

Referência da Classe Reversi

Classe que representa o jogo **Reversi**, derivada de **Jogo**. Mais...

```
#include <Reversi.hpp>
```

Diagrama de hierarquia da classe Reversi:



Membros Públicos

void **inicializarTabuleiro** () override

Inicializa o tabuleiro do jogo **Reversi**, colocando as peças iniciais.

Reversi (**Jogador** *_jogador1, **Jogador** *_jogador2)

Construtor da classe **Reversi**.

bool **validarJogada** (int x, int y, **Jogador** *jogador, char peca) override

Valida se uma jogada é possível de acordo com as regras do **Reversi**.

int **verificarCondicaoVitoria** (char peca) override

Verifica a condição de vitória.

void **imprimirTabuleiro** ()

Imprime o estado atual do tabuleiro no terminal.

void **realizarJogada** (**Jogador** *jogador, char peca) override

Realiza uma jogada no tabuleiro.

void **capturarPecas** (int x, int y, **Jogador** *jogador, char peca)

Captura as peças do oponente em todas as direções possíveis (horizontal, vertical e diagonal).

void **capturarDirecao** (int x, int y, **Jogador** *jogador, int deltaX, int deltaY, char peca)

Captura as peças do oponente em uma direção específica.

► **Membros Públicos herdados de Jogo**

Descrição detalhada

Classe que representa o jogo **Reversi**, derivada de **Jogo**.

A classe **Reversi** contém a lógica específica para o jogo **Reversi**, incluindo métodos para verificar a condição de vitória, realizar jogadas e inicializar o tabuleiro. A classe interage com a classe **Jogador** para rastrear as peças de cada jogador e determinar o vencedor.

Construtores e Destrutores

◆ Reversi()

```
Reversi::Reversi ( Jogador * _jogador1 ,  
                  Jogador * _jogador2 )
```

Construtor da classe **Reversi**.

Parâmetros

_jogador1 Ponteiro para o primeiro jogador.

_jogador2 Ponteiro para o segundo jogador.

Documentação das funções

◆ capturarDirecao()

```
void Reversi::capturarDirecao ( int      x,  
                                int      y,  
                                Jogador * jogador,  
                                int      deltaX,  
                                int      deltaY,  
                                char      peca )
```

Captura as peças do oponente em uma direção específica.

Este método verifica se há peças do oponente em uma sequência em uma direção definida pelos deltas (deltaX, deltaY). Se a sequência termina com uma peça do jogador atual, todas as peças do oponente nessa sequência são capturadas.

Parâmetros

x A coordenada x inicial.

y A coordenada y inicial.

jogador O jogador que realizou a jogada.

deltaX O incremento para a coordenada x na direção.

deltaY O incremento para a coordenada y na direção.

◆ capturarPecas()

```
void Reversi::capturarPecas ( int      x,  
                             int      y,  
                             Jogador * jogador,  
                             char      peca )
```

Captura as peças do oponente em todas as direções possíveis (horizontal, vertical e diagonal).

Este método é chamado após uma peça ser colocada no tabuleiro. Ele verifica em todas as direções a partir da posição (x, y) se existem peças do oponente que podem ser capturadas, ou seja, se estão entre a peça recém-colocada e outra peça do jogador atual.

Parâmetros

- x** A coordenada x da peça que foi colocada.
- y** A coordenada y da peça que foi colocada.
- jogador** Ponteiro para o jogador que realizou a jogada.

◆ inicializarTabuleiro()

```
void Reversi::inicializarTabuleiro ( )
```

override

virtual

Inicializa o tabuleiro do jogo **Reversi**, colocando as peças iniciais.

Reimplementa **Jogo**.

◆ realizarJogada()

```
void Reversi::realizarJogada ( Jogador * jogador,  
                              char      peca )
```

override

virtual

Realiza uma jogada no tabuleiro.

Implementa **Jogo**.

◆ validarJogada()

```
bool Reversi::validarJogada ( int      x,  
                              int      y,  
                              Jogador * jogador,  
                              char      peca )
```

override virtual

Valida se uma jogada é possível de acordo com as regras do **Reversi**.

Parâmetros

x Posição X no tabuleiro.

y Posição Y no tabuleiro.

jogador Ponteiro para o jogador que está fazendo a jogada.

Retorna

true se a jogada for válida, false caso contrário.

Implementa **Jogo**.

◆ verificarCondicaoVitoria()

```
int Reversi::verificarCondicaoVitoria ( char peca )
```

override virtual

Verifica a condição de vitória.

Analisa o tabuleiro para determinar se algum jogador cumpriu as condições de vitória.

Parâmetros

peca A peça do jogador atual.

Retorna

1 se o jogador atual venceu, -1 em caso de empate, 0 se o jogo continua.

Reimplementa **Jogo**.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- /Users/iangodoi/Desktop/TP-jogosTabuleiro-cpp-desenvolvimento 5/include/**Reversi.hpp**
- /Users/iangodoi/Desktop/TP-jogosTabuleiro-cpp-desenvolvimento 5/src/**Reversi.cpp**