

# PROJETO Nº 2

## ÉPOCA NORMAL

### Resumo

Pretende-se desenvolver um programa, em LISP, para jogar o jogo dos pontos e das caixas, utilizando um algoritmo de teoria de jogos: o AlfaBeta. O programa deverá permitir jogar partidas entre um jogador humano e um computador, ou entre dois computadores.

## 1 JOGO DOS PONTOS E DAS CAIXAS

Trata-se de um jogo de tática para 2 jogadores. É constituído por um tabuleiro de 7 x 7 caixas. Cada caixa é delimitada por 4 pontos entre os quais é possível desenhar um arco. Quando os quatro pontos a volta de uma caixa estão conectados por 4 arcos, a caixa é considerada fechada. Uma jogada de um jogador consiste em desenhar um arco entre dois pontos adjacentes da sua escolha. Se este jogador conseguir fechar uma caixa com a sua jogada, poderá voltar a jogar. Senão, passa a ser a vez do adversário jogar. A figura 1 apresenta um exemplo do puzzle com 52 arcos conectados e 10 caixas fechadas. Na figura 1 o jogador 2, cujas jogadas estão representadas a tracejado, está em vantagem (relativa) sobre o jogador 1, cujas jogadas estão representadas por um traço contínuo, na medida em que ele conseguiu fechar 7 caixas até o momento, enquanto que o jogador 1 conseguiu fechar apenas 3 caixas.

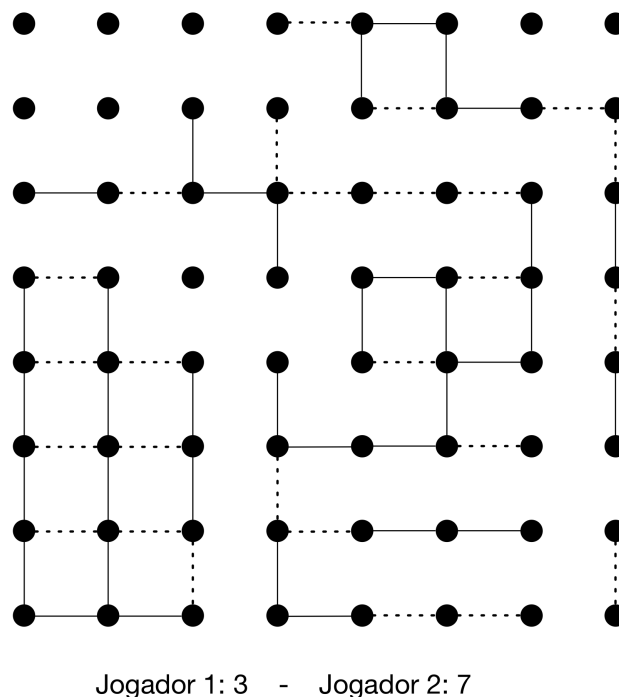


Figura 1: Ilustração de um exemplo do jogo dos pontos e das caixas

## Modos de Funcionamento

O programa deveria funcionar em 2 modos: 1) Máquina vs. Humano e 2) Máquina vs. Máquina. No modo 1, no início de cada partida, o utilizador deve decidir 1) quem começa, computador ou humano, e 2) qual o tempo limite para o computador jogar. As jogadas são efetuadas alternadamente até um dos jogadores vencer a partida. À medida que os jogadores vão fechando caixas, as suas pontuações são atualizadas pelo programa do jogo que mostra os resultados. Quando já não existem caixas por fechar, o jogo termina. O jogador que ganha a partida é o jogador que tiver fechado o maior número de caixas.

## 2 OBJETIVOS DO PROJETO

Pretende-se desenvolver um programa, em LISP, para jogar o jogo dos pontos e das caixas entre um jogador humano e um PC. O programa deverá implementar o algoritmo AlfaBeta e as funções auxiliares que irão permitir realizar partidas do jogo. Pretende-se uma implementação do Alfa-Beta que permite ao computador vencer o jogador humano ou um outro computador. No âmbito deste projeto, será também realizado um campeonato entre os programas entregues pelos alunos. Este campeonato é descrito na secção 8. O campeonato corresponde ao modo 2 do jogo. Para isso, o código deverá ser agrupado em packages, onde os símbolos serão repartidos. Neste 2º modo, será necessário disponibilizar uma função acessível do exterior para exportar o símbolo. A interface com esta função (argumentos e seu significado) será disponibilizadas mais à frente.

## 3 NOTAÇÃO DO JOGO

### 3.1 Tabuleiro

O tabuleiro é representado sob a forma de uma lista composta por duas listas. A primeira destas três listas representa os arcos horizontais, a segunda lista os arcos verticais. A Figura 3 apresenta uma representação em Lisp do tabuleiro vazio Disponível no início de uma partida. A figura 3 ilustra a representação em Lisp do tabuleiro da figura 1 que corresponde a um exemplo de tabuleiro no decorrer de uma partida. Nas duas listas que representam os arcos,

- O valor [NIL] representa a ausência de arco na posição,
- O valor [1] representa a presença de um arco colocado pelo jogador 1 nessa posição,
- O valor [2] representa a presença de um arco colocado pelo jogador 2 nessa posição.

### 3.2 Movimentos

Um movimento de colocação é considerado possível se a posição de destino pertence ao tabuleiro *E* se ainda não existe um arco nessa posição. Cada movimento de colocação é composto por 4 parâmetros:

- Para a colocação do arco na horizontal:
  - o primeiro parâmetro representa o número da linha onde desenhar o arco (entre 1 e 8),
  - o segundo representa o número da coluna onde desenhar o arco (entre 1 e 8),
  - o terceiro representa o tabuleiro ao qual aplicar o operador,

```

1      (
2          ((NIL NIL NIL NIL NIL NIL NIL) (NIL NIL NIL NIL NIL NIL NIL)
3          (NIL NIL NIL NIL NIL NIL NIL) (NIL NIL NIL NIL NIL NIL NIL)
4          (NIL NIL NIL NIL NIL NIL NIL) (NIL NIL NIL NIL NIL NIL NIL)
5          (NIL NIL NIL NIL NIL NIL NIL) (NIL NIL NIL NIL NIL NIL NIL))
6          ((NIL NIL NIL NIL NIL NIL NIL) (NIL NIL NIL NIL NIL NIL NIL)
7          (NIL NIL NIL NIL NIL NIL NIL) (NIL NIL NIL NIL NIL NIL NIL)
8          (NIL NIL NIL NIL NIL NIL NIL) (NIL NIL NIL NIL NIL NIL NIL)
9          (NIL NIL NIL NIL NIL NIL NIL) (NIL NIL NIL NIL NIL NIL NIL))
10     )

```

Figura 2: Exemplo de representação em LISP do tabuleiro vazio do jogo

```

1      (
2          ((NIL NIL NIL 2 1 NIL NIL) (NIL NIL NIL NIL 2 1 2)
3          (1 2 1 2 2 2 NIL) (2 NIL NIL NIL 1 2 NIL)
4          (2 2 NIL NIL 2 1 NIL) (2 2 NIL 1 1 2 NIL)
5          (2 2 NIL 2 1 1 NIL) (1 1 NIL 1 2 2 NIL))
6          ((NIL NIL NIL 1 1 1 1) (NIL NIL NIL 1 1 1 1)
7          (NIL 1 NIL NIL 1 1 2) (NIL 2 1 NIL 1 2 1)
8          (1 NIL NIL 1 NIL NIL NIL) (1 NIL NIL 1 1 NIL NIL)
9          (NIL NIL 1 1 NIL NIL NIL) (NIL 2 1 2 1 NIL 2))
10     )

```

Figura 3: Exemplo de representação em LISP do tabuleiro do jogo ilustrado na figura 1

- o quarto representa o símbolo a colocar [1;2], que corresponde ao jogador que fez a jogada.
- Para a colocação do arco na vertical,
  - o primeiro parâmetro representa o número da coluna onde desenhar o arco (entre 1 e 8),
  - o segundo representa o número da linha onde desenhar o arco (entre 1 e 8),
  - o terceiro representa o tabuleiro ao qual aplicar o operador,
  - o quarto representa o símbolo a colocar [1;2], que corresponde ao jogador que fez a jogada.

Consideramos a origem das coordenadas como sendo o canto superior esquerdo do tabuleiro.

### 3.3 Estrutura do programa

O programa deverá estar dividido em três partes:

- O algoritmo deverá constar de um ficheiro `alfabeta.LISP`
- O ficheiro `jogo.LISP` deve conter as funções que permitem escrever e ler de ficheiros e tratar da interação com o utilizador.

- Os operadores do jogo devem ser escritos num ficheiro com o nome pontosecaixas.LISP

Enquanto que a primeira parte do programa deverá ser genérica para qualquer jogo que recorre ao algoritmo AlfaBeta, a segunda e a terceira parte são específicas ao jogo dos pontos e das caixas.

## 4 ALGORITMO

O algoritmo de jogo a implementar é o AlfaBeta. Antes de devolver cada resultado, o programa deverá acrescentar uma linha a um ficheiro LOG.dat com a jogada realizada (na forma acima referida), o valor da posição para onde jogou, o número de nós analisados, o número de cortes efetuados (de cada tipo) e o tempo gasto. Deve ainda escrever a mesma informação no ecrã. O programa deverá então ler a jogada do adversário, a partir do teclado quando se trata do jogador humano, na forma convencionada, e repetir o procedimento até um dos jogadores ganhar a partida.

O tempo limite para o computador jogar será de X milissegundos. O valor de X deverá ter um valor entre 1000 e 5000 milissegundos mas o valor exato só será comunicado 7 dias antes da data de entrega do projeto.

## 5 GRUPOS

Os projetos deverão ser realizados em grupos de duas pessoas, sendo contudo sempre sujeitos a avaliação oral individual. Em casos excepcionais poderá haver trabalhos realizados individualmente.

## 6 DATAS

**Entrega do projeto :** 30 de janeiro de 2017, até as 08:00 da manhã.

## 7 DOCUMENTAÇÃO A ENTREGAR

A entrega do projeto e sua documentação deverá ser feita através do Moodle, na zona do evento “entrega do primeiro projeto”. Todos os ficheiros a entregar deverão ser devidamente arquivados num ficheiro comprimido (zip com um tamanho máximo de 5Mb), até à data acima indicada. O nome do arquivo deve seguir a nomenclatura seguinte:

nomeAluno1\_numeroAluno1\_nomeAluno2\_numeroAluno2\_P2.zip

Para além de entregar os ficheiros de código, é necessário elaborar 2 manuais (o manual de utilizador e o manual técnico) com a estrutura habitual, que deverão constar em dois ficheiros escritos em formato PDF. Não será preciso entregar uma versão impressa dos relatórios.

### Ficheiros com Código

O programa deverá estar dividido em três partes: uma parte para implementar o algoritmo AlfaBeta, independente do domínio de aplicação, outra para implementar o jogo incluindo a definição

dos operadores específicos do domínio de aplicação, e a terceira parte para fazer a interação com o utilizador e para proceder à escrita e leitura de ficheiros.

**jogo.lisp** Carrega os outros ficheiros de código, escreve e lê de ficheiros e trata da interação com o utilizador.

**pontosecaixas.lisp** Código relacionado com o problema. (operadores, ...)

**alfabeta.lisp** Deve conter a implementação do algoritmo AlfaBeta

## 7.1 Manuais

**ManualTecnico.pdf** O Manual Técnico deverá conter o algoritmo geral devidamente comentado; descrição dos objetos que compõem o projeto, incluindo dados e procedimentos; identificação das limitações e opções técnicas. Deverá ser apresentada uma análise crítica dos resultados das execuções do programa onde deverá transparecer a compreensão das limitações do projeto. Deverão fazer uma análise estatística sobre um conjunto de 5 execuções do programa onde constará os vencedores dos jogos realizados. Poderão utilizar os dados do ficheiro LOG.dat para isso.

**ManualUtilizador.pdf** O Manual do Utilizador deverá conter a identificação dos objectivos do programa, e descrição geral do seu funcionamento; explicação da forma como se usa o programa (acompanhada de exemplos); descrição da informação necessária e da informação produzida (ecrã/teclado e ficheiros); limitações do programa, do ponto de vista do utilizador, de natureza não técnica.

## 8 CAMPEONATO

Após a entrega dos projetos, será realizado um campeonato do jogo do pontos das caixas entre os programas de cada grupo (os alunos deverão manifestar a sua intenção de participar, fazendo um registo no Moodle). Os dois primeiros lugares deste campeonato receberam um bónus de pontos na nota final do projeto 2; respectivamente + 3 valores para os alunos do grupo vencedor e + 2 valores para os alunos do grupo que acaba na 2ª posição. Todos os outros grupos que participam no projeto e cujo programa correrá no programa do campeonato sem bloquear irá receber um bónus de +1 valor, mesmo que percam sempre.

## 9 AVALIAÇÃO

A avaliação do projeto levará em linha de conta os seguintes aspetos:

- Data de entrega final – Existe uma tolerância de uma semana em relação ao prazo de entrega, com a penalização de um valor por cada dia de atraso. Após uma semana de atraso a nota do projeto será zero.
- Correção processual da entrega do projeto - (Moodle, manuais e ficheiros de entrega no formato correto). Anomalias processuais dão origem a uma penalização que pode ir até 3 valores.
- Qualidade técnica - Objectivos atingidos; Código correto; Facilidade de leitura e manutenção do programa; Opções técnicas corretas.
- Qualidade da documentação - Estrutura e conteúdo dos manuais que acompanham o projeto.

- Avaliação oral - Eficácia e eficiência da exposição; compreensão das limitações e possibilidades de desenvolvimento do programa. Nesta fase poderá haver lugar a uma revisão total da nota de projeto.
- Avaliação escrita de LISP - Confirmação da capacidade de compreensão e desenvolvimento de código em LISP.

Tabela 1: Grelha de classificação.

Algoritmo AlfaBeta com cortes alfa e os cortes beta	6
Função de avaliação dos tabuleiros	4
Implementação do limite de tempo para o computador	2
Operadores do jogo	2
Ordenação dos nós	1
Qualidade da Codificação	1
Relatórios (Utilizador e Técnico)	4

Com este projeto pretende-se motivar o paradigma de programação funcional. A utilização de variáveis globais, de instruções de atribuição do tipo set, setq, setf, de ciclos, de funções destrutivas ou de quaisquer funções com efeitos laterais é fortemente desincentivada dado que denota normalmente uma baixa qualidade técnica.