

Jogo de Xadrez

Gerado por Doxygen 1.8.13

Sumário

1	jogo_de_xadrez	1
2	Índice das Estruturas de Dados	3
2.1	Estruturas de Dados	3
3	Estruturas	5
3.1	Referência da Estrutura board	5
3.1.1	Campos	5
3.1.1.1	BlackCheck	5
3.1.1.2	Board	5
3.1.1.3	Weight	5
3.1.1.4	WhiteCheck	6
3.2	Referência da Estrutura list_past_move	6
3.2.1	Campos	6
3.2.1.1	head	6
3.2.1.2	last	6
3.3	Referência da Estrutura ListNode	6
3.3.1	Descrição Detalhada	7
3.4	Referência da Estrutura ListOfMoves	7
3.4.1	Descrição Detalhada	7
3.4.2	Campos	7
3.4.2.1	howmany	7
3.5	Referência da Estrutura ListPastMoves	8
3.5.1	Descrição Detalhada	8

3.6	Referência da Estrutura Move	8
3.6.1	Descrição Detalhada	8
3.6.2	Campos	8
3.6.2.1	destiny	8
3.6.2.2	origin	9
3.7	Referência da Estrutura node_list	9
3.7.1	Campos	9
3.7.1.1	move	9
3.7.1.2	next	9
3.8	Referência da Estrutura NodeList	9
3.9	Referência da Estrutura NodeTree	10
3.9.1	Descrição Detalhada	10
3.9.2	Campos	10
3.9.2.1	board	10
3.9.2.2	child	10
3.9.2.3	n_child	10
3.9.2.4	play	11
3.10	Referência da Estrutura TBoard	11
3.10.1	Descrição Detalhada	11
3.11	Referência da Estrutura Tree	11
3.11.1	Descrição Detalhada	11
3.11.2	Campos	11
3.11.2.1	root	11

Capítulo 1

jogo_de_xadrez

Um jogo de xadrez escrito em C.

Instruções:

Primeiramente, abre-se a pasta src, que está dentro do diretório jogo_de_xadrez, no terminal. Em seguida, compila-se o programa escrevendo make. Para iniciar a execução do programa, digita-se no terminal, dentro da pasta src, a seguinte instrução: ./main.

Em seguida, Aparecerá uma interface com 4 opções, uma para iniciar um jogo com um tabuleiro na posição de peças padrão do xadrez, a segunda que permite recuperar um jogo salvo por meio de um arquivo contendo um tabuleiro salvo, a terceira que permite criar um tabuleiro do zero e a última que permite sair do jogo.

Ao selecionar a primeira opção e a segunda, abrirá outra janela de opções que dispõem de 3 opções, a primeira em que se joga jogadorXjogador, a segunda que é jogadorxComputador e a opção de sair do jogo.

Ao selecionar a terceira opção, primeiro abre-se uma interface para que se crie a sua própria disposição em um tabuleiro de xadrez, usando a linha de comando própria da interface para adicionar as peças, começando pelos reis, e clicando na posição desejada. Ao finalizar esse processo, aparecem as 2 opções de modo de jogo.

Como jogar:

. Modo Jogador x Jogador:

Aparece a interface com o tabuleiro selecionado, começando pelas peças brancas. O jogador deve escrever na linha de comando usando o padrão pré estabelecido para determinar a peça em sua posição inicial e a posição final do movimento. Em seguida, é a vez das peças pretas jogarem, seguindo o mesmo procedimento. O jogo termina ao se chegar em uma condição de checkmate ou empate.

. Modo Jogador x Computador:

Aparece a interface com o tabuleiro selecionado, começando pelas peças brancas. O jogador recebe escrito na tela as melhores jogadas possíveis para o momento, podendo escolhê-las ou não. Em seguida, a máquina joga baseada em sua IA para tentar ganhar do jogador. O jogo termina quando se chega em uma posição de checkmate ou empate.

Capítulo 2

Índice das Estruturas de Dados

2.1 Estruturas de Dados

Aqui estão as estruturas de dados, uniões e suas respectivas descrições:

board	5
list_past_move	6
ListNode	
Estrutura do nó da lista que armazena as jogadas passadas	6
ListOfMoves	
Estrutura representará todas as possíveis jogadas de um tabuleiro	7
ListPastMoves	
Estrutura da lista que armazena as jogadas passadas	8
Move	
Estrutura que irá caracterizar a jogada	8
node_list	9
NodeList	9
NodeTree	
Estrutura de um nó da árvore	10
TBoard	
Estrutura do tabuleiro	11
Tree	
Estrutura da árvore	11

Capítulo 3

Estruturas

3.1 Referência da Estrutura board

Campos de Dados

- char **Board** [8][8]
- int **Weight**
- int **WhiteCheck**
- int **BlackCheck**

3.1.1 Campos

3.1.1.1 BlackCheck

```
int BlackCheck
```

Inteiro que armazena a condição de xeque do rei preto.

3.1.1.2 Board

```
char Board[8][8]
```

Matriz de caracteres que representa o tabuleiro.

3.1.1.3 Weight

```
int Weight
```

Inteiro que armazena o peso ponderado do tabuleiro.

3.1.1.4 WhiteCheck

```
int WhiteCheck
```

Inteiro que armazena a condição de xeque do rei branco.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- tabuleiro.h

3.2 Referência da Estrutura list_past_move

Campos de Dados

- **ListNode * head**
- **ListNode * last**

3.2.1 Campos

3.2.1.1 head

```
ListNode* head
```

Ponteiro para a cabeça da lista.

3.2.1.2 last

```
ListNode* last
```

Ponteiro para o último elemento da lista.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- in-out.h

3.3 Referência da Estrutura ListNode

Estrutura do nó da lista que armazena as jogadas passadas.

```
#include <in-out.h>
```

3.3.1 Descrição Detalhada

Estrutura do nó da lista que armazena as jogadas passadas.

Essa estrutura armazena os elementos necessários para recuperar uma jogada feita anteriormente.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- in-out.h

3.4 Referência da Estrutura ListOfMoves

Estrutura representará todas as possíveis jogadas de um tabuleiro.

```
#include <list_of_moves.h>
```

Campos de Dados

- int **howmany**
- **NodeList** * **first**
- **NodeList** * **current**
- **NodeList** * **last**

3.4.1 Descrição Detalhada

Estrutura representará todas as possíveis jogadas de um tabuleiro.

Lista do tipo **Move** (pag. 8) que é importante para a análise das jogadas.

3.4.2 Campos

3.4.2.1 howmany

```
int howmany
```

Inteiro que guarda quantas jogadas são possíveis de serem feitas.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- list_of_moves.h

3.5 Referência da Estrutura ListPastMoves

Estrutura da lista que armazena as jogadas passadas.

```
#include <in-out.h>
```

3.5.1 Descrição Detalhada

Estrutura da lista que armazena as jogadas passadas.

Essa estrutura serve para recuperação de jogadas passadas e salvamentos de jogos em PGN.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- in-out.h

3.6 Referência da Estrutura Move

Estrutura que irá caracterizar a jogada.

```
#include <list_of_moves.h>
```

Campos de Dados

- int **origin** [2]
- int **destiny** [2]

3.6.1 Descrição Detalhada

Estrutura que irá caracterizar a jogada.

Estrutura que armazena uma maneira de interpretar uma jogada.

3.6.2 Campos

3.6.2.1 destiny

```
int destiny[2]
```

Array que armazena as coordenadas de destino do movimento.

3.6.2.2 origin

```
int origin[2]
```

Array que armazena as coordenadas de origem do movimento.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `list_of_moves.h`

3.7 Referência da Estrutura `node_list`

Campos de Dados

- `char move[7]`
- `struct node_list * next`

3.7.1 Campos

3.7.1.1 move

```
char move[7]
```

String com a jogada feita na notação de xadrez.

3.7.1.2 next

```
struct node_list* next
```

Ponteiro para o próximo elemento da lista.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `in-out.h`

3.8 Referência da Estrutura `NodeList`

Campos de Dados

- `Move play`
- `struct NodeList * next`

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- `list_of_moves.h`

3.9 Referência da Estrutura NodeTree

Estrutura de um nó da árvore.

```
#include <arv_deciso.es.h>
```

Campos de Dados

- **TBoard * board**
- **Move * play**
- **int n_child**
- **NodeTree ** child**

3.9.1 Descrição Detalhada

Estrutura de um nó da árvore.

Estrutura que armazena os componentes básicos de um nó da árvore de decisões

3.9.2 Campos

3.9.2.1 board

```
TBoard* board
```

Ponteiro para uma variável do tipo Tboard que representa a organização do tabuleiro depois de alguma jogada

3.9.2.2 child

```
NodeTree** child
```

Vetor de ponteiros para os filhos do nó

3.9.2.3 n_child

```
int n_child
```

Inteiro que armazena o número de filhos que o nó da árvore tem

3.9.2.4 play

Move* play

Ponteiro para uma variável do tipo **Move** (pag. 8) que representa a jogada que originou a configuração do tabuleiro
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- arv_deciso.es.h

3.10 Referência da Estrutura TBoard

Estrutura do tabuleiro.

```
#include <tabuleiro.h>
```

3.10.1 Descrição Detalhada

Estrutura do tabuleiro.

Estrutura que armazena os elementos necessário para definir um tabuleiro.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- tabuleiro.h

3.11 Referência da Estrutura Tree

Estrutura da árvore.

```
#include <arv_deciso.es.h>
```

Campos de Dados

- **NodeTree *** root

3.11.1 Descrição Detalhada

Estrutura da árvore.

Estrutura que armazena uma referência para a raiz da árvore de decisões

3.11.2 Campos

3.11.2.1 root

NodeTree* root

Ponteiro para uma variável do tipo ponteiro que representa a raiz da árvore

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- arv_deciso.es.h

Índice Remissivo

BlackCheck	play, 10
board, 5	
Board	origin
board, 5	Move, 8
board, 5	
BlackCheck, 5	play
Board, 5	NodeTree, 10
NodeTree, 10	
Weight, 5	root
WhiteCheck, 5	Tree, 11
child	TBoard, 11
NodeTree, 10	Tree, 11
	root, 11
destiny	Weight
Move, 8	board, 5
	WhiteCheck
head	board, 5
list_past_move, 6	
howmany	
ListOfMoves, 7	
last	
list_past_move, 6	
list_past_move, 6	
head, 6	
last, 6	
ListNode, 6	
ListOfMoves, 7	
howmany, 7	
ListPastMoves, 8	
Move, 8	
destiny, 8	
origin, 8	
move	
node_list, 9	
n_child	
NodeTree, 10	
next	
node_list, 9	
node_list, 9	
move, 9	
next, 9	
NodeList, 9	
NodeTree, 10	
board, 10	
child, 10	
n_child, 10	