

Reference Manual

Generated by Doxygen 1.6.1

Sun May 30 23:18:03 2010

Contents

| | | |
|----------|-------------------------------------|----------|
| 1 | Class Index | 1 |
| 1.1 | Class List | 1 |
| 2 | File Index | 3 |
| 2.1 | File List | 3 |
| 3 | Class Documentation | 5 |
| 3.1 | elemento Struct Reference | 5 |
| 3.1.1 | Detailed Description | 5 |
| 3.1.2 | Member Data Documentation | 5 |
| 3.1.2.1 | cor | 5 |
| 3.1.2.2 | letra | 5 |
| 3.2 | erro Struct Reference | 6 |
| 3.2.1 | Member Data Documentation | 6 |
| 3.2.1.1 | err_code | 6 |
| 3.2.1.2 | mensagem | 6 |
| 3.3 | gravados Struct Reference | 7 |
| 3.3.1 | Detailed Description | 7 |
| 3.3.2 | Member Data Documentation | 7 |
| 3.3.2.1 | its | 7 |
| 3.3.2.2 | jgs | 7 |
| 3.3.2.3 | mvs | 7 |
| 3.3.2.4 | totalGr | 7 |
| 3.4 | It Struct Reference | 8 |
| 3.4.1 | Detailed Description | 8 |
| 3.4.2 | Member Data Documentation | 8 |
| 3.4.2.1 | next | 8 |
| 3.4.2.2 | pos | 8 |
| 3.5 | jogadas Struct Reference | 9 |

| | | |
|----------|-----------------------------|-----------|
| 3.5.1 | Detailed Description | 9 |
| 3.5.2 | Member Data Documentation | 9 |
| 3.5.2.1 | col | 9 |
| 3.5.2.2 | color | 9 |
| 3.5.2.3 | next | 9 |
| 3.5.2.4 | row | 9 |
| 3.6 | movimentos Struct Reference | 10 |
| 3.6.1 | Detailed Description | 10 |
| 3.6.2 | Member Data Documentation | 10 |
| 3.6.2.1 | next | 10 |
| 3.6.2.2 | pos | 10 |
| 4 | File Documentation | 11 |
| 4.1 | ajuda.c File Reference | 11 |
| 4.1.1 | Detailed Description | 12 |
| 4.1.2 | Define Documentation | 12 |
| 4.1.2.1 | CHUNK_SIZE | 12 |
| 4.1.3 | Function Documentation | 12 |
| 4.1.3.1 | ajuda | 12 |
| 4.1.3.2 | auto_black | 12 |
| 4.1.3.3 | auto_white | 13 |
| 4.1.3.4 | automatico | 13 |
| 4.1.3.5 | brancasAuto | 13 |
| 4.1.3.6 | change_tc | 14 |
| 4.1.3.7 | control | 14 |
| 4.1.3.8 | copy_board | 15 |
| 4.1.3.9 | corPossivel | 15 |
| 4.1.3.10 | corrigir | 15 |
| 4.1.3.11 | crg | 16 |
| 4.1.3.12 | executa_tc | 16 |
| 4.1.3.13 | imp | 16 |
| 4.1.3.14 | impacto | 17 |
| 4.1.3.15 | p_brancas | 17 |
| 4.1.3.16 | p_preta | 18 |
| 4.1.3.17 | pretasAuto | 18 |
| 4.1.3.18 | resolve | 18 |
| 4.1.3.19 | rsv | 19 |

| | | |
|----------|---------------------------|----|
| 4.1.3.20 | tc | 19 |
| 4.1.3.21 | tc_auto | 19 |
| 4.1.3.22 | tentarFinais | 20 |
| 4.1.3.23 | tentativa | 20 |
| 4.1.3.24 | vc | 21 |
| 4.1.3.25 | verificaAB | 21 |
| 4.1.3.26 | verificaAP | 21 |
| 4.2 | ajuda.h File Reference | 23 |
| 4.2.1 | Function Documentation | 23 |
| 4.2.1.1 | ajuda | 23 |
| 4.2.1.2 | brancasAuto | 23 |
| 4.2.1.3 | control | 23 |
| 4.2.1.4 | count_i | 24 |
| 4.2.1.5 | crg | 24 |
| 4.2.1.6 | executa_tc | 24 |
| 4.2.1.7 | imp | 25 |
| 4.2.1.8 | pretasAuto | 25 |
| 4.2.1.9 | rsv | 25 |
| 4.2.1.10 | vc | 26 |
| 4.3 | comandos.c File Reference | 27 |
| 4.3.1 | Detailed Description | 27 |
| 4.3.2 | Function Documentation | 27 |
| 4.3.2.1 | executa_comando | 27 |
| 4.3.2.2 | executaCriaJogo | 28 |
| 4.3.2.3 | executaGravaJogo | 28 |
| 4.3.2.4 | executaJogada | 28 |
| 4.4 | comandos.h File Reference | 30 |
| 4.4.1 | Function Documentation | 30 |
| 4.4.1.1 | executa_comando | 30 |
| 4.4.1.2 | executaCriaJogo | 30 |
| 4.4.1.3 | executaGravaJogo | 31 |
| 4.4.1.4 | executaJogada | 31 |
| 4.5 | erro.c File Reference | 32 |
| 4.5.1 | Typedef Documentation | 32 |
| 4.5.1.1 | ERRO | 32 |
| 4.5.2 | Function Documentation | 32 |

| | | |
|----------|--|----|
| 4.5.2.1 | <code>mensagem_de_erro</code> | 32 |
| 4.6 | <code>erro.h</code> File Reference | 33 |
| 4.6.1 | Define Documentation | 33 |
| 4.6.1.1 | <code>E_ARGS</code> | 33 |
| 4.6.1.2 | <code>E_COMMAND</code> | 33 |
| 4.6.1.3 | <code>E_COORDS</code> | 33 |
| 4.6.1.4 | <code>E_FILE_FORMAT</code> | 33 |
| 4.6.1.5 | <code>E_INVARGS</code> | 33 |
| 4.6.1.6 | <code>E_NO_BOARD</code> | 33 |
| 4.6.1.7 | <code>E_NO_FILE</code> | 33 |
| 4.6.1.8 | <code>E_NO_MOVES</code> | 33 |
| 4.6.1.9 | <code>E_NOERR</code> | 33 |
| 4.6.1.10 | <code>E_SAVE</code> | 33 |
| 4.6.1.11 | <code>E_WRONG_SOLUTION</code> | 33 |
| 4.6.2 | Function Documentation | 33 |
| 4.6.2.1 | <code>mensagem_de_erro</code> | 33 |
| 4.7 | <code>estado_tabuleiro.c</code> File Reference | 35 |
| 4.7.1 | Detailed Description | 35 |
| 4.7.2 | Function Documentation | 35 |
| 4.7.2.1 | <code>caminho</code> | 35 |
| 4.7.2.2 | <code>free_tab</code> | 35 |
| 4.7.2.3 | <code>prepara_tab</code> | 36 |
| 4.7.2.4 | <code>st</code> | 36 |
| 4.7.2.5 | <code>vb</code> | 36 |
| 4.7.2.6 | <code>vl</code> | 36 |
| 4.7.2.7 | <code>vp</code> | 37 |
| 4.8 | <code>estado_tabuleiro.h</code> File Reference | 38 |
| 4.8.1 | Function Documentation | 38 |
| 4.8.1.1 | <code>caminho</code> | 38 |
| 4.8.1.2 | <code>free_tab</code> | 38 |
| 4.8.1.3 | <code>prepara_tab</code> | 38 |
| 4.8.1.4 | <code>st</code> | 38 |
| 4.8.1.5 | <code>vb</code> | 39 |
| 4.8.1.6 | <code>vl</code> | 39 |
| 4.8.1.7 | <code>vp</code> | 39 |
| 4.9 | <code>jogadas.c</code> File Reference | 40 |

| | | |
|----------|--------------------------|----|
| 4.9.1 | Detailed Description | 40 |
| 4.9.2 | Function Documentation | 41 |
| 4.9.2.1 | anc | 41 |
| 4.9.2.2 | anm | 41 |
| 4.9.2.3 | E_Par | 41 |
| 4.9.2.4 | E_Par_des | 41 |
| 4.9.2.5 | E_trp_col | 42 |
| 4.9.2.6 | E_trp_row | 42 |
| 4.9.2.7 | free_jogadasGR | 43 |
| 4.9.2.8 | freeIts | 43 |
| 4.9.2.9 | freeJgs | 43 |
| 4.9.2.10 | freeMvs | 43 |
| 4.9.2.11 | grava_jogada | 43 |
| 4.9.2.12 | grava_movimento | 43 |
| 4.9.2.13 | inicia_GR | 44 |
| 4.9.2.14 | it | 44 |
| 4.9.2.15 | jogada | 44 |
| 4.9.2.16 | jogadaManual | 44 |
| 4.9.2.17 | pds | 45 |
| 4.9.2.18 | pis | 45 |
| 4.9.2.19 | rb | 46 |
| 4.9.2.20 | snd | 46 |
| 4.9.2.21 | trp | 46 |
| 4.10 | jogadas.h File Reference | 47 |
| 4.10.1 | Typedef Documentation | 48 |
| 4.10.1.1 | Gravados | 48 |
| 4.10.1.2 | IT | 48 |
| 4.10.1.3 | Jogadas | 48 |
| 4.10.1.4 | Movimentos | 48 |
| 4.10.2 | Function Documentation | 48 |
| 4.10.2.1 | anc | 48 |
| 4.10.2.2 | anm | 48 |
| 4.10.2.3 | E_Par | 48 |
| 4.10.2.4 | E_Par_des | 49 |
| 4.10.2.5 | E_trp_col | 49 |
| 4.10.2.6 | E_trp_row | 49 |

| | | |
|-----------|----------------------------|----|
| 4.10.2.7 | free_jogadasGR | 50 |
| 4.10.2.8 | grava_jogada | 50 |
| 4.10.2.9 | grava_movimento | 50 |
| 4.10.2.10 | inicia_GR | 50 |
| 4.10.2.11 | it | 51 |
| 4.10.2.12 | jogada | 51 |
| 4.10.2.13 | jogadaManual | 51 |
| 4.10.2.14 | pds | 52 |
| 4.10.2.15 | pis | 52 |
| 4.10.2.16 | rb | 52 |
| 4.10.2.17 | snd | 53 |
| 4.10.2.18 | trp | 53 |
| 4.11 | main.c File Reference | 54 |
| 4.11.1 | Detailed Description | 54 |
| 4.11.2 | Function Documentation | 54 |
| 4.11.2.1 | main | 54 |
| 4.12 | tabuleiro.c File Reference | 55 |
| 4.12.1 | Detailed Description | 55 |
| 4.12.2 | Function Documentation | 55 |
| 4.12.2.1 | criarTabuleiro | 55 |
| 4.12.2.2 | dimensao | 55 |
| 4.12.2.3 | ficheiroValido | 56 |
| 4.12.2.4 | freeTab | 56 |
| 4.12.2.5 | gravaJogo | 56 |
| 4.12.2.6 | lerficheiro | 56 |
| 4.12.2.7 | printTabuleiro | 57 |
| 4.13 | tabuleiro.h File Reference | 58 |
| 4.13.1 | Detailed Description | 58 |
| 4.13.2 | Typedef Documentation | 58 |
| 4.13.2.1 | Elem | 58 |
| 4.13.2.2 | Elemento | 58 |
| 4.13.3 | Function Documentation | 58 |
| 4.13.3.1 | criarTabuleiro | 58 |
| 4.13.3.2 | dimensao | 58 |
| 4.13.3.3 | ficheiroValido | 59 |
| 4.13.3.4 | freeTab | 59 |

| | | |
|----------|----------------|----|
| 4.13.3.5 | gravaJogo | 59 |
| 4.13.3.6 | lerficheiro | 60 |
| 4.13.3.7 | printTabuleiro | 60 |

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

| | |
|---|----|
| elemento (É constituída por dois chars, que correspondem às características de cada célula do tabuleiro, a cor e a letra) | 5 |
| erro | 6 |
| gravados (Estrutura onde é armazenada toda a informação relativa a movimentos e jogadas guardadas) | 7 |
| It (Estrutura responsável por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada) | 8 |
| jogadas (Estrutura responsável por guardar a coluna, linha e cor de uma célula a gravar) | 9 |
| movimentos (Estrutura responsável por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada) | 10 |

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

| | |
|---|----|
| ajuda.c (Ajuda do jogo) | 11 |
| ajuda.h | 23 |
| comandos.c (Comandos do jogo) | 27 |
| comandos.h | 30 |
| erro.c | 32 |
| erro.h | 33 |
| estado_tabuleiro.c (Comandos de ajuda ao jogo) | 35 |
| estado_tabuleiro.h | 38 |
| jogadas.c (Jogadas possiveis de efectuar e gravar do jogo e) | 40 |
| jogadas.h | 47 |
| main.c (Main do jogo) | 54 |
| tabuleiro.c (Tabuleiro do jogo) | 55 |
| tabuleiro.h (Header file do tabuleiro.c) | 58 |

Chapter 3

Class Documentation

3.1 elemento Struct Reference

É constituída por dois chars, que correspondem às características de cada célula do tabuleiro, a cor e a letra.

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

Public Attributes

- char [letra](#)
Letra da célula do tabuleiro.
- char [cor](#)
Cor da célula do tabuleiro.

3.1.1 Detailed Description

É constituída por dois chars, que correspondem às características de cada célula do tabuleiro, a cor e a letra.

3.1.2 Member Data Documentation

3.1.2.1 elemento::cor

Cor da célula do tabuleiro.

3.1.2.2 elemento::letra

Letra da célula do tabuleiro.

The documentation for this struct was generated from the following file:

- [tabuleiro.h](#)

3.2 erro Struct Reference

Public Attributes

- char * [err_code](#)
- char * [mensagem](#)

3.2.1 Member Data Documentation

3.2.1.1 char* erro::err_code

3.2.1.2 char* erro::mensagem

The documentation for this struct was generated from the following file:

- [erro.c](#)

3.3 gravados Struct Reference

Estrutura onde é armazenada toda a informação relativa a [movimentos](#) e [jogadas](#) guardadas.

```
#include <jogadas.h>
```

Public Attributes

- [int](#) [totalGr](#)
- [IT](#) * [its](#)
- [Jogadas](#) * [jgs](#)
- [Movimentos](#) * [mvs](#)

3.3.1 Detailed Description

Estrutura onde é armazenada toda a informação relativa a [movimentos](#) e [jogadas](#) guardadas.

3.3.2 Member Data Documentation

3.3.2.1 [IT](#)* [gravados::its](#)

3.3.2.2 [Jogadas](#)* [gravados::jgs](#)

3.3.2.3 [Movimentos](#)* [gravados::mvs](#)

3.3.2.4 [int](#) [gravados::totalGr](#)

The documentation for this struct was generated from the following file:

- [jogadas.h](#)

3.4 It Struct Reference

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

```
#include <jogadas.h>
```

Public Attributes

- int [pos](#)
- struct [It](#) * [next](#)

3.4.1 Detailed Description

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

3.4.2 Member Data Documentation

3.4.2.1 struct [It](#)* [It::next](#) [[read](#)]

3.4.2.2 int [It::pos](#)

The documentation for this struct was generated from the following file:

- [jogadas.h](#)

3.5 jogadas Struct Reference

Estrutura responsavel por guardar a coluna,linha e cor de uma celula a gravar.

```
#include <jogadas.h>
```

Public Attributes

- int [col](#)
- int [row](#)
- char [color](#)
- struct [jogadas](#) * [next](#)

3.5.1 Detailed Description

Estrutura responsavel por guardar a coluna,linha e cor de uma celula a gravar.

3.5.2 Member Data Documentation

3.5.2.1 int [jogadas::col](#)

3.5.2.2 char [jogadas::color](#)

3.5.2.3 struct [jogadas](#)* [jogadas::next](#) [[read](#)]

3.5.2.4 int [jogadas::row](#)

The documentation for this struct was generated from the following file:

- [jogadas.h](#)

3.6 movimentos Struct Reference

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

```
#include <jogadas.h>
```

Public Attributes

- int [pos](#)
- struct [movimentos](#) * [next](#)

3.6.1 Detailed Description

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

3.6.2 Member Data Documentation

3.6.2.1 `struct movimentos* movimentos::next` [[read](#)]

3.6.2.2 `int movimentos::pos`

The documentation for this struct was generated from the following file:

- [jogadas.h](#)

Chapter 4

File Documentation

4.1 ajuda.c File Reference

```
ajuda do jogo #include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <ctype.h>
#include "tabuleiro.h"
#include "erro.h"
#include "jogadas.h"
#include "estado_tabuleiro.h"
#include "ajuda.h"
```

Defines

- #define [CHUNK_SIZE](#) 32

Functions

- int [ajuda](#) ()
- int [p_brancas](#) (Elem **Tab, int i, int j, int *DIM, [Gravados](#) *gr, int [erro](#))
- int [p_preta](#) (Elem **Tab, int i, int j, int *DIM, [Gravados](#) *gr, int [erro](#))
- int [verificaAB](#) (int x, int y, Elem **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr, int *cores)
- int [brancasAuto](#) (int x, int y, Elem **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr, int *cores)
- int [verificaAP](#) (int x, int y, Elem **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr, int *cores)
- int [pretasAuto](#) (int x, int y, Elem **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr, int *cores)
- int [auto_black](#) (Elem **Tab, int *DIM, [Gravados](#) *gr, int [erro](#))
- int [auto_white](#) (Elem **Tab, int *DIM, [Gravados](#) *gr, int [erro](#))
- int [control](#) (Elem **Tab, int *DIM, int *flagAB, int *flagAP, [Gravados](#) *gr, int [erro](#))

- void `automatico` (`Elem **Tab`, `int *DIM`, `Gravados *gr`, `int *flagAP`, `int *flagAB`, `int x`, `int y`, `int *cores`)
- int `change_tc` (`int i`, `int j`, `Elem **Tab`, `int *DIM`, `Gravados *gr`, `int *flagAP`, `int *flagAB`, `int *cores`)
- int `tc_auto` (`Elem **Tab`, `int *DIM`, `Gravados *gr`, `int *flagAP`, `int *flagAB`)
- int `tc` (`Elem **Tab`, `int *DIM`, `int x`, `int y`, `Gravados *gr`, `int *flagAP`, `int *flagAB`)
- int `executa_tc` (`char *args`, `int *DIM`, `Elem **Tab`, `int *flagAP`, `int *flagAB`, `Gravados *gr`)
- int `vc` (`Elem **Tab`, `int *DIM`, `Gravados *gr`)
- int `impacto` (`Elem **Tab`, `int *DIM`, `int i`, `int j`, `char *cor`, `Gravados *gr`)
- int `imp` (`Elem **Tab`, `int *DIM`, `Gravados *gr`, `int *flagAP`, `int *flagAB`)
- void `corPossivel` (`int x`, `int y`, `Elem **Tab`, `int *DIM`, `Gravados *gr`, `int *brancas`, `int *pretas`)
- int `tentativa` (`int x`, `int y`, `Elem **Tab`, `int *DIM`, `Gravados *gr`, `int *brancas`, `int *pretas`)
- int `tentarFinais` (`Elem **Tab`, `int *DIM`, `Gravados *gr`, `int *x`, `int *y`)
- int `resolve` (`Elem **Tab`, `int *DIM`, `Gravados *gr`, `int x`, `int y`)
- int `rsv` (`Elem **Tab`, `int *DIM`, `Gravados *gr`)
- `Elem * copy_board` (`Elem **Tab`, `int *DIM`, `Gravados *gr`)
- int `corrige` (`Elem **Tab`, `Gravados *gr`, `char cor`, `int x`, `int y`)
- int `crg` (`Elem **Tab`, `int *DIM`, `Gravados *gr`)

4.1.1 Detailed Description

ajuda do jogo

Author:

André Pimenta, João Gomes, Nelson Carvalho

Date:

1 Maio de 2010

4.1.2 Define Documentation

4.1.2.1 #define CHUNK_SIZE 32

4.1.3 Function Documentation

4.1.3.1 int ajuda ()

Função que vai imprimir as intruções de ajuda guardada no ficheiro com as instruções de ajuda.

4.1.3.2 int auto_black (Elem ** Tab, int * DIM, Gravados * gr, int erro)

A função `auto_black` procura no tabuleiro uma letra preta e quando encontra chama a função `p_brancas`,

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

erro flag que permite ou não enviar mensagens de `erro`

gr apontador para a estrutura de dados onde se encontram guardados todos os `movimentos` e `jogadas`.

Returns:

Retorna um inteiro que irá indicar se houve ou não alguma alteração no tabuleiro.

4.1.3.3 int auto_white (Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int *erro*)

A função auto_white procura no tabuleiro uma letra branca e quando encontra chama a função p_preta,

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

erro flag que permite ou não enviar mensagens de erro

gr apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

Returns:

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

4.1.3.4 void automatico (Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int * *flagAP*, int * *flagAB*, int *x*, int *y*, int * *cores*)

Funcao que recebe as coordenadas de uma celula do tabuleiro e verifica se nessas coordenadas a célula é branca ou preta e se a respectiva flag está activa e caso esteja chama a funcao brancasAuto/pretasAuto.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

flagAB flag do automático branco.

flagAP flag do automático preto.

x coordenada da linha.

y coordenada da coluna.

cores apontador para um inteiro que irá guardar o num de cores que foram alteradas por esta acção

4.1.3.5 int brancasAuto (int *x*, int *y*, Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*, int * *cores*)

A função brancasAuto, infere a cor branca à célula, e chama a função vericaAB para ver se pode inferir outras cores mediante a cor inferida.

Parameters:

x coordenada das linhas.

y coordenada das colunas.

Tab tabuleiro.

DIM dimensão do tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

cores apontador para um inteiro que irá guardar o num de cores que foram alteradas por esta acção

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar se houve ou não alguma alteração no tabuleiro.

4.1.3.6 int change_tc (int *i*, int *j*, Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int * *flagAP*, int * *flagAB*, int * *cores*)

Funcao que testa se não posição que é passada pode ser inserida a cor preta a célula ou cor branca, ou as duas ou nenhum, caso apenas uma delas possa ser inferida a função infere essa mesma cor.

Parameters:

i coordenada da linha.

j coordenada da coluna.

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAB flag do automático branco.

flagAP flag do automático preto.

cores apontador para um inteiro que irá guardar o num de cores que foram alteradas por esta acção

Returns:

Retorna um inteiro que irá indicar que foi introduzida a cor possivel ou se não foi possivel inferir qualquer cor.

4.1.3.7 int control (Elem ** *Tab*, int * *DIM*, int * *flagAB*, int * *flagAP*, Gravados * *gr*, int *erro*)

A função control recebe, o tabuleiro as dimensões do mesmo, bem como as flags dos automaticos, e dependendo das flags que estão activas chama auto_white e/ou auto_black sucessivamente até que estas já não façam nenhuma alteração no tabuleiro

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

erro flag que permite ou não enviar mensagens de [erro](#)

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possiveis.

4.1.3.8 Elem* copy_board (Elem ** *Tab*, int * *DIM*, Gravados * *gr*)

A função 'copy_board' faz uma cópia do tabuleiro original mas coloca todas as células com uma cor indefinida.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que foi inferida uma cor à célula de maior impacto.

4.1.3.9 void corPossivel (int *x*, int *y*, Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int * *brancas*, int * *pretas*)

A função 'corPossivel' vai colocar quantas células muda a cor naquela célula, ou seja vai tentar branco e conta quantas mudou, se não mudar nenhuma devido a ser jogada impossível pois nos apontadores para as mudadas correspondentes. Pode-se então dizer a funcionalidade desta função é dar a cor/cores possíveis de serem [jogadas](#) na célula em questão e o número de alterações que esta provoca.

Parameters:

x linha da célula

y coluna da célula

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAB flag do automático branco.

flagAP flag do automático preto.

brancas apontador destinado a armazenar o número de células mudadas caso a célula com as coordenadas *x* e *y* seja branca.

pretas apontador destinado a armazenar o número de células mudadas caso a célula com as coordenadas *x* e *y* seja preta.

Returns:

Retorna um inteiro que irá indicar que foi inferida uma cor à célula de maior impacto.

4.1.3.10 int corrige (Elem ** *Tab*, Gravados * *gr*, char *cor*, int *x*, int *y*)

A função corrige irá anular [movimentos](#) até que a célula em causa tenha cor indefinida ou a cor do tabuleiro final, que é passada como argumento.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

cor cor a que deve corresponder a célula no tab final

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar se foi corrigida ou já não dá pra corrigir mais.

4.1.3.11 int *crg* (Elem ** *Tab*, int * *DIM*, Gravados * *gr*)

A função '*crg*' chama a função '*copy_board*' que retorna a um tabuleiro igual mas resolvido, depois percorre o tabuleiro original a procura de incompatibilidade, quando encontra uma célula com uma cor diferente da solução retrocede até ao ponto antes da cor ter sido inferida a essa célula, faz o mesmo para todas as células até que todas as células pintadas estejam de acordo com a solução.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que o tabuleiro foi corrigido.

4.1.3.12 int *executa_tc* (char * *args*, int * *DIM*, Elem ** *Tab*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

Função que tal como o nome indica executa a função tc, esta funcao e responsavel por ver s a funcao tc irá receber a célula especifica ou caso contrario indica a funcao tc que irá ter de percorrer todo o tabuleiro

Parameters:

args argumentos a tratar.

DIM dimensão do tabuleiro.

Tab tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possíveis.

4.1.3.13 int *imp* (Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int * *flagAP*, int * *flagAB*)

A função '*imp*' percorre o tabuleiro e onde a cor for indefinida chama a função '*impaco*' que lhe irá retorna o num de células mudadas e vai comparando e guardando caso a célula seguinte tenha alterado mais célula que a anterior, e no fim depois de o tabuleiro todo percorrido infere a cor a célula com mais impacto.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAB flag do automático branco.

flagAP flag do automático preto.

Returns:

Retorna um inteiro que irá indicar que foi inferida uma cor à célula de maior impacto.

4.1.3.14 int impacto (Elem ** *Tab*, int * *DIM*, int *i*, int *j*, char * *cor*, Gravados * *gr*)

Função que vai dar o numero de celulas que são mudadas, devido à célula com as cordenadas *j* e *i*, mudarem de cor, passa tambem para o apontador *cor*, qual a cor que mudou. Para isso a função vai executar a *tc* e ve se muda a cor e se mudar tem na variavel mudadas o total de cores mudadas, após isto voçta a por o tabuleiro como estava, apenas serve para dar o "impacto" de uma cor no tabuleiro

Parameters:

cor apontador para a cor que provaca mudança nas outras.

i linha da celula

j coluna da celula

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possiveis.

4.1.3.15 int p_branca (Elem ** *Tab*, int *i*, int *j*, int * *DIM*, Gravados * *gr*, int *erro*)

A função *p_branca* pinta de branco todas as letras à volta da letra da posição (*i,j*).

Parameters:

Tab tabuleiro.

i linha.

j coluna.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

erro flag que permite ou não enviar mensagens de [erro](#)

Returns:

Retorna um inteiro que irá indicar se houve ou não alguma alteração no tabuleiro.

4.1.3.16 int p_preta (Elem ** *Tab*, int *i*, int *j*, int * *DIM*, Gravados * *gr*, int *erro*)

A função p_preta procura na linha (*i*) e na coluna (*j*) uma letra igual á letra passada e pinta essa letra de preto.

Parameters:

Tab tabuleiro.

i linha.

j coluna.

letra letra.

DIM dimensão do tabuleiro.

erro flag que permite ou não enviar mensagens de [erro](#)

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar se houve ou não alguma alteração no tabuleiro.

4.1.3.17 int pretasAuto (int *x*, int *y*, Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*, int * *cores*)

A função pretasAuto, infere a cor preta a celula, e chama a função vericaAP para ver s pode inferir outras cores sobre a cor inferida

Parameters:

x coordenada das linhas.

y coordenada das colunas.

Tab tabuleiro.

DIM dimensão do tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

cores apontador para um inteiro que irá guardar o num de cores que foram alteradas por esta acção

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

4.1.3.18 int resolve (Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int *x*, int *y*)

A função 'resolve' é chamada quando existe uma celulas que não são pintadas pela tc, ou seja que no estado actual do tabuleiro podem ter qualquer cor. Ela vai tentar pintar de branco e chama a rsv, que vai pintar o resto, caso, não seja a cor, retocede e pinta de preto, esta função vai ser chamada para as celulas que podem ter as duas cores no estado actual do tabuleiro.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

x coordenada do linha

y coordenada da coluna

Returns:

Retorna um inteiro

4.1.3.19 int rsv (Elem ** *Tab*, int * *DIM*, Gravados * *gr*)

A função 'rsv' resolve o tabuleiro, para isso vai chamar a tentar finais , onde são atribuídas as cores finais do tabuleiro que conseguimos saber no momento, caso ainda não esteja resolvido devolve caso tal seja possível as coordenadas de uma célula que pode ter as duas cores e caso seja brnaca tem maior impacto no tabuleiro. Esta célula vai ser passada a função resolve que vai lhe atribuir a cor final, a função resolve funciona de forma recursiva e vai chamar a rsv após atribuir a cor, para que esta atribua as cores as novas células desbloqueadas. Este processo repete-se até o tabuleiro estar resolvido.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que indica se o tabuleiro tem ou não resolução.

4.1.3.20 int tc (Elem ** *Tab*, int * *DIM*, int *x*, int *y*, Gravados * *gr*, int * *flagAP*, int * *flagAB*)

Funcao que tenta inferir uma cor a célula indicada, segundo as regras do jogo

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAB flag do automático branco.

flagAP flag do automático preto.

Returns:

Retorna um inteiro que irá indicar que foi introduzida a cor possível.

4.1.3.21 int tc_auto (Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int * *flagAP*, int * *flagAB*)

Funcao chamada pela função 'tc' caso esta não receba uma posição específica, percorre todo o tabuleiro a procura de cores indefinidas e chama a funcao change_tc para tentar inferir uma cor a essa célula

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAB flag do automático branco.

flagAP flag do automático preto.

Returns:

Retorna um inteiro que irá indicar que foi percorrido todo o tabuleiro.

4.1.3.22 int tentarFinais (Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int * *x*, int * *y*)

A função 'tentarFinais' vai pintando as cores finais possíveis de adivinhar (comando tc), das que não são possíveis as duas vai guardando a que tem maior impacto ao ser branca, se existir uma célula que não pode ter nenhuma cor, então irá returnar o [erro](#) em return -1;

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

x onde irá guardar a coordenada da linha

y onde irá guardar a coordenada da coluna

Returns:

Retorna um inteiro que irá indicar se foi ou não possível inferir uma cor na célula.

4.1.3.23 int tentativa (int *x*, int *y*, Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int * *brancas*, int * *pretas*)

A função 'tentativa' testa se qual a cor a poder ser inferida, e caso seja possível inferir apenas uma única cor, esta é inferida.

Parameters:

x coordenada da linha.

y coordenada da coluna.

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

brancas apontador destinado a armazenar o numero de células mudadas caso a célula com as coordenadas *x* e *y* seja branca.

pretas apontador destinado a armazenar o numero de células mudadas caso a célula com as coordenadas *x* e *y* seja preta.

Returns:

Retorna um inteiro que irá indicar que foi inferida uma cor à célula de maior impacto.

4.1.3.24 int vc (Elem ** *Tab*, int * *DIM*, Gravados * *gr*)

A função vc percorre o tabuleiro e aplica as regras das pretas e das brancas ate que não seja possivel aplicar mais nenhuma regra

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possiveis.

4.1.3.25 int verificaAB (int *x*, int *y*, Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*, int * *cores*)

A função verificaAB verifica se pinta a nova celula ou se esta já está pintada.

Parameters:

x coordenada das linhas.

y coordenada das colunas.

Tab tabuleiro.

DIM dimensão do tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

cores apontador para um inteiro que irá guardar o num de cores que foram alteradas por esta acção

Returns:

Retorna um inteiro que irá indicar se houve ou não alguma alteração no tabuleiro.

4.1.3.26 int verificaAP (int *x*, int *y*, Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*, int * *cores*)

A função verificaAP verifica se pinta a nova celula ou se esta já está pintada.

Parameters:

x coordenada das linhas.

y coordenada das colunas.

Tab tabuleiro.

DIM dimensão do tabuleiro.

flagAB flag do automático branco.

cores apontador para um inteiro que irá guardar o num de cores que foram alteradas por esta acção

flagAP flag do automático preto.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar se houve ou não alguma alteração no tabuleiro.

4.2 ajuda.h File Reference

Functions

- int [pretasAuto](#) (int x, int y, [Elem](#) **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr, int *cores)
- int [brancasAuto](#) (int x, int y, [Elem](#) **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr, int *cores)
- int [ajuda](#) ()
- int [rsv](#) ([Elem](#) **Tab, int *DIM, [Gravados](#) *gr)
- int [control](#) ([Elem](#) **Tab, int *DIM, int *flagAB, int *flagAP, [Gravados](#) *gr, int erro)
- int [executa_tc](#) (char *args, int *DIM, [Elem](#) **Tab, int *flagAP, int *flagAB, [Gravados](#) *gr)
- int [vc](#) ([Elem](#) **Tab, int *DIM, [Gravados](#) *gr)
- int [count_i](#) ([Elem](#) **Tab, int *DIM)
- int [imp](#) ([Elem](#) **Tab, int *DIM, [Gravados](#) *gr, int *flagAP, int *flagAB)
- int [crg](#) ([Elem](#) **Tab, int *DIM, [Gravados](#) *gr)

4.2.1 Function Documentation

4.2.1.1 int ajuda ()

Função que vai imprimir as intruções de ajuda guardada no ficheiro com as instruções de ajuda.

4.2.1.2 int brancasAuto (int x, int y, [Elem](#) ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, [Gravados](#) * *gr*, int * *cores*)

A função brancasAuto, infere a cor branca à célula, e chama a função vericaAB para ver se pode inferir outras cores mediante a cor inferida.

Parameters:

x coordenada das linhas.

y coordenada das colunas.

Tab tabuleiro.

DIM dimensão do tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

cores apontador para um inteiro que irá guardar o num de cores que foram alteradas por esta acção

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar se houve ou não alguma alteração no tabuleiro.

4.2.1.3 int control ([Elem](#) ** *Tab*, int * *DIM*, int * *flagAB*, int * *flagAP*, [Gravados](#) * *gr*, int erro)

A função control recebe, o tabuleiro as dimensões do mesmo, bem como as flags dos automaticos, e dependendo das flags que estão activas chama auto_white e/ou auto_black sucessivamente até que estas já não façam nenhuma alteração no tabuleiro

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

erro flag que permite ou não enviar mensagens de [erro](#)

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possíveis.

4.2.1.4 int count_i (Elem ** *Tab*, int * *DIM*)**4.2.1.5 int crg (Elem ** *Tab*, int * *DIM*, Gravados * *gr*)**

A função 'crg' chama a função 'copy_board' que retorna a um tabuleiro igual mas resolvido, depois percorre o tabuleiro original a procura de incompatibilidade, quando encontra uma célula com uma cor diferente da solução retrocede até ao ponto antes da cor ter sido inferida a essa célula, faz o mesmo para todas as células até que todas as células pintadas estejam de acordo com a solução.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que o tabuleiro foi corrigido.

4.2.1.6 int executa_tc (char * *args*, int * *DIM*, Elem ** *Tab*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

Função que tal como o nome indica executa a função tc, esta funcao e responsavel por ver se a funcao tc irá receber a celula especifica ou caso contrario indica a funcao tc que irá ter de percorrer todo o tabuleiro

Parameters:

args argumentos a tratar.

DIM dimensão do tabuleiro.

Tab tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possíveis.

4.2.1.7 int imp (Elem ** *Tab*, int * *DIM*, Gravados * *gr*, int * *flagAP*, int * *flagAB*)

A função 'imp' percorre o tabuleiro e onde a cor for indefinida chama a função 'impaco' que lhe irá retorna o num de células mudadas e vai comparando e guardando caso a célula seguinte tenha alterado mais célula que a anterior, e no fim depois de o tabuleiro todo percorrido infere a cor a célula com mais impacto.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAB flag do automático branco.

flagAP flag do automático preto.

Returns:

Retorna um inteiro que irá indicar que foi inferida uma cor à célula de maior impacto.

4.2.1.8 int pretasAuto (int *x*, int *y*, Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*, int * *cores*)

A função pretasAuto, infere a cor preta a célula, e chama a função vericaAP para ver se pode inferir outras cores sobre a cor inferida

Parameters:

x coordenada das linhas.

y coordenada das colunas.

Tab tabuleiro.

DIM dimensão do tabuleiro.

flagAB flag do automático branco.

flagAP flag do automático preto.

cores apontador para um inteiro que irá guardar o num de cores que foram alteradas por esta acção

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

4.2.1.9 int rsv (Elem ** *Tab*, int * *DIM*, Gravados * *gr*)

A função 'rsv' resolve o tabuleiro, para isso vai chamar a tentar finais , onde são atribuídas as cores finais do tabuleiro que conseguimos saber no momento, caso ainda não esteja resolvido devolve caso tal seja possível as coordenadas de uma célula que pode ter as duas cores e caso seja brnaca tem maior impacto no tabuleiro. Esta célula vai ser passada a função resolve que vai lhe atribuir a cor final, a função resolve funciona de forma recursiva e vai chamar a rsv após atribuir a cor, para que esta atribua as cores as novas células desbloqueadas. Este processo repete até o tabuleiro estar resolvido.

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que indica se o tabuleiro tem ou não resolução.

4.2.1.10 int vc (Elem ** *Tab*, int * *DIM*, Gravados * *gr*)

A função *vc* percorre o tabuleiro e aplica as regras das pretas e das brancas ate que não seja possivel aplicar mais nenhuma regra

Parameters:

Tab tabuleiro.

DIM dimensão do tabuleiro.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

Returns:

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possiveis.

4.3 comandos.c File Reference

```
Comandos do jogo. #include "tabuleiro.h"
#include "erro.h"
#include "jogadas.h"
#include "ajuda.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include "estado_tabuleiro.h"
```

Functions

- int [executaJogada](#) (char *args, char *cor, int *DIM, [Elem](#) **Tab, int *flagAP, int *flagAB, [Gravados](#) *gr)
- int [executaCriaJogo](#) (char *args, int *DIM, [Elem](#) **Tab, [Gravados](#) *gr, int *flagAP, int *flagAB)
- int [executaGravaJogo](#) (char *args, int *DIM, [Elem](#) **Tab)
- int [executa_comando](#) (char *linha, int *DIM, [Elem](#) **Tab, int *flagAP, int *flagAB, [Gravados](#) *gr)

4.3.1 Detailed Description

Comandos do jogo.

Author:

André Pimenta, João Gomes, Nelson Carvalho

Date:

1 Abril de 2010

4.3.2 Function Documentation

4.3.2.1 int [executa_comando](#) (char * *linha*, int * *DIM*, [Elem](#) ** *Tab*, int * *flagAP*, int * *flagAB*, [Gravados](#) * *gr*)

A função [executa_comando](#) é uma das funções mais importantes pois é ela que vai reagir aos comandos dados pelo utilizador. A função irá receber a linha de comando separa la em dois, no comando e nos argumentos do comando, e passa los para as funções associadas aos comandos.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

linha Recebe uma string que corresponde á linha lida da shell.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as jogadas e moviemntos efectuados.

Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

4.3.2.2 int executaCriaJogo (char * args, int * DIM, Elem ** Tab, Gravados * gr, int * flagAP, int * flagAB)

A função executaCriaJog recebe os argumentos da linha de comando, vai verificar se tem os argumentos, certos se tal ocorre, executa criarJogo com o nome do argumento no formato ".ltr".

Parameters:

args Recebe uma string com os argumentos associados ao comando em questão ("cr").

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

gr apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

4.3.2.3 int executaGravaJogo (char * args, int * DIM, Elem ** Tab)

A função executaGravaJogo recebe os argumentos da linha de comando e vai verificar se tem os argumetos certos, se tal for verdade, executa gravaJogo com o nome do argumento no formato ".ltr".

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

args Recebe uma string com os argumentos associados ao comando em questão("gr").

Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

4.3.2.4 int executaJogada (char * args, char * cor, int * DIM, Elem ** Tab, int * flagAP, int * flagAB, Gravados * gr)

A função executajogada recebe os argumentos da linha de comandos e a cor que vai passar para a jogada. Vai verificar se tem os argumetos certos, e se tal vou verdade, chama a função jogada, para que esta se possa realizar.

Parameters:

args Recebe uma string com os argumentos do comando do utilizador.

cor Corresponde ao próprio comando que irá ser utilizado como a cor que se pretende mudar.

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

Returns:

Retorna um inteiro que verifica se foi ou não bem sucedida a função.

4.4 comandos.h File Reference

Functions

- int [executaJogada](#) (char *args, char *cor, int *DIM, [Elem](#) **Tab, int *flagAP, int *flagAB, [Gravados](#) *gr)
- int [executaCriaJogo](#) (char *args, int *DIM, [Elem](#) **Tab, [Gravados](#) *gr, int *flagAP, int *flagAB)
- int [executaGravaJogo](#) (char *args, int *DIM, [Elem](#) **Tab)
- int [executa_comando](#) (char *linha, int *DIM, [Elem](#) **Tab, int *flagAP, int *flagAB, [Gravados](#) *gr)

4.4.1 Function Documentation

4.4.1.1 int [executa_comando](#) (char * *linha*, int * *DIM*, [Elem](#) ** *Tab*, int * *flagAP*, int * *flagAB*, [Gravados](#) * *gr*)

A função [executa_comando](#) é uma das funções mais importantes pois é ela que vai reagir aos comandos dados pelo utilizador. A função irá receber a linha de comando separa la em dois, no comando e nos argumentos do comando, e passa los para as funções associadas aos comandos.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

linha Recebe uma string que corresponde á linha lida da shell.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as [jogadas](#) e moviemntos efectuados.

Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

4.4.1.2 int [executaCriaJogo](#) (char * *args*, int * *DIM*, [Elem](#) ** *Tab*, [Gravados](#) * *gr*, int * *flagAP*, int * *flagAB*)

A função [executaCriaJog](#) recebe os argumentos da linha de comando, vai verificar se tem os argumentos, certos se tal ocorre, executa [criarJogo](#) com o nome do argumento no formato ".ltr".

Parameters:

args Recebe uma string com os argumentos associados ao comando em questão ("cr").

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

4.4.1.3 int executaGravaJogo (char * *args*, int * *DIM*, Elem ** *Tab*)

A função executaGravaJogo recebe os argumentos da linha de comando e vai verificar se tem os argumetos certos, se tal for verdade, executa gravaJogo com o nome do argumento no formato ".ltr".

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

args Recebe uma string com os argumentos associados ao comando em questão("gr").

Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

4.4.1.4 int executaJogada (char * *args*, char * *cor*, int * *DIM*, Elem ** *Tab*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

A função executajogada recebe os argumentos da linha de comandos e a cor que vai passar para a jogada. Vai verificar se tem os argumetos certos, e se tal vou verdade, chama a função jogada, para que esta se possa realizar.

Parameters:

args Recebe ums string com os argumentos do comando do utilizador.

cor Corresponde ao próprio comado que irá ser utilizado como a cor que se pretende mudar.

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

Returns:

Retorna um inteiro que verifica se foi ou não bem sucedida a função.

4.5 erro.c File Reference

```
#include <stdio.h>
#include "erro.h"
```

Classes

- struct [erro](#)

Typedefs

- typedef struct [erro](#) [ERRO](#)

Functions

- int [mensagem_de_erro](#) (int *erro_num*)

4.5.1 Typedef Documentation

4.5.1.1 typedef struct erro ERRO

4.5.2 Function Documentation

4.5.2.1 int mensagem_de_erro (int *erro_num*)

Reporta o [erro](#) escrevendo uma mensagem no standard error

Parameters:

erro_num O número do [erro](#) que vem da tabela de erros. Usar SEMPRE o define e não o valor numérico para maior legibilidade.

4.6 erro.h File Reference

Defines

- #define [E_NOERR](#) 0
- #define [E_COMMAND](#) 1
- #define [E_ARGS](#) 2
- #define [E_NO_BOARD](#) 3
- #define [E_NO_FILE](#) 4
- #define [E_FILE_FORMAT](#) 5
- #define [E_WRONG_SOLUTION](#) 6
- #define [E_NO_MOVES](#) 7
- #define [E_COORDS](#) 8
- #define [E_SAVE](#) 9
- #define [E_INVARGS](#) 10

Functions

- int [mensagem_de_erro](#) (int erro_num)

4.6.1 Define Documentation

4.6.1.1 #define [E_ARGS](#) 2

4.6.1.2 #define [E_COMMAND](#) 1

4.6.1.3 #define [E_COORDS](#) 8

4.6.1.4 #define [E_FILE_FORMAT](#) 5

4.6.1.5 #define [E_INVARGS](#) 10

4.6.1.6 #define [E_NO_BOARD](#) 3

4.6.1.7 #define [E_NO_FILE](#) 4

4.6.1.8 #define [E_NO_MOVES](#) 7

4.6.1.9 #define [E_NOERR](#) 0

4.6.1.10 #define [E_SAVE](#) 9

4.6.1.11 #define [E_WRONG_SOLUTION](#) 6

4.6.2 Function Documentation

4.6.2.1 int [mensagem_de_erro](#) (int *erro_num*)

Reporta o [erro](#) escrevendo uma mensagem no standard error

Parameters:

erro_num O número do [erro](#) que vem da tabela de erros. Usar SEMPRE o define e não o valor numérico para maior legibilidade.

4.7 estado_tabuleiro.c File Reference

Comandos de ajuda ao jogo. #include "tabuleiro.h"

#include "erro.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

Functions

- int [vb](#) ([Elem](#) **Tab, int *DIM, int [erro](#))
- int [vp](#) ([Elem](#) **Tab, int *DIM, int [erro](#))
- void [free_tab](#) (int **tab_aux, int *DIM)
- int [prepara_tab](#) ([Elem](#) **Tab, int **tab_aux, int *DIM)
- int [caminho](#) (int **tab_aux, int *DIM)
- int [vl](#) ([Elem](#) **Tab, int *DIM, int [erro](#))
- int [st](#) ([Elem](#) **Tab, int *DIM, int [erro](#))

4.7.1 Detailed Description

Comandos de ajuda ao jogo.

Author:

André Pimenta, João Gomes, Nelson Carvalho

Date:

3 Abril de 2010

4.7.2 Function Documentation

4.7.2.1 int [caminho](#) (int ** *tab_aux*, int * *DIM*)

4.7.2.2 void [free_tab](#) (int ** *tab_aux*, int * *DIM*)

Função que procura a existencia de caminho entre todas as células brABas ou indefinidas. Caso não exista retorna uma mensagem de [erro](#).

Parameters:

tab_aux Tabuleiro do jogo

DIM dimensão do tabuleiro

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

4.7.2.3 int prepara_tab (Elem ** *Tab*, int ** *tab_aux*, int * *DIM*)**4.7.2.4 int st (Elem ** *Tab*, int * *DIM*, int *erro*)**

Função que aplica todos os teste ao mesmo tempo de verificação do estado do tabuleiro.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

erro flag que avisa se deve emitir mensagem de *erro* ou não para o utilizador

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um *erro*,é exibido atraves da função mensagem de *erro*.

4.7.2.5 int vb (Elem ** *Tab*, int * *DIM*, int *erro*)

Função que percorre todas as posições do tabuleiro caso sejam pretas, verifica se não têm vizinas na vertical ou na horizontal com a cor preta. Caso existam retorna uma mensagem de *erro*.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

erro flag que avisa se deve emitir mensagem de *erro* ou não para o utilizador

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um *erro*,é exibido atraves da função mensagem de *erro*.

4.7.2.6 int vl (Elem ** *Tab*, int * *DIM*, int *erro*)

Função que percorre todas as posições do tabuleiro e verifica se existe um caminho entre todas elas(indefinidas ou brancas)

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

erro flag que avisa se deve emitir mensagem de *erro* ou não para o utilizador

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um *erro*,é exibido atraves da função mensagem de *erro*.

4.7.2.7 int vp (Elem ** *Tab*, int * *DIM*, int *erro*)

Função que percorre uma linha/coluna e ve se existem duas letras iguais com mesma cor(brABa), fazendo o mesmo para todas as linhas/colunas. Caso existam retorna uma mensagem de [erro](#).

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

erro flag que avisa se deve emitir mensagem de [erro](#) ou não para o utilizador

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

4.8 estado_tabuleiro.h File Reference

Functions

- int [vb](#) ([Elem](#) **Tab, int *DIM, int [erro](#))
- int [vp](#) ([Elem](#) **tab, int *DIM, int [erro](#))
- int [vl](#) ([Elem](#) **Tab, int *DIM, int [erro](#))
- int [st](#) ([Elem](#) **Tab, int *DIM, int [erro](#))
- int [caminho](#) (int *tab_aux, int *DIM)
- int [prepara_tab](#) ([Elem](#) **Tab, int **tab_aux, int *DIM)
- void [free_tab](#) (int **tab_aux, int *DIM)

4.8.1 Function Documentation

4.8.1.1 int [caminho](#) (int * *tab_aux*, int * *DIM*)

4.8.1.2 void [free_tab](#) (int ** *tab_aux*, int * *DIM*)

Função que procura a existencia de caminho entre todas as celulas brABas ou indefinidas. Caso não exista retorna uma mensagem de [erro](#).

Parameters:

tab_aux Tabuleiro do jogo

DIM dimensão do tabuleiro

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

4.8.1.3 int [prepara_tab](#) ([Elem](#) ** *Tab*, int ** *tab_aux*, int * *DIM*)

4.8.1.4 int [st](#) ([Elem](#) ** *Tab*, int * *DIM*, int *erro*)

Função que aplica todos os teste ao mesmo tempo de verificação do estado do tabuleiro.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

[erro](#) flag que avisa se deve emitir mensagem de [erro](#) ou não para o utilizador

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

4.8.1.5 int vb (Elem ** *Tab*, int * *DIM*, int *erro*)

Função que percorre todas as posições do tabuleiro caso sejam pretas, verifica se não têm vizinas na vertical ou na horizontal com a cor preta. Caso existam retorna uma mensagem de [erro](#).

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

erro flag que avisa se deve emitir mensagem de [erro](#) ou não para o utilizador

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

4.8.1.6 int vl (Elem ** *Tab*, int * *DIM*, int *erro*)

Função que percorre todas as posições do tabuleiro e verifica se existe um caminho entre todas elas (indefinidas ou brancas)

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

erro flag que avisa se deve emitir mensagem de [erro](#) ou não para o utilizador

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

4.8.1.7 int vp (Elem ** *Tab*, int * *DIM*, int *erro*)

Função que percorre uma linha/coluna e vê se existem duas letras iguais com mesma cor (brABa), fazendo o mesmo para todas as linhas/colunas. Caso existam retorna uma mensagem de [erro](#).

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

erro flag que avisa se deve emitir mensagem de [erro](#) ou não para o utilizador

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

4.9 jogadas.c File Reference

Jogadas possiveis de efectuar e gravar do jogo e. #include "tabuleiro.h"

```
#include "erro.h"
#include "jogadas.h"
#include "ajuda.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Functions

- int [jogadaManual](#) (char cor, int x, int y, int *DIM, [Elem **Tab](#), [Gravados *gr](#))
- int [jogada](#) (char cor, int x, int y, [Elem **Tab](#), [Gravados *gr](#), int [erro](#), int *cores)
- int [E_trp_col](#) ([Elem **Tab](#), int *DIM, int i, int j, [Gravados *gr](#))
- int [E_trp_row](#) ([Elem **Tab](#), int *DIM, int i, int j, [Gravados *gr](#))
- int [trp](#) ([Elem **Tab](#), int *DIM, int *flagAP, int *flagAB, [Gravados *gr](#))
- int [snd](#) ([Elem **Tab](#), int *DIM, int *flagAP, int *flagAB, [Gravados *gr](#))
- int [E_Par](#) ([Elem **Tab](#), int pos, char letra, int *DIM, char *local)
- int [pis](#) ([Elem **Tab](#), int *DIM, int *flagAP, int *flagAB, [Gravados *gr](#))
- int [E_Par_des](#) ([Elem **Tab](#), char letra1, char letra2, int *DIM, int line1, int cl1, int line2, int cl2)
- int [pds](#) ([Elem **Tab](#), int *DIM, int *flagAP, int *flagAB, [Gravados *gr](#))
- int [grava_jogada](#) (int row, int col, char color, [Gravados *gr](#))
- int [anc](#) ([Elem **Tab](#), [Gravados *gr](#))
- int [it](#) ([Gravados *gr](#))
- int [rb](#) ([Elem **Tab](#), [Gravados *gr](#))
- int [inicia_GR](#) ([Gravados *gr](#))
- int [grava_movimento](#) ([Gravados *gr](#))
- int [anm](#) ([Elem **Tab](#), [Gravados *gr](#))
- int [freeMvs](#) ([Gravados *gr](#))
- int [freeJgs](#) ([Gravados *gr](#))
- int [freeIts](#) ([Gravados *gr](#))
- int [free_jogadasGR](#) ([Gravados *gr](#))

4.9.1 Detailed Description

Jogadas possiveis de efectuar e gravar do jogo e.

Author:

André Pimenta, João Gomes, Nelson Carvalho

Date:

1 Maio de 2010

4.9.2 Function Documentation

4.9.2.1 int anc (Elem ** *Tab*, Gravados * *gr*)

Função responsável por anular uma cor guardada quando é chamada. A função vai anular a ultima cor guardada.

Parameters:

Tab tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.2 int anm (Elem ** *Tab*, Gravados * *gr*)

Função responsável por anular um movimento quando é chamada. A função vai anular cores até chegar á ultima cor gravada quando foi guardado o movimento.

Parameters:

Tab tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.3 int E_Par (Elem ** *Tab*, int *pos*, char *letra*, int * *DIM*, char * *local*)

Função auxiliar que verifica se existem pares de letras numa linha ou numa coluna Para tal vai percorrer a linha ou coluna e verifica se existe. A função tem em conta o caso de tres letras seguidas não ser um par mas sim um triplo.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

pos Linha ou coluna onde vai procurar os pares

letra a letra que se pretende saber se existe pares ou não

local Para decidir se se procura nas linhas ou nas colunas

Returns:

Inteiro que confirma se existe ou não um para na coluna/linha.

4.9.2.4 int E_Par_des (Elem ** *Tab*, char *letra1*, char *letra2*, int * *DIM*, int *line1*, int *cl1*, int *line2*, int *cl2*)

Função auxiliar que verifica se existem pares de letras tanto nas linhas como nas colunas passdas como parametro. Para tal vai percorrer a linha ou coluna e verifica se existe.

Parameters:

Tab Tabuleiro do jogo

letra1 Primeira letra da diagonal.

letra2 Segunda letra da diagonal

DIM dimensão do tabuleiro

line1 Linha da primeira letra

line2 Linha da segunda letra

cl1 Coluna da primeira letra

cl2 Coluna da segunda letra

Returns:

Inteiro que confirma se existe ou não um para na coluna/linha.

4.9.2.5 int E_trp_col (Elem ** *Tab*, int * *DIM*, int *i*, int *j*, Gravados * *gr*)

Função responsável por verificar se existe um triplo a partir das cordenadas *i* e *j* na vertical. Se existir atribui as cores correspondentes.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

i linha da célula.

j coluna da célula.

gr apontador para a estrutura de dados onde se guardas as [jogadas](#) e [movimentos](#).

Returns:

inteiro que irá retornar se houve alguma alteração no tabuleiro ou não

4.9.2.6 int E_trp_row (Elem ** *Tab*, int * *DIM*, int *i*, int *j*, Gravados * *gr*)

Função responsável por verficar se existe um triplo a partir das cordenadas *i* e *j* na horizontal. Se existir atribui as cores correspondentes.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

i linha da celula.

j coluna da celula.

gr apontador para a estrutura de dados onde se guardas as [jogadas](#) e [movimentos](#).

Returns:

inteiro que irá retornar se houve alguma alteração no tabuleiro ou não

4.9.2.7 int free_jogadasGR (Gravados * *gr*)

Função responsável por limpar da memória todos os dados guardados, em relação às cores e movimentos guardados.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.8 int freeIts (Gravados * *gr*)

Função responsável por limpar da memória todos os estados de tabuleiros marcados com o comando it.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.9 int freeJgs (Gravados * *gr*)

Função responsável por limpar da memória todas as [jogadas](#) guardadas em memória.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.10 int freeMvs (Gravados * *gr*)

Função responsável por limpar da memória todos os [movimentos](#) guardados em memória.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.11 int grava_jogada (int *row*, int *col*, char *color*, Gravados * *gr*)

Função que irá guardar a cor a linha e a coluna de um tabuleiro que correspondente à célula que vai ser substituída. Esta função será chamada sempre que se mude uma cor.

Parameters:

row linha da célula que se vai guardar.

col coluna da célula que se vai guardar.

color Cor da célula que vai ser substituída

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.12 int grava_movimento (Gravados * *gr*)

Função que irá guardar na estrutura de dados de [movimentos](#) o número de cores que foram guardadas até ao momento da sua chamada. Para posteriormente poder se recuperar o estado do tabuleiro.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.13 int inicia_GR (Gravados * *gr*)

Função reponsavel por inicializar e alocar espaço para as estruturas de dados responsaveis por guardar informação relativa a [movimentos](#) e [jogadas](#) efectuadas. Aloca espaço para [It](#), para moviemntos e para [jogadas](#)

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.14 int it (Gravados * *gr*)

Função resonsavel tal como a guarda [movimentos](#), de guardar na estrutura de dados de IT's o numero de cores que foram guardadas até ao momento da sua chamada. Para postriormente poder se recoperar o estado do tabuleiro.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.15 int jogada (char *cor*, int *x*, int *y*, Elem ** *Tab*, Gravados * *gr*, int *erro*, int * *cores*)

A função jogada é responsável por alterar os dados do tabuleiro que são atribuidos automaticamente. Ela irá receber o novo dado e as poisções onde vai ficar o novo dado, actualizando assim o tabuleiro com os novos dados

Parameters:

cor nova cor para um [elemento](#) do tabuleiro

x Corresponde á linha que vamos substituir no tabuleiro.

y Corresponde á coluna que vamos substituir no tabuleiro.

Tab Tabuleiro do jogo

gr apontador para a estrutura de dados onde se guardas as [jogadas](#) e [movimentos](#).

erro flag que avisa se deve emitir mensagem de [erro](#) ou não para o utilizador

cores contador de cores mudadas

Returns:

Retorna um inteiro que verifica se a jogada foi ou não bem sucedida.

4.9.2.16 int jogadaManual (char *cor*, int *x*, int *y*, int * *DIM*, Elem ** *Tab*, Gravados * *gr*)

A função jogada é responsável por alterar os dados do tabuleiro introduzidos pelo utilizador. Ela irá receber o novo dado e as poisções onde vai ficar o novo dado, actualizando assim o tabuleiro com os novos dados

Parameters:

cor nova cor para um [elemento](#) do tabuleiro

x Corresponde á linha que vamos substituir no tabuleiro.

y Corresponde á coluna que vamos substituir no tabuleiro.

DIM dimensão do tabuleiro

Tab Tabuleiro do jogo

gr apontador para a estrutura de dados onde se guardas as jogadas e movimentos.

Returns:

Retorna um inteiro que verifica se a jogada foi ou não bem sucedida.

4.9.2.17 int pds (Elem ** Tab, int * DIM, int * flagAP, int * flagAB, Gravados * gr)

Função responsável por ver se existem ou não pares isolados no tabuleiro. Para tal esta função vai pegar em quatro células, formando um quadrado. Neste quadrado verifica se as duas diagonais pertencem ou não a regra pares desencontrados, com a ajuda da função auxiliar EpParDes. Caso pertence ver se a diagonal contrária tem cores contrárias, se tiver dá uma mensagem de erro, se não muda as cores.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as jogadas e movimentos efectuados.

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um erro, é exibido através da função mensagem de erro.

4.9.2.18 int pis (Elem ** Tab, int * DIM, int * flagAP, int * flagAB, Gravados * gr)

Função que procura pares isolados no tabuleiro (numa coluna ou numa linha). Para tal vai percorrer todas as posições do tabuleiro, e ver se a letra é diferente das que a rodeiam na horizontal ou na vertical. Confirmando se a presença de uma letra isola, chama a função auxiliar Epar que vai ver se existe um par dessa letra na coluna ou na linha. Caso se verifiquem as duas condições irá ver se a letra não está de uma cor contrária à que se vai atribuir, caso não esteja atribui-se uma nova cor, se não uma mensagem de erro.

Parameters:

Tab Tabuleiro do jogo

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as jogadas e movimentos efectuados.

DIM dimensão do tabuleiro

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um erro, é exibido através da função mensagem de erro.

4.9.2.19 int rb (Elem ** *Tab*, Gravados * *gr*)

Função responsável por anular cores até chegar ao estado de tabuleiro marcado pela função it. Desta forma será possível retornar o estado do tabuleiro tal como foi marcado na função it.

Parameters:

Tab tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.9.2.20 int snd (Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

Função que procura sandes no tabuleiro(tres letras onde apenas a do meio é diferente na mesma linha/-coluna). Para tal vai se percorrer todas as posições do tabuleiro, ver caso existam quais letras de cima e de baixo, para verificar se existe triplo na vertical, e a da esquerda e a da direita para a horizontal. Confirmando se a presença de uma sandes de letras, atribui se as cores correctas caso nenhuma das celulas não tenha um cor contrária, se não envia uma mensagem de [erro](#).

Parameters:

Tab Tabuleiro do jogo

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as [jogadas](#) e moviemntos efectuados.

DIM dimensão do tabuleiro

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

4.9.2.21 int trp (Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

Função que procura triplos no tabuleiro(tres letras iguais na mesma linha/coluna). Para tal vai se percorrer todas as posições do tabuleiro, ver caso existam quais letras de cima e de baixo, para verificar se existe triplo na vertical, e a da esquerda e a da direita para a horizontal. É também tido em contam o caso de existirem quadroplos não serem considerados triplos. Confirmando se a presença de um triplo atribui se as cores correctas caso nenhuma das celulas não tenha um cor contrária, se não envia uma mensagem de [erro](#).

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as [jogadas](#) e moviemntos efectuados.

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

4.10 jogadas.h File Reference

Classes

- struct [jogadas](#)

Estrutura responsavel por guardar a coluna,linha e cor de uma celula a gravar.

- struct [movimentos](#)

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

- struct [It](#)

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

- struct [gravados](#)

Estrutura onde é armazenada toda a informação relativa a [movimentos](#) e [jogadas](#) guardadas.

Typedefs

- typedef struct [jogadas](#) [Jogadas](#)
- typedef struct [movimentos](#) [Movimentos](#)
- typedef struct [It](#) [IT](#)
- typedef struct [gravados](#) [Gravados](#)

Functions

- int [grava_jogada](#) (int col, int row, char cor, [Gravados](#) *gr)
- int [jogada](#) (char cor, int x, int y, [Elem](#) **Tab, [Gravados](#) *gr, int erro, int *cores)
- int [E_Par_des](#) ([Elem](#) **Tab, char letra1, char letra2, int *DIM, int line1, int cl1, int line2, int col2)
- int [jogadaManual](#) (char cor, int x, int y, int *DIM, [Elem](#) **Tab, [Gravados](#) *gr)
- int [pds](#) ([Elem](#) **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr)
- int [E_trp_row](#) ([Elem](#) **Tab, int *DIM, int i, int j, [Gravados](#) *gr)
- int [E_trp_col](#) ([Elem](#) **Tab, int *DIM, int i, int j, [Gravados](#) *gr)
- int [trp](#) ([Elem](#) **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr)
- int [snd](#) ([Elem](#) **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr)
- int [E_Par](#) ([Elem](#) **Tab, int pos, char letra, int *DIM, char *local)
- int [pis](#) ([Elem](#) **Tab, int *DIM, int *flagAP, int *flagAB, [Gravados](#) *gr)
- int [anc](#) ([Elem](#) **Tab, [Gravados](#) *gr)
- int [free_jogadasGR](#) ([Gravados](#) *gr)
- int [it](#) ([Gravados](#) *gr)
- int [rb](#) ([Elem](#) **Tab, [Gravados](#) *gr)
- int [inicia_GR](#) ([Gravados](#) *gr)
- int [grava_movimento](#) ([Gravados](#) *gr)
- int [anm](#) ([Elem](#) **Tab, [Gravados](#) *gr)

4.10.1 Typedef Documentation

4.10.1.1 typedef struct gravados Gravados

4.10.1.2 typedef struct It IT

4.10.1.3 typedef struct jogadas Jogadas

4.10.1.4 typedef struct movimentos Movimentos

4.10.2 Function Documentation

4.10.2.1 int anc (Elem ** *Tab*, Gravados * *gr*)

Função responsável por anular uma cor guardada quando é chamada. A função vai anular a ultima cor guardada.

Parameters:

Tab tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.10.2.2 int anm (Elem ** *Tab*, Gravados * *gr*)

Função responsável por anular um movimento quando é chamada. A função vai anular cores até chegar á ultima cor gravada quando foi guardado o movimento.

Parameters:

Tab tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.10.2.3 int E_Par (Elem ** *Tab*, int *pos*, char *letra*, int * *DIM*, char * *local*)

Função auxiliar que verifica se existem pares de letras numa linha ou numa coluna Para tal vai percorrer a linha ou coluna e verifica se existe. A função tem em conta o caso de tres letras seguidas não ser um par mas sim um triplo.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

pos Linha ou coluna onde vai procurar os pares

letra a letra que se pretende saber se existe pares ou não

local Para decidir se se procura nas linhas ou nas colunas

Returns:

Inteiro que confirma se existe ou não um para na coluna/linha.

4.10.2.4 int E_Par_des (Elem ** *Tab*, char *letra1*, char *letra2*, int * *DIM*, int *line1*, int *cl1*, int *line2*, int *cl2*)

Função auxiliar que verifica se existem pares de letras tanto nas linhas como nas colunas passdas como parametro. Para tal vai percorrer a linha ou coluna e verifica se existe.

Parameters:

Tab Tabuleiro do jogo
letra1 Primeira letra da diagonal.
letra2 Segunda letra da diagonal
DIM dimensão do tabuleiro
line1 Linha da primeira letra
line2 Linha da segunda letra
cl1 Coluna da primeira letra
cl2 Coluna da segunda letra

Returns:

Inteiro que confirma se existe ou não um para na coluna/linha.

4.10.2.5 int E_trp_col (Elem ** *Tab*, int * *DIM*, int *i*, int *j*, Gravados * *gr*)

Função responsável por verificar se existe um triplo a partir das cordenadas i e j na vertical. Se existir atribui as cores correspondentes.

Parameters:

Tab Tabuleiro do jogo
DIM dimensão do tabuleiro
i linha da célula.
j coluna da célula.
gr apontador para a estrutura de dados onde se guardas as [jogadas](#) e [movimentos](#).

Returns:

inteiro que irá retornar se houve alguma alteração no tabuleiro ou não

4.10.2.6 int E_trp_row (Elem ** *Tab*, int * *DIM*, int *i*, int *j*, Gravados * *gr*)

Função responsável por verificar se existe um triplo a partir das cordenadas i e j na horizontal. Se existir atribui as cores correspondentes.

Parameters:

Tab Tabuleiro do jogo
DIM dimensão do tabuleiro
i linha da celula.
j coluna da celula.

gr apontador para a estrutura de dados onde se guardas as jogadas e movimentos.

Returns:

inteiro que irá retornar se houve alguma alteração no tabuleiro ou não

4.10.2.7 int free_jogadasGR (Gravados * *gr*)

Função responsável por limpar da memória todos os dados guardados, em relação às cores e movimentos guardados.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

4.10.2.8 int grava_jogada (int *row*, int *col*, char *color*, Gravados * *gr*)

Função que irá guardar a cor a linha e a coluna de um tabuleiro que corresponde à célula que vai ser substituída. Esta função será chamada sempre que se mude uma cor.

Parameters:

row linha da célula que se vai guardar.

col coluna da célula que se vai guardar.

color Cor da célula que vai ser substituída

gr apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

4.10.2.9 int grava_movimento (Gravados * *gr*)

Função que irá guardar na estrutura de dados de movimentos o número de cores que foram guardadas até ao momento da sua chamada. Para posteriormente poder se recuperar o estado do tabuleiro.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

4.10.2.10 int inicia_GR (Gravados * *gr*)

Função responsável por inicializar e alocar espaço para as estruturas de dados responsáveis por guardar informação relativa a movimentos e jogadas efectuadas. Aloca espaço para *It*, para movimentos e para jogadas

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

4.10.2.11 int it (Gravados * *gr*)

Função responsável tal como a guarda [movimentos](#), de guardar na estrutura de dados de IT's o numero de cores que foram guardadas até ao momento da sua chamada. Para posteriormente poder se recuperar o estado do tabuleiro.

Parameters:

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.10.2.12 int jogada (char *cor*, int *x*, int *y*, Elem ** *Tab*, Gravados * *gr*, int *erro*, int * *cores*)

A função jogada é responsável por alterar os dados do tabuleiro que são atribuídos automaticamente. Ela irá receber o novo dado e as posições onde vai ficar o novo dado, actualizando assim o tabuleiro com os novos dados

Parameters:

cor nova cor para um [elemento](#) do tabuleiro

x Corresponde á linha que vamos substituir no tabuleiro.

y Corresponde á coluna que vamos substituir no tabuleiro.

Tab Tabuleiro do jogo

gr apontador para a estrutura de dados onde se guardas as [jogadas](#) e [movimentos](#).

erro flag que avisa se deve emitir mensagem de [erro](#) ou não para o utilizador

cores contador de cores mudadas

Returns:

Retorna um inteiro que verifica se a jogada foi ou não bem sucedida.

4.10.2.13 int jogadaManual (char *cor*, int *x*, int *y*, int * *DIM*, Elem ** *Tab*, Gravados * *gr*)

A função jogada é responsável por alterar os dados do tabuleiro introduzidos pelo utilizador. Ela irá receber o novo dado e as posições onde vai ficar o novo dado, actualizando assim o tabuleiro com os novos dados

Parameters:

cor nova cor para um [elemento](#) do tabuleiro

x Corresponde á linha que vamos substituir no tabuleiro.

y Corresponde á coluna que vamos substituir no tabuleiro.

DIM dimensão do tabuleiro

Tab Tabuleiro do jogo

gr apontador para a estrutura de dados onde se guardas as [jogadas](#) e [movimentos](#).

Returns:

Retorna um inteiro que verifica se a jogada foi ou não bem sucedida.

4.10.2.14 int pds (Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

Função responsável por ver se existem ou não pares isolados no tabuleiro. Para tal esta função vai pegar em quatro células, formando um quadrado. Neste quadrado verifica se as duas diagonais pertencem ou não a regra pares desencontrados, com a ajuda da função auxiliar EpParDes. Caso pertence ver se a diagonal contrária tem cores contrárias, se tiver dá uma mensagem de [erro](#), se não muda as cores.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as [jogadas](#) e movimentos efectuados.

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

4.10.2.15 int pis (Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

Função que procura pares isolados no tabuleiro (numa coluna ou numa linha). Para tal vai percorrer todas as posições do tabuleiro, e ver se a letra é diferente das que a rodeiam na horizontal ou na vertical. Confirmando se a presença de uma letra isola, chama a função auxiliar Epar que vai ver se existe um par dessa letra na coluna ou na linha. Caso se verifiquem as duas condições irá ver se a letra não está de uma cor contrária à que se vai atribuir, caso não esteja atribui-se uma nova cor, se não uma mensagem de [erro](#).

Parameters:

Tab Tabuleiro do jogo

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as [jogadas](#) e movimentos efectuados.

DIM dimensão do tabuleiro

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

4.10.2.16 int rb (Elem ** *Tab*, Gravados * *gr*)

Função responsável por anular cores até chegar ao estado de tabuleiro marcado pela função it. Desta forma será possível retornar o estado do tabuleiro tal como foi marcado na função it.

Parameters:

Tab tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

gr apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

4.10.2.17 int snd (Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

Função que procura sandes no tabuleiro(tres letras onde apenas a do meio é diferente na mesma linha/-coluna). Para tal vai se percorrer todas as posições do tabuleiro, ver caso existam quais letras de cima e de baixo, para verificar se existe triplo na vertical, e a daesquerda e a da direita para a horizontal. Confirmando se a presença de uma sandes de letras, atribui se as cores correctas caso nenhuma das celulas não tenha um cor contraria, se não envia uma mensagem de [erro](#).

Parameters:

Tab Tabuleiro do jogo

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as [jogadas](#) e moviemntos efectuados.

DIM dimensão do tabuleiro

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

4.10.2.18 int trp (Elem ** *Tab*, int * *DIM*, int * *flagAP*, int * *flagAB*, Gravados * *gr*)

Função que procura triplos no tabuleiro(tres letras iguais na mesma linha/coluna). Para tal vai se percorrer todas as posições do tabuleiro, ver caso existam quais letras de cima e de baixo, para verificar se existe triplo na vertical, e a daesquerda e a da direita para a horizontal. É também tido em contam o caso de exitirem quadroplos não serem considerados triplos. Confirmando se a presença de um triplo atribui se as cores correctas caso nenhuma das celulas não tenha um cor contraria, se não envia uma mensagem de [erro](#).

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

flagAP flag para controlo do automatico preto.

flagAB flag para controlo do automatico branco.

gr apontador para a estrutura de dados onde se guarda as [jogadas](#) e moviemntos efectuados.

Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

4.11 main.c File Reference

```
main do jogo #include "tabuleiro.h"
#include "jogadas.h"
#include "comandos.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <readline/readline.h>
#include <readline/history.h>
```

Functions

- `int main ()`

4.11.1 Detailed Description

main do jogo

Author:

André Pimenta, João Gomes, Nelson Carvalho

Date:

5 Março de 2010

4.11.2 Function Documentation

4.11.2.1 `int main ()`

A função main é um ciclo que apenas termina quando o utilizador insere o comando "q" Esta função vai então ser responsável por receber os comandos continuamente, e imprimir o resultado enquanto o jogo decorrer

4.12 tabuleiro.c File Reference

```
Tabuleiro do jogo. #include "tabuleiro.h"
#include "erro.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <readline/readline.h>
#include <readline/history.h>
```

Functions

- int [dimensao](#) (FILE *fp, int *DIM)
- int [ficheiroValido](#) (FILE *fp, int *DIM)
- void [freeTab](#) (int *DIM, Elem **Tab)
- int [lerficheiro](#) (int *DIM, Elem **Tab, FILE *fp)
- int [criarTabuleiro](#) (char *nome, int *DIM, Elem **Tab)
- void [printTabuleiro](#) (int *DIM, Elem **Tab)
- int [gravaJogo](#) (char *nome, int *DIM, Elem **Tab)

4.12.1 Detailed Description

Tabuleiro do jogo.

Author:

André Pimenta, João Gomes, Nelson Carvalho

Date:

5 Março de 2010

4.12.2 Function Documentation

4.12.2.1 int criarTabuleiro (char * *nome*, int * *DIM*, Elem ** *Tab*)

4.12.2.2 int dimensao (FILE * *fp*, int * *DIM*)

A função *dimensao* é responsável por carregar a dimensão do tabuleiro que estamos a abrir, para a variável *DIM*. Para isso ela usa o *fscanf* da biblioteca *stdio*.

Parameters:

DIM dimensão do tabuleiro

fp Recebe o ficheiro do qual irá verificar se existe o valor da *dim* e passa-lo para a variável Global *DIM*.

Returns:

retorna um valor lógico 1 ou 0 que confirma se conseguiu ou não ler o valor correspondente á dimensão.

4.12.2.3 int `ficheiroValido` (FILE * *fp*, int * *DIM*)

A função `ficheiroValido` recebe o ficheiro do qual queremos importar o tabuleiro e verifica se este é válido para o nosso jogo. Para ser válido irá verificar se tem o número de elementos correspondentes a dimensão, se são elementos do tipo predefinido do tabuleiro(ex: "[A]")

Parameters:

DIM dimensão do tabuleiro

fp Recebe o ficheiro que está aberto na função `criarTabuleiro`

Returns:

Retorna um inteiro que irá confirmar se o ficheiro é válido ou não.

4.12.2.4 void `freeTab` (int * *DIM*, Elem ** *Tab*)

A função `freeTab` irá apagar da memória do tabuleiro antigo caso estejamos a abrir um novo tabuleiro.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

Remarks:

Esta função apenas é usada se existir um tabuleiro carregado e queremos carregar outro.

4.12.2.5 int `gravaJogo` (char * *nome*, int * *DIM*, Elem ** *Tab*)

A função `gravaJogo` é responsável por gravar o nosso tabuleiro num ficheiro cujo o nome é dado como argumento e que está no formato ".ltr". O tabuleiro será imprimido para o ficheiro seguindo as regras de representação deste, para que possa ser lido de novo e seja considerado válido.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

nome Recebe uma string que representa o nome que vai dar ao ficheiro que vai guardar

4.12.2.6 int `lerficheiro` (int * *DIM*, Elem ** *Tab*, FILE * *fp*)

A função `criarTabuleiro` recebe o nome do ficheiro que queremos abrir e irá tentar abrir este. Depois irá chamar a função `freeTab` caso exista já um tabuleiro carregado e em seguida chama a função `ficheiroValido` para ver verificar se o mesmo é válido. Após todas verificações a função cria espaço para o tabuleiro na memória e carrega este para lá, ficando guardado no apontador *Tab*. Os valores lidos do ficheiro irão ser associados por duas características, a letra e a cor, esta última será "i" correspondente a indefinida caso a letra esteja rodeada por dois espaços brancos, preta caso esteja rodeada por dois parênteses rectos e branca caso esteja rodeada por dois parênteses curvos.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

fp ficheiro a ser lido

Returns:

Retorna um inteiro que verifica se conseguiu ou não criar o Tabuleiro.

4.12.2.7 void printTabuleiro (int * *DIM*, Elem ** *Tab*)

A função printTABuleiro é responsável por imprimir no ecrã o nosso tabuleiro.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

4.13 tabuleiro.h File Reference

header file do [tabuleiro.c](#) `#include <stdio.h>`

Classes

- struct [elemento](#)

É constituída por dois chars, que correspondem às características de cada celula do tabuliro, a cor e aletra.

Typedefs

- typedef struct [elemento](#) [Elemento](#)
- typedef struct [elemento](#) * [Elem](#)

Functions

- int [dimensao](#) (FILE *fp, int *DIM)
- int [ficheiroValido](#) (FILE *fp, int *DIM)
- void [freeTab](#) (int *DIM, [Elem](#) **Tab)
- int [lerficheiro](#) (int *DIM, [Elem](#) **Tab, FILE *fp)
- int [criarTabuleiro](#) (char *nome, int *DIM, [Elem](#) **Tab)
- void [printTabuleiro](#) (int *DIM, [Elem](#) **Tab)
- int [gravaJogo](#) (char *nome, int *DIM, [Elem](#) **Tab)

4.13.1 Detailed Description

header file do [tabuleiro.c](#)

Author:

André Pimenta,Miguel Gomes,Nelson Cravalho

Date:

5 Março de 2010

4.13.2 Typedef Documentation

4.13.2.1 typedef struct elemento* Elem

4.13.2.2 typedef struct elemento Elemento

4.13.3 Function Documentation

4.13.3.1 int criarTabuleiro (char * *nome*, int * *DIM*, Elem ** *Tab*)

4.13.3.2 int dimensao (FILE * *fp*, int * *DIM*)

A unção dimensão é responsável por carregar a dimensão do tabuleiro que estamos a abrir, para a variável DIM. Para isso ela usa o fscanf da biblioteca stdio.

Parameters:

DIM dimensão do tabuleiro

fp Recebe o ficheiro do qual irá verificar se existe o valor da dim e passa-lo para a variável Global DIM.

Returns:

retorna um valor lógico 1 ou 0 que confirma se conseguiu ou não ler o valor correspondente á dimensão.

4.13.3.3 int ficheiroValido (FILE * *fp*, int * *DIM*)

A função ficheiroValido recebe o ficheiro do qual queremos importar o tabuleiro e verifica se este é válido para o nosso jogo. Para ser válido irá verficar se tem o número de elementos correspondentes a dimensão, se são elementos do tipo predefenido do tabuleiro(ex: "[A]")

Parameters:

DIM dimensão do tabuleiro

fp Recebe o ficheiro que está aberto na função criarTabuleiro

Returns:

Retorna um inteiro que irá confirmar se o ficheiro é válido ou não.

4.13.3.4 void freeTab (int * *DIM*, Elem ** *Tab*)

A função freeTab irá apagar da memória do tabuleiro antigo caso estejamos a abrir um novo tabuleiro.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

Remarks:

Esta função apenas é usada se existir um tabuleiro carregado e queremos carregar outro.

4.13.3.5 int gravaJogo (char * *nome*, int * *DIM*, Elem ** *Tab*)

A função gravaJogo é responsável por gravar o nosso tabuleiro num ficheiro cujo o nome é dado como argumento e que está no formato ".ltr". O tabuleiro será imprimido para o ficheiro seguindo as regras de representação deste, para que possa ser lido de novo e seja considerado válido.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

nome Recebe uma string que representa o nome que vai dar ao ficheiro que vai guardar

4.13.3.6 int lerficheiro (int * *DIM*, Elem ** *Tab*, FILE * *fp*)

A função criarTabuleiro recebe o nome do ficheiro que queremos abrir e irá tentar abrir este. Depois irá chamar a função freeTab caso exista já um tabuleiro carregado e em seguida chama a função ficheiroValido para ver verificar se o mesmo é válido. Após todas verificações a função cria espaço para o tabuleiro na memória e carrega este para lá, ficando guardado no apontador Tab. Os valores lidos do ficheiro irão ser associados por duas características, a letra e a cor, esta última será "i" correspondente a indefinida caso a letra esteja rodeada por dois espaços brancos, preta caso esteja rodeada por dois parênteses rectos e branca caso esteja rodeada por dois parênteses curvos.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

fp ficheiro a ser lido

Returns:

Retorna um inteiro que verifica se conseguiu ou não criar o Tabuleiro.

4.13.3.7 void printTabuleiro (int * *DIM*, Elem ** *Tab*)

A função printTabuleiro é responsável por imprimir no ecrã o nosso tabuleiro.

Parameters:

Tab Tabuleiro do jogo

DIM dimensão do tabuleiro

Index

ajuda

ajuda.c, [12](#)
ajuda.h, [23](#)

ajuda.c, [11](#)

ajuda, [12](#)
auto_black, [12](#)
auto_white, [13](#)
automatico, [13](#)
brancasAuto, [13](#)
change_tc, [14](#)
CHUNK_SIZE, [12](#)
control, [14](#)
copy_board, [14](#)
corPossivel, [15](#)
corrige, [15](#)
crg, [16](#)
executa_tc, [16](#)
imp, [16](#)
impacto, [17](#)
p_brancas, [17](#)
p_preta, [17](#)
pretasAuto, [18](#)
resolve, [18](#)
rsv, [19](#)
tc, [19](#)
tc_auto, [19](#)
tentarFinais, [20](#)
tentativa, [20](#)
vc, [20](#)
verificaAB, [21](#)
verificaAP, [21](#)

ajuda.h, [23](#)

ajuda, [23](#)
brancasAuto, [23](#)
control, [23](#)
count_i, [24](#)
crg, [24](#)
executa_tc, [24](#)
imp, [24](#)
pretasAuto, [25](#)
rsv, [25](#)
vc, [26](#)

anc

jogadas.c, [41](#)
jogadas.h, [48](#)

anm

jogadas.c, [41](#)
jogadas.h, [48](#)

auto_black

ajuda.c, [12](#)

auto_white

ajuda.c, [13](#)

automatico

ajuda.c, [13](#)

brancasAuto

ajuda.c, [13](#)
ajuda.h, [23](#)

caminho

estado_tabuleiro.c, [35](#)
estado_tabuleiro.h, [38](#)

change_tc

ajuda.c, [14](#)

CHUNK_SIZE

ajuda.c, [12](#)

col

jogadas, [9](#)

color

jogadas, [9](#)

comandos.c, [27](#)

executa_comando, [27](#)
executaCriaJogo, [28](#)
executaGravaJogo, [28](#)
executaJogada, [28](#)

comandos.h, [30](#)

executa_comando, [30](#)
executaCriaJogo, [30](#)
executaGravaJogo, [30](#)
executaJogada, [31](#)

control

ajuda.c, [14](#)
ajuda.h, [23](#)

copy_board

ajuda.c, [14](#)

cor

elemento, [5](#)

corPossivel

ajuda.c, [15](#)

corrige

- ajuda.c, 15
- count_i
 - ajuda.h, 24
- crg
 - ajuda.c, 16
 - ajuda.h, 24
- criarTabuleiro
 - tabuleiro.c, 55
 - tabuleiro.h, 58
- dimencao
 - tabuleiro.c, 55
 - tabuleiro.h, 58
- E_ARGS
 - erro.h, 33
- E_COMMAND
 - erro.h, 33
- E_COORDS
 - erro.h, 33
- E_FILE_FORMAT
 - erro.h, 33
- E_INVARGS
 - erro.h, 33
- E_NO_BOARD
 - erro.h, 33
- E_NO_FILE
 - erro.h, 33
- E_NO_MOVES
 - erro.h, 33
- E_NOERR
 - erro.h, 33
- E_Par
 - jogadas.c, 41
 - jogadas.h, 48
- E_Par_des
 - jogadas.c, 41
 - jogadas.h, 48
- E_SAVE
 - erro.h, 33
- E_trp_col
 - jogadas.c, 42
 - jogadas.h, 49
- E_trp_row
 - jogadas.c, 42
 - jogadas.h, 49
- E_WRONG_SOLUTION
 - erro.h, 33
- Elem
 - tabuleiro.h, 58
- Elemento
 - tabuleiro.h, 58
- elemento, 5
 - cor, 5
- letra, 5
- err_code
 - erro, 6
- ERRO
 - erro.c, 32
- erro, 6
 - err_code, 6
 - mensagem, 6
- erro.c, 32
 - ERRO, 32
 - mensagem_de_erro, 32
- erro.h, 33
 - E_ARGS, 33
 - E_COMMAND, 33
 - E_COORDS, 33
 - E_FILE_FORMAT, 33
 - E_INVARGS, 33
 - E_NO_BOARD, 33
 - E_NO_FILE, 33
 - E_NO_MOVES, 33
 - E_NOERR, 33
 - E_SAVE, 33
 - E_WRONG_SOLUTION, 33
 - mensagem_de_erro, 33
- estado_tabuleiro.c, 35
 - caminho, 35
 - free_tab, 35
 - prepara_tab, 35
 - st, 36
 - vb, 36
 - vl, 36
 - vp, 36
- estado_tabuleiro.h, 38
 - caminho, 38
 - free_tab, 38
 - prepara_tab, 38
 - st, 38
 - vb, 38
 - vl, 39
 - vp, 39
- executa_comando
 - comandos.c, 27
 - comandos.h, 30
- executa_tc
 - ajuda.c, 16
 - ajuda.h, 24
- executaCriaJogo
 - comandos.c, 28
 - comandos.h, 30
- executaGravaJogo
 - comandos.c, 28
 - comandos.h, 30
- executaJogada
 - comandos.c, 28

- comandos.h, 31
- ficheiroValido
 - tabuleiro.c, 55
 - tabuleiro.h, 59
- free_jogadasGR
 - jogadas.c, 42
 - jogadas.h, 50
- free_tab
 - estado_tabuleiro.c, 35
 - estado_tabuleiro.h, 38
- freeIts
 - jogadas.c, 43
- freeJgs
 - jogadas.c, 43
- freeMvs
 - jogadas.c, 43
- freeTab
 - tabuleiro.c, 56
 - tabuleiro.h, 59
- grava_jogada
 - jogadas.c, 43
 - jogadas.h, 50
- grava_movimento
 - jogadas.c, 43
 - jogadas.h, 50
- Gravados
 - jogadas.h, 48
- gravados, 7
 - its, 7
 - jgs, 7
 - mvs, 7
 - totalGr, 7
- gravaJogo
 - tabuleiro.c, 56
 - tabuleiro.h, 59
- imp
 - ajuda.c, 16
 - ajuda.h, 24
- impacto
 - ajuda.c, 17
- inicia_GR
 - jogadas.c, 43
 - jogadas.h, 50
- IT
 - jogadas.h, 48
- It, 8
 - next, 8
 - pos, 8
- it
 - jogadas.c, 44
 - jogadas.h, 50
- its
 - gravados, 7
- jgs
 - gravados, 7
- jogada
 - jogadas.c, 44
 - jogadas.h, 51
- jogadaManual
 - jogadas.c, 44
 - jogadas.h, 51
- Jogadas
 - jogadas.h, 48
- jogadas, 9
 - col, 9
 - color, 9
 - next, 9
 - row, 9
- jogadas.c, 40
 - anc, 41
 - anm, 41
 - E_Par, 41
 - E_Par_des, 41
 - E_trp_col, 42
 - E_trp_row, 42
 - free_jogadasGR, 42
 - freeIts, 43
 - freeJgs, 43
 - freeMvs, 43
 - grava_jogada, 43
 - grava_movimento, 43
 - inicia_GR, 43
 - it, 44
 - jogada, 44
 - jogadaManual, 44
 - pds, 45
 - pis, 45
 - rb, 45
 - snd, 46
 - trp, 46
- jogadas.h, 47
 - anc, 48
 - anm, 48
 - E_Par, 48
 - E_Par_des, 48
 - E_trp_col, 49
 - E_trp_row, 49
 - free_jogadasGR, 50
 - grava_jogada, 50
 - grava_movimento, 50
 - Gravados, 48
 - inicia_GR, 50
 - IT, 48
 - it, 50

- jogada, 51
- jogadaManual, 51
- Jogadas, 48
- Movimentos, 48
- pds, 51
- pis, 52
- rb, 52
- snd, 52
- trp, 53
- lerficheiro
 - tabuleiro.c, 56
 - tabuleiro.h, 59
- letra
 - elemento, 5
- main
 - main.c, 54
- main.c, 54
 - main, 54
- mensagem
 - erro, 6
- mensagem_de_erro
 - erro.c, 32
 - erro.h, 33
- Movimentos
 - jogadas.h, 48
- movimentos, 10
 - next, 10
 - pos, 10
- mvs
 - gravados, 7
- next
 - It, 8
 - jogadas, 9
 - movimentos, 10
- p_brancas
 - ajuda.c, 17
- p_preta
 - ajuda.c, 17
- pds
 - jogadas.c, 45
 - jogadas.h, 51
- pis
 - jogadas.c, 45
 - jogadas.h, 52
- pos
 - It, 8
 - movimentos, 10
- prepara_tab
 - estado_tabuleiro.c, 35
 - estado_tabuleiro.h, 38
- pretasAuto
 - ajuda.c, 18
 - ajuda.h, 25
- printTabuleiro
 - tabuleiro.c, 57
 - tabuleiro.h, 60
- rb
 - jogadas.c, 45
 - jogadas.h, 52
- resolve
 - ajuda.c, 18
- row
 - jogadas, 9
- rsv
 - ajuda.c, 19
 - ajuda.h, 25
- snd
 - jogadas.c, 46
 - jogadas.h, 52
- st
 - estado_tabuleiro.c, 36
 - estado_tabuleiro.h, 38
- tabuleiro.c, 55
 - criarTabuleiro, 55
 - dimensao, 55
 - ficheiroValido, 55
 - freeTab, 56
 - gravaJogo, 56
 - lerficheiro, 56
 - printTabuleiro, 57
- tabuleiro.h, 58
 - criarTabuleiro, 58
 - dimensao, 58
 - Elem, 58
 - Elemento, 58
 - ficheiroValido, 59
 - freeTab, 59
 - gravaJogo, 59
 - lerficheiro, 59
 - printTabuleiro, 60
- tc
 - ajuda.c, 19
- tc_auto
 - ajuda.c, 19
- tentarFinais
 - ajuda.c, 20
- tentativa
 - ajuda.c, 20
- totalGr
 - gravados, 7
- trp

jogadas.c, [46](#)
jogadas.h, [53](#)

vb
 estado_tabuleiro.c, [36](#)
 estado_tabuleiro.h, [38](#)

vc
 ajuda.c, [20](#)
 ajuda.h, [26](#)

verificaAB
 ajuda.c, [21](#)

verificaAP
 ajuda.c, [21](#)

vl
 estado_tabuleiro.c, [36](#)
 estado_tabuleiro.h, [39](#)

vp
 estado_tabuleiro.c, [36](#)
 estado_tabuleiro.h, [39](#)