**GARDENS OF IO**

Relatório Intercalar



Mestrado Integrado em Engenharia Informática

e Computação

**Grupo Gardens\_of\_Io\_1:**

Filipe Ribeiro – 201104129

Ricardo Neves – 201405868

Faculdade de Engenharia da Universidade do Porto Rua Roberto Frias, sn, 4200-465 Porto, Portugal

16 de Outubro de 2016

1. O Jogo “Gardens of Io”

Este jogo tem como cena, uma das luas de Júpiter, Io. Um grupo de marcianos viajou até lá com o objetivo de colonizar e embelezar a lua, através de um aperfeiçoamento das técnicas de jardinagem. Estas técnicas consistem em plantar flores, com sementes de crescimento instantâneo, de tal ordem que evitam passar por cima delas, evitando que elas se estraguem.

“Gardens of Io” é jogado no seguinte tabuleiro:

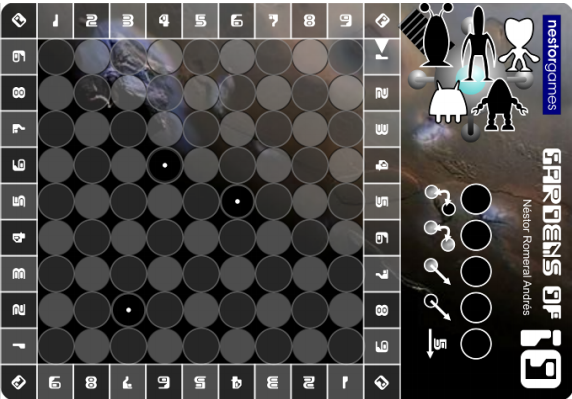


Figura 1 - Tabuleiro de jogo

O jogo pode ser jogado por 2-5 jogadores. Cada jogador escolhe um marciano diferente e os jogadores colocam 3 árvores no tabuleiro pela ordem de jogo (na imagem anterior podem ser observadas as representações de 3 árvores já em jogo). Dependendo do número de jogadores, os jogadores deverão tirar à sorte um determinado número de flores. Há 6 cores diferentes de flores: Branco, Amarelo, Azul, Roxo, Verde e Vermelho.

No seu turno cada jogador deverá, pela seguinte ordem:

1- O jogador poderá, uma vez por jogo, utilizar uma ação especial que ainda não tenha sido utilizada (as ações especiais são os 5 círculos ao lado do tabuleiro na imagem anterior). As ações especiais são todas diferentes. No caso de só haver dois jogadores, cada jogador poderá usar duas ações diferentes.

2- O jogador poderá colocar no tabuleiro qualquer flor que tenha na mão, desde que o caminho para a posição pretendida não esteja bloqueado por uma árvore (caminhos em linha reta ou na diagonal apenas).

3- O jogador ganha o número de pontos correspondente à quantidade de flores, da mesma cor que jogou, que estão juntas ortogonalmente.

4- O jogador move o seu marciano o número de casas correspondente ao número de pontos efetuados nessa jogada. Só contam as casas livres, ou seja, caso o jogador marque 1 ponto, mas na casa seguinte esteja um marciano diferente, o jogador avança duas casas. Caso o jogador complete uma volta, este pode pegar numa flor que já esteja no tabuleiro, exceto duma cor que já tenha sido esgotada, e pode adicioná-la à sua mão.

5- Se o jogador tiver jogado a última flor de uma certa cor, ou seja, já estão todas no tabuleiro, o jogador tem direito a um turno extra.

O jogo termina quando um jogador joga todas as suas flores, ou quando o jogador que está mais adiantado consegue dar uma volta de avanço ao mais atrasado. Ganha o jogador que estiver mais avançado, sendo que o número de voltas dadas conta.

1. Representação do Estado do Jogo

Representação do estado inicial do tabuleiro:

[ [‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’ , ’0’],

[‘9’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘1’],

[‘8’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘2’],

[‘7’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘3’],

[‘6’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘4’],

[‘5’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘5’],

[‘4’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘6’],

[‘3’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘7’],

[‘2’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘8’],

[‘1’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘9’],

[‘0’, ‘9’, ‘8’, ‘7’, ‘6’, ‘5’, ‘4’, ‘3’, ‘2’, ‘1’ , ’0’] ]

[action1, action2, action3, action4, action5]

Representação de um possível estado intermédio do tabuleiro:

[ [‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’ , ’0’],

[‘9’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘1’],

[‘8’, empty, **tree** , empty, empty, empty, **green**, empty, empty, empty, ‘2’],

[‘7’, empty, empty, empty, empty, empty, **green**, empty, **blue** , empty, ‘3’],

[**pl1**, empty, empty, empty, **tree** , empty, empty, empty, empty, empty, ‘4’],

[‘5’, empty, empty, empty, empty, empty, **green**, empty, empty, empty, ‘5’],

[‘4’, empty, empty, **red** , empty, empty, empty, empty, empty, empty, ‘6’],

[‘3’, empty, empty, **red** , empty, empty, **tree** , empty, empty, empty, ‘7’],

[‘2’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘8’],

[‘1’, empty, empty, empty, empty, empty, empty, empty, empty, empty, ‘9’],

[‘0’, **pl2**, **pl3**, ‘7’, ‘6’, ‘5’, ‘4’, ‘3’, ‘2’, ‘1’ , ’0’] ]

[action1, **empty**, **empty**, action4, action5]

1. Visualização do Tabuleiro

Aqui encontra-se a porção de código que vai fazer com que o tabuleiro seja visualizado na consola:

%Trocar varíaveis identificadoras pelos símbolos correspondentes%

getCellSymbol(emptyCell, ' ').

getCellSymbol(number0Cell, '0').

getCellSymbol(number1Cell, '1').

getCellSymbol(number2Cell, '2').

getCellSymbol(number3Cell, '3').

getCellSymbol(number4Cell, '4').

getCellSymbol(number5Cell, '5').

getCellSymbol(number6Cell, '6').

getCellSymbol(number7Cell, '7').

getCellSymbol(number8Cell, '8').

getCellSymbol(number9Cell, '9').

getCellSymbol(action1Cell, 'V').

getCellSymbol(action2Cell, 'W').

getCellSymbol(action3Cell, 'X').

getCellSymbol(action4Cell, 'Y').

getCellSymbol(action5Cell, 'Z').

%Visualizar o tabuleiro%

printBoard([]):-

nl, nl,

printRow(Actions).

printBoard([Head | Tail]):-

write('| '),

printRow(Head), nl,

printBoard(Tail).

printRow([]).

printRow([Head | Tail]):-

getCellSymbol(Head, Piece),

write(Piece), write(' | '),

printRow(Tail).

%Visualizar linha de ações%

printActionLine([]).

printActionLine([Head | Tail]):-

getCellSymbol(Head, Piece),

write(Piece), write(' | '),

printActionLine(Tail).

1. Movimento

Antes de cada movimento é necessário correr algumas verificações:

CheckNPlayers(Game).

CheckIfEmpty(Row1,Column1,Row2,Column2,Board).

CheckMarcianMove(Row1,Column1,Row2,Column2,Board).

CheckAction1(Player,Game).

CheckAction2(Player,Game).

CheckAction3(Player,Game).

CheckAction4(Player,Game).

CheckAction5(Player,Game).

Após a verificação necessária para o movimento correspondente, temos as seguintes possibilidades:

CountFlowers(Player,Board).

PlantFlower(Row,Column, Board).

IsLastFlowerPlant(Player).

PlantTree(Row, Column, Board).

SwapFlowers(Row1,Column1,Row2,Column2,Board).

SwapFlowerTree(Row1,Column1,Row2,Column2,Board).

MoveFlower(Row1,Column1,Row2,Column2,Board).

MoveTree(Row1,Column1,Row2,Column2,Board).

MoveMartian(Row1,Column1,Row2,Column2,Board).

IsLapDone(Player).

RemoveFlowers(Player,Board).