

Virus Wars

PLOG - 2018

Ângelo Miguel Tenreiro Teixeira – 201606516 Henrique Melo Lima – 201606525

Descrição

O jogo é baseado na expansão de vários vírus tendo sido criado algures nos anos 80.

Regras

É jogado por dois jogadores – azul e vermelho - cujo objetivo é deixar o oponente sem jogadas possíveis.

Começando pelo azul, os jogadores têm 5 jogadas por turno, sendo que cada jogada pode ser de 2 tipos:

- Colocar um vírus numa célula acessível vazia no tabuleiro
- Absorver zombificar um vírus oponente em qualquer célula acessível do tabuleiro, p.ex. trocar um vírus adversário por um estado zombificado da cor do jogador, que fica permanente até ao fim do jogo, não podendo ser "acordados", "recuperados" ou removidos do tabuleiro

Uma célula é acessível se:

- Está verticalmente, horizontalmente ou diagonalmente adjacente a um vírus já presente do jogador no tabuleiro, mesmo tendo sido colocado num mesmo turno
- Está verticalmente, horizontalmente ou diagonalmente adjacente a um conjunto de zombies ligados da cor do jogador, mesmo que os zombies tenham aparecido no mesmo turno

Referências úteis:

Wiki Page

http://www.iggamecenter.com/info/en/viruswars.html

Representação Interna do Jogo

Para representar o tabuleiro do jogo, é utilizada uma lista de listas de átomos (valores inteiros), sendo que o valor de cada átomo segue a seguinte estrutura:

- 1: Representa uma célula com um vírus azul no estado normal
- 2: Representa uma célula com um vírus vermelho no estado normal
- 3: Representa uma célula com um zombie azul
- 4: Representa uma célula com um zombie vermelho
- 0: Representa uma célula vazia

Nota: Os valores das personagens do jogador azul são ímpares e os das personagens do jogador vermelho são pares. Esta decisão foi tomada de forma a futuramente facilitar a identificação das personagens de cada jogador.

```
start gameplay(L):-
              [1,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0,2]
     mid gameplay(L):-
              [1,1,3,0,0,0,0,0,0,0,0,0],
              [1,3,0,0,0,0,0,0,0,0,0,0],
[0,0,3,3,0,0,0,0,0,0,0,0],
              [0,0,0,0,2,0,0,0,0,0,0],
              [0,0,0,0,2,2,0,0,0,0,0],
              [0,0,0,3,3,2,0,0,0,0,0],
              [0,2,2,0,0,0,2,2,2,2,2],
              [0,0,0,0,0,0,0,0,0,0,2],
              [0,0,0,0,0,0,0,0,0,0,2],
31
32
33
34
35
36
              [0,0,0,0,0,0,0,0,0,0,2],
              [0,0,0,0,0,0,0,0,0,0,2]
     final_gameplay(L):-
              [1,1,3,0,0,0,0,0,0,0,0,0],
              [1,3,0,0,0,0,0,0,0,0,0],
              [0,0,3,3,0,3,0,0,0,0,0],
              [0,0,1,0,3,3,3,0,0,0,0],
              [0,0,0,0,3,3,3,0,0,0,0],
              [0,0,1,3,3,3,1,1,0,0,0],
              [0,3,3,1,0,0,3,3,3,3,3],
              [0,0,0,0,0,0,1,1,0,3,3],
              [0,0,0,0,0,0,0,0,0,3,3],
              [0,0,0,0,0,0,0,0,0,1,3],
46
              [0,0,0,0,0,0,0,0,0,0,3]
```

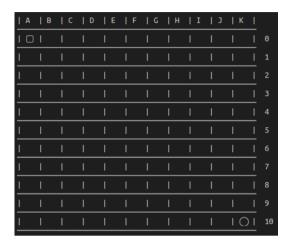
Algoritmo

O tabuleiro é desenhado a partir de uma lista de listas, através do predicado display_game. Este predicado, por sua vez, utiliza predicados auxiliares como:

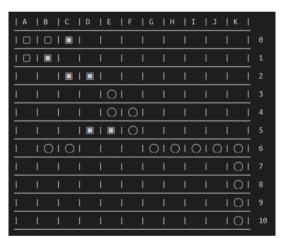
- **traducao**: garante a tradução de certos valores (de forma a obter, por exemplo, o código unicode do caracter correspondente a um vírus)
- **print_cell**: imprime para a consola o caracter unicode que corresponde à tradução de um elemento e separa-o com o caracter "|".
- **display_line**: imprime para a consola, recursivamente, o caracter unicode que corresponde à tradução de cada elemento de uma lista.
- **display_seperated_line**: à semelhança de display line, imprime para a consola, recursivamente, cada elemento de uma lista. No entanto, utiliza o predicado print_cell para cada elemento, resultado numa separação visual.
- **gen_line:** através de recursão, cria uma lista com um número de elementos especificado na qual todos os elementos têm o valor fornecido.
- **gen_column_labels:** através de recursão, cria uma lista com um número de elementos especificado na qual cada elemento é numerado e incrementado em 1 face ao elemento anterior. O valor do elemento inicial é especificado. Este predicado é utilizado para gerar os códigos unicode dos titulos das colunas do tabuleiro.
- **display_matrix:** utilizando os predicados acima referidos recursivamente, desenha as linhas do tabuleiro, numerando-as.

Exemplos

Inicio do jogo:



Durante o jogo:



Fim do jogo:

