Tetris - PS Edition

Gerado por Doxygen 1.8.11

Quinta, 9 de Junho de 2016 18:21:21

Sumário

1 Índice das Estruturas de Dados **2**

1.1 Estruturas de Dados 2

2 Índice dos Arquivos **2**

2.1 Lista de Arquivos 2

3 Estruturas **2**

3.1 Referência da Estrutura Bloco 2

3.1.1 Campos 3

3.2 Referência da Estrutura Peca 4

3.2.1 Campos 4

3.3 Referência da Estrutura Tela 5

3.3.1 Descrição Detalhada 5

3.3.2 Campos 5

4 Arquivos **7**

4.1 Referência do Arquivo bloco.h 7

4.2 Referência do Arquivo engine.c 8

4.2.1 Funções 8

4.3 Referência do Arquivo engine.h 9

4.3.1 Funções 9

4.4 Referência do Arquivo main.c 10

4.5 Referência do Arquivo pecas.c 10

4.5.1 Funções 11

4.5.2 Variáveis 12

4.6 Referência do Arquivo pecas.h 12

4.6.1 Funções 14

4.7 Referência do Arquivo tela.c 15

4.7.1 Funções 16

4.8 Referência do Arquivo tela.h 18

4.8.1 Definições dos tipos 19

4.8.2 Enumerações 20

4.8.3 Funções 20

Índice 23

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 4

Tela 5

2 Índice dos Arquivos

2.1 Lista de Arquivos

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

bloco.h 7

engine.c 8

engine.h 9

main.c 10

pecas.c 10

pecas.h 12

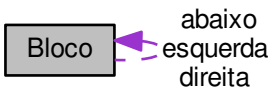
tela.c 15

tela.h 18

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

3.1.1 Campos

3.1.1.1 struct Bloco* Bloco::abaixo

Ponteiro para vizinho abaixo.

3.1.1.2 char Bloco::bolinha

Caractere atual da peça.

3.1.1.3 unsigned short int Bloco::cor

Cor da peça.

3.1.1.4 struct Bloco* Bloco::direita

Ponteiro para vizinho à direita.

3.1.1.5 struct Bloco* Bloco::esquerda

Ponteiro para vizinho à esquerda.

3.1.1.6 unsigned short int Bloco::move

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

3.1.1.7 int Bloco::pos_x

Coordenada cartesiana horizontal do bloco.

3.1.1.8 int Bloco::pos_y

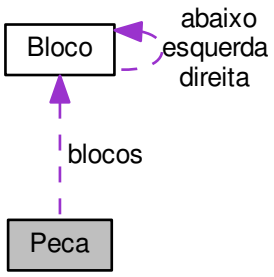
Coordenada cartesiana vertical do bloco.

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

- bloco.h

3.2 Referência da Estrutura Peca

Diagrama de colaboração para Peca:



Campos de Dados

- int tamanho
- unsigned short int cor_peca
- unsigned short int move_peca
- bloco ∗ blocos [ ]

3.2.1 Campos

3.2.1.1 bloco∗ Peca::blocos[ ]

Referência para blocos na tela.

3.2.1.2 unsigned short int Peca::cor_peca

Cor da peça.

3.2.1.3 unsigned short int Peca::move_peca

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

3.2.1.4 int Peca::tamanho

Tamanho da peça.

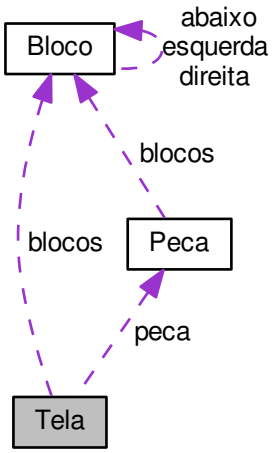
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- pecas.h

3.3 Referência da Estrutura Tela

`#include <tela.h>`

Diagrama de colaboração para Tela:



Campos de Dados

- int estado
- int pontos
- int tempo_m
- int tempo_s
- int comprimento
- int largura
- WINDOW ∗ janela
- struct Peca ∗ peca
- bloco blocos [ ]

3.3.1 Descrição Detalhada

/struct Define a tela do jogo.

3.3.2 Campos

3.3.2.1 bloco Tela::blocos[ ]

Matriz dos blocos na tela.

3.3.2.2 int Tela::comprimento

Comprimento da tela.

3.3.2.3 int Tela::estado

Estado atual do jogo.

3.3.2.4 WINDOW* Tela::janela

Ponteiro para a janela do jogo.

3.3.2.5 int Tela::largura

Largura da tela.

3.3.2.6 struct Peca* Tela::peca

Ponteiro para a peça em movimento.

3.3.2.7 int Tela::pontos

Pontuação do jogador.

3.3.2.8 int Tela::tempo_m

Tempo de execução em minutos.

3.3.2.9 int Tela::tempo_s

Tempo de execução em segundos.

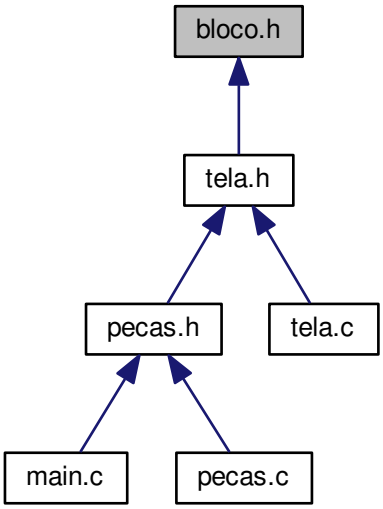
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- tela.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 **COMPRIMENTO** 15
- #define **LARGURA** 25

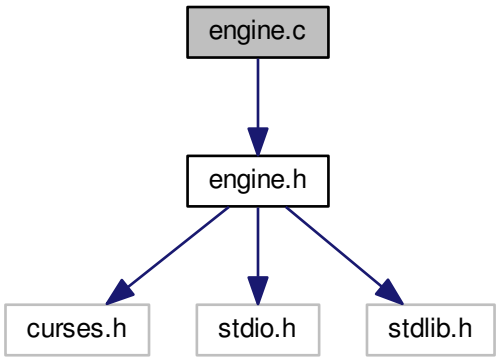
Definições de Tipos

- typedef struct Bloco **bloco**

4.2 Referência do Arquivo engine.c

```
#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.

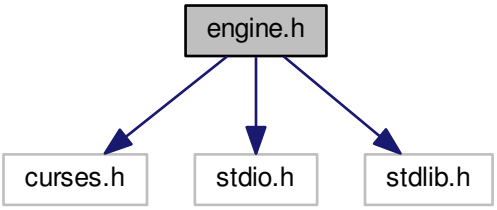
Retorna

Saída convertida.

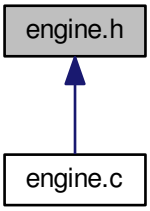
4.3 Referência do Arquivo engine.h

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

Gráfico de dependência de inclusões para 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.

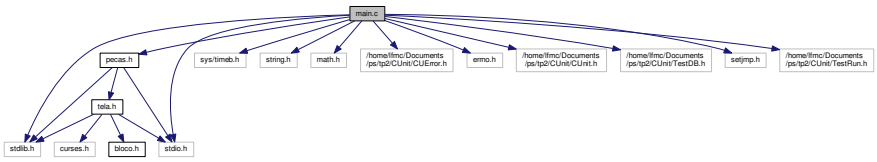
Retorna

Saída convertida.

4.4 Referência do Arquivo main.c

```
#include <stdlib.h>
#include <stdio.h>
#include "pecas.h"
#include <sys/timeb.h>
#include "testes.c"
```

Gráfico de dependência de inclusões para main.c:



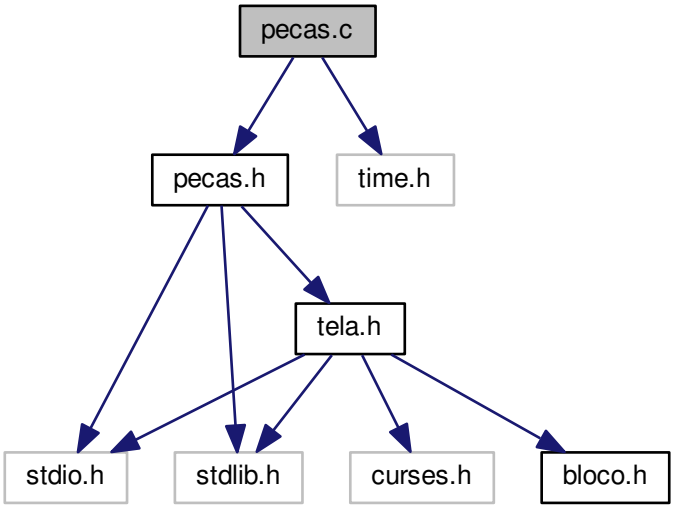
Funções

- int **main** ()

4.5 Referência do Arquivo pecas.c

```
#include "pecas.h"
#include <time.h>
```

Gráfico de dependência de inclusões para pecas.c:



Funções

- void nova_peca (Tela *tela)
- void move_peca_x (peca *p, int x)
- void move_peca_y (peca *p, int y)
- void libera_peca (peca *p)

Variáveis

- unsigned short int cor_nova_peca = 4

4.5.1 Funções

4.5.1.1 void libera_peca (peca * p)

Libera a memória alocada para a peça.

Parâmetros

p	Ponteiro para a peça a ser liberada.

4.5.1.2 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.

4.5.1.3 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.

4.5.1.4 void nova_peca (Tela * tela)

Gera nova peça do jogo. Orientação e tamanho são dados de forma pseudoaleatória. A cor é dada de forma cíclica.

Parâmetros

tela	Ponteiro para tela de jogo.

<Indica a orientação da peça. Se 1, a orientação é vertical. Se 0, a orientação é horizontal

<Indica o tamanho da peça

4.5.2 Variáveis

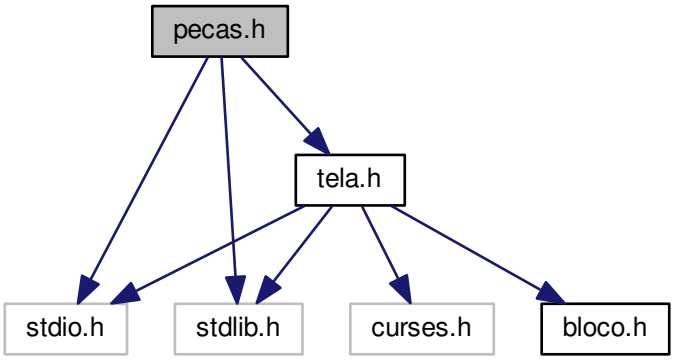
4.5.2.1 unsigned short int cor_nova_peca = 4

Par de cores das peças variam entre 4 e 7.

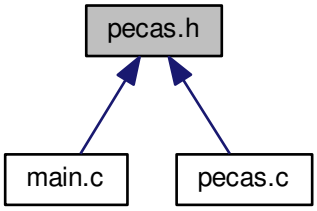
4.6 Referência do Arquivo pecas.h

```
#include <stdio.h>
#include <stdlib.h>
#include "tela.h"
```

Gráfico de dependência de inclusões para 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**

Funções

- void nova_peca (Tela *tela)
- void move_peca_x (peca *peca, int x)
- void move_peca_y (peca *peca, int y)
- void libera_peca (peca *p)

4.6.1 Funções

4.6.1.1 void libera_peca (peca * p)

Libera a memória alocada para a peça.

Parâmetros

p	Ponteiro para a peça a ser liberada.

4.6.1.2 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.

4.6.1.3 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.

4.6.1.4 void nova_peca (Tela * tela)

Gera nova peça do jogo. Orientação e tamanho são dados de forma pseudoaleatória. A cor é dada de forma cíclica.

Parâmetros

tela	Ponteiro para tela de jogo.

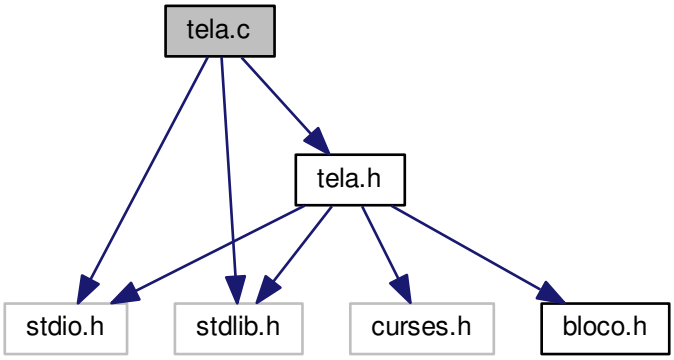
<Indica a orientação da peça. Se 1, a orientação é vertical. Se 0, a orientação é horizontal

<Indica o tamanho da peça

4.7 Referência do Arquivo tela.c

```
#include <stdio.h>
#include <stdlib.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)

4.7.1 Funções

4.7.1.1 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.

4.7.1.2 Tela∗ cria_tela ( )

Cria uma tela de jogo com os parâmetros corretos.

Retorna

Retorna um ponteiro para tal tela.

4.7.1.3 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.

4.7.1.4 void destroi_tela (Tela * t)

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

Parâmetros

t	Ponteiro para a tela de jogo.

4.7.1.5 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.

4.7.1.6 void mostra_pontos (int pontos)

Mostra a pontuação do jogados.

Parâmetros

pontos	O escore atual.

4.7.1.7 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.

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

Mostra o tempo da partida.

Parâmetros

minutos	Tempo em minutos.
segundos	Tempo em segundos.

4.7.1.9 int verifica_linha (Tela * *t*)

Verifica se uma linha horizontal do jogo está completamente preenchida.

Parâmetros

t	Ponteiro para a tela do jogo.

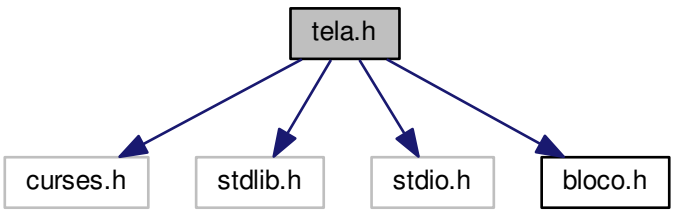
Retorna

100 se a linha estiver preenchida. 0 caso contrário.

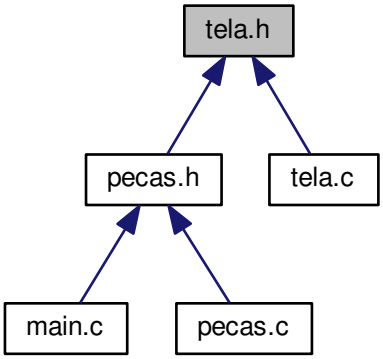
4.8 Referência do Arquivo tela.h

```
#include <curses.h>
#include <stdlib.h>
#include <stdio.h>
#include "bloco.h"
```

Gráfico de dependência de inclusões para 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 limpa_linha (Tela ∗t, int y)
- void desce_linhas (Tela ∗t, int y)
- int verifica_linha (Tela ∗t)
- int checa_fim (Tela ∗t)

4.8.1 Definições dos tipos

4.8.1.1 typedef struct Tela Tela

/struct Define a tela do jogo.

4.8.2 Enumerações

4.8.2.1 enum estado

Variável enumerada que indica o estado do jogo.

4.8.3 Funções

4.8.3.1 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.

4.8.3.2 Tela* cria_tela ( )

Cria uma tela de jogo com os parâmetros corretos.

Retorna

Retorna um ponteiro para tal tela.

4.8.3.3 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.

4.8.3.4 void destroi_tela (Tela * t)

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

Parâmetros

t	Ponteiro para a tela de jogo.

4.8.3.5 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.

4.8.3.6 void mostra_pontos (int pontos)

Mostra a pontuação do jogados.

Parâmetros

pontos	O escore atual.

4.8.3.7 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.

4.8.3.8 void mostra_tempo (int minutos, int segundos)

Mostra o tempo da partida.

Parâmetros

minutos	Tempo em minutos.
segundos	Tempo em segundos.

4.8.3.9 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, 2
    abaixo, 3
    bolinha, 3
    cor, 3
    direita, 3
    esquerda, 3
    move, 3
    pos_x, 3
    pos_y, 3
bloco.h, 7
blocos
    Peca, 4
    Tela, 5
bolinha
    Bloco, 3

checa_fim
    tela.c, 16
    tela.h, 20
comprimento
    Tela, 5
cor
    Bloco, 3
cor_nova_peca
    pecas.c, 12
cor_peca
    Peca, 4
cria_tela
    tela.c, 16
    tela.h, 20

desce_linhas
    tela.c, 17
    tela.h, 20
destroi_tela
    tela.c, 17
    tela.h, 20
direita
    Bloco, 3

engine.c, 8
    finaliza_ncurses, 8
    inicia_ncurses, 8
    pega_input, 8
engine.h, 9
    finaliza_ncurses, 9
    inicia_ncurses, 9
    pega_input, 10
esquerda
    Bloco, 3
estado
    Tela, 6
    tela.h, 20

finaliza_ncurses
    engine.c, 8
    engine.h, 9

inicia_ncurses
    engine.c, 8
    engine.h, 9

janela
    Tela, 6

largura
    Tela, 6
libera_peca
    pecas.c, 11
    pecas.h, 14
limpa_linha
    tela.c, 17
    tela.h, 20

main.c, 10
mostra_pontos
    tela.c, 17
    tela.h, 21
mostra_tela
    tela.c, 17
    tela.h, 21
mostra_tempo
    tela.c, 17
    tela.h, 21
move
    Bloco, 3
move_peca
    Peca, 4
move_peca_x
    pecas.c, 11
    pecas.h, 15
move_peca_y
    pecas.c, 12
    pecas.h, 15

nova_peca
    pecas.c, 12
    pecas.h, 15

Peca, 4
    blocos, 4
    cor_peca, 4
    move_peca, 4
    tamanho, 4
peca
    Tela, 6
pecas.c, 10
    cor_nova_peca, 12
    libera_peca, 11
    move_peca_x, 11

move_peca_y, 12
nova_peca, 12
pecas.h, 12
libera_peca, 14
move_peca_x, 15
move_peca_y, 15
nova_peca, 15
pega_input
engine.c, 8
engine.h, 10
pontos
Tela, 6
pos_x
Bloco, 3
pos_y
Bloco, 3

tamanho
Peca, 4
Tela, 5
blocos, 5
comprimento, 5
estado, 6
janela, 6
largura, 6
peca, 6
pontos, 6
tela.h, 19
tempo_m, 6
tempo_s, 6
tela.c, 15
checa_fim, 16
cria_tela, 16
desce_linhas, 17
destroi_tela, 17
limpa_linha, 17
mostra_pontos, 17
mostra_tela, 17
mostra_tempo, 17
verifica_linha, 18
tela.h, 18
checa_fim, 20
cria_tela, 20
desce_linhas, 20
destroi_tela, 20
estado, 20
limpa_linha, 20
mostra_pontos, 21
mostra_tela, 21
mostra_tempo, 21
Tela, 19
verifica_linha, 21
tempo_m
Tela, 6
tempo_s
Tela, 6

verifica_linha
tela.c, 18
tela.h, 21