

Tetris

Gerado por Doxygen 1.8.11

Sumário

1	Classes	1
1.1	Referência da Estrutura TipoJogador	1
1.1.1	Atributos	1
1.1.1.1	ano	1
1.1.1.2	apelido	1
1.1.1.3	dia	1
1.1.1.4	mes	1
1.1.1.5	pontos	1
1.1.1.6	tempo	1
1.2	Referência da Estrutura TipoPeca	2
1.2.1	Atributos	2
1.2.1.1	pecas	2
1.2.1.2	speed	2
1.2.1.3	x	2
1.2.1.4	y	2
1.3	Referência da Estrutura TipoTela	2
1.3.1	Atributos	3
1.3.1.1	cor	3
1.3.1.2	peca	3

2 Arquivos	5
2.1 Referência do Arquivo engine.c	5
2.1.1 Descrição Detalhada	7
2.1.2 Definições e macros	7
2.1.2.1 BLOCO	7
2.1.2.2 CORFUNDO	7
2.1.2.3 CORVERMELHO	7
2.1.2.4 TAMANHOTELAX	7
2.1.2.5 TAMANHOTELAY	7
2.1.2.6 TEST_MODE	7
2.1.2.7 VAZIO	7
2.1.3 Enumerações	7
2.1.3.1 comandos	7
2.1.4 Funções	7
2.1.4.1 AddBloco(TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])	7
2.1.4.2 DeletaBloco(TipoTela *unidade)	8
2.1.4.3 DeletaLinhas(TipoTela tela[][TAMANHOTELAX], int *pontuacao)	8
2.1.4.4 Gravidade(TipoTela tela[][TAMANHOTELAX], int posicaolinha)	8
2.1.4.5 Loop(TipoTela tela[][TAMANHOTELAX])	8
2.1.4.6 PegalInput()	9
2.1.4.7 PoePecaNoTopo(TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])	9
2.1.4.8 RemoveBloco(TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])	9
2.1.4.9 Temporizador(int milissegundos, TipoPeca *peca, int *flagDesce)	9
2.1.4.10 VerificaColisao(TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])	10
2.1.4.11 VerificaLinhas(TipoTela tela[][TAMANHOTELAX])	10
2.1.4.12 VerificaMorte(TipoTela tela[][TAMANHOTELAX])	10
2.1.5 Variáveis	11
2.1.5.1 com	11
2.2 Referência do Arquivo engine.h	11
2.2.1 Descrição Detalhada	13

2.2.2	Definições e macros	13
2.2.2.1	TAMANHOTELAX	13
2.2.2.2	TAMANHOTELAY	13
2.2.3	Funções	13
2.2.3.1	AddBloco(TipoPeca *, TipoTela[][TAMANHOTELAX])	13
2.2.3.2	DeletaBloco(TipoTela *)	13
2.2.3.3	DeletaLinhas(TipoTela[][TAMANHOTELAX], int *)	14
2.2.3.4	Gravidade(TipoTela[][TAMANHOTELAX], int)	14
2.2.3.5	Loop(TipoTela[][TAMANHOTELAX])	14
2.2.3.6	PegaInput()	14
2.2.3.7	PoePecaNoTopo(TipoPeca *, TipoTela[][TAMANHOTELAX])	15
2.2.3.8	RemoveBloco(TipoPeca *, TipoTela[][TAMANHOTELAX])	15
2.2.3.9	VerificaColisao(TipoPeca *, TipoTela[][TAMANHOTELAX])	15
2.2.3.10	VerificaLinhas(TipoTela[][TAMANHOTELAX])	15
2.2.3.11	VerificaMorte(TipoTela[][TAMANHOTELAX])	16
2.3	Referência do Arquivo main.c	16
2.3.1	Descrição Detalhada	17
2.3.2	Funções	18
2.3.2.1	main()	18
2.4	Referência do Arquivo pecas.c	18
2.4.1	Descrição Detalhada	19
2.4.2	Definições e macros	20
2.4.2.1	B	20
2.4.2.2	COR_FUNDO	20
2.4.2.3	NUM_PECAS	20
2.4.2.4	TAMANHO_X	20
2.4.2.5	TAMANHO_Y	20
2.4.2.6	V	20
2.4.3	Funções	20
2.4.3.1	AlocaPeca()	20

2.4.3.2	CopiaDaMatrizDePecas(char MatrizFonte[][5], TipoTela MatrizAlvo[][5])	20
2.4.3.3	CopiaPeca(TipoPeca *pecaIn, TipoPeca *pecaOut)	20
2.4.3.4	GeraPeca(TipoPeca *ponteiroPeca)	21
2.4.3.5	GeraPecaEspecifica(TipoPeca *ponteiroPeca, int indicePeca, int direcao)	21
2.4.3.6	LiberaPeca(TipoPeca *peca)	21
2.4.3.7	MovePecaX(TipoPeca *targetPeca, int inputX)	21
2.4.3.8	MovePecaY(TipoPeca *targetPeca, int inputY)	21
2.4.3.9	PecaGetBloco(TipoPeca *peca, int y, int x)	22
2.4.3.10	PecaGetCor(TipoPeca *peca)	22
2.4.3.11	PecaGetSpeed(TipoPeca *peca)	22
2.4.3.12	PecaGetX(TipoPeca *peca)	22
2.4.3.13	PecaGetY(TipoPeca *peca)	23
2.4.3.14	RotacionaPeca(TipoPeca *peca)	23
2.4.3.15	SpeedUp(TipoPeca *peca, int multiplicador)	23
2.5	Referência do Arquivo pecas.h	23
2.5.1	Descrição Detalhada	25
2.5.2	Definições e macros	26
2.5.2.1	TAMANHO_X	26
2.5.2.2	TAMANHO_Y	26
2.5.3	Funções	26
2.5.3.1	AlocaPeca()	26
2.5.3.2	CopiaPeca(TipoPeca *, TipoPeca *)	26
2.5.3.3	GeraPeca(TipoPeca *)	26
2.5.3.4	GeraPecaEspecifica(TipoPeca *, int, int)	26
2.5.3.5	LiberaPeca(TipoPeca *)	27
2.5.3.6	MovePecaX(TipoPeca *, int)	27
2.5.3.7	MovePecaY(TipoPeca *, int)	27
2.5.3.8	PecaGetBloco(TipoPeca *, int, int)	27
2.5.3.9	PecaGetCor(TipoPeca *)	27
2.5.3.10	PecaGetSpeed(TipoPeca *)	28

2.5.3.11	PecaGetX(TipoPeca *)	28
2.5.3.12	PecaGetY(TipoPeca *)	28
2.5.3.13	RotacionaPeca(TipoPeca *)	28
2.5.3.14	SpeedUp(TipoPeca *, int)	29
2.6	Referência do Arquivo ranking.c	29
2.6.1	Descrição Detalhada	30
2.6.2	Funções	30
2.6.2.1	AtualizaPlacar(TipoJogador jogador)	30
2.6.2.2	ConverterApelido(int *apelido, TipoJogador *jogador)	30
2.6.2.3	CriaPlacar(TipoJogador jogador)	31
2.6.2.4	ReceberApelido()	31
2.6.2.5	ReceberData(TipoJogador *jogador)	31
2.6.2.6	VerificaPlacar()	31
2.7	Referência do Arquivo ranking.h	31
2.7.1	Descrição Detalhada	33
2.7.2	Definições e macros	33
2.7.2.1	ARQUIVO	33
2.7.2.2	AUX	33
2.7.3	Funções	33
2.7.3.1	AtualizaPlacar(TipoJogador)	33
2.7.3.2	ConverterApelido(int *, TipoJogador *)	33
2.7.3.3	CriaPlacar(TipoJogador)	34
2.7.3.4	ReceberApelido()	34
2.7.3.5	ReceberData(TipoJogador *)	34
2.7.3.6	VerificaPlacar()	34
2.8	Referência do Arquivo tela.c	34
2.8.1	Descrição Detalhada	35
2.8.2	Definições e macros	36
2.8.2.1	BLOCO	36
2.8.3	Funções	36

2.8.3.1	CriarTela(TipoTela tela[][25])	36
2.8.3.2	FimTela(int pontos)	36
2.8.3.3	InicioTela()	36
2.8.3.4	MostrarTela(TipoTela tela[][25], int pontos, int tempoDecorrido)	36
2.8.3.5	SetPeca(TipoTela *pecaDada, char caractereAferido, int corAferida)	37
2.8.3.6	SetPecaCaractere(TipoTela *pecaDada, char caractereAferido)	37
2.8.3.7	SetPecaCor(TipoTela *pecaDada, int corAferida)	37
2.8.3.8	VerificaSeBloco(TipoTela pecaDada)	37
2.9	Referência do Arquivo tela.h	37
2.9.1	Descrição Detalhada	39
2.9.2	Funções	39
2.9.2.1	CriarTela(TipoTela[][25])	39
2.9.2.2	FimTela(int)	39
2.9.2.3	InicioTela()	39
2.9.2.4	MostrarTela(TipoTela[][25], int, int)	40
2.9.2.5	SetPeca(TipoTela *, char, int)	40
2.9.2.6	SetPecaCaractere(TipoTela *, char)	40
2.9.2.7	SetPecaCor(TipoTela *, int)	40
2.9.2.8	VerificaSeBloco(TipoTela)	40
2.10	Referência do Arquivo teste.c	41
2.10.1	Descrição Detalhada	42
2.10.2	Funções	42
2.10.2.1	AdicionarSuite(void)	42
2.10.2.2	main()	42
2.10.2.3	TestaColisaoBloco(void)	42
2.10.2.4	TestaColisaoParede(void)	42
2.10.2.5	TestaMorte(void)	42
2.10.2.6	TestaPecaNoTopo(void)	42
2.10.2.7	TesteAlocaPeca(void)	42
2.10.2.8	TesteCorPecaCorFundo(void)	42
2.10.2.9	TesteCorPecaDiferente(void)	42
2.10.2.10	TesteLimpaLinha(void)	42

Capítulo 1

Classes

1.1 Referência da Estrutura TipoJogador

```
#include <ranking.h>
```

Atributos Públicos

- char [apelido](#) [7]
- int [pontos](#)
- int [dia](#)
- int [mes](#)
- int [ano](#)
- int [tempo](#)

1.1.1 Atributos

1.1.1.1 int TipoJogador::ano

1.1.1.2 char TipoJogador::apelido[7]

1.1.1.3 int TipoJogador::dia

1.1.1.4 int TipoJogador::mes

1.1.1.5 int TipoJogador::pontos

1.1.1.6 int TipoJogador::tempo

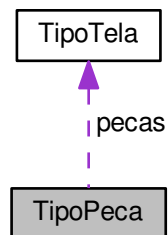
A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [ranking.h](#)

1.2 Referência da Estrutura TipoPeca

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

Diagrama de colaboração para TipoPeca:



Atributos Públicos

- [TipoTela pecas](#) [TAMANHO_Y][TAMANHO_X]
- [int x](#)
- [int y](#)
- [int speed](#)

1.2.1 Atributos

1.2.1.1 **TipoTela** TipoPeca::pecas[TAMANHO_Y][TAMANHO_X]

1.2.1.2 **int** TipoPeca::speed

1.2.1.3 **int** TipoPeca::x

1.2.1.4 **int** TipoPeca::y

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

- [pecas.h](#)

1.3 Referência da Estrutura TipoTela

```
#include <tela.h>
```

Atributos Públicos

- char [peca](#)
- int [cor](#)

1.3.1 Atributos

1.3.1.1 int TipoTela::cor

1.3.1.2 char TipoTela::peca

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

- [tela.h](#)

Capítulo 2

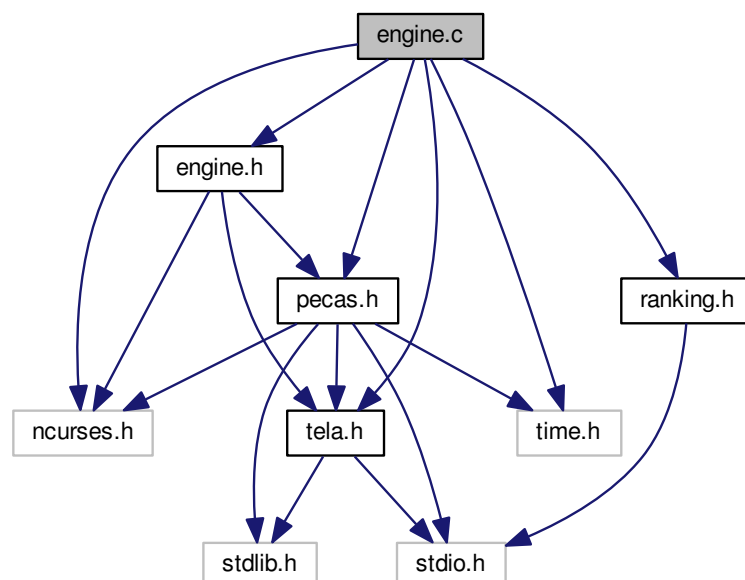
Arquivos

2.1 Referência do Arquivo engine.c

Gerencia o funcionamento completo do jogo.

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

Gráfico de dependência de inclusões para engine.c:



Definições e Macros

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

Enumerações

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

Comandos.

Funções

- int VerificaMorte (TipoTela tela[][TAMANHOTELAX])
Função que verifica se há algum bloco na linha limítrofe.
- int VerificaColisao (TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])
Função que verifica a colisão de blocos.
- void PoePecaNoTopo (TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])
Função que insere uma peça na matriz da tela.
- void DeletaBloco (TipoTela *unidade)
Função que limpa um campo da matriz da tela.
- void AddBloco (TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])
Função que insere um bloco na matriz da tela.
- void RemoveBloco (TipoPeca *peca, TipoTela tela[][TAMANHOTELAX])
Função que remove um bloco da matriz da tela.
- int VerificaLinhas (TipoTela tela[][TAMANHOTELAX])
Função que verifica se há uma linha completa.
- void Gravidade (TipoTela tela[][TAMANHOTELAX], int posicaolinha)
Função que decrementa a posição de uma linha da tela.
- void DeletaLinhas (TipoTela tela[][TAMANHOTELAX], int *pontuacao)
Função que deleta uma linha completa na matriz da tela.
- int PegalInput ()
Função que recebe o comando o usuário.
- int Temporizador (int milissegundos, TipoPeca *peca, int *flagDesce)
Função que coordena o tempo de jogo.
- int Loop (TipoTela tela[][TAMANHOTELAX])
Função que gerencia todas as funções do jogo.

Variáveis

- enum comandos com

2.1.1 Descrição Detalhada

Gerencia o funcionamento completo do jogo.

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

Autor

Cristóvão

Desde

11/04/16

Versão

2.8

2.1.2 Definições e macros

2.1.2.1 `#define BLOCO '0'`

2.1.2.2 `#define CORFUNDO 7`

2.1.2.3 `#define CORVERMELHO 4`

2.1.2.4 `#define TAMANHOTELAX 25`

2.1.2.5 `#define TAMANHOTELAY 15`

2.1.2.6 `#define TEST_MODE`

2.1.2.7 `#define VAZIO ' '`

2.1.3 Enumerações

2.1.3.1 `enum comandos`

Comandos.

Enumeração dos comandos que podem ser inseridos.

Valores de enumerações

ESQUERDA

BAIXO

DIREITA

CIMA

SAIR

2.1.4 Funções

2.1.4.1 `void AddBloco (TipoPeca * peca, TipoTela tela[][TAMANHOTELAX])`

Função que insere um bloco na matriz da tela.

Parâmetros

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

2.1.4.2 void DeletaBloco (TipoTela * *unidade*)

Função que limpa um campo da matriz da tela.

Parâmetros

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

2.1.4.3 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.

Parâmetros

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

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

Função que decrementa a posição de uma linha da tela.

Parâmetros

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

2.1.4.5 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.

Parâmetros

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

Retorna

0 caso não aconteça nenhum erro.

2.1.4.6 int PegalInput ()

Função que recebe o comando o usuário.

Atribui determinados comandos de acordo com as teclas pressionadas.

Retorna

o comando do usuário.

2.1.4.7 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.

Parâmetros

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

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

Função que remove um bloco da matriz da tela.

Parâmetros

* <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.

2.1.4.9 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.

Parâmetros

<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.

Retorna

o comando inserido pelo usuário.

2.1.4.10 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.

Parâmetros

<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.

Retorna

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

2.1.4.11 int VerificaLinhas (TipoTela tela[][TAMANHOTELAX])

Função que verifica se há uma linha completa.

Parâmetros

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

Retorna

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

2.1.4.12 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.

Parâmetros

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

Retorna

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

2.1.5 Variáveis

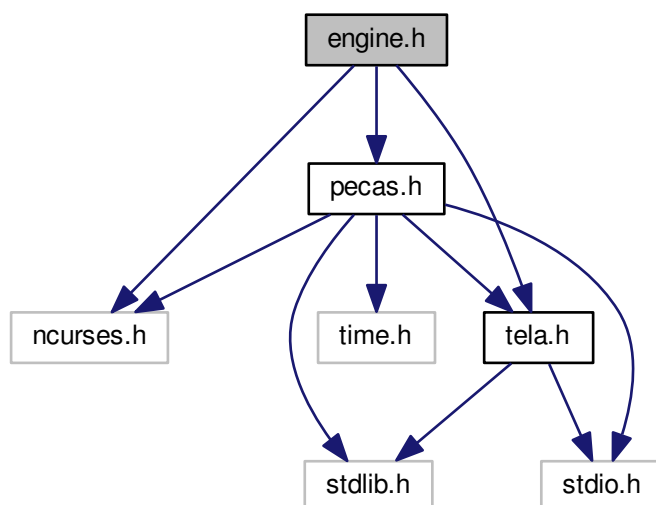
2.1.5.1 enum comandos com

2.2 Referência do Arquivo engine.h

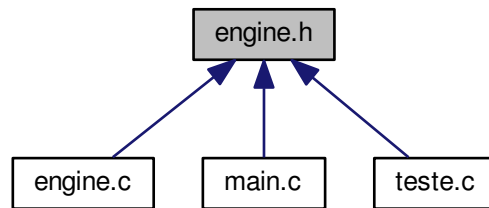
Contém a definição do módulo engine.

```
#include <ncurses.h>
#include "tela.h"
#include "pecas.h"
```

Gráfico de dependência de inclusões para engine.h:



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



Definições e Macros

- `#define TAMANHOTELAY 15`
- `#define TAMANHOTELAX 25`

Funções

- `int VerificaMorte (TipoTela[][TAMANHOTELAX])`
Função que verifica se há algum bloco na linha limítrofe.
- `int VerificaColisao (TipoPeca *, TipoTela[][TAMANHOTELAX])`
Função que verifica a colisão de blocos.
- `void PoePecaNoTopo (TipoPeca *, TipoTela[][TAMANHOTELAX])`
Função que insere uma peça na matriz da tela.
- `void DeletaBloco (TipoTela *)`
Função que limpa um campo da matriz da tela.
- `void AddBloco (TipoPeca *, TipoTela[][TAMANHOTELAX])`
Função que insere um bloco na matriz da tela.
- `void RemoveBloco (TipoPeca *, TipoTela[][TAMANHOTELAX])`
Função que remove um bloco da matriz da tela.
- `int VerificaLinhas (TipoTela[][TAMANHOTELAX])`
Função que verifica se há uma linha completa.
- `void Gravidade (TipoTela[][TAMANHOTELAX], int)`
Função que decrementa a posição de uma linha da tela.
- `void DeletaLinhas (TipoTela[][TAMANHOTELAX], int *)`
Função que deleta uma linha completa na matriz da tela.
- `int PegalInput ()`
Função que recebe o comando o usuário.
- `int Loop (TipoTela[][TAMANHOTELAX])`
Função que gerencia todas as funções do jogo.

2.2.1 Descrição Detalhada

Contém a definição do módulo engine.

Define os protótipos das funções do módulo.

Autor

Cristóvão

Desde

11/04/16

Versão

2.8

2.2.2 Definições e macros

2.2.2.1 #define TAMANHOTELAX 25

2.2.2.2 #define TAMANHOTELAY 15

2.2.3 Funções

2.2.3.1 void AddBloco (TipoPeca * *peca*, TipoTela *tela*[][TAMANHOTELAX])

Função que insere um bloco na matriz da tela.

Parâmetros

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

2.2.3.2 void DeletaBloco (TipoTela * *unidade*)

Função que limpa um campo da matriz da tela.

Parâmetros

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

2.2.3.3 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.

Parâmetros

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

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

Função que decrementa a posição de uma linha da tela.

Parâmetros

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

2.2.3.5 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.

Parâmetros

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

Retorna

0 caso não aconteça nenhum erro.

2.2.3.6 int PegalInput ()

Função que recebe o comando o usuário.

Atribui determinados comandos de acordo com as teclas pressionadas.

Retorna

o comando do usuário.

2.2.3.7 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.

Parâmetros

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

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

Função que remove um bloco da matriz da tela.

Parâmetros

<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.

2.2.3.9 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.

Parâmetros

<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.

Retorna

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

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

Função que verifica se há uma linha completa.

Parâmetros

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

Retorna

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

2.2.3.11 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.

Parâmetros

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

Retorna

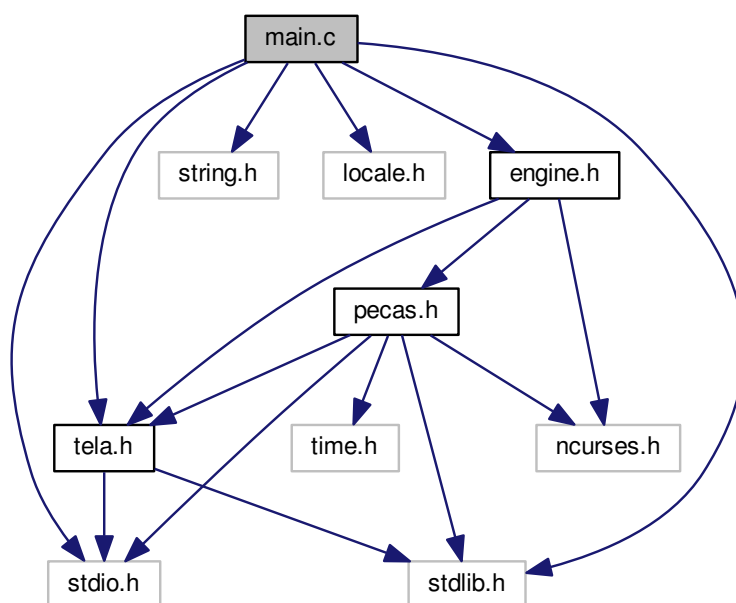
um inteiro com valor 0 ou 1 que indica se a linha de limite foi atingida.

2.3 Referência do Arquivo main.c

Gerencia a criação e manipulação da tela.

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


Gráfico de dependência de inclusões para main.c:



Funções

- int `main` ()

2.3.1 Descrição Detalhada

Gerencia a criação e manipulação da tela.

Este módulo implementa a tela do jogo, bem como gerencia as operações relacionadas à ela.

Autor

Anderson, André, Cristóvão, Pedro Vítor.

Desde

11/04/16

Versão

2.0

2.3.2 Funções

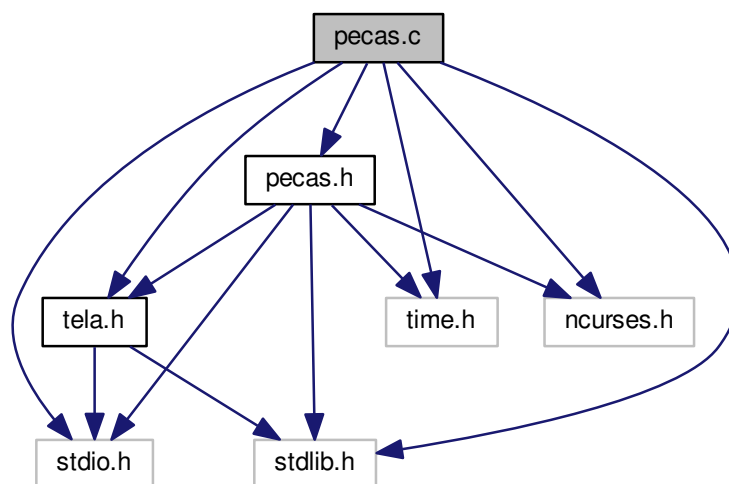
2.3.2.1 int main ()

2.4 Referência do Arquivo pecas.c

Gerencia a criação e manipulação de peças no jogo.

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

Gráfico de dependência de inclusões para pecas.c:



Definições e Macros

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

Funções

- void **SpeedUp** (**TipoPeca** *peca, int multiplicador)
Função que incrementa a velocidade de queda da peça.
- void **RotacionaPeca** (**TipoPeca** *peca)
Função que rotaciona a peça.
- void **CopiaDaMatrizDePecas** (char MatrizFonte[][5], **TipoTela** MatrizAlvo[][5])
Função que copia uma peça de uma matriz à outra.
- void **GeraPeca** (**TipoPeca** *ponteiroPeca)
Função que gera uma peça aleatoriamente.
- void **GeraPecaEspecifica** (**TipoPeca** *ponteiroPeca, int indicePeca, int direcao)
Função que gera uma peça específica.
- **TipoTela** **PecaGetBloco** (**TipoPeca** *peca, int y, int x)
Função que retorna uma peça e sua posição.
- int **PecaGetCor** (**TipoPeca** *peca)
Função que retorna a cor de uma peça.
- void **CopiaPeca** (**TipoPeca** *pecaln, **TipoPeca** *pecaOut)
Função que copia uma peça.
- int **PecaGetX** (**TipoPeca** *peca)
Função que retorna a posição vertical da peça.
- int **PecaGetY** (**TipoPeca** *peca)
Função que retorna a posição horizontal da peça.
- int **PecaGetSpeed** (**TipoPeca** *peca)
Função que retorna a velocidade de queda da peça.
- **TipoPeca** * **AlocaPeca** ()
Função que aloca memória para uma peça.
- void **LiberaPeca** (**TipoPeca** *peca)
Função que desaloca a memória de uma peça.
- void **MovePecaX** (**TipoPeca** *targetPeca, int inputX)
Função que move uma peça verticalmente.
- void **MovePecaY** (**TipoPeca** *targetPeca, int inputY)
Função que move uma peça horizontalmente.

2.4.1 Descrição Detalhada

Gerencia a criação e manipulação de peças no jogo.

Este módulo implementa as peças do jogo, bem como todas as operações relacionadas à elas.

Autor

Anderson

Desde

11/04/16

Versão

2.6

2.4.2 Definições e macros

2.4.2.1 `#define B '0'`

2.4.2.2 `#define COR_FUNDO 7`

2.4.2.3 `#define NUM_PECAS 8`

2.4.2.4 `#define TAMANHO_X 5`

2.4.2.5 `#define TAMANHO_Y 5`

2.4.2.6 `#define V '-'`

2.4.3 Funções

2.4.3.1 `TipoPeca* AlocaPeca ()`

Função que aloca memória para uma peça.

Retorna

a nova peça.

2.4.3.2 `void CopiaDaMatrizDePecas (char MatrizFonte[][5], TipoTela MatrizAlvo[][5])`

Função que copia uma peça de uma matriz à outra.

Parâmetros

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

2.4.3.3 `void CopiaPeca (TipoPeca * pecaIn, TipoPeca * pecaOut)`

Função que copia uma peça.

Parâmetros

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

2.4.3.4 void GeraPeca (TipoPeca * *ponteiroPeca*)

Função que gera uma peça aleatoriamente.

Parâmetros

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

2.4.3.5 void GeraPecaEspecificca (TipoPeca * *ponteiroPeca*, int *indicePeca*, int *direcao*)

Função que gera uma peça específica.

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

Parâmetros

<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.

2.4.3.6 void LiberaPeca (TipoPeca * *peca*)

Função que desaloca a memória de uma peça.

2.4.3.7 void MovePecaX (TipoPeca * *targetPeca*, int *inputX*)

Função que move uma peça verticalmente.

Parâmetros

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

2.4.3.8 void MovePecaY (TipoPeca * *targetPeca*, int *inputY*)

Função que move uma peça horizontalmente.

Parâmetros

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

2.4.3.9 TipoTela PecaGetBloco (TipoPeca * peca, int y, int x)

Função que retorna uma peça e sua posição.

Parâmetros

<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.

Retorna

a matriz da peça.

2.4.3.10 int PecaGetCor (TipoPeca * peca)

Função que retorna a cor de uma peça.

Parâmetros

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

Retorna

um inteiro que indica a cor da peça.

2.4.3.11 int PecaGetSpeed (TipoPeca * peca)

Função que retorna a velocidade de queda da peça.

Parâmetros

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

Retorna

um inteiro com a velocidade da peça.

2.4.3.12 int PecaGetX (TipoPeca * peca)

Função que retorna a posição vertical da peça.

Parâmetros

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

Retorna

um inteiro com a posição vertical da peça.

2.4.3.13 int PecaGetY (TipoPeca * *peca*)

Função que retorna a posição horizontal da peça.

Parâmetros

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

Retorna

um inteiro com a posição horizontal da peça.

2.4.3.14 void RotacionaPeca (TipoPeca * *peca*)

Função que rotaciona a peça.

Parâmetros

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

2.4.3.15 void SpeedUp (TipoPeca * *peca*, int *multiplicador*)

Função que incrementa a velocidade de queda da peça.

Parâmetros

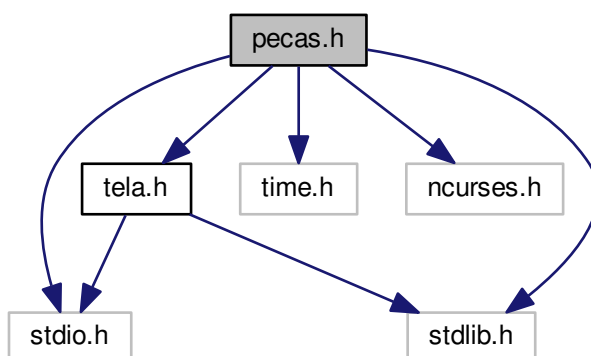
<i>*peça</i>	ponteiro para a peça atual.
<i>multiplicador</i>	inteiro que representa o quanto a velocidade de queda da peça será incrementada.

2.5 Referência do Arquivo pecas.h

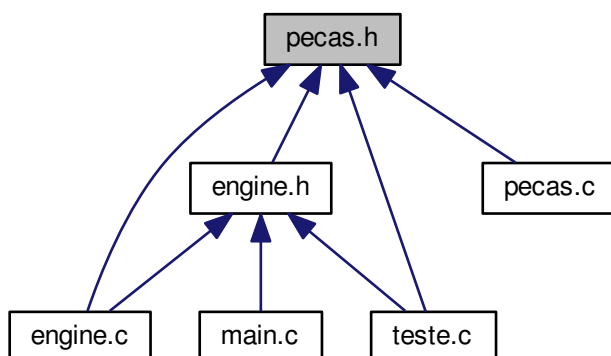
Faz a definição do módulo de peças.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <ncurses.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:



Componentes

- struct [TipoPeca](#)

Definições e Macros

- #define [TAMANHO_X](#) 5
- #define [TAMANHO_Y](#) 5

Funções

- `TipoPeca * AlocaPeca ()`
Função que aloca memória para uma peça.
- `void LiberaPeca (TipoPeca *)`
Função que desaloca a memória de uma peça.
- `void CopiaPeca (TipoPeca *, TipoPeca *)`
Função que copia uma peça.
- `void GeraPeca (TipoPeca *)`
Função que gera uma peça aleatoriamente.
- `void GeraPecaEspecific (TipoPeca *, int, int)`
Função que gera uma peça específica.
- `void MovePecaX (TipoPeca *, int)`
Função que move uma peça verticalmente.
- `void MovePecaY (TipoPeca *, int)`
Função que move uma peça horizontalmente.
- `void RotacionaPeca (TipoPeca *)`
Função que rotaciona a peça.
- `void SpeedUp (TipoPeca *, int)`
Função que incrementa a velocidade de queda da peça.
- `int PecaGetCor (TipoPeca *)`
Função que retorna a cor de uma peça.
- `int PecaGetX (TipoPeca *)`
Função que retorna a posição vertical da peça.
- `int PecaGetY (TipoPeca *)`
Função que retorna a posição horizontal da peça.
- `int PecaGetSpeed (TipoPeca *)`
Função que retorna a velocidade de queda da peça.
- `TipoTela PecaGetBloco (TipoPeca *, int, int)`
Função que retorna uma peça e sua posição.

2.5.1 Descrição Detalhada

Faz a definição do módulo de peças.

Define as estruturas do módulo e os protótipos de funções.

Autor

Anderson

Desde

11/04/16

Versão

2.6

2.5.2 Definições e macros

2.5.2.1 `#define TAMANHO_X 5`

2.5.2.2 `#define TAMANHO_Y 5`

2.5.3 Funções

2.5.3.1 `TipoPeca* AlocaPeca ()`

Função que aloca memória para uma peça.

Retorna

a nova peça.

2.5.3.2 `void CopiaPeca (TipoPeca * pecaln, TipoPeca * pecaOut)`

Função que copia uma peça.

Parâmetros

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

2.5.3.3 `void GeraPeca (TipoPeca * ponteiroPeca)`

Função que gera uma peça aleatoriamente.

Parâmetros

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

2.5.3.4 `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.

Parâmetros

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

2.5.3.5 void LiberaPeca (TipoPeca *)

Função que desaloca a memória de uma peça.

2.5.3.6 void MovePecaX (TipoPeca * *targetPeca*, int *inputX*)

Função que move uma peça verticalmente.

Parâmetros

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

2.5.3.7 void MovePecaY (TipoPeca * *targetPeca*, int *inputY*)

Função que move uma peça horizontalmente.

Parâmetros

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

2.5.3.8 TipoTela PecaGetBloco (TipoPeca * *peca*, int *y*, int *x*)

Função que retorna uma peça e sua posição.

Parâmetros

<i>*peca</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.

Retorna

a matriz da peça.

2.5.3.9 int PecaGetCor (TipoPeca * *peca*)

Função que retorna a cor de uma peça.

Parâmetros

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

Retorna

um inteiro que indica a cor da peça.

2.5.3.10 int PecaGetSpeed (TipoPeca * peca)

Função que retorna a velocidade de queda da peça.

Parâmetros

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

Retorna

um inteiro com a velocidade da peça.

2.5.3.11 int PecaGetX (TipoPeca * peca)

Função que retorna a posição vertical da peça.

Parâmetros

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

Retorna

um inteiro com a posição vertical da peça.

2.5.3.12 int PecaGetY (TipoPeca * peca)

Função que retorna a posição horizontal da peça.

Parâmetros

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

Retorna

um inteiro com a posição horizontal da peça.

2.5.3.13 void RotacionaPeca (TipoPeca * peca)

Função que rotaciona a peça.

Parâmetros

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

2.5.3.14 void SpeedUp (TipoPeca * *peca*, int *multiplicador*)

Função que incrementa a velocidade de queda da peça.

Parâmetros

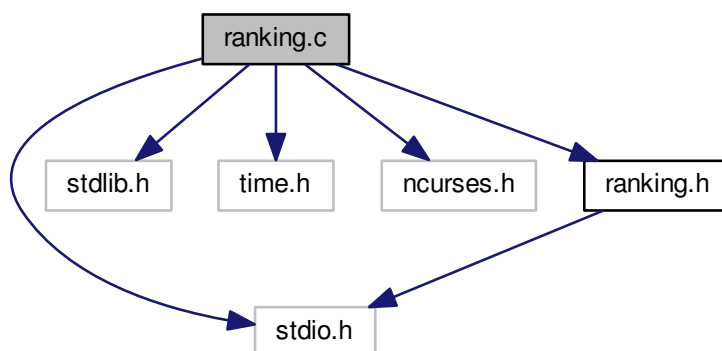
<i>*peça</i>	ponteiro para a peça atual.
<i>multiplicador</i>	inteiro que representa o quanto a velocidade de queda da peça será incrementada.

2.6 Referência do Arquivo ranking.c

Gerencia a criação e manipulação do placar.

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

Gráfico de dependência de inclusões para ranking.c:



Funções

- int * [ReceberApelido](#) ()
Função que recebe o apelido do jogador.
- void [ConverterApelido](#) (int *apelido, [TipoJogador](#) *jogador)

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

 - void **ReceberData** (**TipoJogador** *jogador)

Função que recebe a data em que o jogo foi executado.
- int **VerificaPlacar** ()

Função que verifica a existência de um arquivo com o placar.
- void **CriaPlacar** (**TipoJogador** jogador)

Função que cria um arquivo com o placar.
- void **AtualizaPlacar** (**TipoJogador** jogador)

Função que atualiza o placar.

2.6.1 Descrição Detalhada

Gerencia a criação e manipulação do placar.

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

Autor

André

Desde

11/04/16

Versão

2.1

2.6.2 Funções

2.6.2.1 void AtualizaPlacar (**TipoJogador** jogador)

Função que atualiza o placar.

Parâmetros

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

2.6.2.2 void ConverterApelido (int * *apelido*, **TipoJogador** * *jogador*)

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

Parâmetros

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

2.6.2.3 void CriaPlacar (TipoJogador *jogador*)

Função que cria um arquivo com o placar.

Parâmetros

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

2.6.2.4 int* ReceberApelido ()

Função que recebe o apelido do jogador.

Retorna

um ponteiro para o apelido.

2.6.2.5 void ReceberData (TipoJogador * *jogador*)

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

Parâmetros

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

2.6.2.6 int VerificaPlacar ()

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

Retorna

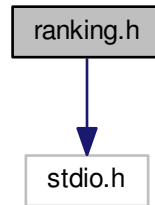
1 ou 0, caso exista ou não um arquivo, respectivamente.

2.7 Referência do Arquivo ranking.h

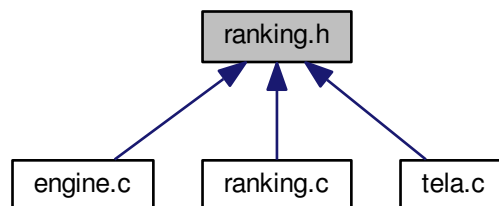
Define o módulo do placar do jogo Este módulo define as estruturas do módulo e os protótipos de funções.

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

Gráfico de dependência de inclusões para ranking.h:



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



Componentes

- struct [TipoJogador](#)

Definições e Macros

- #define [ARQUIVO](#) "ranking.txt"
- #define [AUX](#) "aux.txt"

Funções

- int * [ReceberApelido](#) ()
Função que recebe o apelido do jogador.
- void [ConverterApelido](#) (int *, [TipoJogador](#) *)
Função que converte o apelido de um vetor de caracteres para um vetor de inteiros.
- void [ReceberData](#) ([TipoJogador](#) *)
Função que recebe a data em que o jogo foi executado.

- int [VerificaPlacar](#) ()
Função que verifica a existência de um arquivo com o placar.
- void [CriaPlacar](#) ([TipoJogador](#))
Função que cria um arquivo com o placar.
- void [AtualizaPlacar](#) ([TipoJogador](#))
Função que atualiza o placar.

2.7.1 Descrição Detalhada

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

Autor

André

Desde

11/04/16

Versão

2.1

2.7.2 Definições e macros

2.7.2.1 `#define ARQUIVO "ranking.txt"`

2.7.2.2 `#define AUX "aux.txt"`

2.7.3 Funções

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

Função que atualiza o placar.

Parâmetros

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

2.7.3.2 void [ConverterApelido](#) (int * *apelido*, [TipoJogador](#) * *jogador*)

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

Parâmetros

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

2.7.3.3 void CriaPlacar (TipoJogador *jogador*)

Função que cria um arquivo com o placar.

Parâmetros

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

2.7.3.4 int* ReceberApelido ()

Função que recebe o apelido do jogador.

Retorna

um ponteiro para o apelido.

2.7.3.5 void ReceberData (TipoJogador * *jogador*)

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

Parâmetros

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

2.7.3.6 int VerificaPlacar ()

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

Retorna

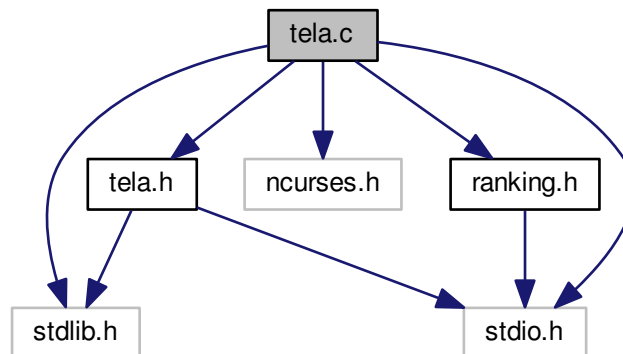
1 ou 0, caso exista ou não um arquivo, respectivamente.

2.8 Referência do Arquivo tela.c

Gerencia a criação e manipulação da tela.

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

Gráfico de dependência de inclusões para tela.c:



Definições e Macros

- `#define BLOCO '0'`

Funções

- void **SetPecaCor** (**TipoTela** *pecaDada, int corAferida)
Função que muda a cor de uma peça.
- void **SetPecaCaractere** (**TipoTela** *pecaDada, char caractereAferido)
Função que indica o caractere a ser usado em uma peça.
- void **SetPeca** (**TipoTela** *pecaDada, char caractereAferido, int corAferida)
Função que faz as alterações em uma peça.
- int **VerificaSeBloco** (**TipoTela** pecaDada)
Função que verifica se uma peça tem o símbolo do fundo.
- void **InicioTela** ()
Função que gera e mostra a tela de início.
- void **CriarTela** (**TipoTela** tela[][25])
Função que cria a tela vazia.
- void **MostrarTela** (**TipoTela** tela[][25], int pontos, int tempoDecorrido)
Função que mostra a tela.
- void **FimTela** (int pontos)
Função que gera e mostra a tela de fim de jogo.

2.8.1 Descrição Detalhada

Gerencia a criação e manipulação da tela.

Este módulo implementa a tela do jogo, bem como gerencia as operações relacionadas a ela.

Autor

André

Desde

11/04/16

Versão

2.3

2.8.2 Definições e macros**2.8.2.1 #define BLOCO '0'****2.8.3 Funções****2.8.3.1 void CriarTela (TipoTela *tela*[][25])**

Função que cria a tela vazia.

Parâmetros

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

2.8.3.2 void FimTela (int *pontos*)

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

2.8.3.3 void InicioTela ()

Função que gera e mostra a tela de início.

2.8.3.4 void MostrarTela (TipoTela *tela*[][25], int *pontos*, int *tempoDecorrido*)

Função que mostra a tela.

Parâmetros

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

2.8.3.5 void SetPeca (TipoTela * pecaDada, char caractereAferido, int corAferida)

Função que faz as alterações em uma peça.

Parâmetros

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

2.8.3.6 void SetPecaCaractere (TipoTela * pecaDada, char caractereAferido)

Função que indica o caractere a ser usado em uma peça.

Parâmetros

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

2.8.3.7 void SetPecaCor (TipoTela * pecaDada, int corAferida)

Função que muda a cor de uma peça.

Parâmetros

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

2.8.3.8 int VerificaSeBloco (TipoTela pecaDada)

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

Parâmetros

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

Retorna

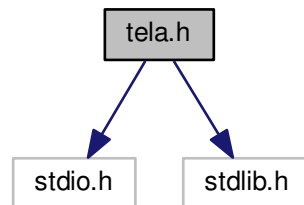
0 ou 1.

2.9 Referência do Arquivo tela.h

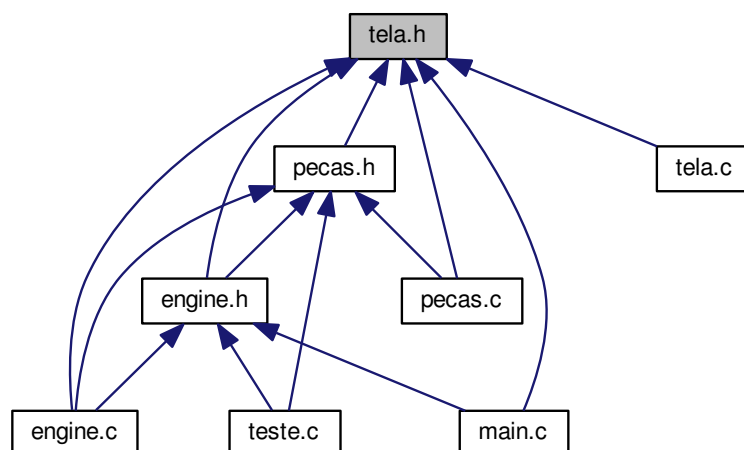
Faz a definição do módulo da tela.

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

Gráfico de dependência de inclusões para tela.h:



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



Componentes

- struct [TipoTela](#)

Funções

- void [InicioTela](#) ()
Função que gera e mostra a tela de início.
- void [CriarTela](#) ([TipoTela](#)[][25])
Função que cria a tela vazia.
- void [MostrarTela](#) ([TipoTela](#)[][25], int, int)

Função que mostra a tela.

- void **SetPecaCor** (**TipoTela** *, int)

Função que muda a cor de uma peça.

- void **SetPecaCaractere** (**TipoTela** *, char)

Função que indica o caractere a ser usado em uma peça.

- void **SetPeca** (**TipoTela** *, char, int)

Função que faz as alterações em uma peça.

- void **FimTela** (int)

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

- int **VerificaSeBloco** (**TipoTela**)

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

2.9.1 Descrição Detalhada

Faz a definição do módulo da tela.

Define as estruturas do módulo e os protótipos de funções.

Autor

André

Desde

11/04/16

Versão

2.3

2.9.2 Funções

2.9.2.1 void CriarTela (TipoTela tela[][25])

Função que cria a tela vazia.

Parâmetros

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

2.9.2.2 void FimTela (int)

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

2.9.2.3 void InicioTela ()

Função que gera e mostra a tela de início.

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

Função que mostra a tela.

Parâmetros

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

2.9.2.5 void SetPeca (TipoTela * pecaDada, char caractereAferido, int corAferida)

Função que faz as alterações em uma peça.

Parâmetros

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

2.9.2.6 void SetPecaCaractere (TipoTela * pecaDada, char caractereAferido)

Função que indica o caractere a ser usado em uma peça.

Parâmetros

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

2.9.2.7 void SetPecaCor (TipoTela * pecaDada, int corAferida)

Função que muda a cor de uma peça.

Parâmetros

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

2.9.2.8 int VerificaSeBloco (TipoTela pecaDada)

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

Parâmetros

*pecaDada	ponteiro para peça.
-----------	---------------------

Retorna

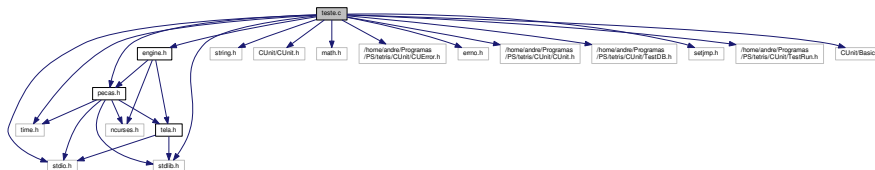
0 ou 1.

2.10 Referência do Arquivo teste.c

Gerencia os testes de funcionamento do programa.

```
#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"
```

Gráfico de dependência de inclusões para teste.c:



Funções

- void [TesteAlocaPeca](#) (void)
Função que testa a alocação de peças.
- void [TesteCorPecaCorFundo](#) (void)
Função que testa as cores das peças.
- void [TesteCorPecaDiferente](#) (void)
Função que testa se as cores das peças geradas são diferentes a cada rodada.
- void [TestaMorte](#) (void)
Função que testa se o jogo acaba com a chegada no limite.
- void [TestaColisaoParede](#) (void)
Função que testa a colisão com as paredes da tela.
- void [TestaColisaoBloco](#) (void)
Função que testa a colisão da peça com os blocos da tela.
- void [TesteLimpaLinha](#) (void)
Função que testa se uma linha completa é limpa.
- void [TestaPecaNoTopo](#) (void)
Função que testa a inclusão de uma nova peça na tela.
- void [AdicionarSuite](#) (void)
Função que cria a suíte de testes.
- int [main](#) ()

2.10.1 Descrição Detalhada

Gerencia os testes de funcionamento do programa.

Autor

Cristóvão

Desde

11/04/16

Versão

2.6

2.10.2 Funções

2.10.2.1 void AdicionarSuite (void)

Função que cria a suíte de testes.

2.10.2.2 int main ()

2.10.2.3 void TestaColisaoBloco (void)

Função que testa a colisão da peça com os blocos da tela.

2.10.2.4 void TestaColisaoParede (void)

Função que testa a colisão com as paredes da tela.

2.10.2.5 void TestaMorte (void)

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

2.10.2.6 void TestaPecaNoTopo (void)

Função que testa a inclusão de uma nova peça na tela.

2.10.2.7 void TesteAlocaPeca (void)

Função que testa a alocação de peças.

2.10.2.8 void TesteCorPecaCorFundo (void)

Função que testa as cores das peças.

2.10.2.9 void TesteCorPecaDiferente (void)

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

2.10.2.10 void TesteLimpaLinha (void)

Função que testa se uma linha completa é limpada.

Índice Remissivo

ARQUIVO

ranking.h, 33

AUX

ranking.h, 33

AddBloco

engine.c, 7

engine.h, 13

AdicionarSuite

teste.c, 42

AlocaPeca

pecas.c, 20

pecas.h, 26

ano

TipoJogador, 1

apelido

TipoJogador, 1

AtualizaPlacar

ranking.c, 30

ranking.h, 33

B

pecas.c, 20

BAIXO

engine.c, 7

BLOCO

engine.c, 7

tela.c, 36

CIMA

engine.c, 7

COR_FUNDO

pecas.c, 20

CORFUNDO

engine.c, 7

CORVERMELHO

engine.c, 7

com

engine.c, 11

comandos

engine.c, 7

ConverterApelido

ranking.c, 30

ranking.h, 33

CopiaDaMatrizDePecas

pecas.c, 20

CopiaPeca

pecas.c, 20

pecas.h, 26

cor

TipoTela, 3

CriaPlacar

ranking.c, 31

ranking.h, 34

CriarTela

tela.c, 36

tela.h, 39

DIREITA

engine.c, 7

DeletaBloco

engine.c, 8

engine.h, 13

DeletaLinhas

engine.c, 8

engine.h, 13

dia

TipoJogador, 1

ESQUERDA

engine.c, 7

engine.c, 5

AddBloco, 7

BAIXO, 7

BLOCO, 7

CIMA, 7

CORFUNDO, 7

CORVERMELHO, 7

com, 11

comandos, 7

DIREITA, 7

DeletaBloco, 8

DeletaLinhas, 8

ESQUERDA, 7

Gravidade, 8

Loop, 8

PegaInput, 9

PoePecaNoTopo, 9

RemoveBloco, 9

SAIR, 7

TAMANHOTELAX, 7

TAMANHOTELAY, 7

TEST_MODE, 7

Temporizador, 9

VAZIO, 7

VerificaColisao, 10

VerificaLinhas, 10

VerificaMorte, 10

engine.h, 11

AddBloco, 13

DeletaBloco, 13

- DeletaLinhas, 13
- Gravidade, 14
- Loop, 14
- PegaInput, 14
- PoePecaNoTopo, 14
- RemoveBloco, 15
- TAMANHOTELAX, 13
- TAMANHOTELAY, 13
- VerificaColisao, 15
- VerificaLinhas, 15
- VerificaMorte, 16
- FimTela
 - tela.c, 36
 - tela.h, 39
- GeraPeca
 - pecas.c, 20
 - pecas.h, 26
- GeraPecaEspecificas
 - pecas.c, 21
 - pecas.h, 26
- Gravidade
 - engine.c, 8
 - engine.h, 14
- InicioTela
 - tela.c, 36
 - tela.h, 39
- LiberaPeca
 - pecas.c, 21
 - pecas.h, 27
- Loop
 - engine.c, 8
 - engine.h, 14
- main
 - main.c, 18
 - teste.c, 42
- main.c, 16
 - main, 18
- mes
 - TipoJogador, 1
- MostrarTela
 - tela.c, 36
 - tela.h, 39
- MovePecaX
 - pecas.c, 21
 - pecas.h, 27
- MovePecaY
 - pecas.c, 21
 - pecas.h, 27
- NUM_PECAS
 - pecas.c, 20
- peca
 - TipoTela, 3
- PecaGetBloco
 - pecas.c, 21
 - pecas.h, 27
- PecaGetCor
 - pecas.c, 22
 - pecas.h, 27
- PecaGetSpeed
 - pecas.c, 22
 - pecas.h, 28
- PecaGetX
 - pecas.c, 22
 - pecas.h, 28
- PecaGetY
 - pecas.c, 23
 - pecas.h, 28
- pecas
 - TipoPeca, 2
- pecas.c, 18
 - AlocaPeca, 20
 - B, 20
 - COR_FUNDO, 20
 - CopiaDaMatrizDePecas, 20
 - CopiaPeca, 20
 - GeraPeca, 20
 - GeraPecaEspecificas, 21
 - LiberaPeca, 21
 - MovePecaX, 21
 - MovePecaY, 21
 - NUM_PECAS, 20
 - PecaGetBloco, 21
 - PecaGetCor, 22
 - PecaGetSpeed, 22
 - PecaGetX, 22
 - PecaGetY, 23
 - RotacionaPeca, 23
 - SpeedUp, 23
 - TAMANHO_X, 20
 - TAMANHO_Y, 20
 - V, 20
- pecas.h, 23
 - AlocaPeca, 26
 - CopiaPeca, 26
 - GeraPeca, 26
 - GeraPecaEspecificas, 26
 - LiberaPeca, 27
 - MovePecaX, 27
 - MovePecaY, 27
 - PecaGetBloco, 27
 - PecaGetCor, 27
 - PecaGetSpeed, 28
 - PecaGetX, 28
 - PecaGetY, 28
 - RotacionaPeca, 28
 - SpeedUp, 29
 - TAMANHO_X, 26
 - TAMANHO_Y, 26
- PegaInput
 - engine.c, 9
 - engine.h, 14

- PoePecaNoTopo
 - engine.c, [9](#)
 - engine.h, [14](#)
- pontos
 - TipoJogador, [1](#)
- ranking.c, [29](#)
 - AtualizaPlacar, [30](#)
 - ConverterApelido, [30](#)
 - CriaPlacar, [31](#)
 - ReceberApelido, [31](#)
 - ReceberData, [31](#)
 - VerificaPlacar, [31](#)
- ranking.h, [31](#)
 - ARQUIVO, [33](#)
 - AUX, [33](#)
 - AtualizaPlacar, [33](#)
 - ConverterApelido, [33](#)
 - CriaPlacar, [34](#)
 - ReceberApelido, [34](#)
 - ReceberData, [34](#)
 - VerificaPlacar, [34](#)
- ReceberApelido
 - ranking.c, [31](#)
 - ranking.h, [34](#)
- ReceberData
 - ranking.c, [31](#)
 - ranking.h, [34](#)
- RemoveBloco
 - engine.c, [9](#)
 - engine.h, [15](#)
- RotacionalPeca
 - pecas.c, [23](#)
 - pecas.h, [28](#)
- SAIR
 - engine.c, [7](#)
- SetPeca
 - tela.c, [36](#)
 - tela.h, [40](#)
- SetPecaCaractere
 - tela.c, [37](#)
 - tela.h, [40](#)
- SetPecaCor
 - tela.c, [37](#)
 - tela.h, [40](#)
- speed
 - TipoPeca, [2](#)
- SpeedUp
 - pecas.c, [23](#)
 - pecas.h, [29](#)
- TAMANHO_X
 - pecas.c, [20](#)
 - pecas.h, [26](#)
- TAMANHO_Y
 - pecas.c, [20](#)
 - pecas.h, [26](#)
- TAMANHOTELAX
 - engine.c, [7](#)
 - engine.h, [13](#)
- TAMANHOTELAY
 - engine.c, [7](#)
 - engine.h, [13](#)
- TEST_MODE
 - engine.c, [7](#)
- tela.c, [34](#)
 - BLOCO, [36](#)
 - CriarTela, [36](#)
 - FimTela, [36](#)
 - InicioTela, [36](#)
 - MostrarTela, [36](#)
 - SetPeca, [36](#)
 - SetPecaCaractere, [37](#)
 - SetPecaCor, [37](#)
 - VerificaSeBloco, [37](#)
- tela.h, [37](#)
 - CriarTela, [39](#)
 - FimTela, [39](#)
 - InicioTela, [39](#)
 - MostrarTela, [39](#)
 - SetPeca, [40](#)
 - SetPecaCaractere, [40](#)
 - SetPecaCor, [40](#)
 - VerificaSeBloco, [40](#)
- tempo
 - TipoJogador, [1](#)
- Temporizador
 - engine.c, [9](#)
- TestaColisaoBloco
 - teste.c, [42](#)
- TestaColisaoParede
 - teste.c, [42](#)
- TestaMorte
 - teste.c, [42](#)
- TestaPecaNoTopo
 - teste.c, [42](#)
- teste.c, [41](#)
 - AdicionarSuite, [42](#)
 - main, [42](#)
 - TestaColisaoBloco, [42](#)
 - TestaColisaoParede, [42](#)
 - TestaMorte, [42](#)
 - TestaPecaNoTopo, [42](#)
 - TesteAlocaPeca, [42](#)
 - TesteCorPecaCorFundo, [42](#)
 - TesteCorPecaDiferente, [42](#)
 - TesteLimpaLinha, [42](#)
- TesteAlocaPeca
 - teste.c, [42](#)
- TesteCorPecaCorFundo
 - teste.c, [42](#)
- TesteCorPecaDiferente
 - teste.c, [42](#)
- TesteLimpaLinha
 - teste.c, [42](#)
- TipoJogador, [1](#)

- ano, [1](#)
- apelido, [1](#)
- dia, [1](#)
- mes, [1](#)
- pontos, [1](#)
- tempo, [1](#)
- TipoPeca, [2](#)
 - pecas, [2](#)
 - speed, [2](#)
 - x, [2](#)
 - y, [2](#)
- TipoTela, [2](#)
 - cor, [3](#)
 - peca, [3](#)
- V
 - pecas.c, [20](#)
- VAZIO
 - engine.c, [7](#)
- VerificaColisao
 - engine.c, [10](#)
 - engine.h, [15](#)
- VerificaLinhas
 - engine.c, [10](#)
 - engine.h, [15](#)
- VerificaMorte
 - engine.c, [10](#)
 - engine.h, [16](#)
- VerificaPlacar
 - ranking.c, [31](#)
 - ranking.h, [34](#)
- VerificaSeBloco
 - tela.c, [37](#)
 - tela.h, [40](#)
- x
 - TipoPeca, [2](#)
- y
 - TipoPeca, [2](#)