

Programação em Lógica

Relatório Intercalar Crab Stack

Mestrado integrado em Engenharia Informática Computação

Nuno Miguel dos Santos Castro (up201406990)

Sara Beatriz Gonçalves Santos (up201402814)

Descrição do jogo

Crab Stack é um jogo de tabuleiro criado pela companhia *Blue Orange Games*, fundada em 2000 por Julien Mayot e Thierry Denoual. O objetivo do jogo é ser o último jogador com pelo menos um caranguejo (peça) que se consiga movimentar. seguindo as instruções originais, as regras e o tabuleiro foram adaptados para a versão de apenas dois jogadores.

Existem diferentes tipos de peças de diferentes cores (verde e vermelho) e tamanhos (pequeno, médio e grande). No início do jogo cada jogador escolhe a cor dos seus caranguejos, baralha-os e de seguida dispõe todas as peças aleatoriamente no tabuleiro apenas nas casas que contêm rochas.

Cada jogador pode mover apenas um dos seus caranguejos no seu respetivo turno, respeitando o tipo de movimento correspondente ao seu tamanho. O jogador só pode mover os seus caranguejos através de outros caranguejos, não podendo movimentá-los através de rochas vazias ou através da água e não podem voltar a uma posição em que já tenham estado nesse turno.

Quando os caranguejos formam uma coluna (Stack), apenas o caranguejo do topo se pode mexer. Formar colunas com os caranguejos segue algumas regras:

- Os caranguejos grandes podem formar colunas com qualquer outro caranguejo, incluindo outros caranguejos grandes.
- Os caranguejos médios podem formar colunas apenas com outros caranguejos médios ou caranguejos pequenos.
- Os caranguejos pequenos apenas podem formar colunas com outros caranguejos pequenos.

Quando os caranguejos ficam separados em 2 grupos (quando existe uma linha no tabuleiro que separe caranguejos impossibilitando o seu movimento), uma “onda” irá atingir os caranguejos e o grupo que ocupa menos espaço no tabuleiro será removido do jogo. Se ocuparem o mesmo espaço então será decidido pelo número de caranguejos de cada grupo e o que tiver menor número será removido. Se ocupar o mesmo espaço e tiver o mesmo número de caranguejos, o jogador que estiver no seu turno decidirá qual grupo será removido.

Se um jogador não conseguir mover nenhum dos seus caranguejos no início do seu turno é eliminado do jogo. O último jogador a ser eliminado é o vencedor. Se o jogo chegar a um ponto em que os jogadores continuam a repetir as mesmas jogadas por não haver outras possíveis é considerado um empate e será jogado mais um jogo para decidir o vencedor.



Figura 1 Início do jogo



Figura 2 Tabuleiro Intermédio



Figura 3 Estado final
Figura 3 (que o jogador azul foi eliminado)

A informação utilizada foi retirada dos seguintes URLs:

- <https://boardgamegeek.com/boardgame/172033/crab-stack>
- <http://www.geekyhobbies.com/crab-stack-review-and-instructions/>

Abordagem Inicial à modelação do jogo

Representação do estado do jogo

O tabuleiro do jogo é representado através de listas. São utilizados diferentes átomos que representam o tabuleiro e os diferentes tipos de peças. Dependendo das suas características os átomos são representados da seguinte maneira:

Tipo	Átomo	Representação no tabuleiro
Rochas	r	O
Caranguejos pequenos jogador 1	cp1	a
Caranguejos médios jogador 1	cm1	A
Caranguejos grandes jogador 1	cg1	*
Caranguejos pequenos jogador 2	cp2	b
Caranguejos médios jogador 2	cm2	B
Caranguejos grandes jogador 2	cg2	+

```

empty_board([[x,x,r,x,r,x,r,x,x],
             [x,r,x,r,x,r,x,r,x],
             [r,x,r,x,x,x,r,x,r],
             [x,r,x,r,x,r,x,r,x],
             [x,x,r,x,r,x,r,x,x]]).

initial_board([[x,x,cg2,x,cp1,x,cm1,x,x],
              [x,cm2,x,cm2,x,cg2,x,cp2,x],
              [cp1,x,cg2,x,x,x,cm1,x,cp2],
              [x,cp1,x,cm2,x,cm2,x,cg1,x],
              [x,x,cm1,x,cg2,x,cp2,x,x]]).

middle_board([[x,x,r,x,cp1,x,cm1,x,x],
              [x,r,x,cm2,x,r,x,cp2,x],
              [cg1,x,cg2,x,x,x,r,x,cp2],
              [x,cp1,x,cm1,x,cm1,x,cg2,x],
              [x,x,r,x,r,x,cp2,x,x]]).

final_board([[x,x,r,x,r,x,cm2,x,x],
             [x,r,x,cm2,x,r,x,r,x],
             [r,x,cg2,x,x,x,r,x,cp2],
             [x,cp2,x,r,x,cm2,x,cg2,x],
             [x,x,r,x,r,x,cp2,x,x]]).

```

Figura 4 Tabuleiro vazio e estados inicial, intermédio e final representados em Prolog

Visualização do tabuleiro em modo de texto

Foi implementado o predicado para a visualização, recebendo como argumento um tabuleiro:

```

display_board([L1|Ls]):-
    display_line(L1),
    nl,
    display_board(Ls).
display_board([]):-
    nl.
display_line([E|Es]):-
    translate(E,V),
    write(V),
    display_line(Es).
display_line([]).

translate(x,' ').
translate(r,'O').
translate(cp1,'a').
translate(cm1,'A').
translate(cg1,'*').
translate(cp2,'b').
translate(cm2,'B').
translate(cg2,'+').

```

Posteriormente poderá ser modificado para conter identificação das células de forma a facilitar a escolha de peça e movimento de cada jogada.

O predicado implementado tem como resultado o seguinte output:

```

  O O O
O O O O
O O   O O
O O O O
  O O O

```

Figura 5 Tabuleiro vazio

```

+ a A
B B + b
a +   A b
a B B *
  A + b

```

Figura 6 Tabuleiro Inicial

```

  O a A
  O B O b
* +   O b
  a A A +
  O O b

```

Figura 7 Tabuleiro Intermédio

```

  O O B
  O B O O
O +   O b
  b O B +
  O O b

```

Figura 8 Tabuleiro Final

Movimentos

Existem 3 tipos de movimentos possíveis dependendo do tipo de peça:

- **Peças grandes:** podem mover-se uma unidade

Predicado moveBig(token)

- **Peças médias:** podem mover-se duas unidades

Predicado moveMedium(token)

- **Peças pequenas:** podem mover-se três unidades

Predicado moveSmall(token)