

Trabalho Laboratorial 2016/2017

1. Âmbito

Pretende-se desenvolver em linguagem Assembly um jogo cujo objectivo consiste em percorrer um determinado labirinto de forma a atingir uma determinada meta.

O jogo é dado por terminado, com vitória, quando a meta for atingida. Quando se tratar de uma vitória é apresentado o tempo gasto pelo jogador e a sua posição no top 10.

2. Funcionamento

▪ Jogo

Sempre que se inicia um jogo deverá ser apresentado um labirinto, com dimensão 40 caracteres x 20 caracteres e um Avatar que deverá estar num dos cantos. O Avatar só pode avançar pelos corredores do labirinto, não sendo permitido atravessar pelas paredes que o constituem. A navegação do Avatar pelo labirinto deverá ser efectuada com recurso ao teclado, nomeadamente às setas (←, ↑, →, ↓).

Deverá ser apresentado o ponto inicial e final do jogo através de marcadores especiais. Quando o programa termina é apresentado o tempo gasto pelo jogador e a sua posição no top 10.

▪ Gestão de pontuação do jogo – TOP 10

Sempre que um jogador vencer, deverá ser-lhe apresentado o “top 10” e verificado se o seu tempo de jogo lhe permite a entrada no top. Em caso afirmativo, deverá ser introduzido o seu nome e respectivo tempo de jogo.

O top 10 deverá ser guardado num ficheiro de texto.

▪ Criação de labirinto

O utilizador poderá criar e/ou seleccionar um labirinto, através de um conjunto de opções disponibilizado para o efeito, constituído no mínimo por:

- “carregar labirinto por omissão” que permite carregar um labirinto base;

- “carregar...”, permite carregar um labirinto previamente criado pelo utilizador;
- “criar/editar”, que permite criar e/ou editar o labirinto actual;

A dimensão do labirinto a criar será sempre de 40x20, em que cada posição será marcada como livre (corredor) ou ocupada (parede). Poderá usar as teclas de cursor para navegar sobre o tabuleiro de suporte à edição do labirinto e a tecla de espaços ou outras para editar as paredes do labirinto.

▪ **Interface**

A interface do programa fica ao critério dos alunos, salvo os seguintes aspectos que devem ser obrigatoriamente representados:

- Acesso a menu com, pelo menos, as seguintes opções:
 - Jogar
 - Top 10
 - Configuração do labirinto
 - Sair

3. Considerações

Este enunciado não explicita todos os pormenores do programa, permitindo que os alunos apresentem soluções alternativas para alguns aspectos não definidos no mesmo. Caso as opções seguidas alterem significativamente alguns dos aspectos apresentados, estas deverão ser colocadas à consideração do docente.

4. Grupos

O trabalho deve ser efectuado em grupo, de 2 elementos.

5. Modo de Entrega

- A entrega dos trabalhos será efectuada através da plataforma de e-Learning do DEIS-ISEC.
- Deve também ser entregue um **relatório técnico, impresso em papel**, com um máximo de 10 páginas e, em anexo, a listagem do programa. Nele deve ser explicada a estrutura, objectivos, funcionamento do programa e o algoritmo subjacente a cada função desenvolvida, bem como as principais opções de desenvolvimento.
- **Data de entrega dos trabalhos:** 4 de Junho de 2017.

6. Defesa

- Todos os trabalhos terão defesa, em que todos os elementos do grupo terão que estar presentes. A defesa destina-se a avaliar o grau de compreensão do código apresentado, incluindo a possibilidade de os alunos efectuarem pequenas alterações. Caso a defesa não seja satisfatória pelo grupo ou por algum dos elementos do grupo, os trabalhos poderão ser anulados ou levarem a uma diferenciação na nota entre elementos.
- A defesa será realizada em dia e hora a combinar posteriormente.

7. Avaliação

Este trabalho é cotado para 4 valores, distribuídos da seguinte forma:

Funcionalidade	Pontuação
Avatar a movimentar-se pelo labirinto base, com detecção de final de jogo.	2 Val.
Contabilização do tempo de jogo e apresentação do "Top 10".	1 Val.
Criação de labirintos e implementação de opções associadas.	1 Val.

8. Funcionalidades Extra (Opcional)

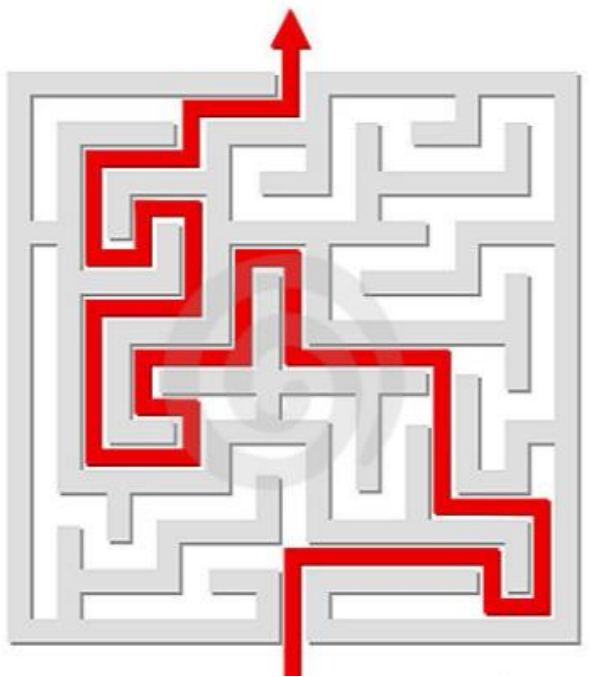
A funcionalidade extra consiste numa forma alternativa de percorrer o tabuleiro. Esta forma consiste na indicação de todo o percurso através de uma sequência de dígitos em hexadecimal. Cada dígito hexadecimal corresponderá a um movimento. Nesta representação hexadecimal de 4 bits, os dois primeiros bits codificam a direção (00-Norte, 01-Sul, 10-Este e 11-Oeste) e os restantes dois bits indicam o número de passos a avançar (00-um, 01-dois, 10-três ou 11-quatro passos).

Considere-se o seguinte exemplo:

Norte-2 passos; Este-4 passos; Sul-1 passo;...

0001; 1011; 0100;...

1B4...



Esta funcionalidade extra é cotada para 2 valores a acrescentar aos 4 valores previstos para o trabalho.