

## My Project

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	TipoJogador Struct Reference . . . . .	5
3.2	TipoPeca Struct Reference . . . . .	5
3.3	TipoTela Struct Reference . . . . .	6
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	engine.c File Reference . . . . .	7
4.1.1	Detailed Description . . . . .	8
4.1.2	Enumeration Type Documentation . . . . .	9
4.1.2.1	comandos . . . . .	9
4.1.3	Function Documentation . . . . .	9
4.1.3.1	AddBloco(TipoPeca *peca, TipoTela tela[][TAMANHOTELAX]) . . . . .	9
4.1.3.2	DeletaBloco(TipoTela *unidade) . . . . .	9
4.1.3.3	DeletaLinhas(TipoTela tela[][TAMANHOTELAX], int *pontuacao) . . . . .	9
4.1.3.4	Gravidade(TipoTela tela[][TAMANHOTELAX], int posicaolinha) . . . . .	9
4.1.3.5	Loop(TipoTela tela[][TAMANHOTELAX]) . . . . .	10
4.1.3.6	PegaInput() . . . . .	10
4.1.3.7	PoePecaNoTopo(TipoPeca *peca, TipoTela tela[][TAMANHOTELAX]) . . . . .	10
4.1.3.8	RemoveBloco(TipoPeca *peca, TipoTela tela[][TAMANHOTELAX]) . . . . .	10

4.1.3.9	Temporizador(int milissegundos, TipoPeca *peca, int *flagDesce)	11
4.1.3.10	VerificaColisao(TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])	11
4.1.3.11	VerificaLinhas(TipoTela tela[][TAMANHOTELAX])	11
4.1.3.12	VerificaMorte(TipoTela tela[][TAMANHOTELAX])	11
4.2	engine.h File Reference	12
4.2.1	Detailed Description	13
4.2.2	Function Documentation	13
4.2.2.1	AddBloco(TipoPeca *, TipoTela[][TAMANHOTELAX])	13
4.2.2.2	DeletaBloco(TipoTela *)	14
4.2.2.3	DeletaLinhas(TipoTela[][TAMANHOTELAX], int *)	14
4.2.2.4	Gravidade(TipoTela[][TAMANHOTELAX], int)	14
4.2.2.5	Loop(TipoTela[][TAMANHOTELAX])	14
4.2.2.6	PegaInput()	14
4.2.2.7	PoePecaNoTopo(TipoPeca *, TipoTela[][TAMANHOTELAX])	15
4.2.2.8	RemoveBloco(TipoPeca *, TipoTela[][TAMANHOTELAX])	15
4.2.2.9	VerificaColisao(TipoPeca *, TipoTela[][TAMANHOTELAX])	15
4.2.2.10	VerificaLinhas(TipoTela[][TAMANHOTELAX])	15
4.2.2.11	VerificaMorte(TipoTela[][TAMANHOTELAX])	16
4.3	main.c File Reference	16
4.3.1	Detailed Description	17
4.4	pecas.c File Reference	17
4.4.1	Detailed Description	18
4.4.2	Function Documentation	18
4.4.2.1	AlocaPeca()	18
4.4.2.2	CopiaDaMatrizDePecas(char MatrizFonte[][5], TipoTela MatrizAlvo[][5])	18
4.4.2.3	CopiaPeca(TipoPeca *pecaIn, TipoPeca *pecaOut)	19
4.4.2.4	GeraPeca(TipoPeca *ponteiroPeca)	19
4.4.2.5	GeraPecaEspecific(TipoPeca *ponteiroPeca, int indicePeca, int direcao)	19
4.4.2.6	MovePecaX(TipoPeca *targetPeca, int inputX)	19
4.4.2.7	MovePecaY(TipoPeca *targetPeca, int inputY)	20

4.4.2.8	PecaGetBloco(TipoPeca *peca, int y, int x)	20
4.4.2.9	PecaGetCor(TipoPeca *peca)	20
4.4.2.10	PecaGetSpeed(TipoPeca *peca)	20
4.4.2.11	PecaGetX(TipoPeca *peca)	21
4.4.2.12	PecaGetY(TipoPeca *peca)	21
4.4.2.13	RotacionaPeca(TipoPeca *peca)	21
4.4.2.14	SpeedUp(TipoPeca *peca, int multiplicador)	21
4.5	pecas.h File Reference	22
4.5.1	Detailed Description	23
4.5.2	Function Documentation	23
4.5.2.1	AlocaPeca()	23
4.5.2.2	CopiaPeca(TipoPeca *, TipoPeca *)	23
4.5.2.3	GeraPeca(TipoPeca *)	24
4.5.2.4	GeraPecaEspecifica(TipoPeca *, int, int)	24
4.5.2.5	MovePecaX(TipoPeca *, int)	24
4.5.2.6	MovePecaY(TipoPeca *, int)	24
4.5.2.7	PecaGetBloco(TipoPeca *, int, int)	25
4.5.2.8	PecaGetCor(TipoPeca *)	25
4.5.2.9	PecaGetSpeed(TipoPeca *)	25
4.5.2.10	PecaGetX(TipoPeca *)	25
4.5.2.11	PecaGetY(TipoPeca *)	26
4.5.2.12	RotacionaPeca(TipoPeca *)	26
4.5.2.13	SpeedUp(TipoPeca *, int)	26
4.6	pecas_inst.c File Reference	26
4.6.1	Detailed Description	27
4.7	pecas_inst.h File Reference	28
4.7.1	Detailed Description	29
4.8	ranking.c File Reference	29
4.8.1	Detailed Description	30
4.8.2	Function Documentation	30

4.8.2.1	AtualizaPlacar(TipoJogador jogador)	30
4.8.2.2	ConverterApelido(int *apelido, TipoJogador *jogador)	30
4.8.2.3	CriaPlacar(TipoJogador jogador)	30
4.8.2.4	ReceberApelido()	31
4.8.2.5	ReceberData(TipoJogador *jogador)	31
4.8.2.6	VerificaPlacar()	31
4.9	ranking.h File Reference	31
4.9.1	Detailed Description	32
4.9.2	Function Documentation	32
4.9.2.1	AtualizaPlacar(TipoJogador)	32
4.9.2.2	ConverterApelido(int *, TipoJogador *)	33
4.9.2.3	CriaPlacar(TipoJogador)	33
4.9.2.4	ReceberApelido()	33
4.9.2.5	ReceberData(TipoJogador *)	33
4.9.2.6	VerificaPlacar()	33
4.10	tela.c File Reference	34
4.10.1	Detailed Description	35
4.10.2	Function Documentation	35
4.10.2.1	CriarTela(TipoTela tela[][25])	35
4.10.2.2	MostrarTela(TipoTela tela[][25], int pontos, int tempoDecorrido)	35
4.10.2.3	SetPeca(TipoTela *pecaDada, char caractereAferido, int corAferida)	35
4.10.2.4	SetPecaCaractere(TipoTela *pecaDada, char caractereAferido)	36
4.10.2.5	SetPecaCor(TipoTela *pecaDada, int corAferida)	36
4.10.2.6	VerificaSeBloco(TipoTela pecaDada)	36
4.11	tela.h File Reference	36
4.11.1	Detailed Description	38
4.11.2	Function Documentation	38
4.11.2.1	CriarTela(TipoTela[][25])	38
4.11.2.2	MostrarTela(TipoTela[][25], int, int)	38
4.11.2.3	SetPeca(TipoTela *, char, int)	38
4.11.2.4	SetPecaCaractere(TipoTela *, char)	39
4.11.2.5	SetPecaCor(TipoTela *, int)	39
4.11.2.6	VerificaSeBloco(TipoTela)	39
4.12	teste.c File Reference	39
4.12.1	Detailed Description	40
4.12.2	Function Documentation	41
4.12.2.1	TestaColisaoCaixaAberta(void)	41
4.12.2.2	TestaMorte(void)	41
4.12.2.3	TesteAtualizaPlacar(void)	41
4.12.2.4	TesteConverteVetor(void)	41
4.12.2.5	TesteCorPecaDiferente(void)	41
4.12.2.6	TesteData(void)	41
4.12.2.7	TesteEntradaPlacar(void)	41







# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">TipoJogador</a>	5
<a href="#">TipoPeca</a>	5
<a href="#">TipoTela</a>	6



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">engine.c</a>	7
<a href="#">engine.h</a>	12
<a href="#">main.c</a>	16
<a href="#">pecas.c</a>	17
<a href="#">pecas.h</a>	22
<a href="#">pecas_inst.c</a>	26
<a href="#">pecas_inst.h</a>	28
<a href="#">ranking.c</a>	29
<a href="#">ranking.h</a>	31
<a href="#">tela.c</a>	34
<a href="#">tela.h</a>	36
<a href="#">teste.c</a>	39



## Chapter 3

# Class Documentation

### 3.1 TipoJogador Struct Reference

#### Public Attributes

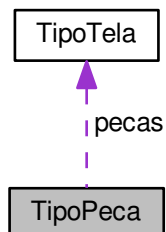
- char **apelido** [7]
- int **pontos**
- int **dia**
- int **mes**
- int **ano**
- int **tempo**

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

- [ranking.h](#)

### 3.2 TipoPeca Struct Reference

Collaboration diagram for TipoPeca:



### Public Attributes

- [TipoTela](#) **pecas** [TAMANHO\_Y][TAMANHO\_X]
- int **x**
- int **y**
- int **speed**

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

- [pecas.h](#)

## 3.3 TipoTela Struct Reference

### Public Attributes

- char **peca**
- int **cor**

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

- [tela.h](#)

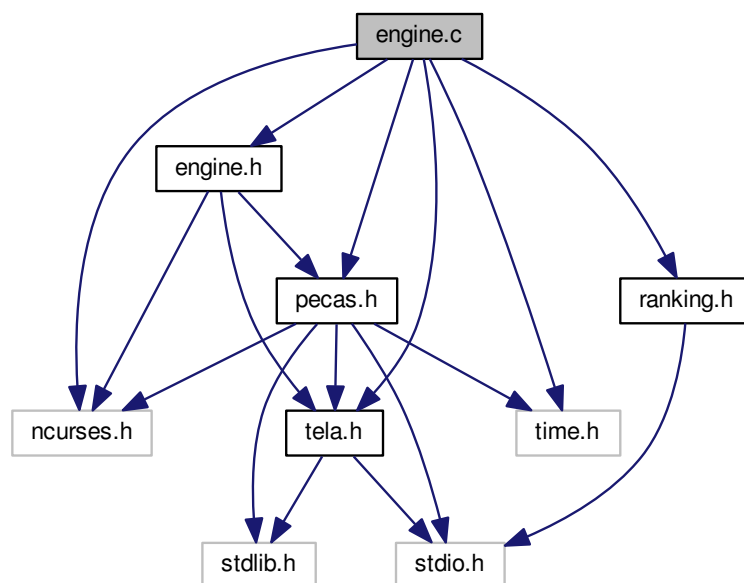
## Chapter 4

# File Documentation

### 4.1 engine.c File Reference

```
#include <ncurses.h>
#include "tela.h"
#include "pecas.h"
#include "time.h"
#include "ranking.h"
#include "engine.h"
```

Include dependency graph for engine.c:



## Macros

- `#define TAMANHOTELAY 15`
- `#define TAMANHOTELAX 25`
- `#define CORVERMELHO 4`
- `#define CORFUNDO 7`
- `#define BLOCO '0'`
- `#define VAZIO '.'`
- `#define TEST_MODE`

## Enumerations

- enum `comandos` {  
    **ESQUERDA**, **BAIXO**, **DIREITA**, **CIMA**,  
    **SAIR** }

## Functions

- int `VerificaMorte` (`TipoTela` tela[ ][TAMANHOTELAX])
- int `VerificaColisao` (`TipoPeca` \*peca, `TipoTela` tela[ ][TAMANHOTELAX])
- void `PoePecaNoTopo` (`TipoPeca` \*peca, `TipoTela` tela[ ][TAMANHOTELAX])
- void `DeletaBloco` (`TipoTela` \*unidade)
- void `AddBloco` (`TipoPeca` \*peca, `TipoTela` tela[ ][TAMANHOTELAX])
- void `RemoveBloco` (`TipoPeca` \*peca, `TipoTela` tela[ ][TAMANHOTELAX])
- int `VerificaLinhas` (`TipoTela` tela[ ][TAMANHOTELAX])
- void `Gravidade` (`TipoTela` tela[ ][TAMANHOTELAX], int posicaoolinha)
- void `DeletaLinhas` (`TipoTela` tela[ ][TAMANHOTELAX], int \*pontuacao)
- int `PegaInput` ()
- int `Temporizador` (int milissegundos, `TipoPeca` \*peca, int \*flagDesce)
- int `Loop` (`TipoTela` tela[ ][TAMANHOTELAX])

## Variables

- enum `comandos` **com**

### 4.1.1 Detailed Description

Gerencia o funcionamento completo do jogo. Este módulo implementa o funcionamento mecânico do jogo e agrega os demais conceitos implementados pelos outros módulos.

#### Author

Cristóvão

#### Since

11/04/16

#### Version

2.8



## 4.1.2 Enumeration Type Documentation

### 4.1.2.1 enum comandos

Comandos. Enumeração dos comandos que podem ser inseridos.

## 4.1.3 Function Documentation

### 4.1.3.1 void AddBloco ( TipoPeca \* *peca*, TipoTela *tela*[[TAMANHOTELAX]] )

Função que insere um bloco na matriz da tela.

#### Parameters

* <i>peca</i>	ponteiro para a peça atual e sua posição na tela.
<i>tela</i> [[TAMANHOTEL↵AX]]	uma matriz que representa a tela do jogo.

### 4.1.3.2 void DeletaBloco ( TipoTela \* *unidade* )

Função que limpa um campo da matriz da tela.

#### Parameters

* <i>unidade</i>	uma posição na tela que deve ser excluída.
------------------	--

### 4.1.3.3 void DeletaLinhas ( TipoTela *tela*[[TAMANHOTELAX]], int \* *pontuacao* )

Função que deleta uma linha completa na matriz da tela. Incrementa a pontuação do jogador em 100 pontos.

#### Parameters

<i>tela</i> [[TAMANHOTEL↵AX]]	uma matriz que representa a tela do jogo.
* <i>pontuacao</i>	um ponteiro para a pontuação atual do jogador.

### 4.1.3.4 void Gravidade ( TipoTela *tela*[[TAMANHOTELAX]], int *posicaolinha* )

Função que decrementa a posição de uma linha da tela.

#### Parameters

<i>tela</i> [[TAMANHOTEL↵AX]]	uma matriz que representa a tela do jogo.
<i>posicaolinha</i>	inteiro que representa a posição de uma linha.

#### 4.1.3.5 int Loop ( TipoTela tela[ ][TAMANHOTELAX] )

Função que gerencia todas as funções do jogo. Mantém o um loop que deve ser percorrido a cada ação realizada, bem como a interpretação dos inputs e as decisões que devem ser tomadas pela engine.

##### Parameters

<i>tela[ ][TAMANHOTELAX]</i>	uma matriz que representa a tela do jogo.
------------------------------	---

##### Returns

0 caso não aconteça nenhum erro.

#### 4.1.3.6 int PegalInput ( )

Função que recebe o comando o usuário. Atribui determinados comandos de acordo com as teclas pressionadas.

##### Returns

o comando do usuário.

#### 4.1.3.7 void PoePecaNoTopo ( TipoPeca \* peca, TipoTela tela[ ][TAMANHOTELAX] )

Função que insere uma peça na matriz da tela. Insere na parte superior da tela a peça da rodada atual do jogo.

##### Parameters

<i>*peca</i>	ponteiro para a peça atual.
<i>tela[ ][TAMANHOTELAX]</i>	uma matriz que representa a tela do jogo.

#### 4.1.3.8 void RemoveBloco ( TipoPeca \* peca, TipoTela tela[ ][TAMANHOTELAX] )

Função que remove um bloco da matriz da tela.

##### Parameters

<i>*peca</i>	ponteiro para a peça atual e sua posição na tela.
<i>tela[ ][TAMANHOTELAX]</i>	uma matriz que representa a tela do jogo.

#### 4.1.3.9 int Temporizador ( int *milissegundos*, TipoPeca \* *peca*, int \* *flagDesce* )

Função que coordena o tempo de jogo. Mantém um loop que incrementa o tempo conforme ele passa. O loop é interrompido caso alguma tecla válida seja pressionada, é chegada a hora de uma peça descer ou quando o contador deve ser atualizado na tela.

##### Parameters

<i>milissegundos</i>	um inteiro que representa um segundo.
* <i>peca</i>	ponteiro para a peça atual do jogo.
* <i>flagdesce</i>	flag que indica se a peça deve descer na tela.

##### Returns

o comando inserido pelo usuário.

#### 4.1.3.10 int VerificaColisao ( TipoPeca \* *peca*, TipoTela *tela*[ ][TAMANHOTELAX] )

Função que verifica a colisão de blocos. Verifica se uma peça colidiu com algum bloco presente na tela.

##### Parameters

* <i>peca</i>	ponteiro para a peça atual e sua posição na tela.
<i>tela</i> [ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.

##### Returns

um inteiro com valor 0 ou 1 que indica se a peça colidiu.

#### 4.1.3.11 int VerificaLinhas ( TipoTela *tela*[ ][TAMANHOTELAX] )

Função que verifica se há uma linha completa.

##### Parameters

<i>tela</i> [ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.
--------------------------------	---

##### Returns

a linha que está completa ou -1 caso não exista uma linha completa.

#### 4.1.3.12 int VerificaMorte ( TipoTela *tela*[ ][TAMANHOTELAX] )

Função que verifica se há algum bloco na linha limítrofe. Verifica se algum bloco atingiu o limite do jogo, ou seja, a posição 5.

## Parameters

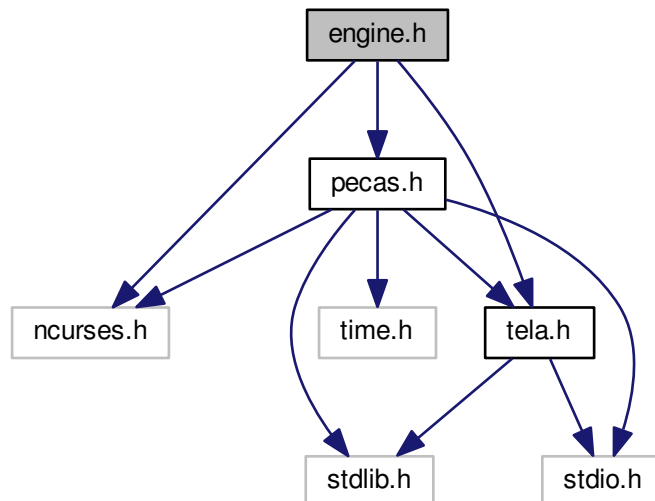
<code>tela[ ][TAMANHOTEL↔AX]</code>	uma matriz que representa a tela do jogo.
-------------------------------------	---

## Returns

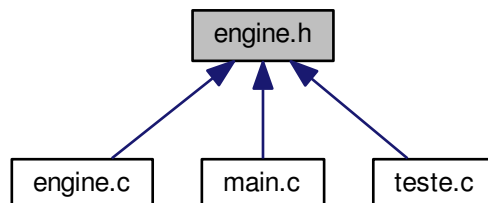
um inteiro com valor 0 ou 1 que indica se a linha de limite foi atingida.

## 4.2 engine.h File Reference

```
#include <ncurses.h>
#include "tela.h"
#include "pecas.h"
Include dependency graph for engine.h:
```



This graph shows which files directly or indirectly include this file:



## Macros

- #define **TAMANHOTELAY** 15
- #define **TAMANHOTELAX** 25

## Functions

- int [VerificaMorte](#) ([TipoTela](#)[][TAMANHOTELAX])
- int [VerificaColisao](#) ([TipoPeca](#) \*, [TipoTela](#)[][TAMANHOTELAX])
- void [PoePecaNoTopo](#) ([TipoPeca](#) \*, [TipoTela](#)[][TAMANHOTELAX])
- void [DeletaBloco](#) ([TipoTela](#) \*)
- void [AddBloco](#) ([TipoPeca](#) \*, [TipoTela](#)[][TAMANHOTELAX])
- void [RemoveBloco](#) ([TipoPeca](#) \*, [TipoTela](#)[][TAMANHOTELAX])
- int [VerificaLinhas](#) ([TipoTela](#)[][TAMANHOTELAX])
- void [Gravidade](#) ([TipoTela](#)[][TAMANHOTELAX], int)
- void [DeletaLinhas](#) ([TipoTela](#)[][TAMANHOTELAX], int \*)
- int [PegaInput](#) ()
- int [Loop](#) ([TipoTela](#)[][TAMANHOTELAX])

### 4.2.1 Detailed Description

Contém a definição do módulo engine. Define os protótipos das funções do módulo.

#### Author

Cristóvão

#### Since

11/04/16

#### Version

2.8

### 4.2.2 Function Documentation

#### 4.2.2.1 void AddBloco ( [TipoPeca](#) \* *peca*, [TipoTela](#) *tela*[][TAMANHOTELAX] )

Função que insere um bloco na matriz da tela.

#### Parameters

<i>*peca</i>	ponteiro para a peça atual e sua posição na tela.
<i>tela</i> [][TAMANHOTELAX]	uma matriz que representa a tela do jogo.

#### 4.2.2.2 void DeletaBloco ( TipoTela \* unidade )

Função que limpa um campo da matriz da tela.

##### Parameters

*unidade	uma posição na tela que deve ser excluída.
----------	--

#### 4.2.2.3 void DeletaLinhas ( TipoTela tela[ ][TAMANHOTELAX], int \* pontuacao )

Função que deleta uma linha completa na matriz da tela. Incrementa a pontuação do jogador em 100 pontos.

##### Parameters

tela[ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.
*pontuacao	um ponteiro para a pontuação atual do jogador.

#### 4.2.2.4 void Gravidade ( TipoTela tela[ ][TAMANHOTELAX], int posicaolinha )

Função que decrementa a posição de uma linha da tela.

##### Parameters

tela[ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.
posicaolinha	inteiro que representa a posição de uma linha.

#### 4.2.2.5 int Loop ( TipoTela tela[ ][TAMANHOTELAX] )

Função que gerencia todas as funções do jogo. Mantém o um loop que deve ser percorrido a cada ação realizada, bem como a interpretação dos inputs e as decisões que devem ser tomadas pela engine.

##### Parameters

tela[ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.
------------------------	---

##### Returns

0 caso não aconteça nenhum erro.

#### 4.2.2.6 int PegalInput ( )

Função que recebe o comando o usuário. Atribui determinados comandos de acordo com as teclas pressionadas.

**Returns**

o comando do usuário.

**4.2.2.7 void PoePecaNoTopo ( TipoPeca \* *peca*, TipoTela *tela*[ ][TAMANHOTELAX] )**

Função que insere uma peça na matriz da tela. Insere na parte superior da tela a peça da rodada atual do jogo.

**Parameters**

<i>*peca</i>	ponteiro para a peça atual.
<i>tela</i> [ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.

**4.2.2.8 void RemoveBloco ( TipoPeca \* *peca*, TipoTela *tela*[ ][TAMANHOTELAX] )**

Função que remove um bloco da matriz da tela.

**Parameters**

<i>*peca</i>	ponteiro para a peça atual e sua posição na tela.
<i>tela</i> [ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.

**4.2.2.9 int VerificaColisao ( TipoPeca \* *peca*, TipoTela *tela*[ ][TAMANHOTELAX] )**

Função que verifica a colisão de blocos. Verifica se uma peça colidiu com algum bloco presente na tela.

**Parameters**

<i>*peca</i>	ponteiro para a peça atual e sua posição na tela.
<i>tela</i> [ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.

**Returns**

um inteiro com valor 0 ou 1 que indica se a peça colidiu.

**4.2.2.10 int VerificaLinhas ( TipoTela *tela*[ ][TAMANHOTELAX] )**

Função que verifica se há uma linha completa.

**Parameters**

<i>tela</i> [ ][TAMANHOTEL↔AX]	uma matriz que representa a tela do jogo.
--------------------------------	---

**Returns**

a linha que está completa ou -1 caso não exista uma linha completa.

#### 4.2.2.11 int VerificaMorte ( TipoTela tela[ ][TAMANHOTELAX] )

Função que verifica se há algum bloco na linha limítrofe. Verifica se algum bloco atingiu o limite do jogo, ou seja, a posição 5.

**Parameters**

<i>tela</i> [ ][TAMANHOTEL↔ AX]	uma matriz que representa a tela do jogo.
------------------------------------	---

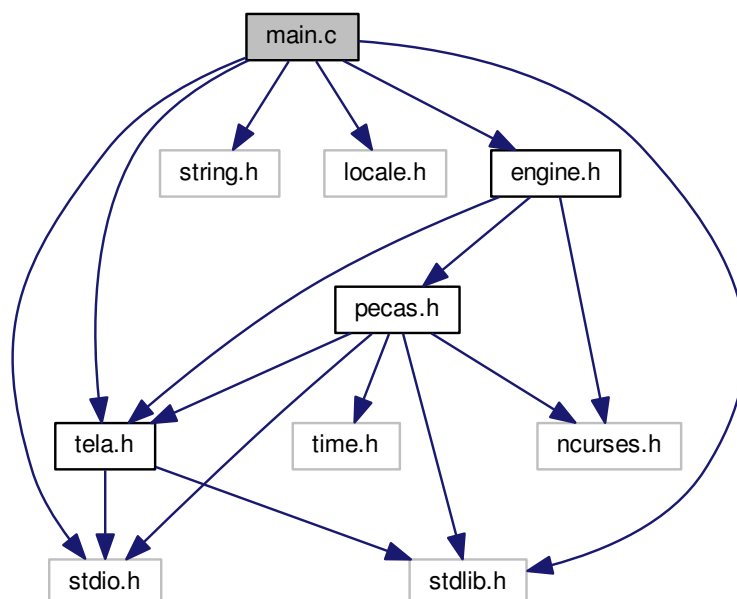
**Returns**

um inteiro com valor 0 ou 1 que indica se a linha de limite foi atingida.

## 4.3 main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>
#include "tela.h"
#include "engine.h"
```

Include dependency graph for main.c:





## Functions

- int **main** ()

### 4.3.1 Detailed Description

Gerencia a criação e manipulação da tela. Este módulo implementa a tela do jogo, bem como gerencia as operações relacionadas à ela.

#### Author

Anderson, André, Cristóvão, Pedro Vítor.

#### Since

11/04/16

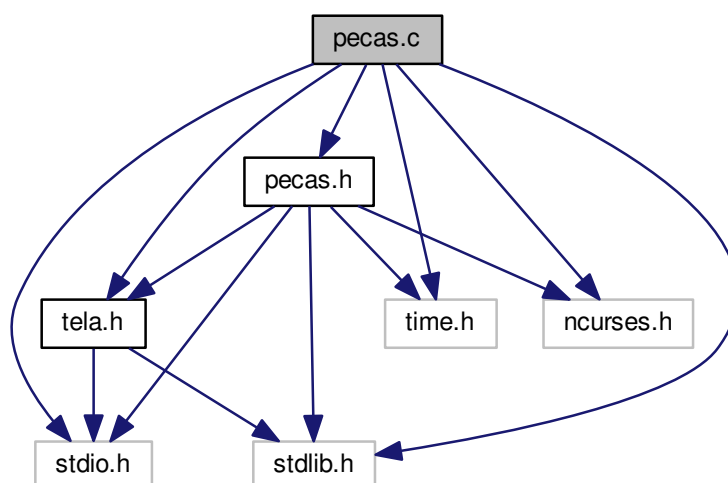
#### Version

2.0

## 4.4 pecas.c File Reference

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

Include dependency graph for pecas.c:



## Macros

- `#define PECAS_OWN`
- `#define TAMANHO_X 5`
- `#define TAMANHO_Y 5`
- `#define COR_FUNDO 7`
- `#define B '0'`
- `#define V '.'`
- `#define NUM_PECAS 8`

## Functions

- void [SpeedUp](#) ([TipoPeca](#) \*peca, int multiplicador)
  - void [RotacionaPeca](#) ([TipoPeca](#) \*peca)
  - void [CopiaDaMatrizDePecas](#) (char MatrizFonte[ ][5], [TipoTela](#) MatrizAlvo[ ][5])
  - void [GeraPeca](#) ([TipoPeca](#) \*ponteiroPeca)
  - void [GeraPecaEspecifica](#) ([TipoPeca](#) \*ponteiroPeca, int indicePeca, int direcao)
  - [TipoTela](#) [PecaGetBloco](#) ([TipoPeca](#) \*peca, int y, int x)
  - int [PecaGetCor](#) ([TipoPeca](#) \*peca)
  - void [CopiaPeca](#) ([TipoPeca](#) \*pecaIn, [TipoPeca](#) \*pecaOut)
  - int [PecaGetX](#) ([TipoPeca](#) \*peca)
  - int [PecaGetY](#) ([TipoPeca](#) \*peca)
  - int [PecaGetSpeed](#) ([TipoPeca](#) \*peca)
  - [TipoPeca](#) \* [AlocaPeca](#) ()
  - void [LiberaPeca](#) ([TipoPeca](#) \*peca)
- Função que desaloca a memória de uma peça.*
- void [MovePecaX](#) ([TipoPeca](#) \*targetPeca, int inputX)
  - void [MovePecaY](#) ([TipoPeca](#) \*targetPeca, int inputY)

### 4.4.1 Detailed Description

Gerencia a criação e manipulação de peças no jogo. Este módulo implementa as peças do jogo, bem como todas as operações relacionadas à elas.

#### Author

Anderson

#### Since

11/04/16

#### Version

2.6

### 4.4.2 Function Documentation

#### 4.4.2.1 [TipoPeca\\*](#) [AlocaPeca](#) ( )

Função que aloca memória para uma peça.

#### Returns

a nova peça.

#### 4.4.2.2 void [CopiaDaMatrizDePecas](#) ( char *MatrizFonte*[ ][5], [TipoTela](#) *MatrizAlvo*[ ][5] )

Função que copia uma peça de uma matriz à outra.

## Parameters

<i>MatrizFonte</i> [][5]	matriz que contém a peça de origem.
<i>MatrizAlvo</i> [][5]	matriz que receberá a peça.

**4.4.2.3 void CopiaPeca ( TipoPeca \* *pecaIn*, TipoPeca \* *pecaOut* )**

Função que copia uma peça.

## Parameters

* <i>pecaIn</i>	peça que será copiada;
* <i>pecaOut</i>	cópia da peça.

**4.4.2.4 void GeraPeca ( TipoPeca \* *ponteiroPeca* )**

Função que gera uma peça aleatoriamente.

## Parameters

* <i>ponteiropeça</i>	ponteiro para uma peça.
-----------------------	-------------------------

**4.4.2.5 void GeraPecaEspecific ( TipoPeca \* *ponteiroPeca*, int *indicePeca*, int *direcao* )**

Função que gera uma peça específica. Gera uma peça não linear, que varia entre modelos específicos do jogo.

## Parameters

* <i>ponteiroPeca</i>	ponteiro para uma peça.
<i>indicePeca</i>	inteiro que indica qual peça específica deve ser gerada.
<i>direcao</i>	inteiro que define a direção da peça a ser gerada.

**4.4.2.6 void MovePecaX ( TipoPeca \* *targetPeca*, int *inputX* )**

Função que move uma peça verticalmente.

## Parameters

* <i>targetPeca</i>	ponteiro para peça.
<i>inputX</i>	nova posição x da peça.

**4.4.2.7 void MovePecaY ( TipoPeca \* *targetPeca*, int *inputY* )**

Função que move uma peça horizontalmente.

**Parameters**

<i>*targetPeca</i>	ponteiro para peça.
<i>inputY</i>	nova posição y da peça.

**4.4.2.8 TipoTela PecaGetBloco ( TipoPeca \* *peca*, int *y*, int *x* )**

Função que retorna uma peça e sua posição.

**Parameters**

<i>*peça</i>	ponteiro para peça.
<i>y</i>	inteiro que indica a posição horizontal da peça.
<i>x</i>	inteiro que indica a posição vertical da peça.

**Returns**

a matriz da peça.

**4.4.2.9 int PecaGetCor ( TipoPeca \* *peca* )**

Função que retorna a cor de uma peça.

**Parameters**

<i>*peca</i>	ponteiro para peça.
--------------	---------------------

**Returns**

um inteiro que indica a cor da peça.

**4.4.2.10 int PecaGetSpeed ( TipoPeca \* *peca* )**

Função que retorna a velocidade de queda da peça.

**Parameters**

<i>*peca</i>	ponteiro para peça.
--------------	---------------------

**Returns**

um inteiro com a velocidade da peça.

**4.4.2.11 int PecaGetX ( TipoPeca \* *peca* )**

Função que retorna a posição vertical da peça

**Parameters**

<i>*peca</i>	ponteiro para peça.
--------------	---------------------

**Returns**

um inteiro com a posição vertical da peça.

**4.4.2.12 int PecaGetY ( TipoPeca \* *peca* )**

Função que retorna a posição horizontal da peça

**Parameters**

<i>*peca</i>	ponteiro para peça.
--------------	---------------------

**Returns**

um inteiro com a posição horizontal da peça.

**4.4.2.13 void RotacionaPeca ( TipoPeca \* *peca* )**

Função que rotaciona a peça.

**Parameters**

<i>*peça</i>	ponteiro para a peça atual.
--------------	-----------------------------

**4.4.2.14 void SpeedUp ( TipoPeca \* *peca*, int *multiplicador* )**

Função que incrementa a velocidade de queda da peça.

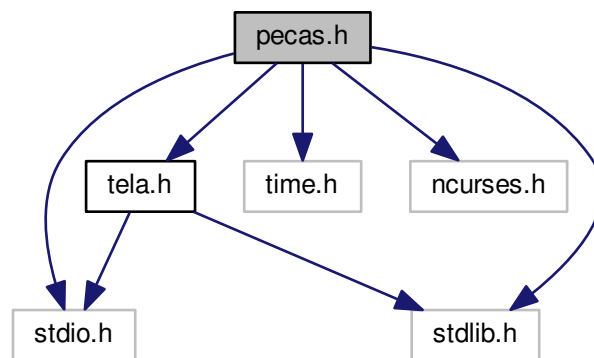
**Parameters**

<i>*peça</i>	ponteiro para a peça atual.
<i>multiplicador</i>	inteiro que representa o quanto a velocidade de queda da peça será incrementada.

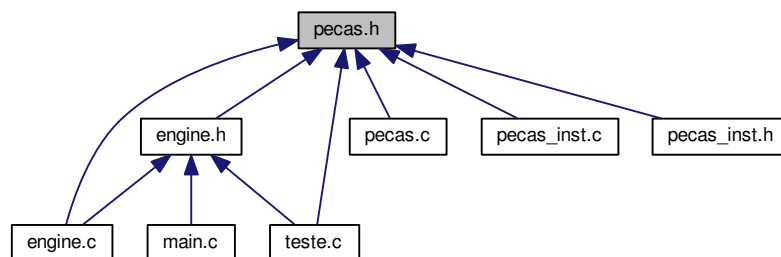
## 4.5 pecas.h File Reference

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

Include dependency graph for pecas.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [TipoPeca](#)

## Macros

- `#define TAMANHO_X 5`
- `#define TAMANHO_Y 5`

## Functions

- [TipoPeca \\*](#) [AlocaPeca](#) ( )
- void [LiberaPeca](#) ([TipoPeca \\*](#))  
*Função que desaloca a memória de uma peça.*
- void [CopiaPeca](#) ([TipoPeca \\*](#), [TipoPeca \\*](#))
- void [GeraPeca](#) ([TipoPeca \\*](#))
- void [GeraPecaEspecificica](#) ([TipoPeca \\*](#), int, int)
- void [MovePecaX](#) ([TipoPeca \\*](#), int)
- void [MovePecaY](#) ([TipoPeca \\*](#), int)
- void [RotacionaPeca](#) ([TipoPeca \\*](#))
- void [SpeedUp](#) ([TipoPeca \\*](#), int)
- int [PecaGetCor](#) ([TipoPeca \\*](#))
- int [PecaGetX](#) ([TipoPeca \\*](#))
- int [PecaGetY](#) ([TipoPeca \\*](#))
- int [PecaGetSpeed](#) ([TipoPeca \\*](#))
- [TipoTela](#) [PecaGetBloco](#) ([TipoPeca \\*](#), int, int)

### 4.5.1 Detailed Description

Faz a definição do módulo de peças. Define as estruturas do módulo e os protótipos de funções.

#### Author

Anderson

#### Since

11/04/16

#### Version

2.6

### 4.5.2 Function Documentation

#### 4.5.2.1 [TipoPeca\\*](#) [AlocaPeca](#) ( )

Função que aloca memória para uma peça.

#### Returns

a nova peça.

#### 4.5.2.2 void [CopiaPeca](#) ( [TipoPeca \\*](#) *pecaIn*, [TipoPeca \\*](#) *pecaOut* )

Função que copia uma peça.

## Parameters

<i>*pecaIn</i>	peça que será copiada;
<i>*pecaOut</i>	cópia da peça.

**4.5.2.3 void GeraPeca ( TipoPeca \* *ponteiroPeca* )**

Função que gera uma peça aleatoriamente.

## Parameters

<i>*ponteiropeça</i>	ponteiro para uma peça.
----------------------	-------------------------

**4.5.2.4 void GeraPecaEspecific ( TipoPeca \* *ponteiroPeca*, int *indicePeca*, int *direcao* )**

Função que gera uma peça específica. Gera uma peça não linear, que varia entre modelos específicos do jogo.

## Parameters

<i>*ponteiroPeca</i>	ponteiro para uma peça.
<i>indicePeca</i>	inteiro que indica qual peça específica deve ser gerada.
<i>direcao</i>	inteiro que define a direção da peça a ser gerada.

**4.5.2.5 void MovePecaX ( TipoPeca \* *targetPeca*, int *inputX* )**

Função que move uma peça verticalmente.

## Parameters

<i>*targetPeca</i>	ponteiro para peça.
<i>inputX</i>	nova posição x da peça.

**4.5.2.6 void MovePecaY ( TipoPeca \* *targetPeca*, int *inputY* )**

Função que move uma peça horizontalmente.

## Parameters

<i>*targetPeca</i>	ponteiro para peça.
<i>inputY</i>	nova posição y da peça.



#### 4.5.2.7 TipoTela PecaGetBloco ( TipoPeca \* *peca*, int *y*, int *x* )

Função que retorna uma peça e sua posição.

##### Parameters

<i>*peça</i>	ponteiro para peça.
<i>y</i>	inteiro que indica a posição horizontal da peça.
<i>x</i>	inteiro que indica a posição vertical da peça.

##### Returns

a matriz da peça.

#### 4.5.2.8 int PecaGetCor ( TipoPeca \* *peca* )

Função que retorna a cor de uma peça.

##### Parameters

<i>*peca</i>	ponteiro para peça.
--------------	---------------------

##### Returns

um inteiro que indica a cor da peça.

#### 4.5.2.9 int PecaGetSpeed ( TipoPeca \* *peca* )

Função que retorna a velocidade de queda da peça.

##### Parameters

<i>*peca</i>	ponteiro para peça.
--------------	---------------------

##### Returns

um inteiro com a velocidade da peça.

#### 4.5.2.10 int PecaGetX ( TipoPeca \* *peca* )

Função que retorna a posição vertical da peça

##### Parameters

<i>*peca</i>	ponteiro para peça.
--------------	---------------------

**Returns**

um inteiro com a posição vertical da peça.

**4.5.2.11 int PecaGetY ( TipoPeca \* *peca* )**

Função que retorna a posição horizontal da peça

**Parameters**

<i>*peca</i>	ponteiro para peça.
--------------	---------------------

**Returns**

um inteiro com a posição horizontal da peça.

**4.5.2.12 void RotacionaPeca ( TipoPeca \* *peca* )**

Função que rotaciona a peça.

**Parameters**

<i>*peça</i>	ponteiro para a peça atual.
--------------	-----------------------------

**4.5.2.13 void SpeedUp ( TipoPeca \* *peca*, int *multiplicador* )**

Função que incrementa a velocidade de queda da peça.

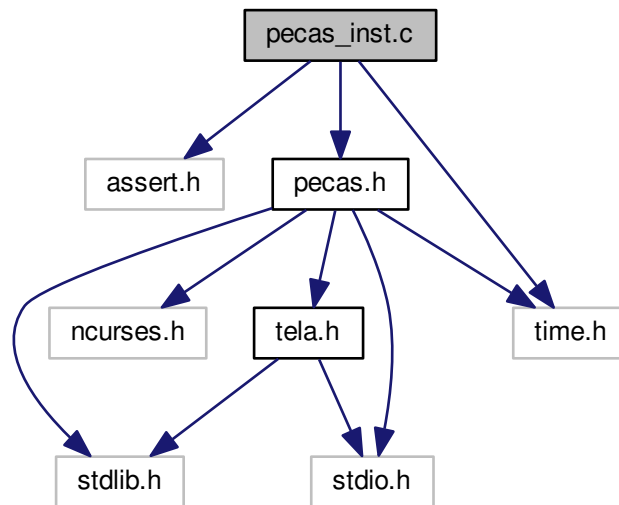
**Parameters**

<i>*peça</i>	ponteiro para a peça atual.
<i>multiplicador</i>	inteiro que representa o quanto a velocidade de queda da peça será incrementada.

## 4.6 pecas\_inst.c File Reference

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

Include dependency graph for pecas\_inst.c:



## Functions

- void **MovePecaXInst** ([TipoPeca](#) \*peca, int x)
- void **MovePecaYInst** ([TipoPeca](#) \*peca, int y)
- [TipoPeca](#) \* **AlocaPecalnst** ()
- void **LiberaPecalnst** ([TipoPeca](#) \*peca)
- void **CopiaPecalnst** ([TipoPeca](#) \*a, [TipoPeca](#) \*b)
- void **GeraPecalnst** ([TipoPeca](#) \*peca)
- void **RotacionaPecalnst** ([TipoPeca](#) \*peca)
- void **SpeedUpInst** ([TipoPeca](#) \*peca, int valor)

### 4.6.1 Detailed Description

Implementa a instrumentação do módulo peças. Garante a validade das entradas e saídas do módulo peças.

#### Author

Cristóvão

#### Since

05/07/16

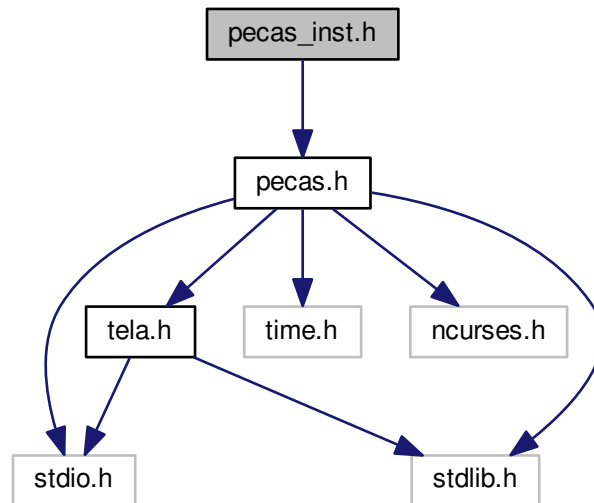
#### Version

1.1

## 4.7 pecas\_inst.h File Reference

```
#include "pecas.h"
```

Include dependency graph for pecas\_inst.h:



### Macros

- `#define MovePecaX MovePecaXInst`
- `#define MovePecaY MovePecaYInst`
- `#define AlocaPeca AlocaPecalnst`
- `#define LiberaPeca LiberaPecalnst`
- `#define CopiaPeca CopiaPecalnst`
- `#define GeraPeca GeraPecalnst`
- `#define RotacionaPeca RotacionaPecalnst`

### Functions

- `TipoPeca * AlocaPecalnst ()`
- `void LiberaPecalnst (TipoPeca *)`
- `void CopiaPecalnst (TipoPeca *, TipoPeca *)`
- `void GeraPecalnst (TipoPeca *)`
- `void MovePecaXInst (TipoPeca *, int)`
- `void MovePecaYInst (TipoPeca *, int)`
- `void RotacionaPecalnst (TipoPeca *)`

### 4.7.1 Detailed Description

Contém a definição do módulo de instrumentação das peças.

**Author**

Cristóvão

**Since**

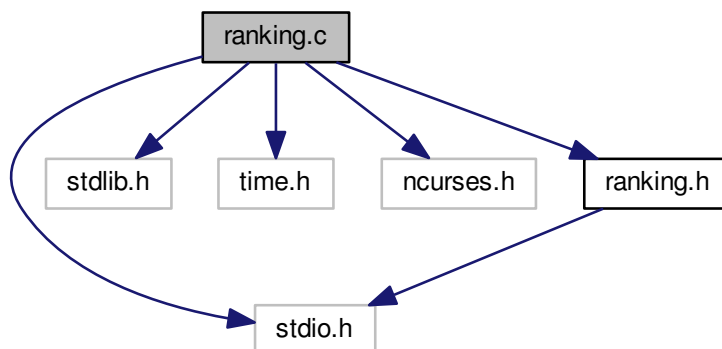
05/07/16

**Version**

1.1

## 4.8 ranking.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <ncurses.h>
#include "ranking.h"
Include dependency graph for ranking.c:
```



### Functions

- int \* [ReceberApelido](#) ()
- void [ConverterApelido](#) (int \*apelido, [TipoJogador](#) \*jogador)
- void [ReceberData](#) ([TipoJogador](#) \*jogador)
- int [VerificaPlacar](#) ()
- void [CriaPlacar](#) ([TipoJogador](#) jogador)
- void [AtualizaPlacar](#) ([TipoJogador](#) jogador)

### 4.8.1 Detailed Description

Gerencia a criação e manipulação do placar. Este módulo implementa o placar do jogo, bem como gerencia as operações relacionados a ele.

#### Author

André

#### Since

11/04/16

#### Version

2.1

### 4.8.2 Function Documentation

#### 4.8.2.1 void AtualizaPlacar ( TipoJogador *jogador* )

Função que atualiza o placar.

##### Parameters

<i>jogador</i>	estrutura contendo os dados do jogador.
----------------	---

#### 4.8.2.2 void ConverterApelido ( int \* *apelido*, TipoJogador \* *jogador* )

Função que converte o apelido de um vetor de inteiros para um vetor de caracteres.

##### Parameters

* <i>apelido</i>	ponteiro para o apelido do jogador
* <i>jogador</i>	ponteiro para a estrutura contendo os dados do jogador.

#### 4.8.2.3 void CriaPlacar ( TipoJogador *jogador* )

Função que cria um arquivo com o placar.

##### Parameters

<i>jogador</i>	estrutura contendo os dados do jogador.
----------------	---

#### 4.8.2.4 int\* ReceberApelido ( )

Função que recebe o apelido do jogador.

##### Returns

um ponteiro para o apelido.

#### 4.8.2.5 void ReceberData ( TipoJogador \* jogador )

Função que recebe a data em que o jogo foi executado.

##### Parameters

*jogador	ponteiro para a estrutura contendo os dados do jogador.
----------	---

#### 4.8.2.6 int VerificaPlacar ( )

Função que verifica a existência de um arquivo com o placar.

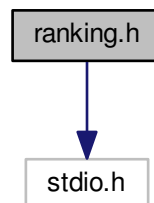
##### Returns

1 ou 0, caso exista ou não um arquivo, respectivamente.

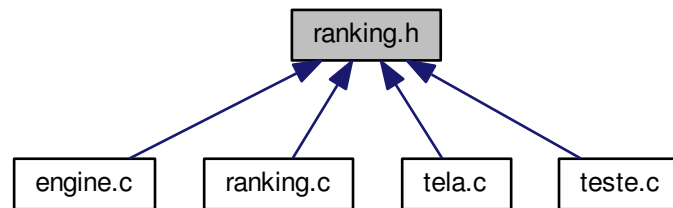
## 4.9 ranking.h File Reference

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

Include dependency graph for ranking.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [TipoJogador](#)

## Macros

- `#define ARQUIVO "ranking.txt"`
- `#define AUX "aux.txt"`

## Functions

- `int * ReceberApelido ()`
- `void ConverterApelido (int *, TipoJogador *)`
- `void ReceberData (TipoJogador *)`
- `int VerificaPlacar ()`
- `void CriaPlacar (TipoJogador)`
- `void AtualizaPlacar (TipoJogador)`

### 4.9.1 Detailed Description

Define o módulo do placar do jogo Este módulo define as estruturas do módulo e os protótipos de funções.

#### Author

André

#### Since

11/04/16

#### Version

2.1

### 4.9.2 Function Documentation

#### 4.9.2.1 void [AtualizaPlacar](#) ( [TipoJogador](#) *jogador* )

Função que atualiza o placar.



## Parameters

<i>jogador</i>	estrutura contendo os dados do jogador.
----------------	---

**4.9.2.2 void ConverterApelido ( int \* *apelido*, TipoJogador \* *jogador* )**

Função que converte o apelido de um vetor de inteiros para um vetor de caracteres.

## Parameters

* <i>apelido</i>	ponteiro para o apelido do jogador
* <i>jogador</i>	ponteiro para a estrutura contendo os dados do jogador.

**4.9.2.3 void CriaPlacar ( TipoJogador *jogador* )**

Função que cria um arquivo com o placar.

## Parameters

<i>jogador</i>	estrutura contendo os dados do jogador.
----------------	---

**4.9.2.4 int\* ReceberApelido ( )**

Função que recebe o apelido do jogador.

## Returns

um ponteiro para o apelido.

**4.9.2.5 void ReceberData ( TipoJogador \* *jogador* )**

Função que recebe a data em que o jogo foi executado.

## Parameters

* <i>jogador</i>	ponteiro para a estrutura contendo os dados do jogador.
------------------	---

**4.9.2.6 int VerificaPlacar ( )**

Função que verifica a existência de um arquivo com o placar.

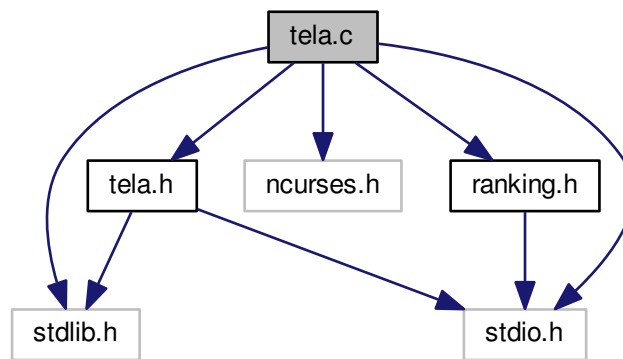
## Returns

1 ou 0, caso exista ou não um arquivo, respectivamente.

## 4.10 tela.c File Reference

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

Include dependency graph for tela.c:



## Macros

- `#define BLOCO '0'`

## Functions

- void `SetPecaCor` (`TipoTela` \*pecaDada, int corAferida)
- void `SetPecaCaractere` (`TipoTela` \*pecaDada, char caractereAferido)
- void `SetPeca` (`TipoTela` \*pecaDada, char caractereAferido, int corAferida)
- int `VerificaSeBloco` (`TipoTela` pecaDada)
- void `InicioTela` ()

*Função que gera e mostra a tela de início.*

- void `CriarTela` (`TipoTela` tela[ ][25])
- void `MostrarTela` (`TipoTela` tela[ ][25], int pontos, int tempoDecorrido)
- void `FimTela` (int pontos)

*Função que gera e mostra a tela de fim de jogo.*

### 4.10.1 Detailed Description

Gerencia a criação e manipulação da tela. Este módulo implementa a tela do jogo, bem como gerencia as operações relacionadas a ela.

**Author**

André

**Since**

11/04/16

**Version**

2.3

### 4.10.2 Function Documentation

#### 4.10.2.1 void CriarTela ( TipoTela tela[ ][25] )

Função que cria a tela vazia.

**Parameters**

<i>tela[ ][25]</i>	matriz da tela.
--------------------	-----------------

#### 4.10.2.2 void MostrarTela ( TipoTela tela[ ][25], int pontos, int tempoDecorrido )

Função que mostra a tela.

**Parameters**

<i>*tela[ ][25]</i>	matriz da tela.
<i>pontos</i>	inteiro contendo a pontuação atual.
<i>tempoDecorrido</i>	inteiro contendo o tempo decorrido.

#### 4.10.2.3 void SetPeca ( TipoTela \* pecaDada, char caractereAferido, int corAferida )

Função que faz as alterações em uma peça.

**Parameters**

<i>*pecaDada</i>	ponteiro para peça.
<i>caractereAferido</i>	caractere a ser passado para a peça.
<i>corAferida</i>	inteiro referente à cor.

#### 4.10.2.4 void SetPecaCaractere ( TipoTela \* *pecaDada*, char *caractereAferido* )

Função que indica o caractere a ser usado em uma peça

##### Parameters

<i>*pecaDada</i>	ponteiro para peça.
<i>caractereAferido</i>	caractere a ser passado para a peça.

#### 4.10.2.5 void SetPecaCor ( TipoTela \* *pecaDada*, int *corAferida* )

Função que muda a cor de uma peça.

##### Parameters

<i>*pecaDada</i>	ponteiro para peça.
<i>corAferida</i>	inteiro referente à cor.

#### 4.10.2.6 int VerificaSeBloco ( TipoTela *pecaDada* )

Função que verifica se uma peça tem o símbolo do fundo.

##### Parameters

<i>*pecaDada</i>	ponteiro para peça.
------------------	---------------------

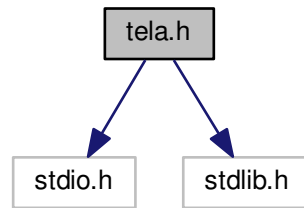
##### Returns

0 ou 1.

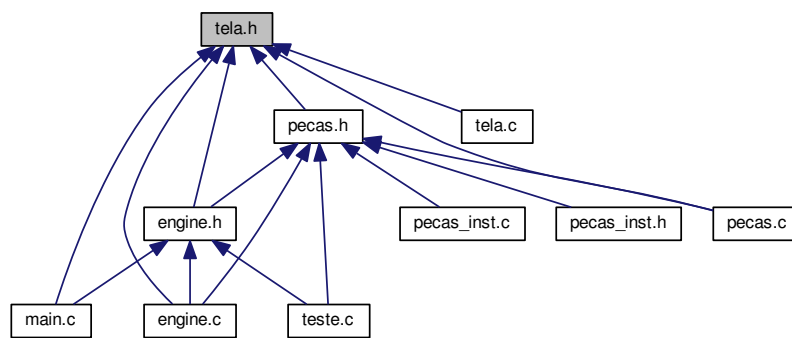
## 4.11 tela.h File Reference

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

Include dependency graph for tela.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [TipoTela](#)

## Functions

- void [InicioTela](#) ()  
*Função que gera e mostra a tela de início.*
- void [CriarTela](#) ([TipoTela](#)[][25])
- void [MostrarTela](#) ([TipoTela](#)[][25], int, int)
- void [SetPecaCor](#) ([TipoTela](#) \*, int)
- void [SetPecaCaractere](#) ([TipoTela](#) \*, char)
- void [SetPeca](#) ([TipoTela](#) \*, char, int)
- void [FimTela](#) (int)  
*Função que gera e mostra a tela de fim de jogo.*
- int [VerificaSeBloco](#) ([TipoTela](#))

### 4.11.1 Detailed Description

Faz a definição do módulo da tela. Define as estruturas do módulo e os protótipos de funções.

#### Author

André

#### Since

11/04/16

#### Version

2.3

### 4.11.2 Function Documentation

#### 4.11.2.1 void CriarTela ( TipoTela tela[ ][25] )

Função que cria a tela vazia.

##### Parameters

<i>tela[ ][25]</i>	matriz da tela.
--------------------	-----------------

#### 4.11.2.2 void MostrarTela ( TipoTela tela[ ][25], int pontos, int tempoDecorrido )

Função que mostra a tela.

##### Parameters

<i>*tela[ ][25]</i>	matriz da tela.
<i>pontos</i>	inteiro contendo a pontuação atual.
<i>tempoDecorrido</i>	inteiro contendo o tempo decorrido.

#### 4.11.2.3 void SetPeca ( TipoTela \* pecaDada, char caractereAferido, int corAferida )

Função que faz as alterações em uma peça.

##### Parameters

<i>*pecaDada</i>	ponteiro para peça.
<i>caractereAferido</i>	caractere a ser passado para a peça.
<i>corAferida</i>	inteiro referente à cor.

#### 4.11.2.4 void SetPecaCaractere ( TipoTela \* *pecaDada*, char *caractereAferido* )

Função que indica o caractere a ser usado em uma peça

##### Parameters

<i>*pecaDada</i>	ponteiro para peça.
<i>caractereAferido</i>	caractere a ser passado para a peça.

#### 4.11.2.5 void SetPecaCor ( TipoTela \* *pecaDada*, int *corAferida* )

Função que muda a cor de uma peça.

##### Parameters

<i>*pecaDada</i>	ponteiro para peça.
<i>corAferida</i>	inteiro referente à cor.

#### 4.11.2.6 int VerificaSeBloco ( TipoTela *pecaDada* )

Função que verifica se uma peça tem o símbolo do fundo.

##### Parameters

<i>*pecaDada</i>	ponteiro para peça.
------------------	---------------------

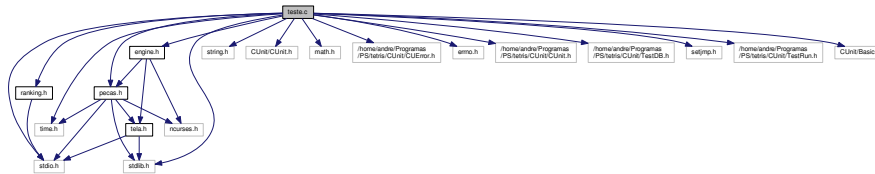
##### Returns

0 ou 1.

## 4.12 teste.c File Reference

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "CUnit/CUnit.h"
#include "CUnit/Basic.h"
#include "engine.h"
#include "pecas.h"
#include "ranking.h"
```

Include dependency graph for teste.c:



## Functions

- void [TesteConverteVetor](#) (void)
- void [TesteData](#) (void)
- void [TesteEntradaPlacar](#) (void)
- void [TesteAtualizaPlacar](#) (void)
- void [AdicionarSuiteRanking](#) (void)  
*Função que cria suíte de testes do módulo ranking.*
- void [TesteAlocaPeca](#) (void)  
*Função que testa a alocação de peças.*
- void [TesteCorPecaCorFundo](#) (void)  
*Função que testa as cores das peças.*
- void [TesteCorPecaDiferente](#) (void)
- void [AdicionarSuitePecas](#) (void)  
*Função que cria suíte de testes do módulo pecas.*
- void [TestaMorte](#) (void)
- void [TestaColisaoParedeCaixaPreta](#) (void)  
*Função que testa a colisão com as paredes da tela.*
- void [TestaColisaoBlocoCaixaPreta](#) (void)  
*Função que testa a colisão da peça com os blocos da tela.*
- void [TestaColisaoCaixaAberta](#) (void)
- void [TesteLimpaLinha](#) (void)  
*Função que testa se uma linha completa é limpada.*
- void [TestaPecaNoTopo](#) (void)  
*Função que testa a inclusão de uma nova peça na tela.*
- void [AdicionarSuiteEngine](#) (void)  
*Função que cria a suíte de testes do modulo de pecas.*
- int **main** ()

### 4.12.1 Detailed Description

Gerencia os testes de funcionamento do programa.

#### Author

Cristóvão

#### Since

11/04/16

#### Version

2.6



### 4.12.2 Function Documentation

#### 4.12.2.1 void TestaColisaoCaixaAberta ( void )

Função que testa os casos de colisão usando caixa aberta, ou seja, considerando a implementação das funções de colisão.

#### 4.12.2.2 void TestaMorte ( void )

Função que testa se o jogo acaba com a chegada no limite.

#### 4.12.2.3 void TesteAtualizaPlacar ( void )

Função que testa a inserção de novos dados no arquivo contendo o placar. O teste verifica se o arquivo contendo o placar foi atualizado.

#### 4.12.2.4 void TesteConverteVetor ( void )

Função que testa a conversão de um vetor de inteiros para uma string. O teste abrange apenas a entrada e a saída da função, avaliando se corresponde ao esperado.

#### 4.12.2.5 void TesteCorPecaDiferente ( void )

Função que testa se as cores das peças geradas são diferentes a cada rodada.

#### 4.12.2.6 void TesteData ( void )

Função que testa a data recebida pelo sistema. Para isso foram considerados os valores limítrofes do dia, do mês e do ano, para com isso determinar se a data recebida pelo sistema é real e assume valores razoáveis.

#### 4.12.2.7 void TesteEntradaPlacar ( void )

Função que testa a criação de um arquivo contendo o placar. O teste verifica se é criado um arquivo após a entrada da primeira pontuação do jogo.



# Index

AddBloco  
  engine.c, 9  
  engine.h, 13

AlocaPeca  
  pecas.c, 18  
  pecas.h, 23

AtualizaPlacar  
  ranking.c, 30  
  ranking.h, 32

comandos  
  engine.c, 9

ConverterApelido  
  ranking.c, 30  
  ranking.h, 33

CopiaDaMatrizDePecas  
  pecas.c, 18

CopiaPeca  
  pecas.c, 19  
  pecas.h, 23

CriaPlacar  
  ranking.c, 30  
  ranking.h, 33

CriarTela  
  tela.c, 35  
  tela.h, 38

DeletaBloco  
  engine.c, 9  
  engine.h, 13

DeletaLinhas  
  engine.c, 9  
  engine.h, 14

engine.c, 7  
  AddBloco, 9  
  comandos, 9  
  DeletaBloco, 9  
  DeletaLinhas, 9  
  Gravidade, 9  
  Loop, 10  
  PegaInput, 10  
  PoePecaNoTopo, 10  
  RemoveBloco, 10  
  Temporizador, 10  
  VerificaColisao, 11  
  VerificaLinhas, 11  
  VerificaMorte, 11

engine.h, 12  
  AddBloco, 13  
  DeletaBloco, 13  
  DeletaLinhas, 14  
  Gravidade, 14  
  Loop, 14  
  PegaInput, 14  
  PoePecaNoTopo, 15  
  RemoveBloco, 15  
  VerificaColisao, 15  
  VerificaLinhas, 15  
  VerificaMorte, 16

GeraPeca  
  pecas.c, 19  
  pecas.h, 24

GeraPecaEspecificas  
  pecas.c, 19  
  pecas.h, 24

Gravidade  
  engine.c, 9  
  engine.h, 14

Loop  
  engine.c, 10  
  engine.h, 14

main.c, 16

MostrarTela  
  tela.c, 35  
  tela.h, 38

MovePecaX  
  pecas.c, 19  
  pecas.h, 24

MovePecaY  
  pecas.c, 19  
  pecas.h, 24

PecaGetBloco  
  pecas.c, 20  
  pecas.h, 24

PecaGetCor  
  pecas.c, 20  
  pecas.h, 25

PecaGetSpeed  
  pecas.c, 20  
  pecas.h, 25

PecaGetX  
  pecas.c, 21  
  pecas.h, 25

PecaGetY  
  pecas.c, 21

- pecas.h, [26](#)
- pecas.c, [17](#)
  - AlocaPeca, [18](#)
  - CopiaDaMatrizDePecas, [18](#)
  - CopiaPeca, [19](#)
  - GeraPeca, [19](#)
  - GeraPecaEspecifica, [19](#)
  - MovePecaX, [19](#)
  - MovePecaY, [19](#)
  - PecaGetBloco, [20](#)
  - PecaGetCor, [20](#)
  - PecaGetSpeed, [20](#)
  - PecaGetX, [21](#)
  - PecaGetY, [21](#)
  - RotacionalPeca, [21](#)
  - SpeedUp, [21](#)
- pecas.h, [22](#)
  - AlocaPeca, [23](#)
  - CopiaPeca, [23](#)
  - GeraPeca, [24](#)
  - GeraPecaEspecifica, [24](#)
  - MovePecaX, [24](#)
  - MovePecaY, [24](#)
  - PecaGetBloco, [24](#)
  - PecaGetCor, [25](#)
  - PecaGetSpeed, [25](#)
  - PecaGetX, [25](#)
  - PecaGetY, [26](#)
  - RotacionalPeca, [26](#)
  - SpeedUp, [26](#)
- pecas\_inst.c, [26](#)
- pecas\_inst.h, [28](#)
- PegaInput
  - engine.c, [10](#)
  - engine.h, [14](#)
- PoePecaNoTopo
  - engine.c, [10](#)
  - engine.h, [15](#)
- ranking.c, [29](#)
  - AtualizaPlacar, [30](#)
  - ConverterApelido, [30](#)
  - CriaPlacar, [30](#)
  - ReceberApelido, [30](#)
  - ReceberData, [31](#)
  - VerificaPlacar, [31](#)
- ranking.h, [31](#)
  - AtualizaPlacar, [32](#)
  - ConverterApelido, [33](#)
  - CriaPlacar, [33](#)
  - ReceberApelido, [33](#)
  - ReceberData, [33](#)
  - VerificaPlacar, [33](#)
- ReceberApelido
  - ranking.c, [30](#)
  - ranking.h, [33](#)
- ReceberData
  - ranking.c, [31](#)
  - ranking.h, [33](#)
- RemoveBloco
  - engine.c, [10](#)
  - engine.h, [15](#)
- RotacionalPeca
  - pecas.c, [21](#)
  - pecas.h, [26](#)
- SetPeca
  - tela.c, [35](#)
  - tela.h, [38](#)
- SetPecaCaractere
  - tela.c, [36](#)
  - tela.h, [39](#)
- SetPecaCor
  - tela.c, [36](#)
  - tela.h, [39](#)
- SpeedUp
  - pecas.c, [21](#)
  - pecas.h, [26](#)
- tela.c, [34](#)
  - CriarTela, [35](#)
  - MostrarTela, [35](#)
  - SetPeca, [35](#)
  - SetPecaCaractere, [36](#)
  - SetPecaCor, [36](#)
  - VerificaSeBloco, [36](#)
- tela.h, [36](#)
  - CriarTela, [38](#)
  - MostrarTela, [38](#)
  - SetPeca, [38](#)
  - SetPecaCaractere, [39](#)
  - SetPecaCor, [39](#)
  - VerificaSeBloco, [39](#)
- Temporizador
  - engine.c, [10](#)
- TestaColisaoCaixaAberta
  - teste.c, [41](#)
- TestaMorte
  - teste.c, [41](#)
- teste.c, [39](#)
  - TestaColisaoCaixaAberta, [41](#)
  - TestaMorte, [41](#)
  - TesteAtualizaPlacar, [41](#)
  - TesteConverteVetor, [41](#)
  - TesteCorPecaDiferente, [41](#)
  - TesteData, [41](#)
  - TesteEntradaPlacar, [41](#)
- TesteAtualizaPlacar
  - teste.c, [41](#)
- TesteConverteVetor
  - teste.c, [41](#)
- TesteCorPecaDiferente
  - teste.c, [41](#)
- TesteData
  - teste.c, [41](#)
- TesteEntradaPlacar
  - teste.c, [41](#)
- TipoJogador, [5](#)

TipoPeca, [5](#)

TipoTela, [6](#)

VerificaColisao

engine.c, [11](#)

engine.h, [15](#)

VerificaLinhas

engine.c, [11](#)

engine.h, [15](#)

VerificaMorte

engine.c, [11](#)

engine.h, [16](#)

VerificaPlacar

ranking.c, [31](#)

ranking.h, [33](#)

VerificaSeBloco

tela.c, [36](#)

tela.h, [39](#)