

Faculdade de Engenharia da Universidade do Porto



Mestrado Integrado em Engenharia Informática e
Computação

Programação em Lógica

Grupo GoRoGo_4:
Bruno Piedade - up201505668
Danny Soares - up201505509

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

1. O Jogo GoRoGo

O GoRoGo é uma variante do jogo Go com origem na China há cerca de 2500 anos. O Go consiste num jogo de estratégia, com um tabuleiro de 19x19 (quadrados) em que dois jogadores jogam, alternadamente, peças pretas e brancas com o objetivo de dominar a maior área ao cercar as peças do oponente. O criador do GoRoGo, Doug Masiel, criou uma nova versão mais simples do Go tradicional, a dimensão do tabuleiro e introduzindo a peça Henge.

É um jogo para dois jogadores com um total de 25 peças (10 pretas, 10 brancas, 5 Henge) e jogado num tabuleiro 5x5 (quadrados). Como o jogo usa 25 peças nunca ultrapassará 24 jogadas, 12 de cada jogador. O objetivo para vencer é conquistar o maior número de peças ao adversário, cercando-lhe 1 ou várias peças. Porém não é possível jogar uma peça Henge como a última jogada e o jogador que não possa fazer uma jogada legal perde imediatamente.

O jogo começa com o tabuleiro vazio. Para iniciar, o jogador das peças brancas coloca a peça Henge excedente na posição que desejar. De seguida as jogadas são alternadas, sendo o jogador das peças pretas o primeiro a posicionar uma das suas 12 peças no tabuleiro.

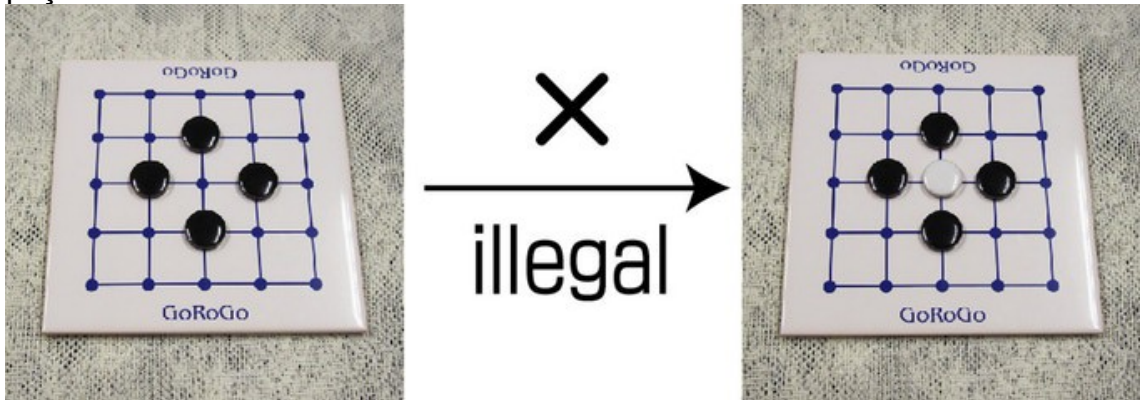
O jogo termina quando nenhum dos jogadores tem peças para jogar ou, se o jogador cujo turno está ativo não tiver nenhuma possibilidade de jogada, perde. Terminando por terem sido jogadas todas as peças, o vencedor é o jogador que tiver conquistado mais peças ao seu adversário. No caso de ocorrer um empate, o vencedor é o jogador das peças brancas.

Regras de jogo:

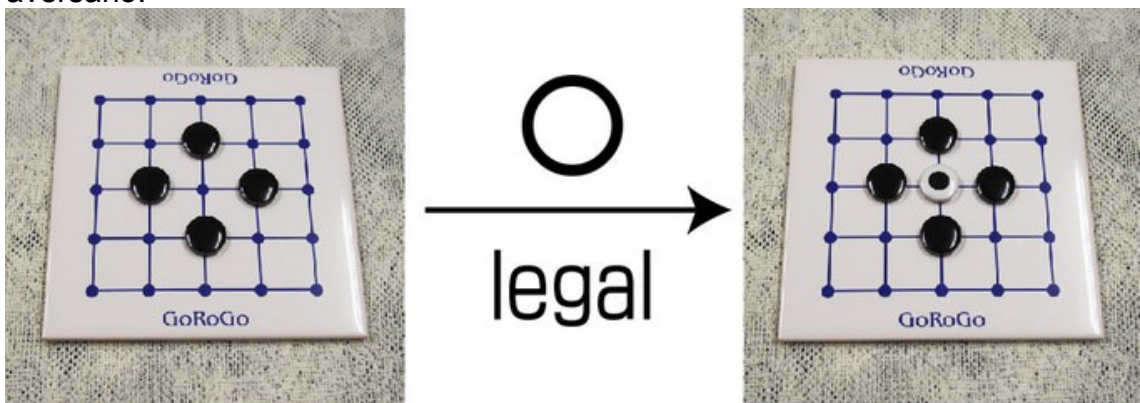
- Cada jogador começa com 10 peças do seu tipo e 2 Henge, sendo que a Henge excedente é a primeira a ser jogada.
- Peças Henge são consideradas pretas ou brancas de acordo com o turno do jogador. Ou seja, quando é o turno do jogador das peças brancas, consideram-se brancas, quando é o turno do jogador das peças pretas, consideram-se pretas.
- O jogador conquista peças do adversário quando estas estiverem cercadas pelas suas peças (incluindo os limites do tabuleiro) e estas peças são retiradas do tabuleiro. As peças Henge nunca podem ser capturadas.
- O jogador poderá colocar a sua peça em qualquer local do tabuleiro vazio com a seguinte exceção:
 - Não pode colocar uma peça regular num espaço que esteja cercado pelas peças do adversário, a menos que isso o leve a conquistar-lhe peças. Porém, as peças Henge podem ser colocadas em qualquer posição.
- As peças não podem ser movidas depois de colocadas.

Demonstração de algumas regras:

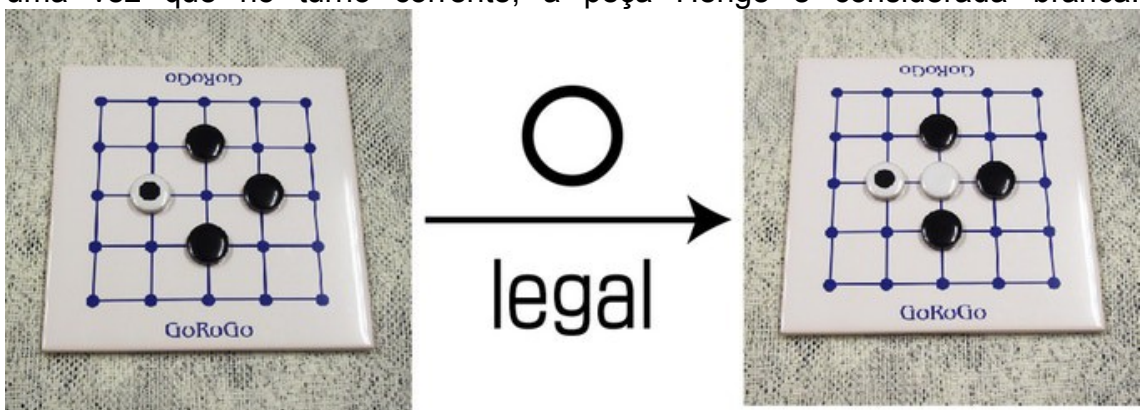
Não é permitido colocar uma peça regular num espaço cercado por peças do adversário.



É permitido colocar uma peça Henge num espaço cercado por peças do adversário.



Como a peça Henge é considerada branca no turno das peças brancas e preta no turno das peças pretas, é possível, na situação da imagem a baixo, colocar uma peça branca no espaço cercado por peças pretas e pela Henge, uma vez que no turno corrente, a peça Henge é considerada branca.



2. O Estado de Jogo

O tabuleiro de jogo será representado através de uma matriz (lista de listas) 5x5, utilizando algarismos (na matriz) e letras (na representação exibida ao utilizador) diferentes para representar as diferentes possibilidades para cada casa:

- 0 ou ' ' > casa vazia
- 1 ou 'B' > peça preta
- 2 ou 'W' > peça branca
- 3 ou 'H' > peça Henge.

O estado inicial do jogo é representado pela matriz preenchida com '0' em todas as posições:

gamestate([[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]]);
e a sua representação para o utilizador é:

As jogadas provocam alterações à matriz do estado inicial, colocando peças em posições do tabuleiro (estados intermédios). Na imagem a baixo está representado um possível estado intermédio, em que cada jogador posicionou 5 das suas peças e ainda nenhum jogador capturou peças do adversário, portanto ainda sobram ao jogador das peças brancas 5 peças normais e 2 Henge e ao jogador das peças pretas 7 peças normais e nenhuma Henge.

	W		H	H
W		W	B	
	H		B	
			W	
		W	B	

O jogo pode terminar em vários estados, seja por impossibilidade de o jogador posicionar a sua peça, seja por não haver mais peças para posicionar. A imagem a baixo representa uma das possibilidades, na qual as peças de ambos os jogadores se esgotaram e ambos os jogadores capturaram 4 peças.

	W		H	H
W		W	B	W
	H	B	B	B
	W	B	H	B
		W	H	

Tendo em conta que o jogo terminou empatado, como dizem as regras, o vencedor é o jogador das peças brancas.

3. Visualização do Tabuleiro

O tabuleiro é guardado na forma de uma matriz (lista de listas) de 5x5, em que cada posição (i,j) da matriz representa a posição respetiva no tabuleiro, considerando (0,0) a posição superior esquerda e (4,4) a posição inferior direita. Para visualizar o tabuleiro utilizamos os predicados presentes na imagem a baixo, em que “initialstate” é a matriz que representa o tabuleiro, “p_u”, “p_s”, “p_l” e “p_m” representam predicados que imprimem o tabuleiro e as peças já jogadas. O predicado “print_state” trata de chamar os predicados necessários para a visualização do tabuleiro.

```
initialstate([[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]]).

convert(Dado, Simbolo):- Dado = 0, Simbolo = ' '.
convert(Dado, Simbolo):- Dado = 1, Simbolo = 'B'.
convert(Dado, Simbolo):- Dado = 2, Simbolo = 'W'.
convert(Dado, Simbolo):- Dado = 3, Simbolo = 'H'.

p_u:- write(' '), nl.
p_s:- write(' | | | | | '), nl.
p_m([]).
p_m([L|T]):- p_l(L), p_m(T).
p_l([C|[]]):- convert(C,S),write(' | '), write(S),write(' | '), nl, p_s.
p_l([C|T]):- convert(C,S),write(' | '), write(S), write(' | '), p_l(T).
print_state:- initialstate(L), p_u, p_m(L).
```

4. Jogadas

As jogadas no GoRoGo são relativamente simples, consistindo apenas no posicionamento de uma peça numa posição livre e não cercada por peças adversárias. Para proceder ao posicionamento das peças utilizaremos o predicado “jogada/3”, que receberá como argumentos o número referente ao tipo de peça e as coordenadas “x” e “y” que representam a posição do tabuleiro onde se quer colocar a peça. Para verificar a validade da jogada utilizaremos o predicado “updateTable/2” que receberá como argumentos o estado atual do tabuleiro e o jogador que jogou.

5. Referências

- <https://www.kickstarter.com/projects/1287555371/gorogo-a-variation-on-the-classic-game-of-go> (“GoRoGo – An artisanal variation of the classic game Go”, Kickstarter)
- <https://pt.wikipedia.org/wiki/Go> (“Go”, Wikipedia)