

# Projeto Nº 2: Época Especial

---

Inteligência Artificial - Escola Superior de Tecnologia de Setúbal  
2019/2020

Prof. Joaquim Filipe  
Eng. Filipe Mariano

## 1. Jogo da Torre: Descrição Geral

O Jogo da Torre é uma variante do problema matemático conhecido como o Passeio do Cavalo, cujo objetivo é, através dos movimentos do Cavalo, visitar todas as casas possíveis de um tabuleiro similar ao de xadrez. Nesta versão a peça de xadrez utilizada é a **Torre** e decorrerá num tabuleiro de **7** linhas e **7** colunas (7x7), em que cada casa possui uma pontuação. Nesta secção pretende-se dar uma perspetiva geral do que é este jogo, deixando para a secção seguinte a explicação do que se pretende que seja desenvolvido no projeto de Inteligência Artificial na vertente de Teoria de Jogos.

### 1.1. Tabuleiro

O jogo possui as seguintes características:

- O tabuleiro tem as dimensões 7x7 em que os valores de cada casa são entre **11** e **77**, sem repetição e sem números que incluam os dígitos 0, 8 e 9. Por exemplo, 59 é um valor que não pode existir.
- Cada vez que é iniciado um jogo é construído um novo tabuleiro com o valor das casas distribuído aleatoriamente.
- O objectivo do jogo é acumular mais pontos que o adversário, usando uma Torre de xadrez. Cada jogador tem uma Torre da sua cor (branco ou preto).

### 1.2. Desenrolar do Jogo

- O jogo começa com a colocação da Torre branca numa casa da **1ª linha (A1-G1 do tabuleiro)**. Esta casa é escolhida pelo jogador com a Torre branca.
- Se a casa escolhida tiver um número com dois dígitos diferentes, por exemplo 57, então, em consequência, o número simétrico 75 é apagado do tabuleiro, tornando esta casa inacessível durante o resto do jogo. Ou seja, nenhuma Torre pode terminar outra jogada nessa casa.
- Se uma Torre for colocado numa casa com um número "duplo", por exemplo 66, então **deve ser removido o número duplo de maior valor**. Depois de um jogador deixar a casa para se movimentar para outra, a casa onde estava fica também inacessível para o jogo, ficando o número da casa apagado.
- Após a primeira jogada (colocar Torre branca) segue-se a jogada do adversário com colocação da Torre preta numa casa da **7ª linha (A7-G7)**.
- Depois de ambas as Torres serem colocadas, todas as jogadas seguintes são efectuadas através de um movimento da Torre (usando as regras tradicionais do Xadrez para essa peça). **Uma Torre não pode saltar para uma casa vazia (sem número) e também não pode fazê-lo para uma casa que esteja ameaçada pela Torre adversária.**
- A cada jogada de um jogador repete-se a regra do simétrico ou duplo.
- Um jogador ganha pontos por cada casa visitada pela sua Torre (igual ao valor da casa). Os pontos são contabilizados apenas pelas casas visitadas, não pelos números simétricos ou duplos removidos.

### 1.3. Determinar o Vencedor

O jogo termina quando não for possível movimentar qualquer uma das Torres no tabuleiro, sendo o vencedor o jogador que ganhou **mais** pontos.

## 2. Fase do Projeto: Versão Dois Jogadores

### 2.1. Objetivo

No âmbito deste projeto vamos considerar o Jogo da Torre na versão de dois jogadores, possibilitando um enquadramento teórico-prático com os conhecimentos adquiridos no âmbito da Teoria de Jogos. Neste caso aplicam-se as regras de jogo descritas na Secção 1.

Tal como na 1ª fase do Projecto, pretende-se que os alunos desenvolvam um programa, em **Lisp**. O programa deverá implementar o algoritmo AlfaBeta ou Negamax com cortes alfa-beta, e as funções auxiliares que permitirão realizar as partidas do jogo.

Pretende-se que o programa permita ao computador vencer o jogador humano ou um outro computador, pelo que deverá funcionar em dois modos:

1. Humano vs Computador
2. Computador vs Computador

No modo 1, no início de cada partida, o utilizador deve decidir quem começa, humano ou computador, e qual o tempo limite para o computador jogar, enquanto que no modo 2 apenas é necessário definir o tempo limite.

O jogo é iniciado pelo Jogador 1 colocando a Torre branca na casa de maior valor na 1ª linha do tabuleiro. O processo é análogo para o Jogador 2, que coloca a Torre preta na casa de maior valor da 7ª linha do tabuleiro. A cada jogada, mediante a casa em que se colocou a Torre, é aplicada a regra do número simétrico ou a regra do número duplo.

Caso um dos jogadores não consiga movimentar a Torre, cede a vez ao jogador oposto. O jogo termina quando nenhum jogador consegue movimentar a Torre.

### 2.2. Funcionamento

No início de cada partida, o utilizador deve decidir quem começa, humano ou computador, e qual o tempo limite para o computador jogar, o qual será de **X** milissegundos. O valor de **X** deverá ser um valor compreendido entre 1000 e 5000 milissegundos.

No que diz respeito ao jogador humano, o programa deverá ler a jogada inserida através do teclado (coordenadas da linha e coluna na forma <letra> <numero de 1 a 7> - exemplo: **G5**). Depois disso, o computador deve apresentar a sua própria jogada no ecrã (no mesmo formato em que lê a do humano) e repetir o procedimento até a partida terminar. O programa deverá ainda ir escrevendo num ficheiro **log.dat** em cada jogada: o número de nós analisados, o número de cortes efetuados (de cada tipo), o tempo gasto em cada jogada e o tabuleiro atual.

## 3. Formulação do Problema

### 3.1. Tabuleiro

O tabuleiro é representado sob a forma de uma lista composta por 7 outras listas, cada uma delas com 7 átomos. Cada uma das listas representa uma linha do tabuleiro, enquanto que cada um dos átomos possui o valor da respetiva casa. Por outras palavras, o tabuleiro é representado por uma lista de listas em LISP, sendo que cada átomo representa uma casa em que a linha corresponde ao índice da sublista e a coluna ao índice do átomo dentro da linha. Cada célula do tabuleiro poderá ser representada pelo seguinte valor:

- De 11 a 77 significa que a casa ainda não foi visitada;
- NIL significa que a casa já foi visitada;
- -1 significa que a Torre branca (Jogador 1) se encontra nessa casa;
- -2 significa que a Torre preta (Jogador 2) se encontra nessa casa.

O estado do jogo é representado por um tabuleiro. Um nó no grafo de jogo corresponde a um estado.

### 3.2. Estrutura do Programa

O programa deverá estar dividido em três partes, cada uma num ficheiro diferente:

1. Uma parte para o algoritmo AlfaBeta ou Negamax (`algoritmo.lisp`).
2. Outra que deve conter as funções que permitem escrever e ler em ficheiros e tratar da interação com o utilizador (`interact.lisp`).
3. E a terceira parte corresponde aos operadores do jogo (`jogo.lisp`).

Enquanto que a primeira parte do programa deverá ser genérica para qualquer jogo que recorre ao algoritmo AlfaBeta ou Negamax com cortes alfa-beta (independente do domínio de aplicação), a segunda e a terceira parte são específicas do Jogo da Torre (dependente do domínio de aplicação).

Na parte 2 deverá ser definida uma função com o nome `jogar` com os parâmetros `estado` e `tempo`, e que devolva uma lista com o tabuleiro em que o deverá ser feita a próxima jogada.

## 4. Grupos

Os projetos deverão ser realizados em grupos de, no máximo, duas pessoas sendo contudo sempre sujeitos a avaliação oral individual para confirmação da capacidade de compreensão dos algoritmos e de desenvolvimento de código em Lisp.

Os grupos deverão ser os mesmos que realizaram a 1ª fase do Projeto.

## 5. Datas

**Entrega do projeto:** 7 de Setembro de 2020, até as 23:55.

## 6. Documentação a Entregar

A entrega do projeto e da respetiva documentação deverá ser feita através do Moodle, na zona do evento "Entrega do Projeto 2 de Época Especial". 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 estrutura `nomeAluno1_numeroAluno1_nomeAluno2_numeroAluno2_E2`.

### 6.1. Código Fonte

Os ficheiros de código devem ser devidamente comentados e organizados da seguinte forma:

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

**jogo.lisp** Código relacionado com o problema.

**algoritmo.lisp** Deve conter a implementação do algoritmo de jogo independente do domínio.

## 6.2. Manuais

No âmbito da Unidade Curricular de Inteligência Artificial pretende-se que os alunos pratiquem a escrita de documentos recorrendo à linguagem de marcação **Markdown**, que é amplamente utilizada para os ficheiros **ReadMe** no **GitHub**. Na Secção 4 do guia de Laboratório nº 2, encontrará toda a informação relativa à estrutura recomendada e sugestões de ferramentas de edição para **Markdown**.

Para além de entregar os ficheiros de código, é necessário elaborar e entregar 2 manuais (o manual de utilizador e o manual técnico), em formato **PDF** juntamente com os sources em **MD**, incluídos no arquivo acima referido:

**ManualTecnico.pdf** O Manual Técnico deverá conter o algoritmo implementado, devidamente comentado e respetivas funções auxiliares; descrição dos tipos abstratos usados no programa; 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. Deverá ser apresentada uma análise estatística de uma execução do programa contra um adversário humano, indicando os elementos solicitados na secção 2.2. Poderão utilizar os dados do ficheiro **log.dat** para isso.

**ManualUtilizador.pdf** O Manual do Utilizador deverá conter a identificação dos objetivos do programa, juntamente com 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).

## 7. Avaliação

Tabela 1: Grelha de classificação.

Funcionalidade	Valores
Algoritmo AlfaBeta ou Negamax com Cortes alfa-beta e determinar Jogada do Computador	5
Jogada Humano	1
Apresentar Estatísticas a cada Jogada (ecrã e ficheiro)	2
Função de Avaliação	2
Implementação do limite de tempo para o computador	1
Procura quiescente	1
Memoização	1
Operadores do jogo	2
Ordenação dos nós	1

Funcionalidade	Valores
Qualidade do código	2
Manuais (utilizador e técnico)	2
<b>Total</b>	<b>20</b>

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

- Data de entrega final – Existe uma tolerância de 2 dias em relação ao prazo de entrega, com a penalização de 1 valor por cada dia de atraso. Findo este período a nota do projeto será 0.
- Correção processual da entrega do projeto – (Moodle; manuais no formato correto). Anomalias processuais darão origem a uma penalização que pode ir até 3 valores.
- Qualidade técnica – Objetivos 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.

## 8. Recomendações Finais

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. Exceptua-se a utilização de `setf` em conjugação com `gethash`.

**ATENÇÃO:** Suspeitas confirmadas de plágio serão penalizadas com a anulação de ambos os projetos envolvidos (fonte e destino), e os responsáveis ficam sujeitos à instauração de processo disciplinar