

Jogo Cage

Relatório Final



Mestrado Integrado em Engenharia Informática e
Computação

Programação em Lógica

Grupo 2:

José Peixoto - 200603103

Luís Cruz - 201303248

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

8 de Novembro de 2016

Resumo

Resumo sucinto do trabalho com 150 a 250 palavras (problema abordado, objetivo, como foi o problema resolvido/abordado, principais resultados e conclusões).

Conteúdo

1	Introdução	4
2	O Jogo Cage	4
2.1	Regras	4
2.1.1	Objetivo	4
2.1.2	Movimentos	4
3	Lógica do Jogo	5
3.1	Representação do Estado do Jogo	5
3.1.1	Representação do estado inicial do tabuleiro:	5
3.2	Visualização do Tabuleiro	6
3.3	Lista de Jogadas Válidas	6
3.4	Execução de Jogadas	6
3.5	Avaliação do Tabuleiro	6
3.6	Final do Jogo	6
3.7	Jogada do Computador	6
4	Interface com o Utilizador	7
5	Conclusões	7
	Bibliografia	8
A	Nome do Anexo	8

1 Introdução

Descrever os objetivos e motivação do trabalho. Descrever num parágrafo breve a estrutura do relatório.

2 O Jogo Cage

O Cage é um jogo de estratégia em tabuleiro semelhante às damas que foi inventado por Mark Steere em maio de 2010. O autor descreve-o como um jogo para dois jogadores sem qualquer informação oculta. É um jogo abstrato sem fator de sorte nem empates. É jogado num tabuleiro de damas 10x10 ou 8x8 e, ao contrário do jogo original das damas, todo tabuleiro está preenchido, no início, com peças já promovidas a “damas”. “Jogo de aniquilação de alta energia” é a frase escolhida pelo autor para caricaturar o jogo, uma vez que o movimento para o centro do tabuleiro assegura a aniquilação, de pelo menos, uma das cores.

2.1 Regras

O Cage é jogado por dois jogadores num tabuleiro de damas com 50 damas vermelhas e 50 damas azuis na versão de tabuleiro 10x10 ou com 32 damas vermelhas e 32 damas azuis na versão de 8x8 tabuleiro. O tabuleiro é iniciado preenchendo todas as casas com damas de cor alternada.

2.1.1 Objetivo

Para vencer é necessário capturar todas as damas inimigas. No final, pode ganhar-se mesmo que se perca a última peça que se está a movimentar (saltar) para capturar todas as damas inimigas ainda em jogo.

2.1.2 Movimentos

Existem quatro tipos de movimentos:

1. Restrito
2. Centralizador
3. Adjacente
4. Salto

Durante um turno, um jogador apenas pode utilizar um tipo de movimento.

Restrição 1 Nunca se pode colocar uma dama ortogonalmente (horizontal ou verticalmente) adjacente a uma dama de cor idêntica. Nem de forma transitória durante um turno de vários movimentos.

Restrição 2 Nunca se pode movimentar uma dama que tenha adjacências ortogonais com damas inimigas para uma casa onde tal não aconteça.

Centralizador Este movimento de uma casa, permite à dama deslocar-se na horizontal, vertical ou diagonal para uma casa vazia e que permite que a dama se aproxime do centro do tabuleiro.

Adjacente Uma dama que não tenha adjacências ortogonais com damas inimigas pode mover-se apenas uma casa em qualquer direção que contenha adjacências ortogonais com uma ou mais damas inimigas.

Salto O movimento de salto permite capturar uma dama inimiga, movimentando a dama do jogador de uma casa ortogonalmente adjacente de um lado da dama inimiga para a casa vazia adjacente do lado oposto. É possível capturar uma dama inimiga nas casas periféricas do tabuleiro de uma casa adjacente e do lado oposto da dama inimiga na borda do tabuleiro. O resultado é que quer a dama capturada quer a dama que captura são removidas do tabuleiro.

3 Lógica do Jogo

Descrever o projeto e implementação da lógica do jogo em Prolog, incluindo a forma de representação do estado do tabuleiro e sua visualização, execução de movimentos, verificação do cumprimento das regras do jogo, determinação do final do jogo e cálculo das jogadas a realizar pelo computador utilizando diversos níveis de jogo. Sugere-se a estruturação desta secção da seguinte forma:

3.1 Representação do Estado do Jogo

Podem ser idêntico ao descrito no relatório intercalar.) Na representação de jogo usar-se-ão listas de listas que apenas incluem átomos para os diferentes tipos de peças (*red* e *blue*) e a casa vazia (*empty*). Para simplificação do desenvolvimento do jogo, escolheu-se a versão mais pequena do tabuleiro 8x8 com um total de 64 damas no início do jogo.

3.1.1 Representação do estado inicial do tabuleiro:

```
[ [blue,red,blue,red,blue,red,blue,red],  
  [red,blue,red,blue,red,blue,red,blue],  
  [blue,red,blue,red,blue,red,blue,red],  
  [red,blue,red,blue,red,blue,red,blue],  
  [blue,red,blue,red,blue,red,blue,red],  
  [red,blue,red,blue,red,blue,red,blue],  
  [blue,red,blue,red,blue,red,blue,red],  
  [red,blue,red,blue,red,blue,red,blue]  
]).
```

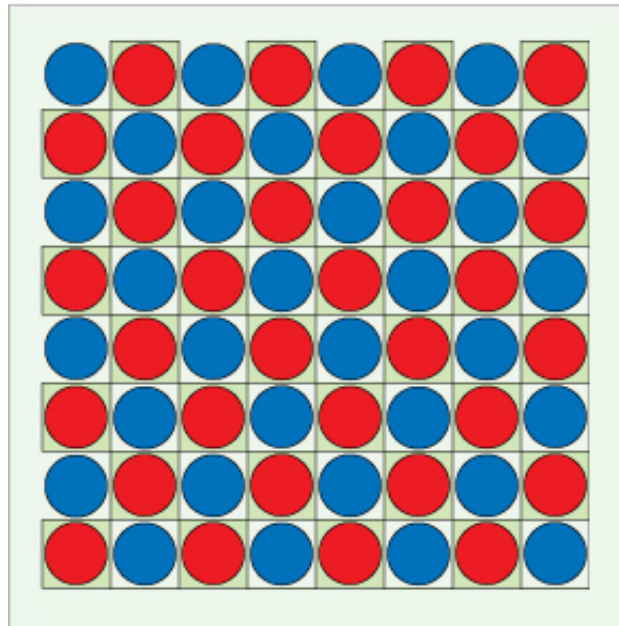


Figura 1: Estado inicial do jogo

3.2 Visualização do Tabuleiro

(Pode ser idêntico ao descrito no relatório intercalar.)

3.3 Lista de Jogadas Válidas

Obtenção de uma lista de jogadas possíveis. Exemplo: *valid_moves(+Board, -ListOfMoves)*.

3.4 Execução de Jogadas

Validação e execução de uma jogada num tabuleiro, obtendo o novo estado do jogo. Exemplo: *move(+Move, +Board, -NewBoard)*.

3.5 Avaliação do Tabuleiro

Avaliação do estado do jogo, que permitirá comparar a aplicação das diversas jogadas disponíveis. Exemplo: *value(+Board, +Player, -Value)*.

3.6 Final do Jogo

Verificação do fim do jogo, com identificação do vencedor. Exemplo: *game_over(+Board, -Winner)*.

3.7 Jogada do Computador

Escolha da jogada a efetuar pelo computador, dependendo do nível de dificuldade. Por exemplo: *choose_move(+Level, +Board, -Move)*.

4 Interface com o Utilizador

Descrever o módulo de interface com o utilizador em modo de texto.

5 Conclusões

Que conclui deste projecto? Como poderia melhorar o trabalho desenvolvido?

A Código fonte

Código Prolog implementado devidamente comentado e outros elementos úteis que não sejam essenciais ao relatório.