Londrina, Londrina, 2023.

RESUMO

Com o desligamento dos projetos de IA, foi solicitado a escrita de um relatório com os avanços do grupo em relação ao projeto para que em algum momento mais para frente uma possível reativação do projeto ocorra com base nas informações do relatório

Palavras-chave: IA. Xadrez. Engine.

Contato do Ramo: sb.uel@ieee.org
Institute of Electrical and Electronics Engineers – IEEE
Universidade Estadual de Londrina - UEL • Paraná - Brasil



SUMÁRIO

Sumário

1. INTRODUÇÃO	5
2. FUNDAMENTAÇÃO TEÓRICA	5
2.1 O QUE FOI FEITO	5
2.1.1 Ferramentas	6
2.2 O que mais foi pensado e o que pode ser feito	7
3. Conclusões	8



1. INTRODUÇÃO

1.1 Sobre o encerramento do projeto

Com a notificação da desativação dos projetos de inteligência artificial, foi solicitado uma compilação das ideias do projeto: o que foi feito e como poderia continuar. Então, a seguir seguem-se os conteúdos que são necessários para a possível realização do projeto.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 O que foi feito

Inicialmente, o grupo decidiu por qual linguagem de programação usar. Por se tratar de um projeto que mexe com inteligência artificial, por meio de uma votação decidiu-se o uso da linguagem Python, devido a sua alta relação com IA, machine learning e seus outros recursos.

Um dos trabalhos iniciais foi o desenvolvimento do tabuleiro que era basicamente um tabuleiro “printado” e que a representação das peças eram suas respectivas iniciais.

Pouco depois, descobriu-se a biblioteca de xadrez em python que basicamente tinha a mesma ideia, porém já tinha “pré-programado” os movimentos de cada peça. Um dos seus “limitadores” era o “input” de comandos que era limitado a escrever o movimento da peça em forma de código. Como por exemplo para mover o peão:

```
board.push_san("e4")  
Move.from_uci('e2e4')
```

Para contornar esse “problema” parecia mais interessante fazer um tabuleiro com interface gráfica que responderia aos comandos do utilizador. Para isso seria usada a biblioteca pygame.

Com essas ideias em mente, os próximos passos seriam:

- Aprender python;
- Desenvolver o formato do tabuleiro na janela;
- Colocar as figuras das peças;
- Programar os movimentos legais das peças;
- Programar o “xeque”, “xeque-mate”, “promoção”, “roque”;
- Programar a engine;



Com essas ideias em mente, os membros do grupo precisavam ainda ter um primeiro contato com o python, visto que, a linguagem vista nas aulas era C. Então, cada um precisava ir atrás do básico da sintaxe da linguagem python, além de precisar correr atrás dos conhecimentos básicos de funcionamento de IA.

Devido ao tempo, o projeto ficou muito mais na teoria.

2.2 FERRAMENTAS

2.1.1 VS Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft. Ele tem como principais funcionalidades editar códigos (suporta diversas linguagens de programação), possui um terminal de comandos integrado, possui diversas extensões e é de simples uso.

2.1.2 Python

Python é uma linguagem de programação de alto nível, interpretada por script, imperativa, orientada a objetos, funcional de tipagem dinâmica e forte.

A escolha dessa linguagem foi motivada pela sua simples leitura, possivelmente menor uso de linhas no código, variedade de bibliotecas e sua forte relação com machine learning, IA e Programação Orientada a Objetos.

2.1.3 Minimax

Um algoritmo minimax é muito comum em jogos de estratégia por turnos. A ideia por trás desse algoritmo é “uma árvore” na qual cada ramificação é um possível estado do jogo após determinada ação. O objetivo dele é observar todas as possíveis jogadas e escolher a que maximiza seus resultados mesmo com a melhor jogada do oponente.

Uma maneira mais simplificada de entender esse algoritmo é:

- Max é a jogada da máquina/jogador, nesse nó é sempre buscado o valor de maior retorno.
- Min é a jogada do humano/oponente, nesse nó é sempre buscado o menor valor (melhor jogada dele)

Sempre entendendo que o computador vai considerar o adversário como escolhendo sua melhor jogada.

Porém, o algoritmo de Minimax possui algumas limitações, quando a árvore possui muitas ramificações (profundidade), ela demora mais para retornar uma jogada, custando tempo e processamento. Para isso, existe um algoritmo nomeado “Poda alfa-beta” que reduz a quantidade de ramificações a serem analisadas.

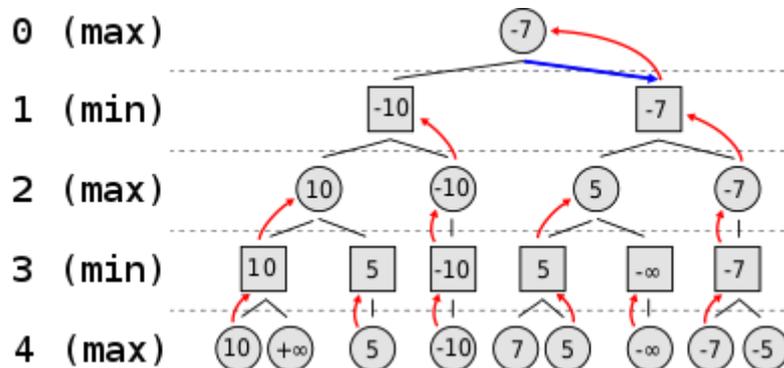


Figura 1: Representação da árvore Minimax

Fonte: <https://pcodinomebzero.neocities.org/Imagens/minimax2.png>

Poda alfa beta

Uma das melhorias do algoritmo minimax é não pesquisar um ramo da árvore que com certeza não leva a um resultado melhor do que se tem. Essa melhoria chama-se poda alfa-beta.

É atribuído α para o valor de melhor escolha (mais alto) até determinado momento (para o caminho MAX) e β é o valor da melhor escolha (mais baixo) até o momento (ao longo do caminho para MIN). Dessa forma sabendo que o valor do nó é pior do que o valor corrente, esse nó é cortado da busca, poupando tempo e processamento.

2.2 O que mais foi pensado e o que pode ser feito

Com base nessas ideias e no estudo sobre IA, foi possível encontrar alguns conteúdos de auxílio para a realização do projeto que serão repassados a seguir:

- [How to Make Chess in Python!](#)
- [Creating a Chess Engine in Python - YouTube](#)

O conteúdo que seria usado como base para o projeto seria a playlist “Creating a Chess Engine in Python” de Eddie Sharick. Nela, há 16 vídeos que servem como auxílio no processo de desenvolvimento da IA de xadrez. Segue também um código baseado nesses vídeos, com algumas implementações a mais, encontrado no github:

- [GitHub - mikolaj-skrzypczak/chess-engine: Implementation of chess in python created mainly by following a step-by-step YouTube tutorial by Eddie Sharick, with added image processing project for the 5th semester of my Computer Science studies.](#)



Ramo Estudantil IEEE - UEL



3. CONCLUSÕES

Devido ao grupo ser composto apenas por calouros com pouco contato com a programação e sem o devido contato com matérias como IA (vista somente no 5º semestre). O projeto não teve muitos avanços que não fossem teoria. Grande parte do que foi feito foram pesquisas teóricas que guiassem o grupo.

Com a notificação dos encerramentos dos projetos de IA foi pedido um relatório que servisse como guia para que no futuro, um possível outro grupo utilizasse como base os materiais encontrados e estudados pelo relatório desse grupo.



REFERÊNCIAS BIBLIOGRÁFICAS

Creating a chess engine in python .Disponível em:
<https://www.youtube.com/playlist?list=PLBwF487qi8MGU81nDGaeNE1EnNEPYWKY>

Inteligência artificial. Busca competitiva em jogos:
<https://www.inf.ufrgs.br/~engel/data/media/file/inf01048/jogos.pdf>

Python Chess Engine. Disponível em:
<https://github.com/mikolaj-skrzypczak/chess-engine>.