

# CARTÕES DE RESONSABILIDADES DE CLASSE (CRC)

Classe: JogoReversi

## **RESPONSABILIDADES:**

- 1.Inicializar o tabuleiro com valores padrão.
- 2.Verificar se a posição é válida para jogar.
- 3.Capturar peças e atualizar os contadores de capturas.
- 4.Exibir a pontuação dos jogadores.
- 5.Determinar se há uma condição de vitória

## **COLABORAÇÕES:**

**JogoTabuleiro:** Fornece a estrutura e lógica básica para o jogo de tabuleiro.

Classe: JogoLig4

## **RESPONSABILIDADES:**

- 1.Inicializar o tabuleiro com valores padrão.
- 2.Realizar uma jogada e verificar erros.
- 3.Verificar condições de vitória.
- 4.Exibir o estado atual do tabuleiro.

## **COLABORAÇÕES:**

**JogoTabuleiro:** Fornece a estrutura e lógica básica para o jogo de tabuleiro.

Classe: JogoDaVelha

## **RESPONSABILIDADES:**

- 1.Inicializar o tabuleiro com valores padrão e configurar o modo de jogo.
- 2.Realizar jogadas e registrar no histórico.
- 3.Verificar condições de vitória.
- 4.Expandir o tabuleiro dinamicamente.

Classe: JogoDaVelha
5. Permitir que jogadores desfaçam jogadas.
<b>COLABORAÇÕES:</b>
<b>JogoTabuleiro</b> : Fornece a estrutura e lógica básica para o jogo de tabuleiro.

Classe: Player
RESPONSABILIDADES:
1. Armazenar informações sobre o jogador.
2. Gerenciar estatísticas de vitórias e derrotas.
3. Fornecer métodos e modificações das estatísticas.
4. Exibir informações de vitórias e derrotas.
COLABORAÇÕES:
<b>ListaDeJogadores</b> : armazenamento e gerenciamento de instâncias de Player.
<b>JogoDeTabuleiro</b> : atualizações após fim de jogo.

Classe: ListaDeJogadores
RESPONSABILIDADES:
1. Armazenar lista de players usando um map.
2. Adicionar/remover players.
3. Ler e carregar dados em um arquivo.
4. Ordenar e exibir histórico.
5. Salvar dados dos jogadores em um arquivo.
COLABORAÇÕES:
<b>Player</b> : manipular instâncias de jogadores.
<b>ifstream</b> e <b>ofstream</b> : leitura e escrita de arquivos.

Classe: JogoDeTabuleiro
RESPONSABILIDADES:
1. Representar o tabuleiro(método abstrato).
2. Inicializar o tabuleiro(método abstrato).
3. Verificar se há um vencedor(método abstrato).
4. Exibir estado atual.
COLABORAÇÕES:

Classes derivadas (ex: `Reversi` , `Velha` , `Lig4` ): Para implementar comportamentos específicos de cada jogo.

`Player` : Para atualizar estatísticas após o fim de um jogo.