

Trabalho Final – Estrutura de Dados e Análise de Algoritmos

Título: Simulador de Xadrez com Listas, Filas e Pilhas

Grupo:

- Andrey Justen Junior (1072316310)
 - Bárbara Prim de Souza (10723114770)
 - Leonardo Alves Silva (10723113466)
 - Mateus Nunes Lehmkuh (1072312497)
-

1. Descrição Geral do Projeto

Este simulador de xadrez foi desenvolvido em Python com Tkinter para a interface gráfica. O tabuleiro 8×8 é representado em um Canvas, e cada peça possui uma classe dedicada para lógica de movimento. O usuário clica para selecionar peças, visualiza as casas possíveis e confirma o movimento. A função de desfazer (undo) é acionada por Ctrl+Z, retornando ao estado anterior.

2. Uso de Estruturas de Dados

1. Listas

- Tabuleiro: matriz 8×8 (self.board.grid) armazenando referências às peças ou None.
- Conjunto de movimentos: lista dinâmica com as posições válidas de destino.
- Sequência de classes: lista fixa define a ordem inicial das peças.
- Vantagem: acesso direto por índices e flexibilidade para coleções de tamanho variável.

2. Filas

- Controle de turnos: lista tratada como fila circular (self.turns).
- Operação: removemos o primeiro jogador e adicionamos ao fim após cada movimento.

- Vantagem: mantém a alternância de jogadores de forma simples e extensível.

3. Pilhas

- Histórico de movimentos: lista que armazena o estado anterior antes de cada jogada.
- Operação: adicionamos (push) antes de mover e removemos (pop) ao desfazer.
- Vantagem: o último movimento feito é o primeiro a ser revertido.

Justificativa Geral: As operações nativas de listas em Python (append, pop, indexação) garantem simplicidade, eficiência e código legível.

3. Análise de Complexidade de Tempo

1. Fila (Controle de Turnos)

- Pop do início da lista (fila): $O(n)$ no pior caso, pois os elementos subsequentes são deslocados para preencher o espaço.
- Append ao fim da lista: $O(1)$ amortizado.
- Conclusão: Em jogos com muitos jogadores, a alternância de turnos pode custar $O(n)$, mas com dois jogadores o custo é constante na prática.

2. Pilha (Desfazer Movimento)

- Push (adicionar estado): $O(1)$, pois adiciona ao final da lista que representa a pilha.
- Pop (remover estado): $O(1)$, pois remove o último elemento da lista.
- Conclusão: As operações de desfazer são eficientes e de tempo constante, ideais para ações reversíveis.

3. Lista (Tabuleiro e Movimentos Possíveis)

- Acesso direto à posição do tabuleiro: $O(1)$, por indexação em uma matriz 8×8 .
- Inserção/remoção em listas de movimentos: $O(1)$ para append, $O(n)$ para remove ou inserções em posições específicas.

- Conclusão: O uso de listas garante flexibilidade e eficiência para manipular dados estruturados e dinâmicos do jogo.
-

4. Instruções de Uso

1. Pré-requisitos:

- Python instalado (versão 3.x) e adicionado ao PATH.

2. Execução:

3. git clone <https://github.com/AndreyJR777/Xadrez-Lista-Fila-Pilha.git>

4. cd Xadrez-Lista-Fila-Pilha

python chess.py

5. Controles:

- Clique na peça para selecionar.
 - Clique em uma casa válida para mover.
 - Pressione Ctrl+Z para desfazer o último movimento.
-

5. Links Úteis

- **Quadro Trello:** <https://trello.com/b/wvLccU9P/a3-estrutura-de-dados-e-an%C3%A1lise-de-algoritmos>
 - **Repositório Git:** <https://github.com/AndreyJR777/Xadrez-Lista-Fila-Pilha>
-

Data de Entrega: 17 de junho de 2025