



INSTITUTO SUPERIOR TÉCNICO

LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES

PROJETO DE INTRODUÇÃO À ARQUITETURA DE COMPUTADORES

# MASTER-MIND

PARTE II



2017 / 2018

## ÍNDICE

Objetivo.....	3
Descrição do Jogo.....	3
Espaço de Jogo .....	3
Periféricos .....	3
Início do Jogo.....	3
Fim do Jogo .....	4
Implementação .....	4
Representação do código e da jogada .....	4
Temporizações .....	4
Valores aleatórios.....	4
Qualidade do código .....	4
Dúvidas na especificação do enunciado.....	4
Plano de Entrega .....	5
Dia 30 de Novembro, até às 23h59, no Fénix .....	5
Semanas de 4 e 11 de Dezembro, na aula de laboratório.....	5

## OBJETIVO

O objetivo da segunda parte do projeto consiste no desenvolvimento da interface de jogo com o jogador. Deve ser usado o motor de jogo desenvolvido na primeira parte do projeto. O jogo decorre numa janela de texto onde vão sendo mostradas as jogadas e o resultado.

Neste documento são descritos os detalhes de funcionamento pretendidos para o jogo. O jogo será programado usando a linguagem Assembly para o P3. O desenvolvimento e teste do programa serão realizados usando o simulador do P3 (p3sim), sendo usados os diversos recursos disponibilizados, como por exemplo a janela de texto, os mostradores (*displays*) de 7 segmentos, os LEDs, e o mostrador LCD.

O projeto deverá ser demonstrado no laboratório com a versão física (*hardware*) do P3.

## DESCRIÇÃO DO JOGO

O jogo mantém a especificação descrita no enunciado da primeira parte do projeto. Tal como nessa especificação, cada jogo termina quando o jogador D descobre a sequência secreta ou deixa de ter posições no tabuleiro para jogar. Mas, ao contrário da primeira parte, após o final de cada jogo inicia-se um novo jogo com um novo código escolhido de forma aleatória.

Mais detalhes sobre o jogo original e variantes podem ser encontradas aqui:

[https://en.wikipedia.org/wiki/Mastermind\\_\(board\\_game\)](https://en.wikipedia.org/wiki/Mastermind_(board_game))

## ESPAÇO DE JOGO

O jogo irá desenrolar-se na janela de texto do simulador do P3. Caso ainda não tenha implementado a saída (output) do programa pela janela de texto, deverá fazê-lo agora na segunda parte.

## PERIFÉRICOS

Para além da funcionalidade principal do jogo na janela de texto, são usados também os seguintes dispositivos periféricos:

- O LCD deverá mostrar, na primeira linha, a melhor pontuação no final de cada jogo obtida até ao momento.
- Os displays de 7 segmentos deverão indicar a pontuação atual (número total de jogadas realizadas até ao momento em cada jogo).
- Os leds são usados para indicar o tempo remanescente para realizar a jogada. Em cada jogada, os LEDs iniciam todos acesos, e vão-se apagando sucessivamente a cada 500ms. O tempo de jogada esgota-se quando não existir mais nenhum LED aceso.
- O botão IA permite iniciar um novo jogo.
- Os botões de I1 até I6 permitem introduzir a jogada do jogador D.

## INÍCIO DO JOGO

Quando arranca, o programa deverá escrever a mensagem “Carregue no botão IA para iniciar”. Quando esse botão for premido:

- a janela de texto deverá ser limpa (através da escrita de várias linhas em branco);
- a pontuação atual deve ser colocada a 0;
- o programa fica a aguardar a jogada que será inserida pelo utilizador.

## FIM DO JOGO

Cada jogo termina quando o jogador D acertar na sequência secreta, ou expirar o tempo máximo para fazer uma jogada, ou atingir o número máximo de jogadas. Nesta altura, o programa deverá escrever as mensagens “Fim do Jogo” e “Carregue em IA para recomeçar”. Quando o botão IA for premido, deve-se iniciar um novo jogo.

## IMPLEMENTAÇÃO

### REPRESENTAÇÃO DO CÓDIGO E DA JOGADA

Na segunda parte do projeto há uma série de requisitos adicionais sobre a representação do código e da jogada nos registos R1 e R2:

- Deverá fazer a compressão da representação do código, jogada e resultado usando um máximo 12 bits (para cada caso);
- Deverá armazenar esta informação em variáveis mantidas em memória, ou seja, usar os registos para guardar variáveis temporárias e a memória para variáveis globais. Naturalmente, do ponto de vista de eficiência da execução do código, deverá trazer as variáveis da memória (ou da pilha) para os registos antes destas serem manipuladas (usualmente no início de uma rotina), e voltar a escrevê-las em memória após a sua utilização (usualmente no fim da rotina). Da mesma forma que, na primeira parte, cada variável teria de ser representada num único registo de 16 bits, terá agora a restrição que cada uma destas variáveis não deve ocupar mais do que uma palavra em memória;

### TEMPORIZAÇÕES

A contagem do tempo de cada jogada é controlado através do temporizador disponível no simulador do P3. O temporizador terá por isso que ser programado adequadamente.

Note que em geral as interrupções devem ser rotinas muito curtas, normalmente limitando-se a ajustar o valor de uma ou mais variáveis, que depois serão utilizadas para controlar a execução do programa principal.

### VALORES ALEATÓRIOS

Como nesta fase o utilizador joga vários jogos de seguida, o algoritmo de geração de números aleatórios que foi implementado na primeira parte será agora usado para gerar uma sequência de números aleatórios (um número diferente para cada jogo). Para gerar o primeiro número aleatório, em vez de um valor inicial constante (“hardcoded”), deverá ser usado um contador do número de iterações de um ciclo que começa após a inicialização do jogo, e termina quando o utilizador carrega no botão IA pela primeira vez.

### QUALIDADE DO CÓDIGO

Tal como na fase anterior, será avaliada a qualidade do código, nomeadamente no que diz respeito à sua modularidade (e.g., uso de funções reutilizáveis e fáceis de manter), à passagem de parâmetros pela pilha de forma que obedeça às convenções ensinadas na cadeira, ou à utilidade dos comentários.

### DÚVIDAS NA ESPECIFICAÇÃO DO ENUNCIADO

Sempre que o enunciado não defina completamente como deve implementar alguma parte da solução, deve tomar a decisão que achar mais apropriada, tendo sempre em mente o objetivo de melhorar a experiência de jogo. Estas decisões devem ser descritas e justificadas no relatório final.

A implementação de qualquer funcionalidade não pedida neste enunciado não será valorizada na nota final.

## PLANO DE ENTREGA

### DIA 30 DE NOVEMBRO, ATÉ ÀS 23H59, NO FÉNIX

Entrega electrónica:

A entrega final deverá ser submetida num ficheiro zip com o nome no formato **tAaBBgC.zip**, em que: A é o dia da semana do turno de laboratório (2 a 6); BB é a hora de início (basta a hora, com 2 dígitos); C é o número do grupo. Este ficheiro deverá conter:

- Breve relatório em PDF (máximo 2 páginas em 11pt) com a descrição do projeto realizado, organização do programa e explicação dos aspectos mais relevantes da implementação. Na conclusão deverá ser feito um balanço do que foi realizado, com indicação dos aspectos nos quais o projeto tenha divergido do enunciado base (funcionalidades adicionais implementadas, funcionalidades não implementadas, outras variações ou divergências, etc.).
- Código, quer ficheiro fonte, quer um PDF gerado com a aplicação p3print fornecida na página da cadeira.

### SEMANAS DE 4 E 11 DE DEZEMBRO, NA AULA DE LABORATÓRIO

Demonstração do funcionamento do jogo concebido e discussão final, com perguntas individuais a cada elemento do grupo, e que podem também incluir perguntas sobre os laboratórios. O calendário das discussões será acordado com o docente do turno respectivo. A discussão terá lugar, preferencialmente, na semana de 11 de dezembro.