

Nudge

Programação em Lógica

Trabalho Prático 1 - Relatório intercalar

História

Nudge é um jogo de tabuleiro bastante recente, criado em 2019, que tem o intuito de trazer diversão a bares e conectar pessoas. Embora seja um jogo simples, tem bastantes variantes que o tornam interessante. Orgulha-se também de ser um jogo amigo do ambiente, constituído por peças recicláveis.

Regras

De acordo com o website oficial do Nudge, as regras são as seguintes:

1. O jogo consiste num tabuleiro 5x5, 3 discos pretos e 3 discos brancos.
2. Discos individuais podem ser mexidos um quadrado de cada vez ou dois quadrados com os dois movimentos disponíveis, para a frente, trás, direita ou esquerda e qualquer combinação das mesmas, mas nunca na diagonal.
3. Linhas de discos ligados podem ser movidos juntos longitudinalmente, como vagões de um comboio, um quadrado de cada vez, contando como um movimento, mas não se pode mover linhas não longitudinalmente - exemplos em baixo.
4. Para empurrar (Nudge) um disco do oponente, é preciso superá-lo em número de discos na linha longitudinalmente - 2 contra 1 e por aí em diante.
5. O jogo acaba quando um dos jogadores empurrar o disco do oponente para fora do tabuleiro.

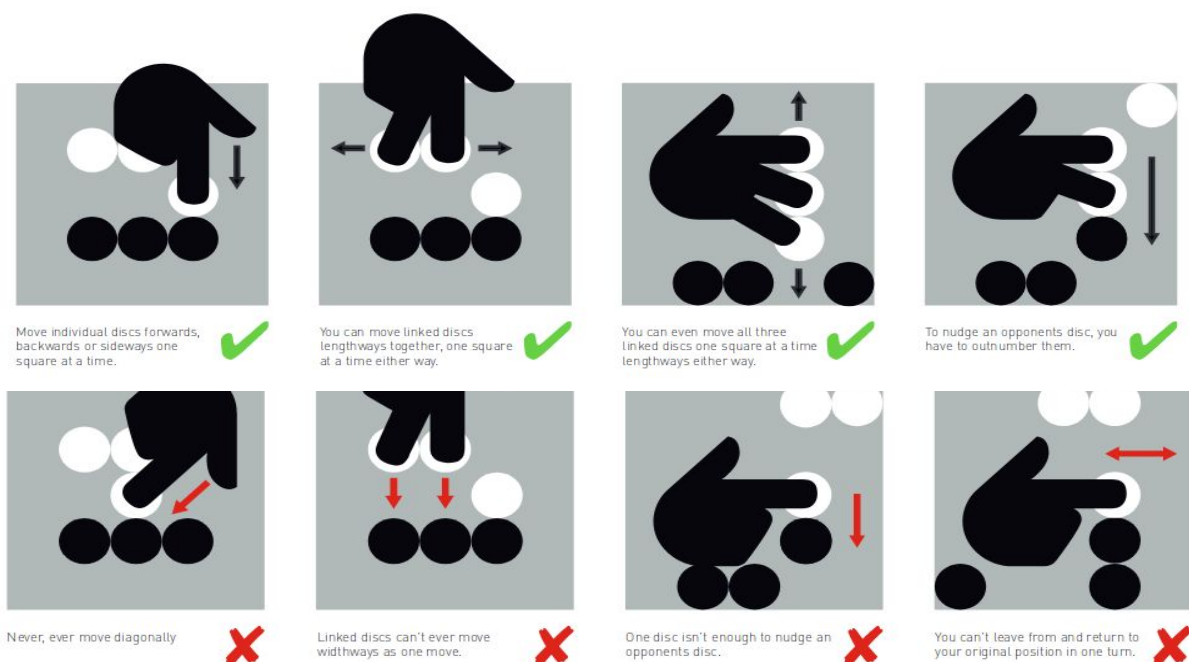


Figura 1 - Movimentos permitidos e não permitidos no Nudge

Fonte: <https://nudge.greenwich-design.co.uk>

Abordagem inicial à modelação do jogo

Representação interna do estado do jogo

O tabuleiro de jogo é representado por uma lista de listas, sendo que cada lista interior constitui uma linha no tabuleiro e os elementos das listas interiores que partilhem o mesmo índice constituem uma coluna no tabuleiro. Cada célula do tabuleiro, contenha ela uma peça preta, branca, ou esteja vazia, é representada por um átomo (*black*, *white*, *empty*).

A representação visual do tabuleiro é efetuada de forma simples e clara, com fronteiras bem definidas e claras. Para a representação das peças, optamos por representar peças pretas com o carácter 'X', e peças brancas com o carácter 'O'. Contudo, no seguimento do projeto, pretendemos recorrer a bibliotecas de cores para alterar a cor dos caracteres impressos no output da consola, de modo a que seja mais clara a distinção entre peças.

```
initialBoard(  
  [  
    [empty, empty, empty, empty, empty, empty, empty],  
    [empty, empty, empty, empty, empty, empty, empty],  
    [empty, empty, black, black, black, empty, empty],  
    [empty, empty, empty, empty, empty, empty, empty],  
    [empty, empty, white, white, white, empty, empty],  
    [empty, empty, empty, empty, empty, empty, empty],  
    [empty, empty, empty, empty, empty, empty, empty]  
  ]  
).
```

```
cell(white, C) :-  
    C = 'O'.  
cell(black, C) :-  
    C = 'X'.  
cell(empty, C) :-  
    C = ' '.
```

Figura 2 - Estrutura da lista de listas do tabuleiro e significado de cada átomo.

Visualização do tabuleiro em modo de texto

Para representação do tabuleiro, é chamado um predicado `display_game(Board,Player)`, tal como especificado, que representa o tabuleiro tendo em conta o seu estado atual (Board) e o jogador em causa (Player). Por sua vez, este predicado irá invocar outro, `printBoard(Board)`, que por sua vez imprime o cabeçalho com as coordenadas (`printCoordLetters`) e a matriz em si (`PrintMatrix([Head|Tail],Index)`). Neste último predicado, é, recursivamente, impressa cada uma das linhas, sendo que para esse efeito existe o predicado `printLineWithCell([Head|Tail])`, que por sua vez irá, também recursivamente, imprimir cada elemento da linha do tabuleiro (listas internas). Além deste, também tem o predicado `printLineWithCellOut([Head|Tail])` que faz o mesmo que o predicado mencionado anteriormente, mas para elementos que estão fora do tabuleiro.

Deste modo, recorrendo a recursividade dupla, é impresso o tabuleiro de uma maneira simples e eficaz.

	A	B	C	D	E
1		X		X	
2			X		
3					
4			0		
5		0		0	

	A	B	C	D	E
1					
2		X	X	X	
3					
4		0	0	0	
5					

	A	B	C	D	E
1	X		X		X
2					
3					
4					
5	0		0		0

	A	B	C	D	E
1			X		
2		X		X	
3					
4		0		0	
5			0		

Figura 3 - Todos os possíveis estados de tabuleiro iniciais

	A	B	C	D	E
1					
2		X	X	0	0
3		X	0		
4					
5					

	A	B	C	D	E
1					
2					
3					
4				X	0
5		X		0	0

Figura 4 - Exemplos de estados de tabuleiro a meio do jogo e no fim do jogo, respetivamente

```

printMatrix([Head|Tail], Index) :-
    Index = 0 -> ( write(' '), printBlank,
    printBlank, NextIndex is Index + 1,
    write(' '), printLineWithCellOut(Head), nl,
    printBlank, write(' '), printHeader,
    printMatrix(Tail, NextIndex));

    Index = 6 -> ( write(' '), printBlank,
    printBlank, NextIndex is Index + 1, write(' '),
    printLineWithCellOut(Head), nl, write(' '),
    printMatrix(Tail, NextIndex));

    write(' '),
    printEmpty,
    write(' '), write(Index), write(' '),
    NextIndex is Index + 1,
    printLineWithCell(Head, 1), nl,
    write(' '),
    printEmpty,
    write(' '),
    printHeader,
    printMatrix(Tail, NextIndex).

```

```

printLineWithCell([Head|Tail], Ind) :-
    Ind = 7 -> (cell(Head, Char), write(' '), write(Char), write(' '), printLineWithCell(Tail));
    cell(Head, Char),
    write(' '), write(Char), write(' |'),
    NextInd is Ind + 1,
    printLineWithCell(Tail, NextInd).

printLineWithCellOut([Head|Tail]) :-
    cell(Head, Char),
    write(' '), write(Char), write(' '),
    printLineWithCellOut(Tail).

```

Figura 5 - Função printMatrix([Head|Tail],Index), printLineWithCell([Head|Tail]) e printLineWithCellOut([Head|Tail]), respetivamente

Bibliografia

Website oficial do Nudge: <https://nudge.greenwich-design.co.uk>