

# Anatomia do motor de um programa de xadrez

Hugo Vinicius M. D. Santana  
Orientador: José Coelho de Pina

# Conteúdo

- Objetivo
- O que é um motor de xadrez?
- Arquitetura
- Entrada e saída
- Representação do tabuleiro
- Buscas
- Função de avaliação

# Objetivo

- Entender como funciona um motor de xadrez.
- Mostrar como é a implementação de cada componente do programa (somente na monografia).

# O que é um motor de xadrez?

- Um motor de xadrez é um programa capaz de jogar um jogo de xadrez.
- Recebe uma posição do tabuleiro de xadrez, analisa esta posição, considera um conjunto de movimentos válidos e retorna um movimento que é o melhor possível de acordo com algum critério.

# Arquitetura

- Um motor de xadrez possui quatro componentes principais.
- Entrada e saída para comunicação.
- Tabuleiro que contem informações necessárias.
- Busca para analisar a árvore do jogo.
- Avaliação que calcula o valor de uma movimentação.

# Uma iteração do motor

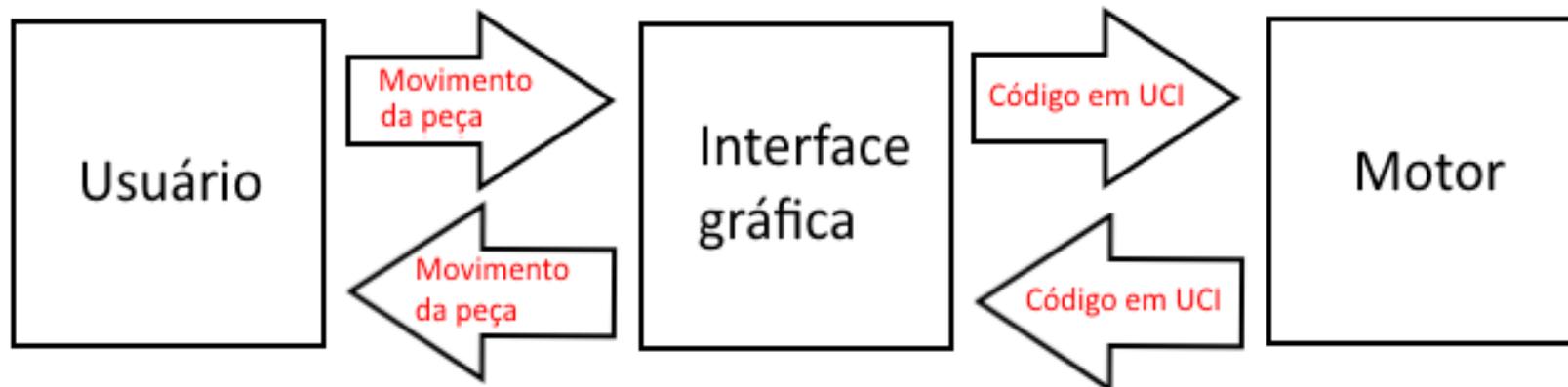
- Dado uma posição do tabuleiro, recebe o próximo movimento do oponente.
- Faz uma busca na árvore do jogo.
- A cada tempo determinado, calcula a pontuação dos nós folha para decidir o melhor movimento. Se há mais tempo, continua a busca.
- Devolve o proxima movimento.

# Entrada e saída para comunicação

- Em geral, não possui uma interface gráfica.
- Recebe comandos pela entrada padrão e devolve respostas para a saída padrão.
- Para facilitar o uso do programa, conectamos o motor à uma interface gráfica externa.

# Interface gráfica

- A troca de informações do motor com a interface gráfica é feita respeitando um protocolo de comunicação.
- Os mais usados são: Interface universal de xadrez (UCI, sigla em inglês) e Protocolo de comunicação do motor de xadrez (CECP).



# Representação do tabuleiro

- Para decidir que peça mover, é necessário ter informações sobre o tabuleiro.
- Precisa de alguma estrutura de dados para representar as posições do tabuleiro.
- Existem três tipos principais de representação.
- Centrada nas peças.
- Centrada nas casas.
- Híbrida.

# Representação centrado nas peças

- Lista de peças:  
Mantêm a posição de todas as peças do tabuleiro em uma lista ou vetor.
- *Bitboards*:  
Uma palavra de 64bit para cada tipo de peça.  
Casa ocupada tem valor 1, caso contrário 0.  
Para verificar se uma casa está ocupada, basta calcular a conjugação binária com o número binário da casa.

# Representação centrado nas casas

- Tabuleiro 8x8:  
Mantêm se a casa está ocupada ou não.
- 0x88 (10001000 em binário):  
Cada casa é representado pelo número binário 0LLL0CCC, onde LLL é a linha e CCC é a coluna.  
A operação (casa & 0x88) vale zero se a casa é válida.  
Exemplo:  
01000111 (terceira linha e oitava coluna)  
 $01000111 \ \& \ 10001000 = 00000000$   
  
00101001 (terceira linha e nona coluna)  
 $00101001 \ \& \ 10001000 = 00001000$

# Representação híbrida

- Muitas vezes, por eficiência, é usado mais de uma representação do tabuleiro.
- Exemplo:  
Usar lista de peças para obter a casa de uma peça e usar 0x88 para verificar se o movimento é válido.

# Buscas

- Para decidir o próximo movimento a ser feito, o motor consulta uma árvore de busca.
- Uma árvore de busca é um grafo direcionado com o nó contendo a pontuação desta posição e de quem é a vez, e a aresta representa o movimento feito.
- A busca é reponsável pela força tática do motor. Pois os programas mais fortes explora mais profundamente a árvore em tempo limitado, isto é, tem mais informações para decidir a melhor movimento.

# Aprimoramentos de busca

- Para um jogo de duas pessoas, é usado a busca minimax.
- Para melhor eficiência, usamos o algoritmo alfa-beta, que define um intervalo  $(\alpha, \beta)$  para cada nó. Se a pontuação do nó não pertencer em  $(\alpha, \beta)$ , a subárvore é removido da árvore.

# Aprimoramentos de busca

- Tabela de transposição:

Guarda os resultados de buscas anteriores. Uma transposição significa uma mesma posição de tabuleiro, mas com sequência de movimentos diferentes. Sempre que a busca encontra uma posição repetida, em vez de continuar, consulta esta tabela para decidir o próximo movimento.

- Busca em profundidade iterativa:

Expande nós de mesma altura para cada iteração. Melhor controle de tempo e facilita a ordenação das arestas a ser visitado na sequência de melhor movimento.

# Função de avaliação

- A função de avaliação é usada para calcular o valor de cada posição, isto é, a chance de vencer. Para cada nó é calculado o valor e usando este resultado o motor decide o próximo movimento.

# Função de avaliação

- Existem várias funções:
- Material, que considera a soma do valor das peças do tabuleiro.
- Mobilidade, que considera o número de movimentos possíveis para um jogador.
- Estrutura dos peões, que considera a posição de todos os peões.
- Segurança do rei, que, usando alguns critérios, calcula se o rei está seguro.

# Referências

- T. A. Marsland. *Computer chess and search*
- *Stuart J. Russell e Peter Norvig. Artificial Intelligence A Modern Approach. third edition*
- *Claude Shannon. “Programming a Computer for Playing Chess”*
- *Chess Programming WIKI - <https://chessprogramming.wikispaces.com/>, 2014*
- *Wikipedia, the free encyclopedia.*

