

## Tetris - PS Edition

Gerado por Doxygen 1.8.11

Segunda, 13 de Junho de 2016 21:12:12

## Sumário

<b>1</b>	<b>Índice das Estruturas de Dados</b>	<b>2</b>
1.1	Estruturas de Dados	2
<b>2</b>	<b>Índice dos Arquivos</b>	<b>2</b>
2.1	Lista de Arquivos	2
<b>3</b>	<b>Estruturas</b>	<b>3</b>
3.1	Referência da Estrutura Bloco	3
3.1.1	Campos	3
3.2	Referência da Estrutura Peca	4
3.2.1	Campos	5
3.3	Referência da Estrutura Placar	5
3.3.1	Campos	6
3.4	Referência da Estrutura tela	7
3.4.1	Descrição Detalhada	8
3.4.2	Campos	8
3.5	Referência da Estrutura timeb	9
<b>4</b>	<b>Arquivos</b>	<b>9</b>
4.1	Referência do Arquivo bloco.h	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	12
4.4.1	Funções	13
4.4.2	Variáveis	15
4.5	Referência do Arquivo pecas.h	15
4.5.1	Enumerações	17
4.5.2	Funções	17
4.6	Referência do Arquivo placar.c	20
4.6.1	Funções	21
4.7	Referência do Arquivo placar.h	22
4.7.1	Funções	22
4.8	Referência do Arquivo tela.c	23
4.8.1	Funções	24
4.9	Referência do Arquivo tela.h	26
4.9.1	Definições dos tipos	27
4.9.2	Enumerações	28
4.9.3	Funções	28

<b>Índice</b>	<b>31</b>
---------------	-----------

## 1 Índice das Estruturas de Dados

### 1.1 Estruturas de Dados

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

<b>Bloco</b>	<b>3</b>
<b>Peca</b>	<b>4</b>
<b>Placar</b>	<b>5</b>
<b>tela</b>	<b>7</b>
<b>timeb</b>	<b>9</b>

## 2 Índice dos Arquivos

### 2.1 Lista de Arquivos

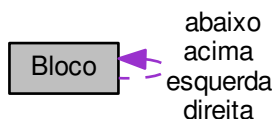
Esta é a lista de todos os arquivos documentados e suas respectivas descrições:

<b>bloco.h</b>	<b>9</b>
<b>cdefs.h</b>	<b>??</b>
<b>engine.c</b>	<b>10</b>
<b>engine.h</b>	<b>11</b>
<b>features.h</b>	<b>??</b>
<b>pecas.c</b>	<b>12</b>
<b>pecas.h</b>	<b>15</b>
<b>placar.c</b>	<b>20</b>
<b>placar.h</b>	<b>22</b>
<b>stubs-64.h</b>	<b>??</b>
<b>stubs.h</b>	<b>??</b>
<b>tela.c</b>	<b>23</b>
<b>tela.h</b>	<b>26</b>
<b>testes.h</b>	<b>??</b>
<b>timeb.h</b>	<b>??</b>
<b>wordsize.h</b>	<b>??</b>

## 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**

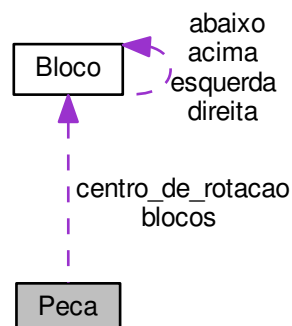
Coordenada cartesiana vertical do bloco.

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

- [bloco.h](#)

## 3.2 Referência da Estrutura Peca

Diagrama de colaboração para Peca:



### Campos de Dados

- int [tamanho](#)
- unsigned short int [cor\\_pecas](#)
- unsigned short int [move\\_pecas](#)
- float [velocidade](#)
- tp\_pecas [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_pecas`

Cor da peça.

#### 3.2.1.4 `unsigned short int Peca::move_pecas`

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_pecas 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.

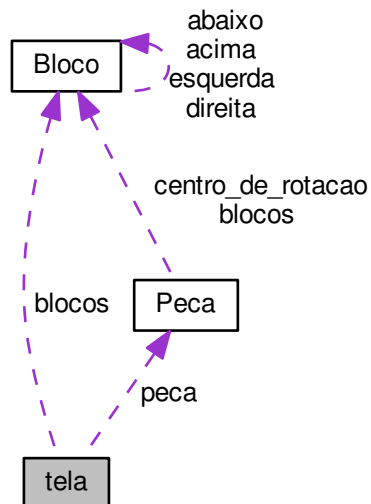
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [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 **int tela::letra**

Auxiliar na escolha de nome do jogador.

#### 3.4.2.8 **struct Peca\* tela::peca**

Ponteiro para a peça em movimento.

#### 3.4.2.9 **int tela::pontos**

Pontuação do jogador.

#### 3.4.2.10 **int tela::tempo\_m**

Tempo de execução em minutos.

## 3.4.2.11 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**

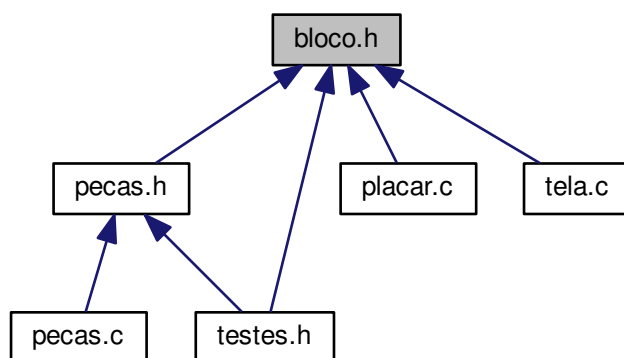
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- 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`

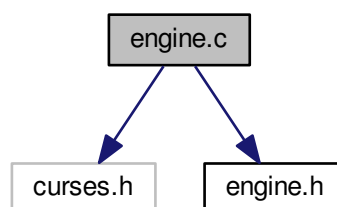
### Definições de Tipos

- `typedef struct Bloco 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

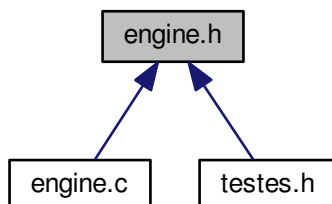
<i>input</i>	Entrada.
--------------	----------

## 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

<i>input</i>	Entrada.
--------------	----------

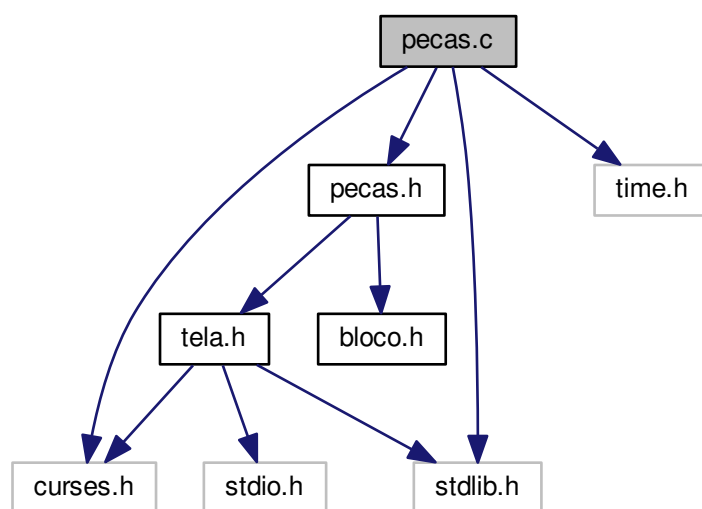
**Retorna**

Saída convertida.

**4.4 Referência do Arquivo pecas.c**

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

Gráfico de dependência de inclusões para pecas.c:

**Funções**

- void [nova\\_pec](#) ([Tela](#) \*[tela](#))
- [peca](#) \* [cria\\_pec](#)\_I ([Tela](#) \*[tela](#))
- [peca](#) \* [cria\\_pec](#)\_Z ([Tela](#) \*[tela](#))
- [peca](#) \* [cria\\_pec](#)\_T ([Tela](#) \*[tela](#))
- [peca](#) \* [cria\\_pec](#)\_O ([Tela](#) \*[tela](#))
- [peca](#) \* [cria\\_pec](#)\_L ([Tela](#) \*[tela](#))
- void [move\\_pec](#)\_x ([peca](#) \*[p](#), int [x](#))
- void [move\\_pec](#)\_y ([peca](#) \*[p](#), int [y](#))
- void [rotaciona\\_pec](#) ([peca](#) \*[peca](#))
- void [speed\\_up](#) ([peca](#) \*[peca](#), int [y](#))
- void [libera\\_pec](#) ([peca](#) \*[p](#))

**Variáveis**

- unsigned short int [cor\\_nova\\_pec](#) = 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

<i>tela</i>	Ponteiro para tela.
-------------	---------------------

###### Retorna

Ponteiro para a peça.

##### 4.4.1.2 `peca* cria_peca_L ( Tela * tela )`

Cria peça L na tela.

###### Parâmetros

<i>tela</i>	Ponteiro para a tela.
-------------	-----------------------

###### Retorna

Ponteiro para peça.

##### 4.4.1.3 `peca* cria_peca_O ( Tela * tela )`

Cria peça O na tela.

###### Parâmetros

<i>tela</i>	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

<i>tela</i>	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**

<i>tela</i>	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**

<i>p</i>	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**

<i>p</i>	A peça a ser movimentada.
<i>x</i>	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**

<i>p</i>	Peça a ser movida.
<i>y</i>	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

<i>tela</i>	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

<i>peca</i>	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

<i>peca</i>	Ponteiro para peça.
<i>y</i>	Inteiro qualquer.

## 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

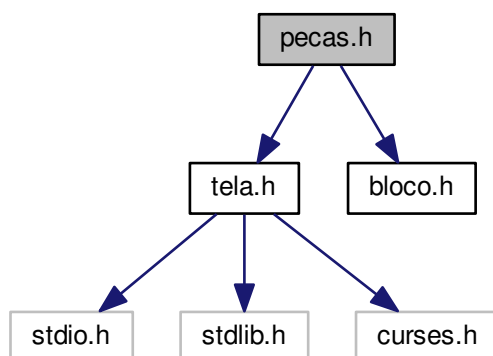
Indica quantas chamadas bem-sucedidas para a função speed\_up foi feita para uma peça. Valor máximo de 5

## 4.5 Referência do Arquivo pecas.h

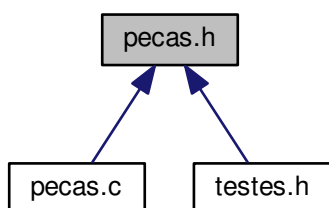
```
#include "tela.h"
#include "bloco.h"
```



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 enum [Tipo\\_Peca](#) **tp\_pecas**
- typedef struct [Peca](#) **peca**

#### Enumerações

- enum [Tipo\\_Peca](#) {  
    **Tipo\_I**, **Tipo\_Z**, **Tipo\_T**, **Tipo\_O**,  
    **Tipo\_L** }

## Funções

- void `nova_pec`a (Tela \*tela)
- void `move_pec`a\_x (pec a \*pec a, int x)
- void `move_pec`a\_y (pec a \*pec a, int y)
- void `rotaciona_pec`a (pec a \*pec a)
- void `speed_up` (pec a \*pec a, int y)
- void `libera_pec`a (pec a \*p)
- pec a \* `cria_pec`a\_I (Tela \*tela)
- pec a \* `cria_pec`a\_Z (Tela \*tela)
- pec a \* `cria_pec`a\_T (Tela \*tela)
- pec a \* `cria_pec`a\_O (Tela \*tela)
- pec a \* `cria_pec`a\_L (Tela \*tela)

### 4.5.1 Enumerações

#### 4.5.1.1 enum Tipo\_Peca

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

### 4.5.2 Funções

#### 4.5.2.1 pec a\* `cria_pec`a\_I ( Tela \* tela )

Cria uma peça do tipo I na tela.

##### Parâmetros

<i>tela</i>	Ponteiro para tela.
-------------	---------------------

##### Retorna

Ponteiro para a peça.

#### 4.5.2.2 pec a\* `cria_pec`a\_L ( Tela \* tela )

Cria peça L na tela.

##### Parâmetros

<i>tela</i>	Ponteiro para a tela.
-------------	-----------------------

##### Retorna

Ponteiro para peça.

#### 4.5.2.3 peca\* cria\_peca\_O ( Tela \* tela )

Cria peça O na tela.

## Parâmetros

<i>tela</i>	Ponteiro para a tela de jogo.
-------------	-------------------------------

## Retorna

Ponteiro para a peça.

4.5.2.4 `peca* cria_peca_T ( Tela * tela )`

Cria peça T na tela.

## Parâmetros

<i>tela</i>	Ponteiro para tela de jogo.
-------------	-----------------------------

## Retorna

Ponteiro para a peça.

4.5.2.5 `peca* cria_peca_Z ( Tela * tela )`

Cria uma peça do tipo Z na tela.

## Parâmetros

<i>tela</i>	Ponteiro para a tela de jogo.
-------------	-------------------------------

## Retorna

Ponteiro para a peça.

4.5.2.6 `void libera_peca ( peca * p )`

Libera a memória alocada para a peça.

## Parâmetros

<i>p</i>	Ponteiro para a peça a ser liberada.
----------	--------------------------------------

4.5.2.7 `void move_peca_x ( peca * p, int x )`

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

## Parâmetros

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

#### 4.5.2.8 void move\_pecas\_y ( peca \* p, int y )

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

##### Parâmetros

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

#### 4.5.2.9 void nova\_pecas ( 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

<i>tela</i>	Ponteiro para tela de jogo.
-------------	-----------------------------

<Indica o tipo da peça

#### 4.5.2.10 void rotaciona\_pecas ( peca \* peca )

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

##### Parâmetros

<i>pecas</i>	Ponteiro para a peça.
--------------	-----------------------

#### 4.5.2.11 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.

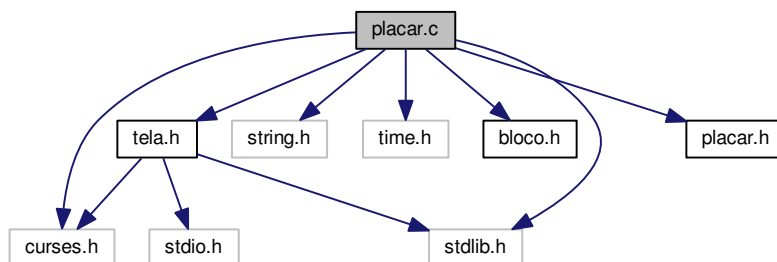
##### Parâmetros

<i>pecas</i>	Ponteiro para peça.
<i>y</i>	Inteiro qualquer.

## 4.6 Referência do Arquivo placar.c

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

<i>pontuacao</i>	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 )

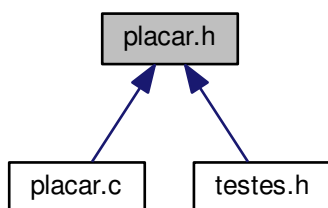
Copia os dados da variável local de escore para a tela de execução.

### Parâmetros

<i>t</i>	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](#)

### Definições de Tipos

- typedef struct [Placar](#) **placar**

### Funções

- void [cria\\_placar](#) ()
- void [atualiza\\_placar](#) (int pontuacao)
- void [mostra\\_placar](#) ()
- void [destroi\\_placar](#) ()
- void [seta\\_jogador](#) ([Tela](#) \*t)

#### 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

<i>pontuacao</i>	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 score para a tela de execução.

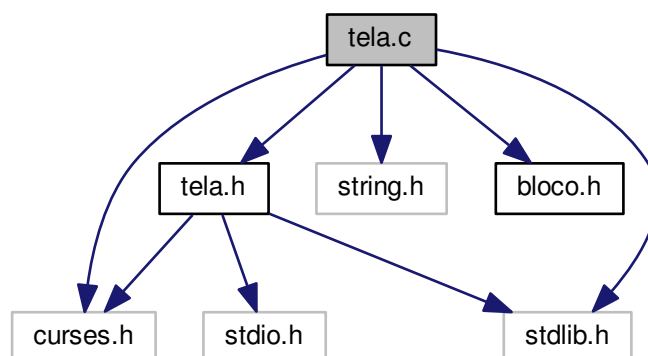
## Parâmetros

<i>t</i>	Ponteiro para tela.
----------	---------------------

## 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

<code>t</code>	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

<code>t</code>	Ponteiro para tela.
----------------	---------------------

#### 4.8.1.4 `void desce_linhas ( Tela * t, int y )`

Desce determinada linha da tela.

## Parâmetros

<i>t</i>	Ponteiro para a tela de jogo.
<i>y</i>	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

<i>t</i>	Ponteiro para a tela de jogo.
----------	-------------------------------

4.8.1.6 void limpa\_linha ( Tela \* *t*, int *y* )

Limpa uma determinada linha do jogo.

## Parâmetros

<i>t</i>	Ponteiro para a tela do jogo.
<i>y</i>	Posição da linha a ser eliminada.

4.8.1.7 void mostra\_pontos ( int *pontos* )

Mostra a pontuação do jogadores.

## Parâmetros

<i>pontos</i>	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

<i>t</i>	Ponteiro para tela a ser mostrada.
----------	------------------------------------

4.8.1.9 void mostra\_tempo ( int *minutos*, int *segundos* )

Mostra o tempo da partida.

## Parâmetros

<i>minutos</i>	Tempo em minutos.
<i>segundos</i>	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

<i>t</i>	Ponteiro para tela.
<i>valor</i>	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

<i>t</i>	Ponteiro para tela.
<i>valor</i>	Quantidade a ser incrementada.

#### 4.8.1.12 int verifica\_linha ( Tela \* *t* )

Verifica se uma linha horizontal do jogo está completamente preenchida.

##### Parâmetros

<i>t</i>	Ponteiro para a tela do jogo.
----------	-------------------------------

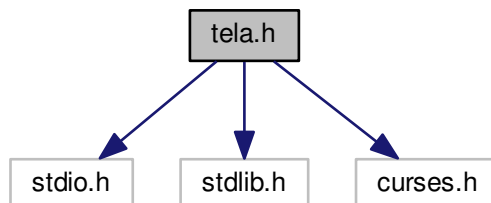
##### Retorna

100 se a linha estiver preenchida. 0 caso contrário.

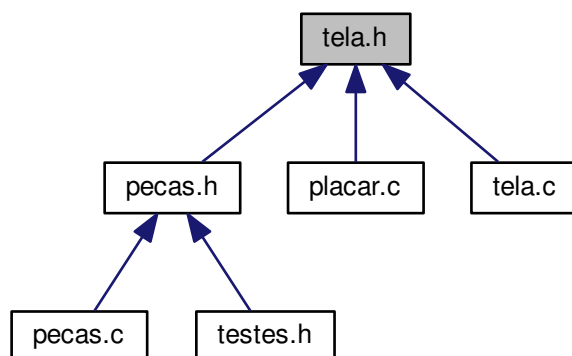
## 4.9 Referência do Arquivo tela.h

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

<i>t</i>	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

<i>t</i>	Ponteiro para tela.
----------	---------------------

##### 4.9.3.4 void desce\_linhas ( Tela \* t, int y )

Desce determinada linha da tela.

###### Parâmetros

<i>t</i>	Ponteiro para a tela de jogo.
<i>y</i>	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**

<i>t</i>	Ponteiro para a tela de jogo.
----------	-------------------------------

**4.9.3.6 void limpa\_linha ( Tela \* t, int y )**

Limpa uma determinada linha do jogo.

**Parâmetros**

<i>t</i>	Ponteiro para a tela do jogo.
<i>y</i>	Posição da linha a ser eliminada.

**4.9.3.7 void mostra\_pontos ( int pontos )**

Mostra a pontuação do jogadores.

**Parâmetros**

<i>pontos</i>	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**

<i>t</i>	Ponteiro para tela a ser mostrada.
----------	------------------------------------

**4.9.3.9 void mostra\_tempo ( int minutos, int segundos )**

Mostra o tempo da partida.

**Parâmetros**

<i>minutos</i>	Tempo em minutos.
<i>segundos</i>	Tempo em segundos.

**4.9.3.10 void muda\_letra ( Tela \* t, int valor )**

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

**Parâmetros**

<i>t</i>	Ponteiro para tela.
<i>valor</i>	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**

<i>t</i>	Ponteiro para tela.
<i>valor</i>	Quantidade a ser incrementada.

**4.9.3.12 int verifica\_linha ( Tela \* *t* )**

Verifica se uma linha horizontal do jogo está completamente preenchida.

**Parâmetros**

<i>t</i>	Ponteiro para a tela do jogo.
----------	-------------------------------

**Retorna**

100 se a linha estiver preenchida. 0 caso contrário.

## Índice Remissivo

abaixo  
    Bloco, 3  
acima  
    Bloco, 3  
anos  
    Placar, 6  
arquivo  
    Placar, 6  
atualiza\_placar  
    placar.c, 21  
    placar.h, 22  
  
Bloco, 3  
    abaixo, 3  
    acima, 3  
    bolinha, 3  
    cor, 3  
    direita, 3  
    esquerda, 3  
    move, 4  
    pos\_x, 4  
    pos\_y, 4  
bloco.h, 9  
blocos  
    Peca, 5  
    tela, 8  
bolinha  
    Bloco, 3  
  
centro\_de\_rotacao  
    Peca, 5  
checa\_fim  
    tela.c, 24  
    tela.h, 28  
comprimento  
    tela, 8  
contador\_jogadores  
    Placar, 6  
cor  
    Bloco, 3  
cor\_nova\_pecas  
    pecas.c, 15  
cor\_pecas  
    Peca, 5  
cria\_pecas\_I  
    pecas.c, 13  
    pecas.h, 17  
cria\_pecas\_L  
    pecas.c, 13  
    pecas.h, 17  
cria\_pecas\_O  
    pecas.c, 13  
    pecas.h, 17  
cria\_pecas\_T  
    pecas.c, 13  
    pecas.h, 19  
cria\_pecas\_Z  
    pecas.c, 14  
    pecas.h, 19  
cria\_placar  
    placar.c, 21  
    placar.h, 23  
cria\_tela  
    tela.c, 24  
    tela.h, 28  
  
define\_jogador  
    tela.c, 24  
    tela.h, 28  
desce\_linhas  
    tela.c, 24  
    tela.h, 28  
destroi\_placar  
    placar.c, 21  
    placar.h, 23  
destroi\_tela  
    tela.c, 25  
    tela.h, 28  
dias  
    Placar, 6  
direita  
    Bloco, 3  
  
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  
estado  
    tela, 8  
    tela.h, 28  
  
finaliza\_ncurses  
    engine.c, 10  
    engine.h, 11  
  
inicia\_ncurses  
    engine.c, 10  
    engine.h, 11  
  
janela  
    tela, 8  
jogador  
    Placar, 6  
    tela, 8  
jogadores



Placar, 6  
 largura  
   tela, 8  
 letra  
   tela, 8  
 libera\_pecas  
   pecas.c, 14  
   pecas.h, 19  
 limpa\_linha  
   tela.c, 25  
   tela.h, 29  
 meses  
   Placar, 6  
 mostra\_placar  
   placar.c, 21  
   placar.h, 23  
 mostra\_pontos  
   tela.c, 25  
   tela.h, 29  
 mostra\_tela  
   tela.c, 25  
   tela.h, 29  
 mostra\_tempo  
   tela.c, 25  
   tela.h, 29  
 move  
   Bloco, 4  
 move\_pecas  
   Peca, 5  
 move\_pecas\_x  
   pecas.c, 14  
   pecas.h, 19  
 move\_pecas\_y  
   pecas.c, 14  
   pecas.h, 20  
 muda\_letra  
   tela.c, 26  
   tela.h, 29  
 nova\_pecas  
   pecas.c, 14  
   pecas.h, 20  
 Peca, 4  
   blocos, 5  
   centro\_de\_rotacao, 5  
   cor\_pecas, 5  
   move\_pecas, 5  
   tamanho, 5  
   tipo, 5  
   velocidade, 5  
 pecas  
   tela, 8  
 pecas.c, 12  
   cor\_nova\_pecas, 15  
   cria\_pecas\_I, 13  
   cria\_pecas\_L, 13  
   cria\_pecas\_O, 13  
   cria\_pecas\_T, 13  
   cria\_pecas\_Z, 14  
   libera\_pecas, 14  
   move\_pecas\_x, 14  
   move\_pecas\_y, 14  
   nova\_pecas, 14  
   rotaciona\_pecas, 15  
   speed\_up, 15  
   speed\_ups, 15  
   pecas.h, 15  
   cria\_pecas\_I, 17  
   cria\_pecas\_L, 17  
   cria\_pecas\_O, 17  
   cria\_pecas\_T, 19  
   cria\_pecas\_Z, 19  
   libera\_pecas, 19  
   move\_pecas\_x, 19  
   move\_pecas\_y, 20  
   nova\_pecas, 20  
   rotaciona\_pecas, 20  
   speed\_up, 20  
   Tipo\_Peca, 17  
 pega\_input  
   engine.c, 10  
   engine.h, 11  
 Placar, 5  
   anos, 6  
   arquivo, 6  
   contador\_jogadores, 6  
   dias, 6  
   jogador, 6  
   jogadores, 6  
   meses, 6  
   pontuacao, 6  
   pontuacoes, 6  
   tempo\_m, 6  
   tempo\_s, 6  
   tempos\_m, 6  
   tempos\_s, 6  
   placar.c, 20  
     atualiza\_placar, 21  
     cria\_placar, 21  
     destroi\_placar, 21  
     mostra\_placar, 21  
     seta\_jogador, 21  
   placar.h, 22  
     atualiza\_placar, 22  
     cria\_placar, 23  
     destroi\_placar, 23  
     mostra\_placar, 23  
     seta\_jogador, 23  
   pontos  
     tela, 8  
   pontuacao  
     Placar, 6  
   pontuacoes  
     Placar, 6

pos\_x  
  Bloco, 4  
pos\_y  
  Bloco, 4  
rotaciona\_pecas  
  pecas.c, 15  
  pecas.h, 20  
seta\_jogador  
  placar.c, 21  
  placar.h, 23  
speed\_up  
  pecas.c, 15  
  pecas.h, 20  
speed\_ups  
  pecas.c, 15  
tamanho  
  Peca, 5  
Tela  
  tela.h, 27  
tela, 7  
  blocos, 8  
  comprimento, 8  
  estado, 8  
  janela, 8  
  jogador, 8  
  largura, 8  
  letra, 8  
  peca, 8  
  pontos, 8  
  tempo\_m, 8  
  tempo\_s, 8  
tela.c, 23  
  checa\_fim, 24  
  cria\_tela, 24  
  define\_jogador, 24  
  desce\_linhas, 24  
  destroi\_tela, 25  
  limpa\_linha, 25  
  mostra\_pontos, 25  
  mostra\_tela, 25  
  mostra\_tempo, 25  
  muda\_letra, 26  
  troca\_letra, 26  
  verifica\_linha, 26  
tela.h, 26  
  checa\_fim, 28  
  cria\_tela, 28  
  define\_jogador, 28  
  desce\_linhas, 28  
  destroi\_tela, 28  
  estado, 28  
  limpa\_linha, 29  
  mostra\_pontos, 29  
  mostra\_tela, 29  
  mostra\_tempo, 29  
  muda\_letra, 29  
  Tela, 27  
  troca\_letra, 30  
  verifica\_linha, 30  
tempo\_m  
  Placar, 6  
  tela, 8  
tempo\_s  
  Placar, 6  
  tela, 8  
tempos\_m  
  Placar, 6  
tempos\_s  
  Placar, 6  
timeb, 9  
tipo  
  Peca, 5  
Tipo\_Peca  
  pecas.h, 17  
troca\_letra  
  tela.c, 26  
  tela.h, 30  
velocidade  
  Peca, 5  
verifica\_linha  
  tela.c, 26  
  tela.h, 30