

Frozen Forest

Relatório Intercalar

Programação em Lógica

(21 de Outubro de 2018)

Bruno Miguel da Silva Barbosa de Sousa

João Manuel Angélico Gonçalves

up201604145@fe.up.pt

up201604245@fe.up.pt

Índice

Descrição do jogo: “Frozen Forest”	2
Representação do Estado do Jogo	4
Visualização do Tabuleiro	6

O Jogo: “Frozen Forest”

Frozen Forest é um jogo de tabuleiro desenvolvido por **Néstor Romeral Andrés** com inspiração num jogo de Silvia Romeral Andrés.

Consiste num jogo de estratégia para 2 jogadores, que vão movimentando a sua peça alternadamente, até que não haja nenhum movimento possível, ou até que o tabuleiro não contenha árvores.

Enquanto que um jogador está a esconder-se (Mina), o outro está a tentar procurá-lo (Yuki). O espaço para onde quem procura se movimenta, vai imediatamente transformar-se num espaço vazio, de forma a diminuir os esconderijos possíveis para o outro jogador, sendo que a Mina está escondida se tiver um obstáculo (árvore) entre a linha de visão desta e de Yuki.

Yuki e Mina têm uma linha de visão clara se não houver nenhuma árvore na linha reta que conecta a localização dos 2 jogadores, sendo que a árvore aonde Mina se encontra não conta como um bloqueio da linha de visão.

O jogo vai ser organizado num tabuleiro com 10x10 espaços, com uma peça para cada jogador (a letra “m” a representar Mina e a letra “y” a representar Yuki), várias peças com a letra “w” que representam um espaço vazio (sem árvores ou jogadores), e vai ser completado por peças com letra “t” que vão corresponder á localização das árvores. Cada peça ocupa um espaço no tabuleiro.

Preparação

Inicialmente, os jogadores vão escolher o número de árvores presentes no tabuleiro (10x10, 9x9, ...), sendo que estas vão estar centradas no tabuleiro e, o resto do tabuleiro vai ser completado com espaços vazios.

De seguida, o jogador que está a controlar o Yuki (quem está a procurar) começa por colocar a sua peça em qualquer espaço no tabuleiro com uma árvore e depois, o outro jogador coloca a sua peça num espaço válido (fora da linha de visão do Yuki).

Objetivo

O objetivo deste jogo é fazer com que o outro jogador não tenha nenhuma casa possível para onde se movimentar ou, para Yuki, este também pode optar por tentar comer todas as árvores.

Movimentação de quem procura (Yuki):

Relativamente á movimentação de quem procura, este só se vai poder deslocar uma casa ortogonalmente ou diagonalmente, sendo que este movimento também terá de ser para uma peça que contenha uma árvore e que tenha uma linha de visão clara para Mina. À medida que este vai jogando, o número de casas vazias vai aumentando visto que Yuki vai comendo as árvores.



(Exemplo em que os movimentos válidos de Yuki (peça cinzenta) estão marcados com verde, os locais sem uma linha de visão clara para Mina (peça vermelha) estão representados por uma cruz vermelha, enquanto que os locais vazios estão marcados com uma cruz roxa.)

Movimentação de quem se esconde (Mina):

Quanto á movimentação do jogador que se esconde, este poderá movimentar-se o número de casas que desejar numa linha ortogonal ou diagonal. A única outra restrição no movimento de Mina é que apenas se pode deslocar para um espaço com a linha de visão bloqueada para Yuki.



(Exemplo em que os movimentos válidos de Mina (peça vermelha) estão marcados com verde e os locais com uma linha de visão clara para Yuki (peça cinzenta) estão identificados com uma cruz vermelha).

Fim

Tal como referido anteriormente, um jogador perde o jogo assim que não tenha nenhuma movimentação possível, aquando a sua vez de jogar.

Mas, quando o tabuleiro já estiver sem árvores, se nenhum jogador tiver perdido até lá, o vencedor é o jogador que estava a procurar (Yuki).

No caso de haver 2 jogos, em que os 2 jogadores alternaram a sua função:

- Se ambos os jogadores ganharam com o Yuki, ganha o que comeu menos árvores.
- Se ambos os jogadores ganharam com a Mina, ganha o que comeu mais árvores.
- Se um jogador ganhar os 2 jogos é claramente o vencedor.

Referência: [Rulebook](#)

Representação interna do estado do jogo

O tabuleiro será representado utilizando listas de listas (10x10) tendo os seguintes átomos como peças:

- t – Árvore
- w – Neve (Sitio onde foi comida árvore)
- y – Yuki
- m – Mina

Para além da representação do tabuleiro, será também guardado:

- O número de vitórias que cada jogador tem.
- O número de árvores que cada jogador comeu jogando como Yuki (para desempate)
- O “personagem” que cada jogador representa (Yuki/Mina)
- O próximo jogador a jogar (p1/p2)

Representação em Prolog

Estado Inicial

```
wins(0,0).
treesEaten(0,0).
players(y,m).
tab([[t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t]]).
nextPlayer(p1).
```

No início, todas as “casas” do tabuleiro serão árvores, e será o jogador1 (como Yuki) a jogar.

Estado Intermédio

```
wins(0,0).
treesEaten(8,0).
players(y,m).
tab([[t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,w,w,w,w,t,t,t],
     [t,t,t,t,t,t,w,t,t,t],
     [t,t,t,t,t,t,w,t,t,t],
     [t,t,t,t,t,t,w,y,t,t],
     [t,t,t,t,t,t,t,t,t,t],
     [t,t,t,t,t,t,t,t,m,t],
     [t,t,t,t,t,t,t,t,t,t]]).
nextPlayer(p2).
```

Ainda no primeiro jogo, após 8 jogadas, o Yuki comeu oito árvores (contando com o espaço onde está actualmente). Dado que tem linha de visão sobre a Mina ainda não perdeu. É a vez de a Mina jogar, terá que se esconder do Yuki.

Estado Final

```
wins(0,1).
treesEaten(40,32).
players(m,y).
tab([[t,t,t,t,t,w,t,w,t,t],
     [t,t,w,t,w,t,w,t,w,t],
     [t,w,t,w,t,w,w,w,t,t],
     [t,t,w,t,w,w,y,w,w,t],
     [t,t,t,w,t,w,w,w,t,t],
     [t,t,w,t,t,t,t,t,w,t],
     [t,w,t,m,t,t,t,w,t,t],
     [t,t,w,t,w,t,w,t,w,t],
     [t,t,t,w,t,w,t,w,t,t],
     [t,t,t,t,t,t,t,t,t,t]]).
nextPlayer(p2).
```

No primeiro jogo, a Mina acabou por conseguir escapar do Yuki, tendo o Yuki comido 40 árvores. Neste segundo jogo, é a vez do Yuki jogar. Dado que está preso (nenhuma casa à sua volta é uma árvore), perdeu o jogo. Sendo assim, o jogo ficaria empatado 1-1, sendo necessário recorrer ao número de árvores comida por cada jogador para desempatar. Como ganharam os dois jogadores jogando com a Mina, ganha quem comeu mais árvores. Assim sendo, o jogador 1 foi o vencedor.

Visualização do tabuleiro em modo de texto

Implementação *display_game(+Board,+Player)* em Prolog

```
display_game(Board,Player):-
    display_board(0,Board),
    format('~n~nPlayer to move: ~p ',Player),
    display_player(Player),
    nl.

display_board(Counter,[Head]):-
    write(' -----'),
    nl,
    format('~d ',Counter),
    display_line(Head),
    nl,
    write(' -----'),
    nl,
    write('  0  1  2  3  4  5  6  7  8  9 ').

display_board(Counter,[Head|Tail]):-
    write(' -----'),
    nl,
    format('~d ',Counter),
    display_line(Head),
    nl,
    Next is Counter+1,
    display_board(Next,Tail).
```

```
display_line([Head]):-
    format('~p| ',Head).

display_line([Head|Tail]):-
    format('~p| ',Head),
    display_line(Tail).

display_player(Player):-
    players(P1,P2),
    (Player=p1,
     write_name(P1));
    (Player=p2,
     write_name(P2)).
```

Output produzido:

```
| ?- tab(_t),nextPlayer(_n),display_game(_t,_n).
0 |t||t||t||t||t||t||t||t||t||t|
1 |t||t||t||t||t||t||t||t||t||t|
2 |t||t||t||t||t||t||t||t||t||t|
3 |t||t||t||t||t||t||t||t||t||t|
4 |t||t||t||t||t||t||t||t||t||t|
5 |t||t||t||t||t||y||t||t||t||t|
6 |t||t||t||t||t||t||t||t||t||t|
7 |t||t||t||t||t||t||t||t||t||t|
8 |t||t||t||t||t||t||t||m||t||t|
9 |t||t||t||t||t||t||t||t||t||t|
  0  1  2  3  4  5  6  7  8  9
```

Player to move: p1 playing as Yuki