

Reference Manual

Generated by Doxygen 1.6.1

Sun Mar 7 21:51:23 2010

Contents

1	Todo List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	elemento Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Data Documentation	7
4.1.2.1	cor	7
4.1.2.2	letra	7
5	File Documentation	9
5.1	main.c File Reference	9
5.1.1	Detailed Description	9
5.1.2	Function Documentation	9
5.1.2.1	main	9
5.2	tabuleiro.c File Reference	10
5.2.1	Detailed Description	10
5.2.2	Function Documentation	10
5.2.2.1	ajuda	10
5.2.2.2	criarTabuleiro	11
5.2.2.3	dimensao	11
5.2.2.4	executa_comando	11
5.2.2.5	executaCriaJogo	11
5.2.2.6	executaGravaJogo	12
5.2.2.7	executaJogada	12

5.2.2.8	ficheiroValido	12
5.2.2.9	freeTab	12
5.2.2.10	gravaJogo	13
5.2.2.11	jogada	13
5.2.2.12	printTabuleiro	13
5.3	tabuleiro.h File Reference	14
5.3.1	Detailed Description	14
5.3.2	Typedef Documentation	15
5.3.2.1	Elem	15
5.3.2.2	Elemento	15
5.3.3	Function Documentation	15
5.3.3.1	ajuda	15
5.3.3.2	criarTabuleiro	15
5.3.3.3	dimensao	15
5.3.3.4	executa_comando	15
5.3.3.5	executaCriaJogo	16
5.3.3.6	executaGravaJogo	16
5.3.3.7	executaJogada	16
5.3.3.8	ficheiroValido	17
5.3.3.9	freeTab	17
5.3.3.10	gravaJogo	17
5.3.3.11	jogada	17
5.3.3.12	printTabuleiro	17
5.3.4	Variable Documentation	18
5.3.4.1	DIM	18
5.3.4.2	Tab	18

Chapter 1

Todo List

Member [ajuda](#) por a função a carregar o ficheiro ajuda e a imprimir, para a proxima etapa

Chapter 2

Class Index

2.1 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)	7
--	---

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

main.c (Main do jogo)	9
tabuleiro.c (Tabuleiro do jogo)	10
tabuleiro.h (Header file do tabuleiro.c)	14

Chapter 4

Class Documentation

4.1 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.

4.1.1 Detailed Description

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

4.1.2 Member Data Documentation

4.1.2.1 elemento::cor

Cor da célula do tabuleiro.

4.1.2.2 elemento::letra

Letra da célula do tabuleiro.

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

- [tabuleiro.h](#)

Chapter 5

File Documentation

5.1 main.c File Reference

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

Functions

- int `main` ()

5.1.1 Detailed Description

main do jogo

Author:

André Pimenta, João Gomes, Nelson Carvalho

Date:

5 Março de 2010

5.1.2 Function Documentation

5.1.2.1 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

5.2 tabuleiro.c File Reference

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

Functions

- int [dimensao](#) (FILE *fp)
- int [ficheiroValido](#) (FILE *fp)
- void [freeTab](#) ()
- int [criarTabuleiro](#) (char *nome)
- int [ajuda](#) ()
- int [jogada](#) (char cor, int x, int y)
- void [printTabuleiro](#) ()
- void [gravaJogo](#) (char *nome)
- int [executaJogada](#) (char *args, char *cor)
- int [executaCriaJogo](#) (char *args)
- int [executaGravaJogo](#) (char *args)
- int [executa_comando](#) (char *linha)

5.2.1 Detailed Description

Tabuleiro do jogo.

Author:

André Pimenta, João Gomes, Nelson Carvalho

Date:

5 Março de 2010

5.2.2 Function Documentation

5.2.2.1 int ajuda ()

A função ajuda irá imprimir no ecrã o ficheiro com as instruções de ajuda

Todo

por a função a carregar o ficheiro ajuda e a imprimir, para a proxima etapa

5.2.2.2 int criarTabuleiro (char * *nome*)

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:

nome Recebe uma string com o nome do ficheiro que vai abrir.

Returns:

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

5.2.2.3 int dimensao (FILE * *fp*)

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

Parameters:

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 à dimensão.

5.2.2.4 int executa_comando (char * *linha*)

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-os para as funções associadas aos comandos.

Parameters:

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

Returns:

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

5.2.2.5 int executaCriaJogo (char * *args*)

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").

Returns:

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

5.2.2.6 int executaGravaJogo (char * args)

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:

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.

5.2.2.7 int executaJogada (char * args, char * cor)

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.

Returns:

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

5.2.2.8 int ficheiroValido (FILE * fp)

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 vereficar se tem o número de elementos correspondentes a dimensão, se são elementos do tipo predefenido do tabuleiro(ex: "[A]")

Parameters:

fp Recebe o ficheiro que está aberto na função criarTabuleiro

Returns:

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

5.2.2.9 void freeTab ()

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

Remarks:

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

5.2.2.10 void gravaJogo (char * nome)

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:

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

5.2.2.11 int jogada (char cor, int x, int y)

A função jogada é responsável por alterar os dados do tabuleiro. 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.

Returns:

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

5.2.2.12 void printTabuleiro ()

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

5.3 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 célula do tabuleiro, a cor e a letra.

Typedefs

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

Functions

- int [dimensao](#) (FILE *fp)
- int [ficheiroValido](#) (FILE *fp)
- void [freeTab](#) ()
- int [criarTabuleiro](#) (char *nome)
- int [ajuda](#) ()
- int [jogada](#) (char cor, int x, int y)
- void [printTabuleiro](#) ()
- void [gravaJogo](#) (char *nome)
- int [executaJogada](#) (char *args, char *cor)
- int [executaCriaJogo](#) (char *args)
- int [executaGravaJogo](#) (char *args)
- int [executa_comando](#) (char *linha)

Variables

- [Elem](#) * [Tab](#)

é o apontador associado ao Tabuleiro, através deste podemos ter a referência ao tabuleiro guardado na memória.

- int [DIM](#)

é a variável global onde se guarda a dimensão do tabuleiro quando esta é lida.

5.3.1 Detailed Description

header file do [tabuleiro.c](#)

Author:

André Pimenta,Miguel Gomes,Nelson Cravalho

Date:

5 Março de 2010

5.3.2 Typedef Documentation

5.3.2.1 typedef struct elemento* Elem

5.3.2.2 typedef struct elemento Elemento

5.3.3 Function Documentation

5.3.3.1 int ajuda ()

A função ajuda irá imprimir no ecrã o ficheiro com as instruções de ajuda

Todo

por a função a carregar o ficheiro ajuda e a imprimir, para a proxima etapa

5.3.3.2 int criarTabuleiro (char * *nome*)

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, eta ultima 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:

nome Recebe uma string com o nome do ficheiro que vai abrir.

Returns:

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

5.3.3.3 int dimensao (FILE * *fp*)

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

Parameters:

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.

5.3.3.4 int executa_comando (char * *linha*)

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:

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

Returns:

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

5.3.3.5 int executaCriaJogo (char * args)

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").

Returns:

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

5.3.3.6 int executaGravaJogo (char * args)

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

Parameters:

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.

5.3.3.7 int executaJogada (char * args, char * cor)

A função executajogada recebe os argumentos da linha de comandos e a cor que vai passar para a jogada. Vai verificar se tem os argumentos 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.

Returns:

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

5.3.3.8 int `ficheiroValido` (FILE * *fp*)

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:

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

Returns:

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

5.3.3.9 void `freeTab` ()

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

Remarks:

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

5.3.3.10 void `gravaJogo` (char * *nome*)

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:

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

5.3.3.11 int `jogada` (char *cor*, int *x*, int *y*)

A função `jogada` é responsável por alterar os dados do tabuleiro. 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.

Returns:

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

5.3.3.12 void `printTabuleiro` ()

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

5.3.4 Variable Documentation

5.3.4.1 DIM

é a variavel global onde se guarda a dimensão do tabuleiro quando esta é lida.

5.3.4.2 Tab

é o apontador associado ao Tabuleiro, atravez deste podemos ter a referencia ao tabuleiro guardado na memoria.

Index

ajuda
 tabuleiro.c, [10](#)
 tabuleiro.h, [15](#)

cor
 elemento, [7](#)
criarTabuleiro
 tabuleiro.c, [10](#)
 tabuleiro.h, [15](#)

DIM
 tabuleiro.h, [18](#)

dimencao
 tabuleiro.c, [11](#)
 tabuleiro.h, [15](#)

Elem
 tabuleiro.h, [15](#)

Elemento
 tabuleiro.h, [15](#)

elemento, [7](#)
 cor, [7](#)
 letra, [7](#)

executa_comando
 tabuleiro.c, [11](#)
 tabuleiro.h, [15](#)

executaCriaJogo
 tabuleiro.c, [11](#)
 tabuleiro.h, [16](#)

executaGravaJogo
 tabuleiro.c, [12](#)
 tabuleiro.h, [16](#)

executaJogada
 tabuleiro.c, [12](#)
 tabuleiro.h, [16](#)

ficheiroValido
 tabuleiro.c, [12](#)
 tabuleiro.h, [16](#)

freeTab
 tabuleiro.c, [12](#)
 tabuleiro.h, [17](#)

gravaJogo
 tabuleiro.c, [12](#)
 tabuleiro.h, [17](#)

jogada
 tabuleiro.c, [13](#)
 tabuleiro.h, [17](#)

letra
 elemento, [7](#)

main
 main.c, [9](#)
main.c, [9](#)
 main, [9](#)

printTabuleiro
 tabuleiro.c, [13](#)
 tabuleiro.h, [17](#)

Tab
 tabuleiro.h, [18](#)

tabuleiro.c, [10](#)
 ajuda, [10](#)
 criarTabuleiro, [10](#)
 dimencao, [11](#)
 executa_comando, [11](#)
 executaCriaJogo, [11](#)
 executaGravaJogo, [12](#)
 executaJogada, [12](#)
 ficheiroValido, [12](#)
 freeTab, [12](#)
 gravaJogo, [12](#)
 jogada, [13](#)
 printTabuleiro, [13](#)

tabuleiro.h, [14](#)
 ajuda, [15](#)
 criarTabuleiro, [15](#)
 DIM, [18](#)
 dimencao, [15](#)
 Elem, [15](#)
 Elemento, [15](#)
 executa_comando, [15](#)
 executaCriaJogo, [16](#)
 executaGravaJogo, [16](#)
 executaJogada, [16](#)
 ficheiroValido, [16](#)
 freeTab, [17](#)
 gravaJogo, [17](#)
 jogada, [17](#)

printTabuleiro, [17](#)
Tab, [18](#)