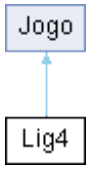
Referência da Classe Lig4

Classe que representa o jogo **Lig4**, derivada da classe base **Jogo**. Mais...

`#include <Lig4.hpp>`

Diagrama de hierarquia da classe Lig4:



Membros Públicos

	Lig4 () Construtor da classe **Lig4**.
void	**inicializarTabuleiro** () override Inicializa o tabuleiro do jogo **Lig4**.
bool	**validarJogada** (int linha, int coluna, **Jogador** *jogador) override Valida a jogada de um jogador no jogo **Lig4**.
int	**verificarCondicaoVitoria** () override Verifica a condição de vitória no jogo **Lig4**.
void	**realizarJogada** () override Realiza a jogada de um jogador em uma coluna no jogo **Lig4**.

▸ **Membros Públicos herdados de Jogo**

Descrição detalhada

Classe que representa o jogo **Lig4**, derivada da classe base **Jogo**.

A classe **Lig4** implementa as regras específicas para o jogo **Lig4**, incluindo a inicialização do tabuleiro, validação das jogadas, verificação das condições de vitória e a realização das jogadas.

Construtores e Destrutores

◆ Lig4()

Lig4::Lig4 ( )

Construtor da classe **Lig4**.

Inicializa o tabuleiro com as dimensões específicas de **Lig4** e define as peças dos jogadores.

```
6           {
7       inicializarTabuleiro();
8   }
```

Documentação das funções

◆ inicializarTabuleiro()

void Lig4::inicializarTabuleiro ( ) override virtual

Inicializa o tabuleiro do jogo **Lig4**.

Sobrescreve o método da classe base para definir o estado inicial do tabuleiro de **Lig4**.

Reimplementa **Jogo**.

```
10                                  {
11      tabuleiro->configurarTabuleiro(altura, largura);
12  }
```

◆ realizarJogada()

void Lig4::realizarJogada ( ) override virtual

Realiza a jogada de um jogador em uma coluna no jogo **Lig4**.

Sobrescreve o método da classe base para realizar uma jogada em uma coluna específica do tabuleiro.

Implementa **Jogo**.

```
95                              {
96      int coluna;
97      std::cout << "Digite a coluna a ser jogada" << std::endl;
98      std::cin >> coluna;
99      if (!validarJogada(0, coluna, getJogadorAtual())) {
100         return;
101     }
102
103     // Realiza a jogada, ou seja, coloca a peça na posição correta na coluna
    especificada
104     for (int i = altura - 1; i >= 0; i--) {
105         if (tabuleiro->obterPeca(i, coluna) == '.') {
106             tabuleiro->definirPosicao(i, coluna, getJogadorAtual()->
    minhaPeca());
107             break;
108         }
109     }
110
111 }
```

◆ validarJogada()

bool Lig4::validarJogada (int linha,
int coluna,
Jogador * jogador) override virtual

Valida a jogada de um jogador no jogo **Lig4**.

Parâmetros

linha Linha onde o jogador deseja jogar.

coluna Coluna onde o jogador deseja jogar.

jogador Objeto que representa o jogador que está realizando a jogada.

Retorna

true se a jogada for válida, false caso contrário.

Verifica se a jogada feita é válida com base nas regras do **Lig4**.

Implementa **Jogo**.

```
14                                                                  {
15      if (coluna < 0 || coluna >= largura) {
16          std::cout << "Coluna inválida!" << std::endl;
17          return false;
18      }
19      if (tabuleiro->obterPeca(0, coluna) != '.') {
20          std::cout << "Coluna cheia!" << std::endl;
21          return false;
22      }
23      return true;
24  }
```

◆ verificarCondicaoVitoria()

int Lig4::verificarCondicaoVitoria ( )

override virtual

Verifica a condição de vitória no jogo **Lig4**.

Analisa o tabuleiro de **Lig4** para determinar o estado atual do jogo.

Retorna

int Retorna 1 se algum jogador venceu, −1 em caso de empate, ou 0 se o jogo deve continuar.

Reimplementa **Jogo**.

```
                                    {

    char jogador = getJogadorAtual() -> minhaPeca();
    char oponente = (jogador == jogador1Peca) ? jogador2Peca : jogador1Peca;

    // Verifica horizontalmente
    for (int i = 0; i < altura; i++) {
        for (int j = 0; j < largura - 3; j++) {
            if (tabuleiro->obterPeca(i, j) == jogador &&
                tabuleiro->obterPeca(i, j+1) == jogador &&
                tabuleiro->obterPeca(i, j+2) == jogador &&
                tabuleiro->obterPeca(i, j+3) == jogador) {
                return 1;  // Jogador atual venceu
            }
        }
    }

    // Verifica verticalmente
    for (int i = 0; i < altura - 3; i++) {
        for (int j = 0; j < largura; j++) {
            if (tabuleiro->obterPeca(i, j) == jogador &&
                tabuleiro->obterPeca(i+1, j) == jogador &&
                tabuleiro->obterPeca(i+2, j) == jogador &&
                tabuleiro->obterPeca(i+3, j) == jogador) {
                return 1;  // Jogador atual venceu
            }
        }
    }

    // Verifica diagonal (cima para baixo - DIAGONAL DESCENDENTE)
    for (int i = 0; i < altura - 3; i++) {
        for (int j = 0; j < largura - 3; j++) {
            if (tabuleiro->obterPeca(i, j) == jogador &&
                tabuleiro->obterPeca(i+1, j+1) == jogador &&
                tabuleiro->obterPeca(i+2, j+2) == jogador &&
                tabuleiro->obterPeca(i+3, j+3) == jogador) {
                return 1;  // Jogador atual venceu
            }
        }
    }

    // Verifica diagonal (baixo para cima - DIAGONAL ASCENDENTE)
    for (int i = 3; i < altura; i++) {
        for (int j = 0; j < largura - 3; j++) {
            if (tabuleiro->obterPeca(i, j) == jogador &&
                tabuleiro->obterPeca(i-1, j+1) == jogador &&
                tabuleiro->obterPeca(i-2, j+2) == jogador &&
                tabuleiro->obterPeca(i-3, j+3) == jogador) {
                return 1;  // Jogador atual venceu
            }
        }
    }
}
```

```
78
79      // Verifica se o tabuleiro está cheio
80      bool tabuleiroCheio = true;
81      for (int j = 0; j < largura; j++) {
82          if (tabuleiro->obterPeca(0, j) == '.') {
83              tabuleiroCheio = false;
84              break;
85          }
86      }
87
88      if (tabuleiroCheio) {
89          return -1;  // Jogo terminou em empate
90      }
91
92      return 0;  // O jogo deve continuar
93  }
```

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

- /Users/iangodoi/Desktop/TP-jogosTabuleiro-cpp-desenvolvimento/include/**Lig4.hpp**
- /Users/iangodoi/Desktop/TP-jogosTabuleiro-cpp-desenvolvimento/src/**Lig4.cpp**