Tetris - PS Edition

Gerado por Doxygen 1.8.11

Segunda, 13 de Junho de 2016 17:12:03

Sumário

1 Índice das Estruturas de Dados **1**

1.1 Estruturas de Dados 1

2 Índice dos Arquivos **1**

2.1 Lista de Arquivos 1

3 Estruturas **2**

3.1 Referência da Estrutura Bloco 2

3.1.1 Campos 3

3.2 Referência da Estrutura Peca 3

3.2.1 Campos 4

3.3 Referência da Estrutura Placar 4

3.3.1 Campos 5

3.4 Referência da Estrutura tela 6

3.4.1 Descrição Detalhada 6

3.4.2 Campos 7

3.5 Referência da Estrutura timeb 8

4 Arquivos **8**

4.1 Referência do Arquivo bloco.h 8

4.1.1 Variáveis 9

4.2 Referência do Arquivo engine.c 10

4.2.1 Funções 10

4.3 Referência do Arquivo engine.h 11

4.3.1 Funções 11

4.4 Referência do Arquivo pecas.c 11

4.4.1 Funções 12

4.4.2 Variáveis 15

4.5 Referência do Arquivo pecas.h 15

4.5.1 Definições dos tipos 16

4.5.2 Enumerações . . . 16
4.5.3 Funções . . . 16
4.6 Referência do Arquivo placar.c . . . 19
4.6.1 Funções . . . 19
4.7 Referência do Arquivo placar.h . . . 20
4.7.1 Funções . . . 21
4.7.2 Variáveis . . . 21
4.8 Referência do Arquivo tela.c . . . 22
4.8.1 Funções . . . 23
4.9 Referência do Arquivo tela.h . . . 25
4.9.1 Definições dos tipos . . . 26
4.9.2 Enumerações . . . 26
4.9.3 Funções . . . 26

Índice **31**

1 Índice das Estruturas de Dados

1.1 Estruturas de Dados

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

Bloco **2**
Peca **3**
Placar **4**
tela **6**
timeb **8**

2 Índice dos Arquivos

2.1 Lista de Arquivos

Esta é a lista de todos os arquivos documentados e suas respectivas descrições:

bloco.h **8**

cdefs.h **??**

engine.c **10**

engine.h **11**

features.h **??**

pecas.c **11**

pecas.h **15**

placar.c **19**

placar.h **20**

stubs-64.h **??**

stubs.h **??**

tela.c **22**

tela.h **25**

testes.h **??**

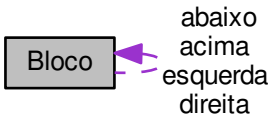
timeb.h **??**

wordsize.h **??**

3 Estruturas

3.1 Referência da Estrutura Bloco

Diagrama de colaboração para Bloco:



Campos de Dados

- char bolinha
- unsigned short int cor
- int pos_x
- int pos_y
- unsigned short int move
- struct Bloco * esquerda
- struct Bloco * direita
- struct Bloco * abaixo
- struct Bloco * acima

3.1.1 Campos

3.1.1.1 struct Bloco* Bloco::abaixo

Ponteiro para vizinho abaixo.

3.1.1.2 struct Bloco* Bloco::acima

Ponteiro para vizinho acima.

3.1.1.3 char Bloco::bolinha

Caractere atual da peça.

3.1.1.4 unsigned short int Bloco::cor

Cor da peça.

3.1.1.5 struct Bloco* Bloco::direita

Ponteiro para vizinho à direita.

3.1.1.6 struct Bloco* Bloco::esquerda

Ponteiro para vizinho à esquerda.

3.1.1.7 unsigned short int Bloco::move

Valor booleano que indica se o bloco está em movimento ou não.

3.1.1.8 int Bloco::pos_x

Coordenada cartesiana horizontal do bloco.

3.1.1.9 int Bloco::pos_y

Coordenada cartesiana vertical do bloco.

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

- bloco.h

3.2 Referência da Estrutura Peca

Campos de Dados

- int tamanho
- unsigned short int cor_peca
- unsigned short int move_peca
- float velocidade
- tp_peca tipo
- bloco * centro_de_rotacao
- bloco ** blocos

3.2.1 Campos

3.2.1.1 bloco** Peca::blocos

Referência para blocos na tela.

3.2.1.2 bloco* Peca::centro_de_rotacao

Ponteiro para o bloco de centro de rotação.

3.2.1.3 unsigned short int Peca::cor_peca

Cor da peça.

3.2.1.4 unsigned short int Peca::move_peca

Booleano que checa se a peça está em movimento ou não.

3.2.1.5 int Peca::tamanho

Tamanho da peça.

3.2.1.6 tp_peca Peca::tipo

Qual o formato da peça.

3.2.1.7 float Peca::velocidade

Velocidade na qual a peça está caindo.

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

- pecas.h

3.3 Referência da Estrutura Placar

Campos de Dados

- char jogadores [5][4]
- int pontuacoes [5]
- int tempos_m [5]
- int tempos_s [5]
- int anos [5]
- int meses [5]
- int dias [5]
- int contador_jogadores
- FILE * arquivo
- char jogador [3]
- int pontuacao
- int tempo_m
- int tempo_s

3.3.1 Campos

3.3.1.1 int Placar::anos[5]

Vetor de anos dos jogadores.

3.3.1.2 FILE* Placar::arquivo

Ponteiro para o arquivo usado para abrir o placar (pontuacao.txt) para leitura e escrita.

3.3.1.3 int Placar::contador_jogadores

Contador de jogadores presentes no placar (máximo 5).

3.3.1.4 int Placar::dias[5]

Vetor de dias dos jogadores no placar.

3.3.1.5 char Placar::jogador[3]

Nome do jogador atual.

3.3.1.6 char Placar::jogadores[5][4]

Vetor de jogadores.

3.3.1.7 int Placar::meses[5]

Vetor de meses dos jogadores no placar.

3.3.1.8 int Placar::pontuacao

Pontuacao do jogador atual.

3.3.1.9 int Placar::pontuacoes[5]

Vetor de pontuação dos jogadores no placar.

3.3.1.10 int Placar::tempo_m

Tempo em minutos do jogador atual.

3.3.1.11 int Placar::tempo_s

Tempo em segundos do jogador atual.

3.3.1.12 int Placar::tempos_m[5]

Vetor de tempo em minutos dos jogadores.

3.3.1.13 int Placar::tempos_s[5]

Vetor de tempo em segundos dos jogadores.

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

- placar.h

3.4 Referência da Estrutura tela

`#include <tela.h>`

Diagrama de colaboração para tela:

Campos de Dados

- char jogador [3]
- int **letra**
- int estado
- int pontos
- int tempo_m
- int tempo_s
- int comprimento
- int largura
- WINDOW * janela
- struct Peca * peca
- bloco blocos [ ]

3.4.1 Descrição Detalhada

/struct Define a tela do jogo.

3.4.2 Campos

3.4.2.1 bloco tela::blocos[ ]

Matriz dos blocos na tela.

3.4.2.2 int tela::comprimento

Comprimento da tela.

3.4.2.3 int tela::estado

Estado atual do jogo.

3.4.2.4 WINDOW* tela::janela

Ponteiro para a janela do jogo.

3.4.2.5 char tela::jogador[3]

Nome do atual jogador.

3.4.2.6 int tela::largura

Largura da tela.

3.4.2.7 struct Peca* tela::peca

Ponteiro para a peça em movimento.

3.4.2.8 int tela::pontos

Pontuação do jogador.

3.4.2.9 int tela::tempo_m

Tempo de execução em minutos.

3.4.2.10 int tela::tempo_s

Tempo de execução em segundos.

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

- tela.h

3.5 Referência da Estrutura timeb

Campos de Dados

- time_t **time**
- unsigned short int **millitm**
- short int **timezone**
- short int **dstflag**

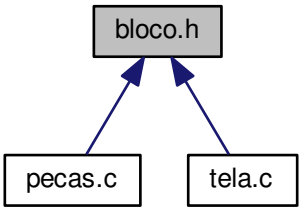
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- timeb.h

4 Arquivos

4.1 Referência do Arquivo bloco.h

Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:



Estruturas de Dados

- struct Bloco

Definições e Macros

- #define **BLOCO_H**
- #define **COMPRIMENTO** 15
- #define **LARGURA** 25

Variáveis

- char bolinha
- unsigned short int cor
- int pos_x
- int pos_y
- unsigned short int move
- struct Bloco * esquerda
- struct Bloco * direita
- struct Bloco * abaixo
- struct Bloco * acima
- **bloco**

4.1.1 Variáveis

4.1.1.1 struct Bloco* abaixo

Ponteiro para vizinho abaixo.

4.1.1.2 struct Bloco* acima

Ponteiro para vizinho acima.

4.1.1.3 char bolinha

Caractere atual da peça.

4.1.1.4 unsigned short int cor

Cor da peça.

4.1.1.5 struct Bloco* direita

Ponteiro para vizinho à direita.

4.1.1.6 struct Bloco* esquerda

Ponteiro para vizinho à esquerda.

4.1.1.7 unsigned short int move

Valor booleano que indica se o bloco está em movimento ou não.

4.1.1.8 int pos_x

Coordenada cartesiana horizontal do bloco.

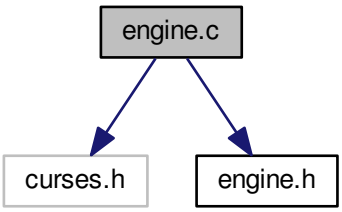
4.1.1.9 int pos_y

Coordenada cartesiana vertical do bloco.

4.2 Referência do Arquivo engine.c

```
#include <curses.h>
#include "engine.h"
```

Gráfico de dependência de inclusões para engine.c:



Funções

- void inicia_ncurses ()
- void finaliza_ncurses ()
- int pega_input (int input)

4.2.1 Funções

4.2.1.1 void finaliza_ncurses ( )

Finaliza o modo ncurses.

4.2.1.2 void inicia_ncurses ( )

Inicializa o modo ncurses e determina as funcionalidades dele que serão usadas.

4.2.1.3 int pega_input (int *input*)

Determina como interpretar a entrada do teclado.

Parâmetros

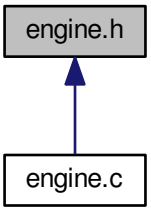
input	Entrada.

Retorna

Saída convertida.

4.3 Referência do Arquivo engine.h

Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:



Funções

- void inicia_ncurses ()
- void finaliza_ncurses ()
- int pega_input (int input)

4.3.1 Funções

4.3.1.1 void finaliza_ncurses ( )

Finaliza o modo ncurses.

4.3.1.2 void inicia_ncurses ( )

Inicializa o modo ncurses e determina as funcionalidades dele que serão usadas.

4.3.1.3 int pega_input (int *input*)

Determina como interpretar a entrada do teclado.

Parâmetros

input	Entrada.

Retorna

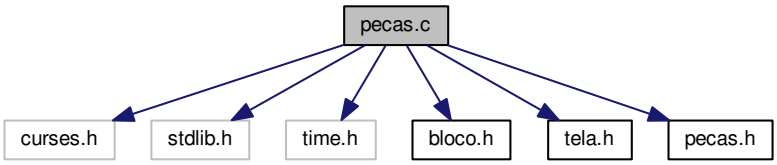
Saída convertida.

4.4 Referência do Arquivo pecas.c

```
#include <curses.h>
```

```
#include <stdlib.h>
#include <time.h>
#include "bloco.h"
#include "tela.h"
#include "pecas.h"
```

Gráfico de dependência de inclusões para pecas.c:



Funções

- void nova_peca (Tela *tela)
- peca * cria_peca_I (Tela *tela)
- peca * cria_peca_Z (Tela *tela)
- peca * cria_peca_T (Tela *tela)
- peca * cria_peca_O (Tela *tela)
- peca * cria_peca_L (Tela *tela)
- void move_peca_x (peca *p, int x)
- void move_peca_y (peca *p, int y)
- void rotaciona_peca (peca *peca)
- void speed_up (peca *peca, int y)
- void libera_peca (peca *p)

Variáveis

- unsigned short int cor_nova_peca = 4
- unsigned short int speed_ups = 0

4.4.1 Funções

4.4.1.1 peca* cria_peca_I (Tela * *tela*)

Cria uma peça do tipo I na tela.

Parâmetros

tela	Ponteiro para tela.

Retorna

Ponteiro para a peça.

4.4.1.2 peca* cria_peca_L (Tela * *tela*)

Cria peça L na tela.

Parâmetros

tela	Ponteiro para a tela.

Retorna

Ponteiro para peça.

4.4.1.3 peca* cria_peca_O (Tela * *tela*)

Cria peça O na tela.

Parâmetros

tela	Ponteiro para a tela de jogo.

Retorna

Ponteiro para a peça.

4.4.1.4 peca* cria_peca_T (Tela * *tela*)

Cria peça T na tela.

Parâmetros

tela	Ponteiro para tela de jogo.

Retorna

Ponteiro para a peça.

4.4.1.5 peca* cria_peca_Z (Tela * *tela*)

Cria uma peça do tipo Z na tela.

Parâmetros

tela	Ponteiro para a tela de jogo.

Retorna

Ponteiro para a peça.

4.4.1.6 void libera_peca (peca * p)

Libera a memória alocada para a peça.

Parâmetros

p	Ponteiro para a peça a ser liberada.

4.4.1.7 void move_peca_x (peca * p, int x)

Movimenta a peça no eixo x, ou seja no sentido horizontal.

Parâmetros

p	A peça a ser movimentada.
x	Indica a direção do movimento. Se positivo, para a direita. Se negativo, para a esquerda.

4.4.1.8 void move_peca_y (peca * p, int y)

Movimenta a peça no eixo y, ou seja, na direção vertical.

Parâmetros

p	Peça a ser movida.
y	Sempre deve ser positivo, pois a peça só pode se movimentar para baixo.

4.4.1.9 void nova_peca (Tela * tela)

Cria nova peça na tela do jogo. O tipo de peça a ser criado é randomizado. A coloração das peças se dá de forma cíclica.

Parâmetros

tela	Ponteiro para tela de jogo.

<Indica o tipo da peça

4.4.1.10 void rotaciona_peca (peca * peca)

Rotaciona a peça no sentido horário em torno de seu centro de rotação.

Parâmetros

peca	Ponteiro para a peça.

4.4.1.11 void speed_up (peca * *peca,* int *y*)

Dobra a velocidade da peça. Pode ser chamada com sucesso no máximo 5 vezes para a mesma peça.

Parâmetros

peca	Ponteiro para peça.

4.4.2 Variáveis

4.4.2.1 unsigned short int cor_nova_peca = 4

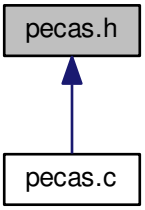
Par de cores das peças variam entre 4 e 7.

4.4.2.2 unsigned short int speed_ups = 0

Indica quantas chamadas bem-sucedidas para a função speed_up foi feita para uma peça. Valor máximo de 5

4.5 Referência do Arquivo pecas.h

Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:



Estruturas de Dados

- struct Peca

Definições de Tipos

- typedef struct Peca peca

Enumerações

- enum Tipo_Peca {
  Tipo_I, **Tipo_Z**, **Tipo_T**, **Tipo_O**,
  Tipo_L }

Funções

- void nova_peca (Tela *tela)
- void move_peca_x (peca *peca, int x)
- void move_peca_y (peca *peca, int y)
- void rotaciona_peca (peca *peca)
- void speed_up (peca *peca, int y)
- void libera_peca (peca *p)
- peca * cria_peca_I (Tela *tela)
- peca * cria_peca_Z (Tela *tela)
- peca * cria_peca_T (Tela *tela)
- peca * cria_peca_O (Tela *tela)
- peca * cria_peca_L (Tela *tela)

4.5.1 Definições dos tipos

4.5.1.1 typedef struct Peca peca

/typedef Peca::peca Informações da peça.

4.5.2 Enumerações

4.5.2.1 enum Tipo_Peca

Indica qual é o formato da peça a ser mostrada na tela.

4.5.3 Funções

4.5.3.1 peca* cria_peca_I (Tela * *tela*)

Cria uma peça do tipo I na tela.

Parâmetros

tela	Ponteiro para tela.

Retorna

Ponteiro para a peça.

4.5.3.2 peca* cria_peca_L (Tela * *tela*)

Cria peça L na tela.

Parâmetros

tela	Ponteiro para a tela.

Retorna

Ponteiro para peça.

4.5.3.3 peca* cria_peca_O (Tela * *tela*)

Cria peça O na tela.

Parâmetros

tela	Ponteiro para a tela de jogo.

Retorna

Ponteiro para a peça.

4.5.3.4 peca* cria_peca_T (Tela * *tela*)

Cria peça T na tela.

Parâmetros

tela	Ponteiro para tela de jogo.

Retorna

Ponteiro para a peça.

4.5.3.5 peca* cria_peca_Z (Tela * *tela*)

Cria uma peça do tipo Z na tela.

Parâmetros

tela	Ponteiro para a tela de jogo.

Retorna

Ponteiro para a peça.

4.5.3.6 void libera_peca (peca * *p*)

Libera a memória alocada para a peça.

Parâmetros

p	Ponteiro para a peça a ser liberada.

4.5.3.7 void move_peca_x (peca * p, int x)

Movimenta a peça no eixo x, ou seja no sentido horizontal.

Parâmetros

p	A peça a ser movimentada.
x	Indica a direção do movimento. Se positivo, para a direita. Se negativo, para a esquerda.

4.5.3.8 void move_peca_y (peca * p, int y)

Movimenta a peça no eixo y, ou seja, na direção vertical.

Parâmetros

p	Peça a ser movida.
y	Sempre deve ser positivo, pois a peça só pode se movimentar para baixo.

4.5.3.9 void nova_peca (Tela * tela)

Cria nova peça na tela do jogo. O tipo de peça a ser criado é randomizado. A coloração das peças se dá de forma cíclica.

Parâmetros

tela	Ponteiro para tela de jogo.

<Indica o tipo da peça

4.5.3.10 void rotaciona_peca (peca * peca)

Rotaciona a peça no sentido horário em torno de seu centro de rotação.

Parâmetros

peca	Ponteiro para a peça.

4.5.3.11 void speed_up (peca * peca, int y)

Dobra a velocidade da peça. Pode ser chamada com sucesso no máximo 5 vezes para a mesma peça.

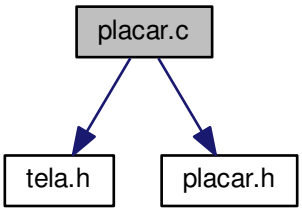
Parâmetros

peca	Ponteiro para peça.

4.6 Referência do Arquivo placar.c

```
#include "tela.h"
#include "placar.h"
```

Gráfico de dependência de inclusões para placar.c:



Funções

- void cria_placar ()
- void atualiza_placar (int pontuacao)
- void mostra_placar ()
- void seta_jogador (Tela *t)
- void destroi_placar ()

4.6.1 Funções

4.6.1.1 void atualiza_placar (int *pontuacao*)

Ordena o placar por ordem de pontuação. O limite é de 5 jogadores, sendo excluída a menor pontuação para manter este padrão.

Parâmetros

pontuacao	Pontuação do jogador atual.

4.6.1.2 void cria_placar ( )

Cria o placar, inicializando ou lendo o arquivo de pontuação, carregando pro programa os valores obtidos.

4.6.1.3 void destroi_placar ( )

Libera a memória alocada para o placar.

4.6.1.4 void mostra_placar ( )

Mostra o placar ao fim do jogo.

4.6.1.5 void seta_jogador (Tela * t)

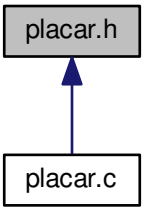
Copia os dados da variável local de escore para a tela de execução.

Parâmetros

t	Ponteiro para tela.

4.7 Referência do Arquivo placar.h

Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:



Estruturas de Dados

- struct Placar

Funções

- void cria_placar ()
- void atualiza_placar (int pontuacao)
- void mostra_placar ()
- void destroi_placar ()
- void seta_jogador (Tela *t)

Variáveis

- char jogadores [5][4]
- int pontuacoes [5]
- int tempos_m [5]
- int tempos_s [5]
- int anos [5]
- int meses [5]
- int dias [5]
- int contador_jogadores
- FILE * arquivo
- char jogador [3]
- int pontuacao
- int tempo_m
- int tempo_s
- **placar**

4.7.1 Funções

4.7.1.1 void atualiza_placar (int *pontuacao*)

Ordena o placar por ordem de pontuação. O limite é de 5 jogadores, sendo excluída a menor pontuação para manter este padrão.

Parâmetros

pontuacao	Pontuação do jogador atual.

4.7.1.2 void cria_placar ( )

Cria o placar, inicializando ou lendo o arquivo de pontuação, carregando pro programa os valores obtidos.

4.7.1.3 void destroi_placar ( )

Libera a memória alocada para o placar.

4.7.1.4 void mostra_placar ( )

Mostra o placar ao fim do jogo.

4.7.1.5 void seta_jogador (Tela ∗ *t*)

Copia os dados da variável local de escore para a tela de execução.

Parâmetros

t	Ponteiro para tela.

4.7.2 Variáveis

4.7.2.1 int anos[5]

Vetor de anos dos jogadores.

4.7.2.2 FILE∗ arquivo

Ponteiro para o arquivo usado para abrir o placar (pontuacao.txt) para leitura e escrita.

4.7.2.3 int contador_jogadores

Contador de jogadores presentes no placar (máximo 5).

4.7.2.4 int dias[5]

Vetor de dias dos jogadores no placar.

4.7.2.5 char jogador[3]

Nome do jogador atual.

4.7.2.6 char jogadores[5][4]

Vetor de jogadores.

4.7.2.7 int meses[5]

Vetor de meses dos jogadores no placar.

4.7.2.8 int pontuacao

Pontuacao do jogador atual.

4.7.2.9 int pontuacoes[5]

Vetor de pontuação dos jogadores no placar.

4.7.2.10 int tempo_m

Tempo em minutos do jogador atual.

4.7.2.11 int tempo_s

Tempo em segundos do jogador atual.

4.7.2.12 int tempos_m[5]

Vetor de tempo em minutos dos jogadores.

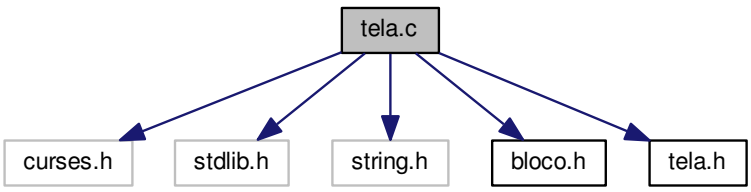
4.7.2.13 int tempos_s[5]

Vetor de tempo em segundos dos jogadores.

4.8 Referência do Arquivo tela.c

```
#include <curses.h>
#include <stdlib.h>
#include <string.h>
#include "bloco.h"
#include "tela.h"
```

Gráfico de dependência de inclusões para tela.c:



Funções

- Tela ∗ cria_tela ()
- void mostra_tela (Tela ∗t)
- void mostra_pontos (int pontos)
- void mostra_tempo (int minutos, int segundos)
- int verifica_linha (Tela ∗t)
- void limpa_linha (Tela ∗t, int y)
- void desce_linhas (Tela ∗t, int y)
- int checa_fim (Tela ∗t)
- void destroi_tela (Tela ∗t)
- void define_jogador (Tela ∗t)
- void troca_letra (Tela ∗t, int valor)
- void muda_letra (Tela ∗t, int valor)

4.8.1 Funções

4.8.1.1 int checa_fim (Tela ∗ *t*)

Verifica se as peças ultrapassaram o limite superior do jogo.

Parâmetros

t	Ponteiro para a tela de jogo.

Retorna

Verdadeiro se ultrapassou o limite. Falso caso contrário.

4.8.1.2 Tela∗ cria_tela ( )

Cria uma tela de jogo com os parâmetros corretos.

Retorna

Retorna um ponteiro para tal tela.

4.8.1.3 void define_jogador (Tela ∗ *t*)

Estado da tela onde o jogador escolhe o seu apelido.

Parâmetros

t	Ponteiro para tela.

4.8.1.4 void desce_linhas (Tela ∗ *t*, int *y*)

Desce determinada linha da tela.

Parâmetros

t	Ponteiro para a tela de jogo.
y	Posição para a linha.

4.8.1.5 void destroi_tela (Tela * t)

Libera o espaço de memória reservado para a tela de jogo.

Parâmetros

t	Ponteiro para a tela de jogo.

4.8.1.6 void limpa_linha (Tela * t, int y)

Limpa uma determinada linha do jogo.

Parâmetros

t	Ponteiro para a tela do jogo.
y	Posição da linha a ser eliminada.

4.8.1.7 void mostra_pontos (int pontos)

Mostra a pontuação do jogados.

Parâmetros

pontos	O escore atual.

4.8.1.8 void mostra_tela (Tela * t)

Mostra a tela de jogo, conforme seu atual estado. Também inicializa os pares de cores a serem utilizados.

Parâmetros

t	Ponteiro para tela a ser mostrada.

4.8.1.9 void mostra_tempo (int minutos, int segundos)

Mostra o tempo da partida.

Parâmetros

minutos	Tempo em minutos.
segundos	Tempo em segundos.

4.8.1.10 void muda_letra (Tela * t, int valor)

Função que alatera o caracter da atual letra selecionada para escolha do apelido.

Parâmetros

t	Ponteiro para tela.
valor	Quantidade a ser incrementada.

4.8.1.11 void troca_letra (Tela * t, int valor)

Função que altera a atual letra selecionada para escolha do apelido.

Parâmetros

t	Ponteiro para tela.
valor	Quantidade a ser incrementada.

4.8.1.12 int verifica_linha (Tela * t)

Verifica se uma linha horizontal do jogo está completamente preenchida.

Parâmetros

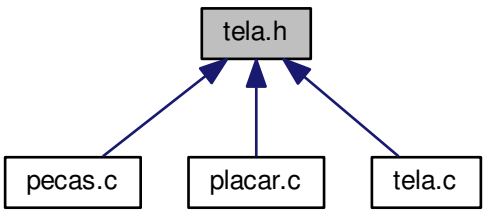
t	Ponteiro para a tela do jogo.

Retorna

100 se a linha estiver preenchida. 0 caso contrário.

4.9 Referência do Arquivo tela.h

Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:



Estruturas de Dados

- struct tela

Definições de Tipos

- typedef struct tela Tela

Enumerações

- enum estado { **INICIO**, **JOGO**, **FINAL** }

Funções

- Tela ∗ cria_tela ()
- void mostra_tela (Tela ∗t)
- void mostra_pontos (int pontos)
- void mostra_tempo (int minutos, int segundos)
- void destroi_tela (Tela ∗t)
- void define_jogador (Tela ∗t)
- void troca_letra (Tela ∗t, int valor)
- void muda_letra (Tela ∗t, int valor)
- void limpa_linha (Tela ∗t, int y)
- void desce_linhas (Tela ∗t, int y)
- int verifica_linha (Tela ∗t)
- int checa_fim (Tela ∗t)

4.9.1 Definições dos tipos

4.9.1.1 typedef struct tela Tela

/struct Define a tela do jogo.

4.9.2 Enumerações

4.9.2.1 enum estado

Variável enumerada que indica o estado do jogo.

4.9.3 Funções

4.9.3.1 int checa_fim (Tela ∗ *t*)

Verifica se as peças ultrapassaram o limite superior do jogo.

Parâmetros

t	Ponteiro para a tela de jogo.

Retorna

Verdadeiro se ultrapassou o limite. Falso caso contrário.

4.9.3.2 Tela* cria_tela ( )

Cria uma tela de jogo com os parâmetros corretos.

Retorna

Retorna um ponteiro para tal tela.

4.9.3.3 void define_jogador (Tela * t)

Estado da tela onde o jogador escolhe o seu apelido.

Parâmetros

t	Ponteiro para tela.

4.9.3.4 void desce_linhas (Tela * t, int y)

Desce determinada linha da tela.

Parâmetros

t	Ponteiro para a tela de jogo.
y	Posição para a linha.

4.9.3.5 void destroi_tela (Tela * t)

Libera o espaço de memória reservado para a tela de jogo.

Parâmetros

t	Ponteiro para a tela de jogo.

4.9.3.6 void limpa_linha (Tela * t, int y)

Limpa uma determinada linha do jogo.

Parâmetros

t	Ponteiro para a tela do jogo.
y	Posição da linha a ser eliminada.

4.9.3.7 void mostra_pontos (int *pontos*)

Mostra a pontuação do jogados.

Parâmetros

pontos	O escore atual.

4.9.3.8 void mostra_tela (Tela * *t*)

Mostra a tela de jogo, conforme seu atual estado. Também inicializa os pares de cores a serem utilizados.

Parâmetros

t	Ponteiro para tela a ser mostrada.

4.9.3.9 void mostra_tempo (int *minutos,* int *segundos*)

Mostra o tempo da partida.

Parâmetros

minutos	Tempo em minutos.
segundos	Tempo em segundos.

4.9.3.10 void muda_letra (Tela * *t,* int *valor*)

Função que alatera o caracter da atual letra selecionada para escolha do apelido.

Parâmetros

t	Ponteiro para tela.
valor	Quantidade a ser incrementada.

4.9.3.11 void troca_letra (Tela * *t,* int *valor*)

Função que altera a atual letra selecionada para escolha do apelido.

Parâmetros

t	Ponteiro para tela.
valor	Quantidade a ser incrementada.

4.9.3.12 int verifica_linha (Tela * t)

Verifica se uma linha horizontal do jogo está completamente preenchida.

Parâmetros

t	Ponteiro para a tela do jogo.

Retorna

100 se a linha estiver preenchida. 0 caso contrário.

Índice Remissivo

abaixo
    Bloco, 3
    bloco.h, 9
acima
    Bloco, 3
    bloco.h, 9
anos
    Placar, 5
    placar.h, 21
arquivo
    Placar, 5
    placar.h, 21
atualiza_placar
    placar.c, 19
    placar.h, 21

Bloco, 2
    abaixo, 3
    acima, 3
    bolinha, 3
    cor, 3
    direita, 3
    esquerda, 3
    move, 3
    pos_x, 3
    pos_y, 3
bloco.h, 8
    abaixo, 9
    acima, 9
    bolinha, 9
    cor, 9
    direita, 9
    esquerda, 9
    move, 9
    pos_x, 9
    pos_y, 9
blocos
    Peca, 4
    tela, 7
bolinha
    Bloco, 3
    bloco.h, 9

centro_de_rotacao
    Peca, 4
checa_fim
    tela.c, 23
    tela.h, 26
comprimento
    tela, 7
contador_jogadores
    Placar, 5
    placar.h, 21
cor
    Bloco, 3
    bloco.h, 9
cor_nova_peca
    pecas.c, 15
cor_peca
    Peca, 4
cria_peca_I
    pecas.c, 12
    pecas.h, 16
cria_peca_L
    pecas.c, 12
    pecas.h, 16
cria_peca_O
    pecas.c, 13
    pecas.h, 17
cria_peca_T
    pecas.c, 13
    pecas.h, 17
cria_peca_Z
    pecas.c, 13
    pecas.h, 17
cria_placar
    placar.c, 19
    placar.h, 21
cria_tela
    tela.c, 23
    tela.h, 27

define_jogador
    tela.c, 23
    tela.h, 27
desce_linhas
    tela.c, 23
    tela.h, 27
destroi_placar
    placar.c, 19
    placar.h, 21
destroi_tela
    tela.c, 24
    tela.h, 27
dias
    Placar, 5
    placar.h, 21
direita
    Bloco, 3
    bloco.h, 9

engine.c, 10
    finaliza_ncurses, 10
    inicia_ncurses, 10
    pega_input, 10
engine.h, 11
    finaliza_ncurses, 11
    inicia_ncurses, 11
    pega_input, 11
esquerda

Bloco, 3
bloco.h, 9
estado
tela, 7
tela.h, 26

finaliza_ncurses
engine.c, 10
engine.h, 11

inicia_ncurses
engine.c, 10
engine.h, 11

janela
tela, 7
jogador
Placar, 5
placar.h, 21
tela, 7
jogadores
Placar, 5
placar.h, 22

largura
tela, 7
libera_peca
pecas.c, 13
pecas.h, 17
limpa_linha
tela.c, 24
tela.h, 27

meses
Placar, 5
placar.h, 22
mostra_placar
placar.c, 19
placar.h, 21
mostra_pontos
tela.c, 24
tela.h, 28
mostra_tela
tela.c, 24
tela.h, 28
mostra_tempo
tela.c, 24
tela.h, 28
move
Bloco, 3
bloco.h, 9
move_peca
Peca, 4
move_peca_x
pecas.c, 14
pecas.h, 17
move_peca_y
pecas.c, 14
pecas.h, 18

muda_letra
tela.c, 25
tela.h, 28

nova_peca
pecas.c, 14
pecas.h, 18

Peca, 3
blocos, 4
centro_de_rotacao, 4
cor_peca, 4
move_peca, 4
tamanho, 4
tipo, 4
velocidade, 4
peca
pecas.h, 16
tela, 7
pecas.c, 11
cor_nova_peca, 15
cria_peca_I, 12
cria_peca_L, 12
cria_peca_O, 13
cria_peca_T, 13
cria_peca_Z, 13
libera_peca, 13
move_peca_x, 14
move_peca_y, 14
nova_peca, 14
rotaciona_peca, 14
speed_up, 14
speed_ups, 15
pecas.h, 15
cria_peca_I, 16
cria_peca_L, 16
cria_peca_O, 17
cria_peca_T, 17
cria_peca_Z, 17
libera_peca, 17
move_peca_x, 17
move_peca_y, 18
nova_peca, 18
peca, 16
rotaciona_peca, 18
speed_up, 18
Tipo_Peca, 16
pega_input
engine.c, 10
engine.h, 11
Placar, 4
anos, 5
arquivo, 5
contador_jogadores, 5
dias, 5
jogador, 5
jogadores, 5
meses, 5
pontuacao, 5

pontuacoes, 5
tempo_m, 5
tempo_s, 5
tempos_m, 5
tempos_s, 5
placar.c, 19
atualiza_placar, 19
cria_placar, 19
destroi_placar, 19
mostra_placar, 19
seta_jogador, 19
placar.h, 20
anos, 21
arquivo, 21
atualiza_placar, 21
contador_jogadores, 21
cria_placar, 21
destroi_placar, 21
dias, 21
jogador, 21
jogadores, 22
meses, 22
mostra_placar, 21
pontuacao, 22
pontuacoes, 22
seta_jogador, 21
tempo_m, 22
tempo_s, 22
tempos_m, 22
tempos_s, 22
pontos
tela, 7
pontuacao
Placar, 5
placar.h, 22
pontuacoes
Placar, 5
placar.h, 22
pos_x
Bloco, 3
bloco.h, 9
pos_y
Bloco, 3
bloco.h, 9

rotaciona_peca
pecas.c, 14
pecas.h, 18

seta_jogador
placar.c, 19
placar.h, 21
speed_up
pecas.c, 14
pecas.h, 18
speed_ups
pecas.c, 15

tamanho
Peca, 4
Tela
tela.h, 26
tela, 6
blocos, 7
comprimento, 7
estado, 7
janela, 7
jogador, 7
largura, 7
peca, 7
pontos, 7
tempo_m, 7
tempo_s, 7
tela.c, 22
checa_fim, 23
cria_tela, 23
define_jogador, 23
desce_linhas, 23
destroi_tela, 24
limpa_linha, 24
mostra_pontos, 24
mostra_tela, 24
mostra_tempo, 24
muda_letra, 25
troca_letra, 25
verifica_linha, 25
tela.h, 25
checa_fim, 26
cria_tela, 27
define_jogador, 27
desce_linhas, 27
destroi_tela, 27
estado, 26
limpa_linha, 27
mostra_pontos, 28
mostra_tela, 28
mostra_tempo, 28
muda_letra, 28
Tela, 26
troca_letra, 28
verifica_linha, 29
tempo_m
Placar, 5
placar.h, 22
tela, 7
tempo_s
Placar, 5
placar.h, 22
tela, 7
tempos_m
Placar, 5
placar.h, 22
tempos_s
Placar, 5
placar.h, 22
timeb, 8
tipo

Peca, 4
Tipo_Peca
pecas.h, 16
troca_letra
tela.c, 25
tela.h, 28

velocidade
Peca, 4
verifica_linha
tela.c, 25
tela.h, 29