

Mapello



Turma 1 Mapello_5

Trabalho realizado por:

Nome	Número
Pedro Miguel Oliveira Azevedo	201603816
Diogo Ferreira de Sousa	201706409

Descrição:

O mapello é uma variante do reversi no qual é adicionado algumas peças extras em comparação com o original, sendo elas as Walls (peças que servem como as paredes do tabuleiro e não podem ser colocadas peças em cima), Bonuses (peças que dão mais pontos a quem os captura) e Jokers (peças normais que pertencem ao jogador atual). O objetivo do jogo é terminar o jogo com o maior número de pontos (pontos = peças + pontos bonus) isto faz-se capturando peças inimigas, tornando as da cor do jogador.

Regras

- Apenas pode colocar uma peça por turno
- No início do jogo, o tabuleiro começa com 4 peças no centro (Ex.: B W | W B), 8 jokers nas bordas do tabuleiro e 8 bonus e paredes no meio do tabuleiro.
- Só pode colocar a peça se consegue capturar peças inimigas (Ex.:B W W _)
- Se não conseguir colocar nenhuma peça passa a jogada para o próximo jogador.
- Se os 2 jogadores passarem consecutivos o jogo acaba e ganha quem tiver mais pontos.

- O jogo acaba quando não houver mais jogadas possíveis (Ex.: Tabuleiro completamente cheio).

Links

-https://nestorgames.com/#mapello_detail -<https://cardgames.io/reversi/>

Representação de Interna do estado do Jogo

O estado do tabuleiro está guardado numa lista de listas (10 x 10) em qual cada posição vai guardar a peça que está aí. No jogador iremos guardar a sua pontuação ,tendo em conta que esta diminui ou aumenta de acordo com as peças que captura e perde, e se tinha passado a ronda anterior.

Peças

As peças na lista de listas são representadas da seguinte forma:


- white : peças brancas
- black : peças pretas
- none : espaço vazio
- wall : parede
- joker : joker
- bonus : bonus

Estado Inicial

No estado inicial temos um tabuleiro $N * N$ gerado pela função na qual preenchemos todos os espaços da lista com 'none' , que em seguida vamos inicializar as posições das peças iniciais (2 peças brancas, 2 peças pretas, 8 paredes, 8 jokers, 8 bonus).

```
% play
% Starts a new game
play:-
    initRandom,
    createEmptyBoard(B0),
    setInitialPieces(B0,B1),
    nl, write('Performing random Wall and Bonus pieces placement'), nl,
    wallSetupPhase(B1, 8, B2),
    bonusSetupPhase(B2, 8, B3),
    jokerSetupPhase(B3, 8, B4),
    playGame(B4, 0, 16, _).
```

Setting Random Walls and Bonus

 Walls and Bonus

Setting Jokers



Estado Intermédio

O estado intermédio do jogo é quando ambos os jogadores conseguem fazer jogadas ou pelo menos um deles consegue jogar, nesta fase vamos adicionando peças brancas e pretas que por sua vez capturam peças dos inimigos ou bonus tornando-as da mesma cor. Esta fase ainda não está completamente implementada, pois ainda é necessário realizar a validação da jogada e a mudança da cor das peças.

Play Loop



Estado Final

O jogo entra no estado final quando:

- Não existem mais espaços vazios no tabuleiro.
- Ambos os jogadores passam a sua vez. Ainda não foi iniciado o trabalho nesta fase, uma vez que ainda é necessário terminar a fase prévia.

Visualização dos estados do jogo

A visualização do tabuleiro é feita chamando a função:

```
% display_game(+Board, +Player)
% Displays the current game state, and announces next player turn

display_game(Board, Player):-
    printBoard(Board),
    write('Player '),
    write(Player),
    write(' Turn'), nl.
```

Esta função por sua vez chama a função printBoard que por sua vez vai linha a linha e depois elemento a elemento imprimir o conteúdo do tabuleiro.

```
% printBoard(+Board, )
% Displays the current board state
printBoard(Board):-
    nl,
    printHeader,
    printRows(Board, 10), nl.
```

```
% printRows(+Board, +N)
% Displays all the rows on the given board
printRows(_, 0).
printRows([FirstElem|OtherElem], N):-
    Number is 10 - N,
    printRowNumber(Number),
    N1 is N-1,
    printRow(FirstElem),
    printRowSep,
    printRows(OtherElem,N1).

% printRow(+Board)
% Displays a single row of the given board
printRow([]):-
    nl.
printRow([FirstElem|OtherElem]):-
    getRep(FirstElem,Label),
    printRep(Label),
    printColumnSep,
    printRow(OtherElem).
```

Cada peça é representada no tabuleiro da seguinte forma:

- 'W' : peças brancas
- 'B' : peças pretas
- '.' : espaço vazio
- 'G' : parede
- 'J' : joker
- 'P' : bonus