GoRoGo

Relatório Intercalar



Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

**Grupo Ordo\_1:**

Sérgio Salgado – 201406136

Nuno Manuel Ferreira Corte-Real – 201405158

Faculdade de Engenharia da Universidade do Porto

Rua Roberto Frias, sn, 42, 4200-465 Porto, Portugal

16 de Outubro de 2016

**Introdução**

O objectivo deste relatório é descrever a fase inicial da implementação do jogo GoRoGo em Prolog, uma linguagem de progamação em lógica. Como o nome indica, este jogo é baseado no clássico jogo de tabuleiro japonês Go, no entanto possui uma reviravolta na sua estratégia: 5 novas peças denominadas por “Henge” que oferecem uma miríade de possibilidades inéditas.

**O Jogo GoRoGo**

Desenvolvido por Doug Masiel, GoRoGo surgiu inicialmente como uma variante de menores dimensões e mais acessível a principiantes do clássico jogo secular japonês Go. No entanto, estas variantes acabavam por ser desprovidas de apelo e originalidade, visto que o jogo original é, já em si, bastante minimalista. Em 2016, ocorreu a Doug a ideia de adicionar peças neutras a um jogo Go de dimensões reduzidas. Denominou a estas adições “Henge” (pronunciado “Hain-gay”, ou  変化駒 em japonês Kanji). O nome alude à natureza volátil destas peças.

Após concebido, o jogo foi testado no Tokyo Game Market em Dezembro de 2016, onde foi universalmente aclamado e requisitado. GoRoGo mostra-se, portanto, como um excelente jogo de transição para o Go original para principiantes interessados.

**Regras**

GoRoGo é jogado por 2 jogadores num tabuleiro quadrado semelhante a um tabuleiro de Go (onde os vértices dos quadrados são as casas onde se colocam as peças) de 5x5. No total, existem 25 peças de jogo: 10 pretas, 10 brancas e 5 Henge. O objectivo principal do jogo é obter um número maior de peças do adversário (ao invés do Go normal, cujo objectivo é adquirir uma área maior que a do adversário). Como o jogo possui 25 peças, cada jogo de GoRoGo nunca excede 24 turnos. Um jogador não pode jogar uma peça Henge na última jogada. Um jogador não pode passar a a vez: caso esteja desprovido de qualquer jogada legal, perde.

Cada jogador possui 10 peças de uma das 2 cores e 2 peças Henge. As restantes peças Henge são colocadas no tabuleiro pelo jogador das peças brancas de modo a completar o tabuleiro inicial.

O jogador das peças pretas é o primeiro a jogar. Na primeira jogada, este pode colocar no tabuleiro ou uma peça preta ou uma peça Henge. Após a jogada inicial, as peças jogadas alternam entre pretas e brancas. Peça jogadas não podem ser movidas.Uma peça é capturada por um jogador quando não possuem nenhuma interseção à sua volta desocupada por uma peça do adversário. A exceção a esta regra são as peças Henge: estas adquirem a cor do jogador que está a jogar.

Um jogador não pode colocar uma peça numa casa rodeada por peças adversárias, a não ser que consiga capturar uma das peças que a rodeiam. No entanto, as peças Henge podem ser colocadas em qualquer interseção e não podem ser capturadas.

No final do jogo, se os jogadores empatarem, o jogador das peças brancas é considerado o vencedor.



Tabuleiro de GoRoGo, 5x5 num possível setup inicial (a peça Henge, colocada pelo jogador branco, poderia ocupar qualquer posição)



Captura de uma peça. A peça assinalada encontra-se com as interseções que a rodeiam bloqueadas (por peças adversárias e/ou por fronteiras do tabuleiro)



Outro exemplo de peça capturada, no turno do jogador preto (quando as peças Henge são consideradas pretas)

**Lógica do Jogo**

Para dar início ao jogo, será preciso um predicado para tal, invocado pelo jogador das peças brancas para que possa colocar a peça Henge inicial no tabuleiro.

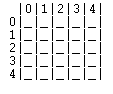
Será preciso um predicado para efetuar uma jogada, que será invocado alternadamente pelos dois jogadores. O predicado que faz uma jogada deverá receber informações sobre a cor da peça que o jogador quer colocar no tabuleiro, as coordenadas da casa na qual o jogador quer colocar a peça e deverá ainda validar a jogada. Para se realizar essa validação, dever-se-á criar um predicado específico para tal.

Será interessante existir um predicado que atualize o estado atual do tabuleiro, verificando se o jogo acabou. Para se verificar o fim do jogo, construir-se-á um predicado especializado nessa função, que deverá contar o número de peças de cada jogador, as peças adversárias que cada jogador capturou e deverá averiguar se algum jogador está inibido de realizar alguma jogada. Para estas 3 últimas funções mencionadas dever-se-ão construir 3 predicados especializados que as realizem, respetivamente.

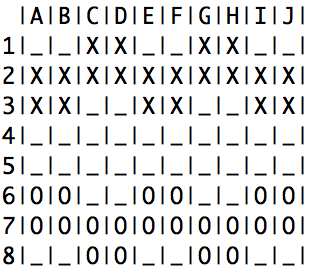
Deverá existir ainda um predicado que imprima o tabuleiro, um predicado de final de jogo e um predicado que sirva de menu/início do jogo.

**Representação do Estado de Jogo**

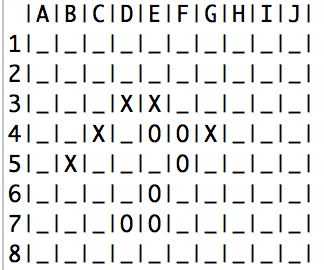
A representação do tabuleiro, em Prolog foi efectuada da seguinte maneira:



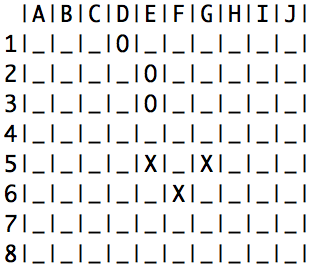
Acrescentando as peças de cada jogador, representamos deste modo o setup inicial de um jogo de GoRoGo:



Na imagem seguinte, vemos uma representação de uma possível situação intermédia no jogo, em que o jogador com as peças “X” será forçado a conectar novamente as suas peças:



Finalmente, a imagem seguinte mostra-nos uma situação de final do jogo, em que o jogador com as peças “O” consegue chegar com sucesso à linha inicial do adversário:



**Visualização do Tabuleiro**

O tabuleiro vazio pode ser visualizado invocando os seguinte predicado:

printboard.

Este predicado é constituído pelo seguinte código:

printboard :- board(X), write(' |0|1|2|3|4|5|6|7|8|9|'), nl, display\_board(X, 0).

O tabuleiro vazio é representado através do seguinte código:

board([[s, s, s, s, s],

[s, s, s, s, s],

[s, s, s, s, s],

[s, s, s, s, s],

[s, s, s, s, s]]).

O tabuleiro já pronto a jogar pode ser visualizado invocando o seguinte predicado:

printboardsetup.

Este predicado, por sua vez, é constituído pelo seguinte código:

printboardsetup :- boardsetup(X), write(' |0|1|2|3|4|5|6|7|8|9|'), nl, display\_board(X, 0).

Código relativo às funções relacionadas com a impressão dos vários tabuleiros:

display\_board([L1|Ls], N) :-

write(N),

N1 is N+1,

display\_line\_aux(L1), nl,

display\_board(Ls, N1).

display\_board([], \_) :- nl.

display\_line\_aux(L1) :- write('|'), display\_line(L1).

display\_line([E|Es]) :-

translate(E, V),

write(V),

write('|'),

display\_line(Es).

display\_line([]).

**Movimentos**

Como já foi previamente referido, para a movimentação serão usados os predicados singleton e ordo e os respectivos predicados para validação de movimentação validate\_singleton e validate\_ordo.

O predicado singleton requesita ao utilizado uma peça do tabuleiro através das suas coordenadas, e um aposição do tabuleiro de destino.

Uma peça é manipulada através do predicado get\_piece e suas funções auxiliares, realizadas com o seguinte código:

Definiram-se ainda as seguintes traduções para possível futuro uso:

translate(s, '\_').

translate(w, '|').

translate(t, '-').

translate(b, ' ').

translate(x, 'X').

translate(o, 'O').

translate(h, ‘H’).

translate(1, '1').

translate(2, '2').

translate(3, '3').

translate(4, '4').

translate(5, '5').

translate(6, '6').

translate(7, '7').

translate(8, '8').

translate(9, '9').

translate(0, '0').

translate(X, X).

**Bibliografia**

Referências:

https://www.kickstarter.com/projects/1287555371/gorogo-a-variation-on-the-classic-game-of-go https://www.kickstarter.com/projects/1287555371/gorogo-a-variation-on-the-classic-game-of-go

<https://boardgamegeek.com/boardgame/216789/gorogo>

<http://www.logygames.com/english/GoRoGo.html>

<https://www.youtube.com/watch?v=zAW9PfXKmTU>