Faculdade de Engenharia da Universidade do Porto

Programação em Lógica - Trabalho Prático 1

Aplicação em Prolog para um Jogo de Tabuleiro - Nudge

Grupo Nudge_1:

João Paulo Monteiro Leite - up201705312
Márcia Isabel Reis Teixeira - up201706065

2019/2020

Índice

1. Descrição do jogo

1.1. História

Nudge é um jogo de estratégia abstrata, com regras simples mas uma mecânica inovadora. Destaca-se principalmente pela preocupação ambiental na sua produção e embalamento, sendo as peças feitas de bioplástico biodegradável, e o tabuleiro de cartão totalmente reciclado.

1.2. Regras

O jogo deve ser jogado por 2 jogadores, e é constituído por um tabuleiro quadrado com 5x5 posições, por 3 discos brancos e 3 discos pretos.

Iniciar o jogo

Há 3 disposições dos discos possíveis para iniciar o jogo, ilustradas na figura 1.

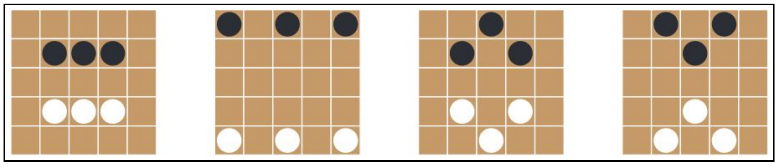

Figura 1 - Posições iniciais do jogo

O primeiro jogador é escolhido aleatoriamente através do lançamento de uma moeda ou de um dos discos. Os jogadores vão depois alternando a vez entre si, fazendo dois movimentos a cada jogada.

Cada disco pode ser movido um quadrado a cada movimento, em qualquer direção que não diagonal (frente, trás ou lados). Uma fila de vários discos pode ser movida longitudinalmente também um quadrado por movimento. Na mesma jogada não é permitido fazer um movimento e depois outro que faça retornar à posição original.

O jogador pode também com um movimento arrastar um disco do adversário (“nudge”). Para o fazer, o jogador deve ter uma fila de discos longitudinalmente maior do que a do adversário (por exemplo, 2 discos contra 1 do adversário).

Os movimentos válidos e inválidos encontram-se ilustrados na figura 2, com explicações em inglês.

O primeiro jogador a conseguir arrastar um disco do adversário para fora do tabuleiro é o vencedor.

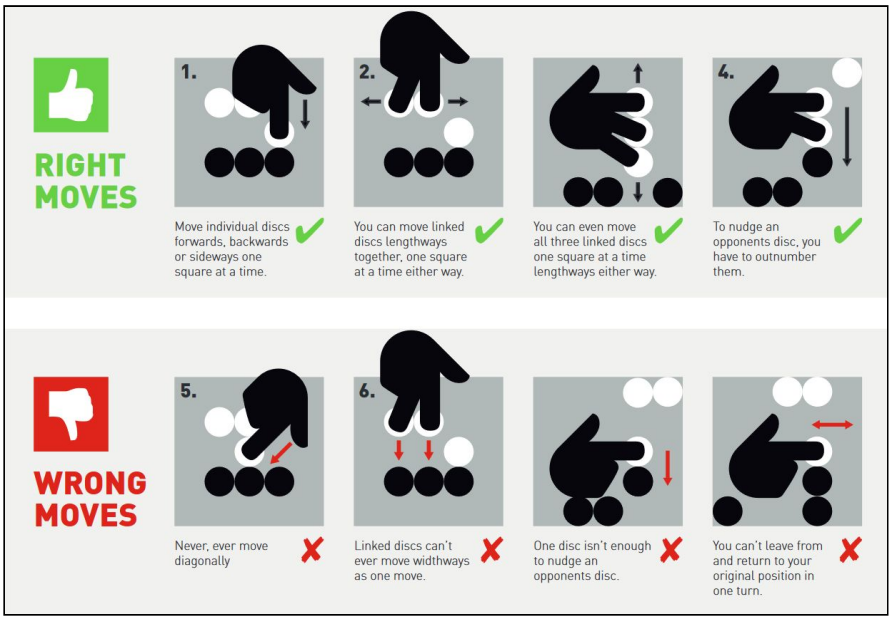

Figura 2 - Ilustração e explicação dos movimentos válidos e inválidos

A história e regras do jogo, bem como as figuras 1 e 2, foram retiradas do site oficial do *Nudge* (nudgegames.co.uk).

2. Modelação em ProLog

2.1. Representação interna do estado do jogo

O estado do tabuleiro do jogo é representado, internamente, por uma lista de listas. O tabuleiro tem 5 listas, cada uma delas correspondente a uma fila, e cada lista tem 5 números, que indicam o estado de cada célula. Estes números podem ser 0, caso a célula esteja vazia, 1 caso esteja ocupada por um disco branco, e 2 caso esteja ocupada por um disco preto.

O jogador atual é guardado também numa variável que vai sendo passada entre os vários predicados, podendo o seu valor ser 1, caso seja a vez do jogador dos discos brancos, ou 2, caso seja a vez do jogador dos discos pretos.

Seguem-se exemplificações de vários estados do tabuleiro em ProLog (inicial, intermédio e final), acompanhadas de imagens que mostram um tabuleiro de Nudge nesse estado (figuras 3-5, retiradas também do site oficial do Nudge).

Código que inicializa o tabuleiro, criando um tabuleiro com as peças numa das posições iniciais permitidas pelo jogo:

```
initBoard(Board):-
    Board=[[0,0,0,0,0],
           [0,1,1,1,0],
           [0,0,0,0,0],
           [0,2,2,2,0],
           [0,0,0,0,0]].
```

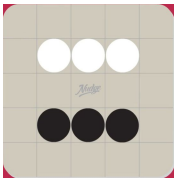

Figura 3

Exemplificação de um possível estado intermédio:

```
Board=[[0,0,0,0,0],
       [0,0,0,1,1],
       [0,0,2,1,0],
       [0,2,2,0,0],
       [0,0,0,0,0]].
```

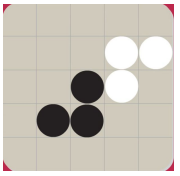

Figura 4

Exemplificação de um possível estado final (dado que o jogo acaba no momento em que uma peça é empurrada para fora do tabuleiro, não se considera, pelo menos nesta fase inicial de desenvolvimento, que seja necessário guardar as peças que estão fora do tabuleiro):

```
Board=[[0,0,0,0,0],
       [0,0,0,0,0],
       [0,0,0,0,0],
       [0,0,0,2,1],
       [0,2,0,1,1]].
```

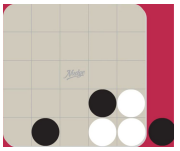

Figura 5

2.2. Visualização do tabuleiro em modo de texto

A visualização do tabuleiro é feita através do predicado displayGame(+Board, +Player). Este predicado começa por imprimir as coordenadas horizontais do tabuleiro (A-E), depois utiliza o predicado printBoard(+Board, +LineNr) para imprimir o tabuleiro recursivamente, imprimindo por fim o símbolo do jogador atual ('B' para o jogador dos discos pretos e 'W' para o jogador dos discos brancos).

No predicado printBoard(+Board, +LineNr), Board corresponde ao tabuleiro (lista de listas) e LineNr ao código ASCII correspondente ao caracter do número da linha a ser impressa no momento, ou seja, a primeira linha terá LineNr 49, a segunda LineNr 50 e assim sucessivamente.
printBoard imprime o tabuleiro recursivamente, imprimindo os divisores entre linhas (ou a margem superior, no caso da primeira linha) e as coordenadas verticais (1-5), usando o predicado printLine(+Line) para imprimir a linha recursivamente e, por fim, efetua a chamada recursiva.
O caso base de printBoard(+Board, +LineNr) é o caso em que Board é uma lista vazia; nesta situação é impressa apenas a margem inferior do tabuleiro.

O predicado printLine(+Line) recebe uma linha do tabuleiro, ou seja, uma lista, e imprime-a, começando por imprimir o divisor das células (linha dupla vertical), usando depois o predicado printCell(+Cell) para imprimir o conteúdo da célula e, por fim, efetua a chamada recursiva.
O case base de printLine(+Line) é o caso em que Line é uma lista vazia, caso em que é impresso apenas o divisor das células.

Por último, o predicado printCell(+Cell) imprime apenas o símbolo associado ao conteúdo da célula: no caso da célula vazia (0) é um espaço, um disco branco (1) é a letra W e um disco preto (2) é a letra B.

Para tornar o código mais legível e organizado existem também os predicados printDivider(_) e printTopBorder(_), que apenas imprimem os divisores entre filas e a margem superior do tabuleiro, respetivamente.
Para impressão no ecrã são usados os predicados *built-in* put_code(+Code) e write(+Term).

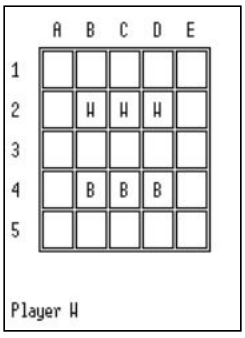

Figura 6 - Output produzido por displayGame(+Board, +Player), mostrando neste caso o tabuleiro no seu estado inicial.