Jogo da velha

Generated by Doxygen 1.8.11

Contents

Chapter 1

Main Page

Código fonte do jogo da velha.

Esse código, juntamente com a Main, representa uma série de funções que controlam o jogo da velha!

Author

Lucas Ribeiro

Date

25/05/2017

Chapter 2

README

Jogo da Velha I.A

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

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

Chapter 5

Data Structure Documentation

5.1 node Struct Reference

Collaboration diagram for node:

Data Fields

- int chave
- int jogador
- int altura
- struct node * esq
- struct node * dir

5.1.1 Field Documentation

5.1.1.1 int altura

Variável inteiro que armazena a alteura do nó na árvore

5.1.1.2 int chave

Variável inteira que armazena uma chave para reconhecer o nó

5.1.1.3 struct node* dir

Ponteiro que aponta para o nó á direita do nó

5.1.1.4 struct node* esq

Ponteiro que aponta para o nó á esquerda do nó

5.1.1.5 int jogador

Variável inteiro que armazena qual jogado fez a jogada

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

- avl_jogo_da_velha.h

5.2 struct Struct Reference

Essa estrutura representa um nó que irá formar a árvore AVL.

5.2.1 Detailed Description

Essa estrutura representa um nó que irá formar a árvore AVL.

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

- avl_jogo_da_velha.h

Chapter 6

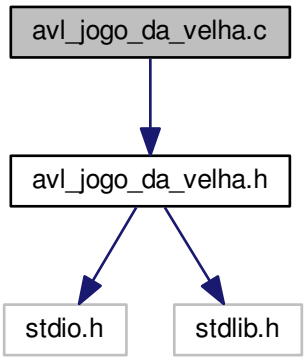
File Documentation

6.1 avl_jogo_da_velha.c File Reference

Implementação das funções descritas na biblioteca avl_jogo_da_velha.h.

```
#include "avl_jogo_da_velha.h"
```

Include dependency graph for avl_jogo_da_velha.c:



Functions

- void **clear** ()
- int maior (int esq, int dir)

  Função que ao receber 2 inteiro, retorna o maior deles.
- int altura (Arvore *a)

  Função retorna a altura de uma árvore.
- int atualizar_altura (Arvore *a)

  Função recursiva que atualiza as altura da árvore.
- int balanceamento (Arvore *a)

Função que retorna a diferença das alturas do nós.

- Arvore ∗ rotacao_simples_esq (Arvore ∗r)

  Função que realiza uma rotação simples a esquerda na AVL.
- Arvore ∗ rotacao_simples_dir (Arvore ∗r)

  Função que realiza uma rotação simples a direita na AVL.
- Arvore ∗ rotacao_dupla_esq (Arvore ∗r)

  Função que realiza uma rotação dupla a esquerda na AVL.
- Arvore ∗ rotacao_dupla_dir (Arvore ∗r)

  Função que realiza uma rotação dupla a direita na AVL.
- Arvore ∗ atualizar_fb_dir (Arvore ∗r)
- Arvore ∗ atualizar_fb_esq (Arvore ∗r)
- Arvore ∗ inserir (Arvore ∗a, int chave, int jogador)

  Insere um novo nó na ávore AVL.
- Arvore ∗ remover (Arvore ∗a, int chave)

  Remove um nó da árvore AVL baseado na sua chave.
- int buscar (Arvore ∗a, int chave)

  Função que encontra um nó específico baseado em sua chave e retorna o valor da variável jogador.
- void **imprimir_in_order** (Arvore ∗a, int nivel)
- void **arvore_para_vetor** (Arvore ∗a, int ∗tabuleiro)
- void **imprimir_tabuleiro** (Arvore ∗a)
- int verifica_ganhador (Arvore ∗a, int jogador)

  Essa função verifica se alguém ganhou o jogo e retorna um inteiro.
- int **conta_nos** (Arvore ∗a)
- void **calcula_passos** (Arvore ∗∗verificacao, Arvore ∗a, Arvore ∗b, int ∗passos, int index)
- Arvore ∗ **jogada_computador** (Arvore ∗a, Arvore ∗verificacao)
- int **jogada_jogador** ()
- int **menu_principal** ()

6.1.1 Detailed Description

Implementação das funções descritas na biblioteca avl_jogo_da_velha.h.

Author

Lucas Ribeiro

Date

25/05/2017

6.1.2 Function Documentation

6.1.2.1 int altura (Arvore ∗ *a*)

Função retorna a altura de uma árvore.

Precondition

O ponteiro para a árvore não deve ter valor NULL

Warning

Caso o pré requisito não seja atendito a funão retornará NULL

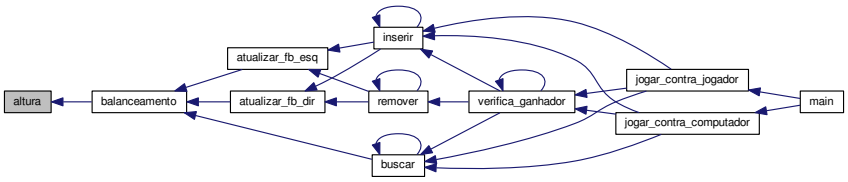
Parameters

a	Ponteiro que aponta para a árvore AVL.

Returns

Retorna um inteiro que representa

Here is the caller graph for this function:



6.1.2.2 int atualizar_altura (Arvore * a)

Função recursiva que atualiza as altura da árvore.

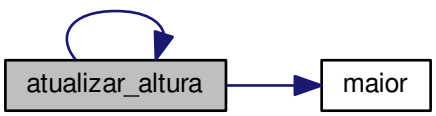
Parameters

a	Ponteiro que aponta para o início da AVL

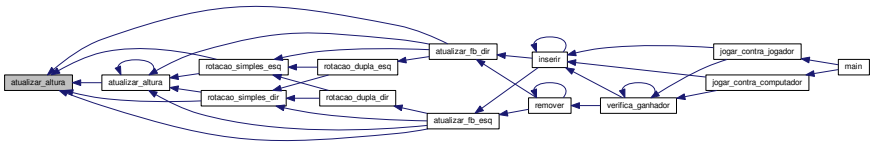
Returns

O retorno inteiro serve para o funcionamento da recursividade.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.3 Arvore * atualizar_fb_dir (Arvore * r)

Função que atualiza a altura e faz o balanceamento dos nós casos os dá direita estejam desbalanceados.

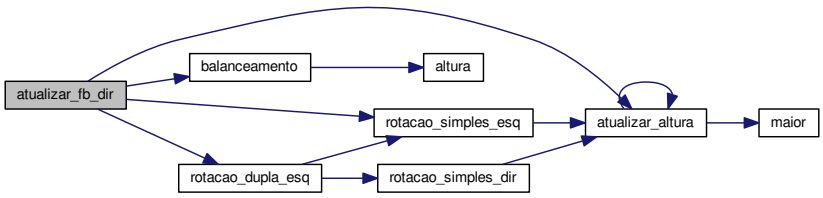
Parameters

r	Ponteiro que aponta para um nó da árvore AVL.

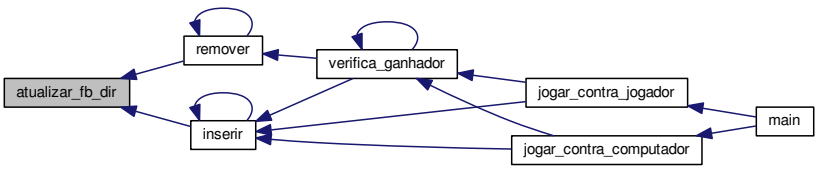
Returns

Retorna o ponteiro para o nó após as devidas atualizações.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.4 Arvore * atualizar_fb_esq (Arvore * r)

Função que atualiza a altura e faz o balancemanto dos nós casos os daá esquerda estejam desbalanceados.

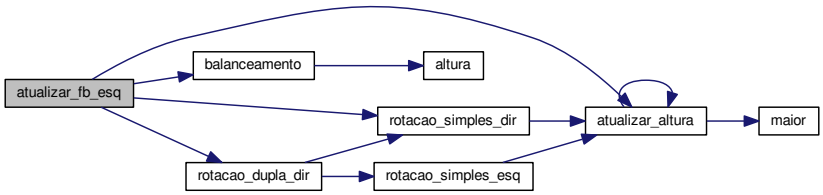
Parameters

r	Ponteiro que aponta para um nó da árvore AVL.

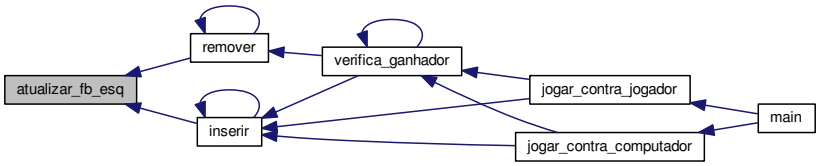
Returns

Retorna o ponteiro para o nó após as devidas alterações feitas.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.5 int balanceamento (Arvore * *a*)

Função que retorna a diferença das alturas do nós.

A função de balancemanto serve para retornar um valor que indica se o nó está balanceado ou não, retornando um valor que vária de -2 a 2.

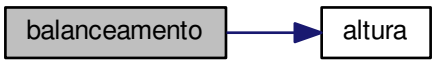
Parameters

a	Ponteiro que aponta para um nó da árvore AVL;

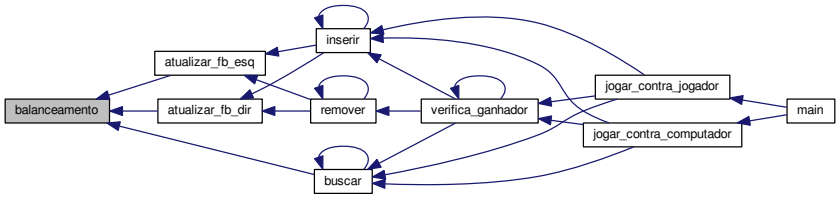
Returns

Retorna o valor do balanceamento

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.6 int buscar (Arvore * *a*, int *chave*)

Função que encontra um nó específico baseado em sua chave e retorna o valor da variável jogador.

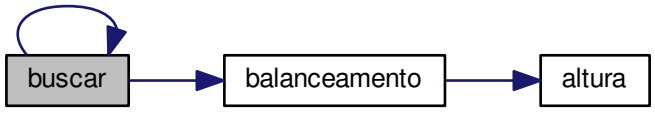
Parameters

a	Ponteiro que aponta para a ávore em que se deseja buscar.
chave	Inteiro que representa a chave do nó que se deseja achar.

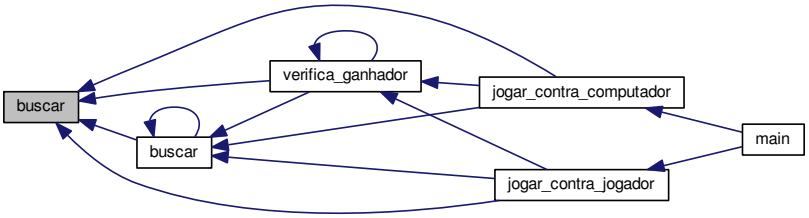
Returns

Retorna um inteiro com o mesmo valor da variável jogador do nó que se buscava.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.7 Arvore * inserir (Arvore * *a*, int *chave*, int *jogador*)

Insere um novo nó na ávore AVL.

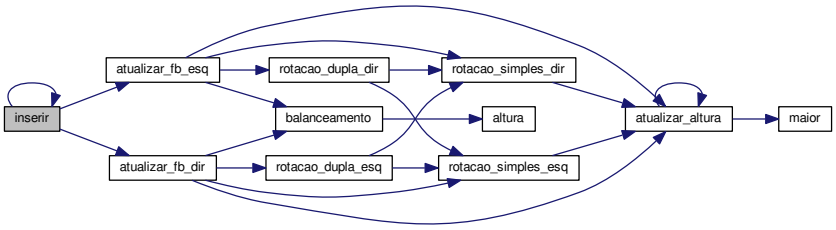
Parameters

a	É um ponteiro que representa a árvore onde se quer inserir.
chave	É um inteiro que representa o valor do nó.
jogador	É um inteiro que representa o jogador que fez a jogada.

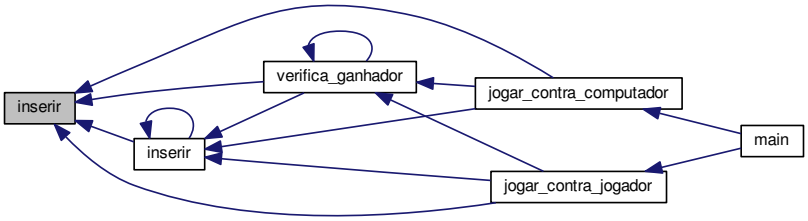
Returns

Retorna o ponteiro para árvore após a inserção ter sido feita.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.8 int maior (int *esq*, int *dir*)

Função que ao receber 2 inteiro, retorna o maior deles.

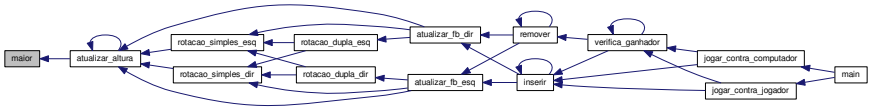
Parameters

esq	Inteiro que representa o valor a esquerda da AVL.
dir	Inteiro que representa o valor a direita da AVL.

Returns

Retorno inteiro que representa o maior de ambos os valores.

Here is the caller graph for this function:



6.1.2.9 Arvore * remover (Arvore * *a*, int *chave*)

Remove um nó da árvore AVL baseado na sua chave.

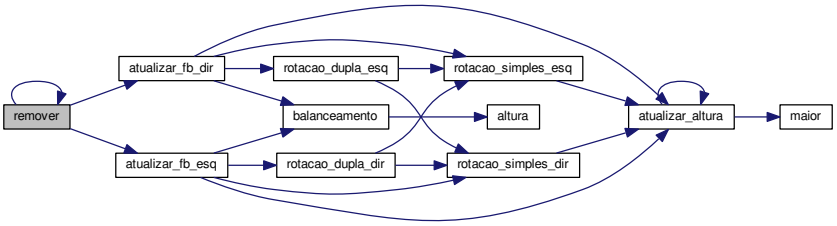
Parameters

a	É um ponteiro que aponta para a árvore ao qual se deseja remover o nó.
chave	É um inteiro que representa a chave do nó que se pretende remover

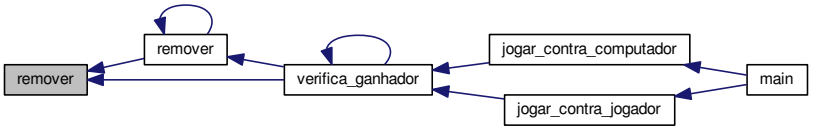
Returns

Retorna o ponteiro para a árvore, após ter sido removido o nó.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.10 Arvore * rotacao_dupla_dir (Arvore * *r*)

Função que realiza uma rotação dupla a direita na AVL.

Parameters

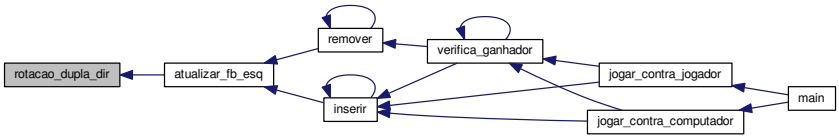
r	Ponteiro que aponta para um nó da árvore AVL.

Returns

Retorna o ponteiro do nó da AVL após a rotação feita.

Here is the call graph for this function:

Here is the caller graph for this function:



6.1.2.11 Arvore * rotacao_dupla_esq (Arvore * r)

Função que realiza uma rotação dupla a esquerda na AVL.

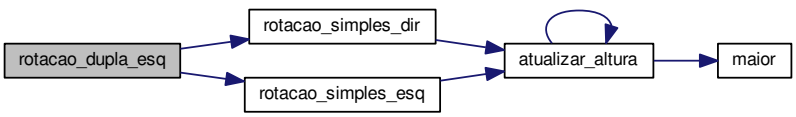
Parameters

r	Ponteiro que aponta para um nó da árvore AVL.

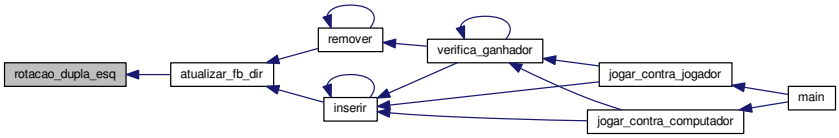
Returns

Retorna o ponteiro do nó da AVL após a rotação feita.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.12 Arvore * rotacao_simples_dir (Arvore * r)

Função que realiza uma rotação simples a direita na AVL.

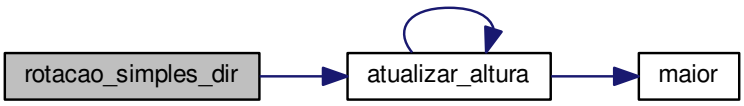
Parameters

r	Ponteiro que aponta para um nó da árvore AVL.

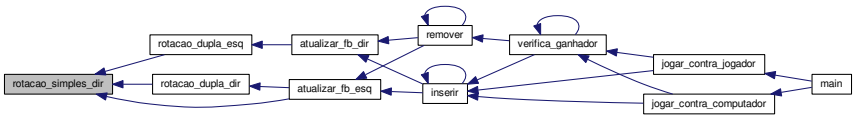
Returns

Retorna o ponteiro do nó da AVL após a rotação feita

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.13 Arvore * rotacao_simples_esq (Arvore * r)

Função que realiza uma rotação simples a esquerda na AVL.

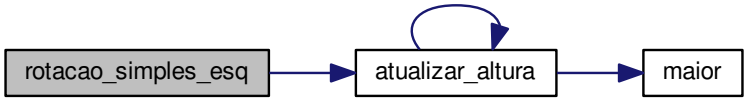
Parameters

r	Ponteiro que aponta para um nó da árvore AVL!

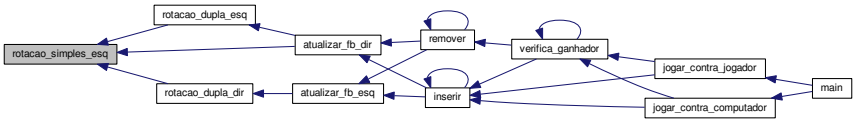
Returns

Retorna o ponteiro do nó da AVL após a rotação feita

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.14 int verifica_ganhador (Arvore * *a*, int *jogador*)

Essa função verifica se alguém ganhou o jogo e retorna um inteiro.

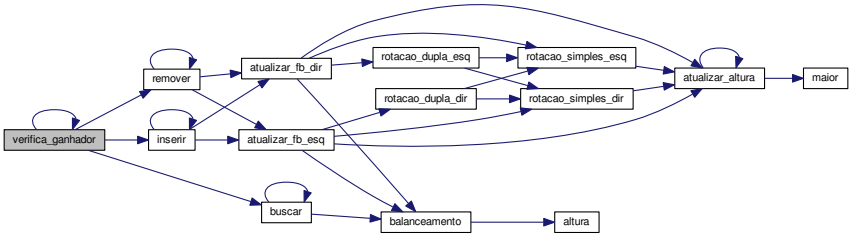
Parameters

a	É um ponteiro para a árvore AVL.
jogador	É um inteiro que representa qual jogador se está verificando.

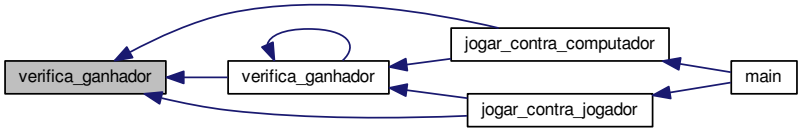
Returns

Inteiro de retorno

Here is the call graph for this function:



Here is the caller graph for this function:

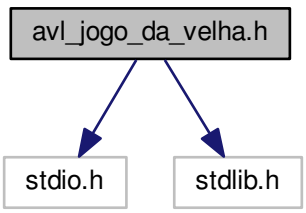


6.2 avl_jogo_da_velha.h File Reference

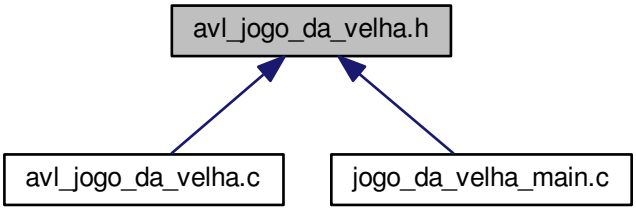
É uma biblioteca para manipulação de AVL.

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

Include dependency graph for avl_jogo_da_velha.h:



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



Data Structures

- struct node

Typedefs

- typedef struct node **No**
- typedef struct node **Arvore**

Functions

- int maior (int esq, int dir)

  Função que ao receber 2 inteiro, retorna o maior deles.
- int altura (Arvore *a)

  Função retorna a altura de uma árvore.
- int atualizar_altura (Arvore *a)

  Função recursiva que atualiza as altura da árvore.
- int balanceamento (Arvore *a)

  Função que retorna a diferença das alturas do nós.
- Arvore * rotacao_simples_esq (Arvore *r)

  Função que realiza uma rotação simples a esquerda na AVL.
- Arvore * rotacao_simples_dir (Arvore *r)

  Função que realiza uma rotação simples a direita na AVL.
- Arvore * rotacao_dupla_esq (Arvore *r)

  Função que realiza uma rotação dupla a esquerda na AVL.
- Arvore * rotacao_dupla_dir (Arvore *r)

  Função que realiza uma rotação dupla a direita na AVL.
- Arvore * atualizar_fb_dir (Arvore *r)
- Arvore * atualizar_fb_esq (Arvore *r)
- Arvore * inserir (Arvore *a, int chave, int jogador)

  Insere um novo nó na ávore AVL.
- Arvore * remover (Arvore *a, int chave)

  Remove um nó da árvore AVL baseado na sua chave.
- int buscar (Arvore *a, int chave)

Função que encontra um nó específico baseado em sua chave e retorna o valor da variável jogador.

- void **imprimir_in_order** (Arvore ∗a, int nivel)
- void **arvore_para_vetor** (Arvore ∗a, int ∗tabuleiro)
- void **imprimir_tabuleiro** (Arvore ∗a)
- int verifica_ganhador (Arvore ∗a, int jogador)

  Essa função verifica se alguém ganhou o jogo e retorna um inteiro.
- int **conta_nos** (Arvore ∗a)
- void **calcula_passos** (Arvore ∗∗verificacao, Arvore ∗a, Arvore ∗b, int ∗passos, int index)
- Arvore ∗ **jogada_computador** (Arvore ∗a, Arvore ∗verificacao)
- int **jogada_jogador** ()
- int **menu_principal** ()

6.2.1 Detailed Description

É uma biblioteca para manipulação de AVL.

Essa biblioteca para manipulação de uma AVL especial que guarda e gerência dados para o funcionamento do jogo da velha.

Author

Lucas Ribeiro

Date

25/05/2017

6.2.2 Function Documentation

6.2.2.1 int altura (Arvore ∗ *a*)

Função retorna a altura de uma árvore.

Precondition

O ponteiro para a árvore não deve ter valor NULL

Warning

Caso o pré requisito não seja atendito a funão retornará NULL

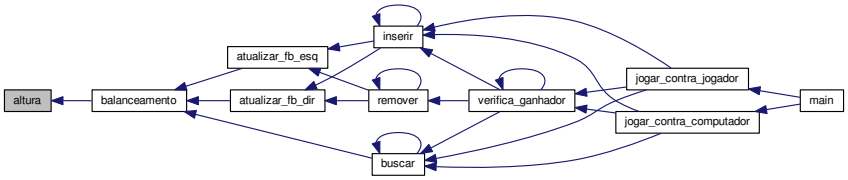
Parameters

a	Ponteiro que aponta para a árvore AVL.

Returns

Retorna um inteiro que representa

Here is the caller graph for this function:



6.2.2.2 int atualizar_altura (Arvore * a)

Função recursiva que atualiza as altura da árvore.

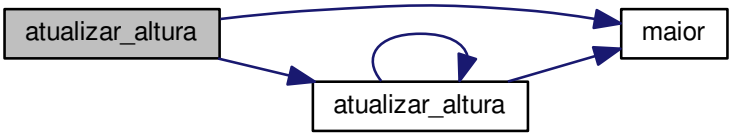
Parameters

a	Ponteiro que aponta para o início da AVL

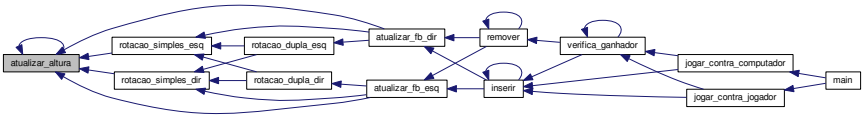
Returns

O retorno inteiro serve para o funcionamento da recursividade.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.3 Arvore* atualizar_fb_dir (Arvore * r)

Função que atualiza a altura e faz o balanceamento dos nós casos os dá direita estejam desbalanceados.

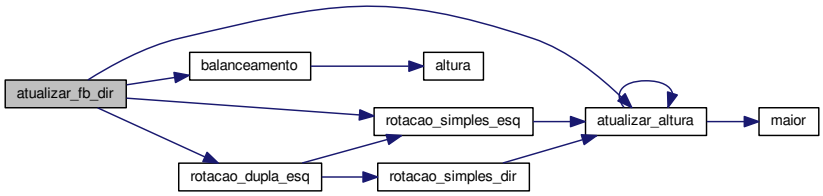
Parameters

r	Ponteiro que aponta para um nó da árvore AVL.

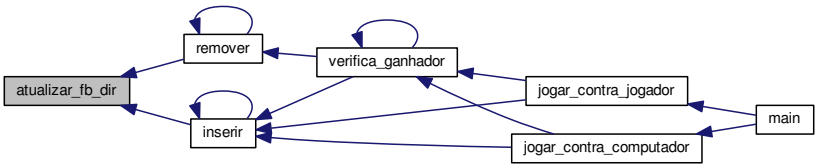
Returns

Retorna o ponteiro para o nó após as devidas atualizações.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.4 Arvore* atualizar_fb_esq (Arvore * r)

Função que atualiza a altura e faz o balancemanto dos nós casos os daá esquerda estejam desbalanceados.

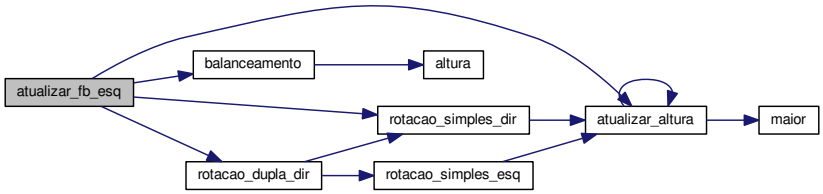
Parameters

r	Ponteiro que aponta para um nó da árvore AVL.

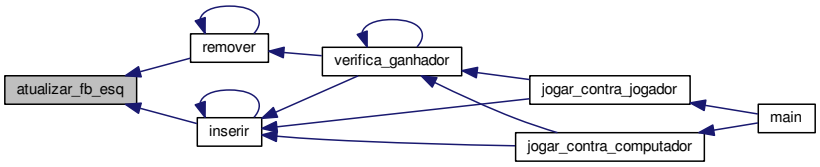
Returns

Retorna o ponteiro para o nó após as devidas alterações feitas.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.5 int balanceamento (Arvore * a)

Função que retorna a diferença das alturas do nós.

A função de balancemanto serve para retornar um valor que indica se o nó está balanceado ou não, retornando um valor que vária de -2 a 2.

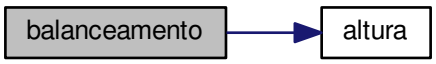
Parameters

a	Ponteiro que aponta para um nó da árvore AVL;

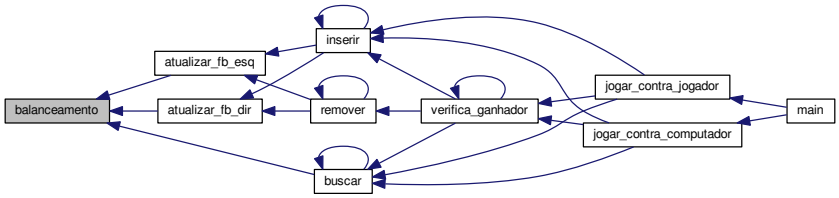
Returns

Retorna o valor do balanceamento

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.6 int buscar (Arvore * *a*, int *chave*)

Função que encontra um nó específico baseado em sua chave e retorna o valor da variável jogador.

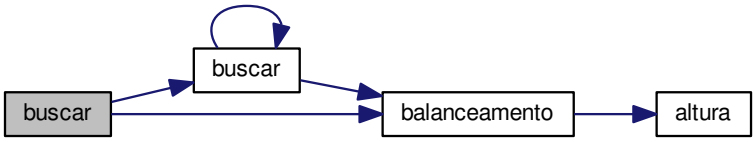
Parameters

a	Ponteiro que aponta para a ávore em que se deseja buscar.
chave	Inteiro que representa a chave do nó que se deseja achar.

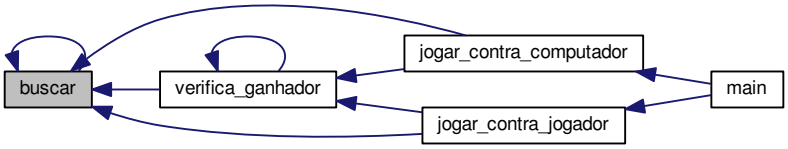
Returns

Retorna um inteiro com o mesmo valor da variável jogador do nó que se buscava.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.7 Arvore* inserir (Arvore * *a*, int *chave*, int *jogador*)

Insere um novo nó na ávore AVL.

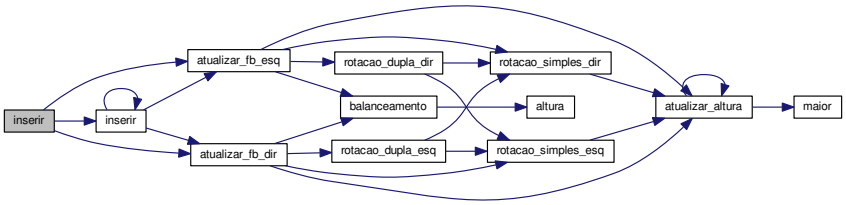
Parameters

a	É um ponteiro que representa a árvore onde se quer inserir.
chave	É um inteiro que representa o valor do nó.
jogador	É um inteiro que representa o jogador que fez a jogada.

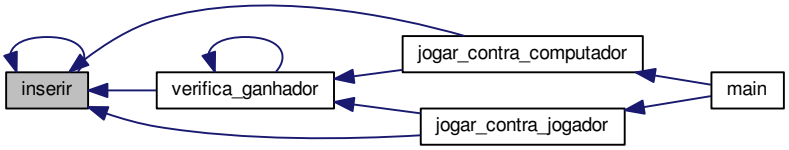
Returns

Retorna o ponteiro para árvore após a inserção ter sido feita.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.8 int maior (int *esq*, int *dir*)

Função que ao receber 2 inteiro, retorna o maior deles.

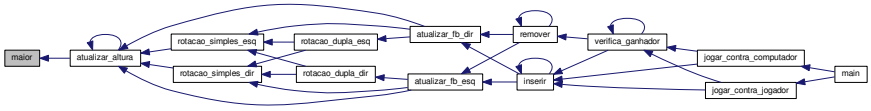
Parameters

esq	Inteiro que representa o valor a esquerda da AVL.
dir	Inteiro que representa o valor a direita da AVL.

Returns

Retorno inteiro que representa o maior de ambos os valores.

Here is the caller graph for this function:



6.2.2.9 Arvore* remover (Arvore * a, int chave)

Remove um nó da árvore AVL baseado na sua chave.

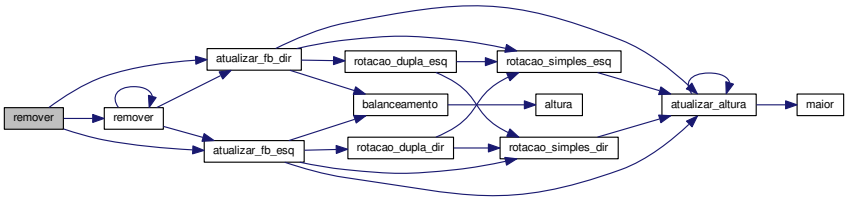
Parameters

a	É um ponteiro que aponta para a árvore ao qual se deseja remover o nó.
chave	É um inteiro que representa a chave do nó que se pretende remover

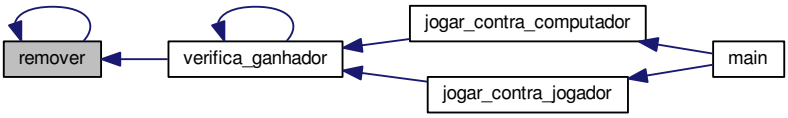
Returns

Retorna o ponteiro para a árvore, após ter sido removido o nó.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.10 Arvore* rotacao_dupla_dir (Arvore * r)

Função que realiza uma rotação dupla a direita na AVL.

Parameters

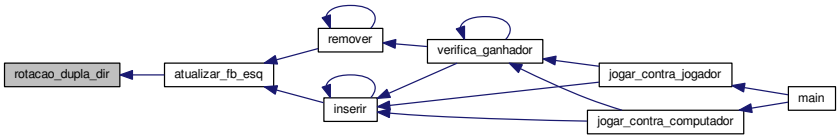
r	Ponteiro que aponta para um nó da árvore AVL.

Returns

Retorna o ponteiro do nó da AVL após a rotação feita.

Here is the call graph for this function:

Here is the caller graph for this function:



6.2.2.11 Arvore* rotacao_dupla_esq (Arvore * r)

Função que realiza uma rotação dupla a esquerda na AVL.

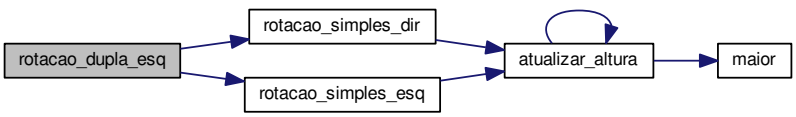
Parameters

r	Ponteiro que aponta para um nó da árvore AVL.

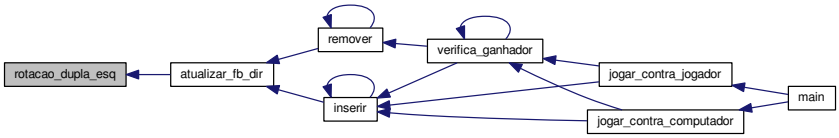
Returns

Retorna o ponteiro do nó da AVL após a rotação feita.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.12 Arvore* rotacao_simples_dir (Arvore * r)

Função que realiza uma rotação simples a direita na AVL.

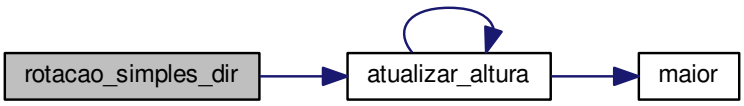
Parameters

r	Ponteiro que aponta para um nó da árvore AVL.

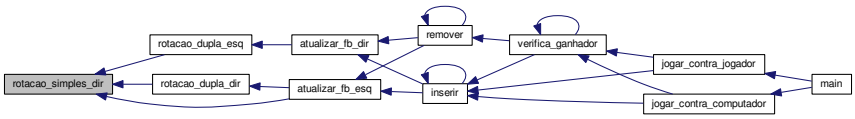
Returns

Retorna o ponteiro do nó da AVL após a rotação feita

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.13 Arvore* rotacao_simples_esq (Arvore * r)

Função que realiza uma rotação simples a esquerda na AVL.

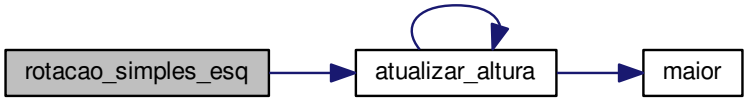
Parameters

r	Ponteiro que aponta para um nó da árvore AVL!

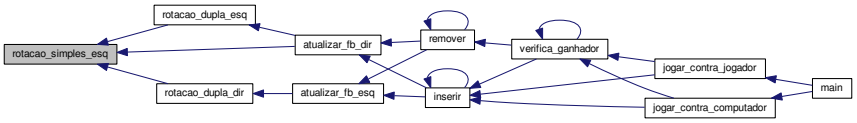
Returns

Retorna o ponteiro do nó da AVL após a rotação feita

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.2.14 int verifica_ganhador (Arvore * *a*, int *jogador*)

Essa função verifica se alguém ganhou o jogo e retorna um inteiro.

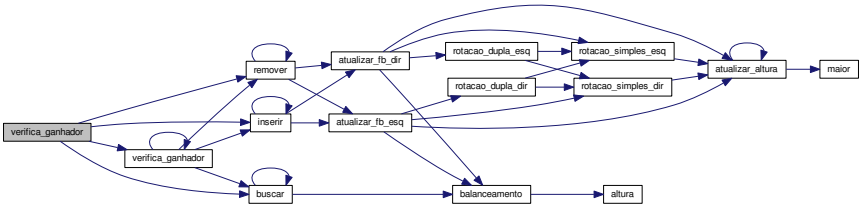
Parameters

a	É um ponteiro para a árvore AVL.
jogador	É um inteiro que representa qual jogador se está verificando.

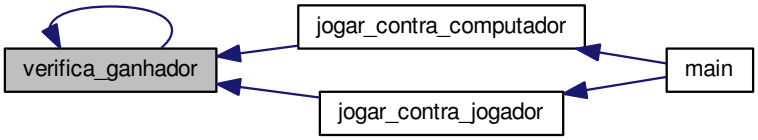
Returns

Inteiro de retorno

Here is the call graph for this function:



Here is the caller graph for this function:

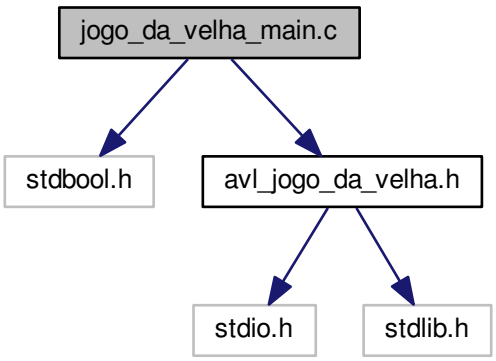


6.3 jogo_da_velha_main.c File Reference

Código fonte do jogo da velha.

```
#include <stdbool.h>
#include "avl_jogo_da_velha.h"
```

Include dependency graph for jogo_da_velha_main.c:



Functions

- void jogar_contra_computador (Arvore *a)

  Função que controla o jogo quando selecionado a opção Player vs IA.
- void jogar_contra_jogador (Arvore *a)

  Função que controla o jogo quando é selecionado a opção de Player vc Player.
- int main ()

  Função Main do programa C.

6.3.1 Detailed Description

Código fonte do jogo da velha.

Author

Lucas Ribeiro

Date

25/05/2017

6.3.2 Function Documentation

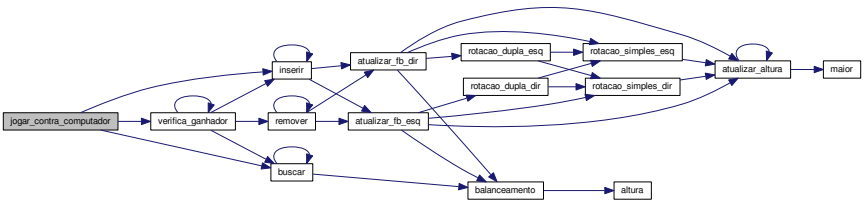
6.3.2.1 void jogar_contra_computador (Arvore * a)

Função que controla o jogo quando selecionado a opção Player vs IA.

Parameters

a	Ponteiro para a árvore AVL

Here is the call graph for this function:



Here is the caller graph for this function:

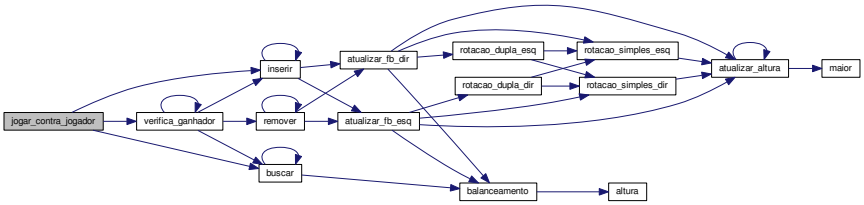
6.3.2.2 void jogar_contra_jogador (Arvore * a)

Função que controla o jogo quando é selecionado a opção de Player vc Player.

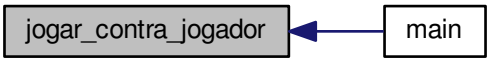
Parameters

a	Ponteiro para a ávore AVL.

Here is the call graph for this function:



Here is the caller graph for this function:



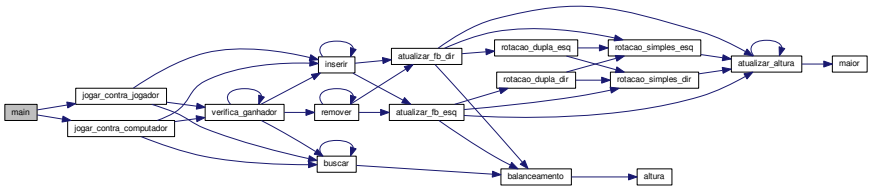
6.3.2.3 int main ( )

Função Main do programa C.

Returns

Retorno da função Main

Here is the call graph for this function:



Index