



## Tecnologias e Arquitetura de Computadores

### Trabalho Laboratorial 2020/2021

Docente:

Francisco José Simões Duarte

Alunos:

Paulo Henrique Figueira Pestana de Gouveia nº 2020121705 – LEI

João Filipe Silva de Almeida nº 2020144466 – LEI

Coimbra, 6 de Junho de 2021

# Índice

<b>Introdução</b>	<b>3</b>
<b>Objetivos</b>	<b>3</b>
<b>Funcionamento</b>	<b>3</b>
Menu	3
Jogo	3
TOP10	4
Editor	4
<b>Funções Usadas</b>	<b>7</b>
<b>Estrutura</b>	<b>8</b>

## Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Tecnologias e Arquitetura de Computadores e consiste na criação de um jogo feito na linguagem Assembly 8086.

O jogo tem base num labirinto fechado, que no seu interior apresenta todas as letras do alfabeto. O utilizador tem como objetivo completar a palavra que lhe é apresentada antes que o tempo acabe.

## Objetivos

Criação de um jogo de labirinto com letras no seu interior em que o objetivo é completar a palavra pedida pelo o jogo. Implementar uma condição de derrota quando o tempo termina, e uma condição de completar todos os níveis, em ambos os casos o jogador tem a opção de introduzir a pontuação que obteve no TOP10. Por final um extra para o jogador ter a liberdade de criar o seu próprio labirinto.

## Funcionamento

### Menu

O jogo apresenta um menu inicial para o jogador (Figura 1).

### Jogo

Primeira opção manda o utilizador para o labirinto de letras (Figura 2), em que o avatar (cursor do jogador) começa numa posição random, para mover o avatar o jogador usa as setas ( $\rightarrow$ ,  $\leftarrow$ ,  $\uparrow$ ,  $\downarrow$ ) do teclado. O jogador é limitado pelas paredes do labirinto, e tem que o percorrer até completar a palavra pedida antes que o temporizador chegue a zero. Ao completar a palavra, o jogador passa para o nível seguinte. A palavra pode ser completa por qualquer ordem, bastando ao jogador apanhar uma vez a letra mesmo estando repetida na palavra objetivo.

Ao completar todos os cinco níveis, o jogo termina e apresenta um menu de vitória (Figura 3), em que pode introduzir a sua pontuação no TOP10, voltar ao menu principal ou sair do jogo. Caso o jogador perca, vai para um menu de derrota (Figura 4) onde tem as mesmas opções.

## TOP10

Na opção TOP10, é pedido ao jogador o nome que quer introduzir para aparecer no ficheiro de texto (figura 5), é necessário obter uma pontuação de valor superior aos TOP10 para se qualificar, para ser consultado use a opção 2 no menu principal. Caso não se qualifique, o programa avisa ao jogador que a sua pontuação não é suficiente (Figura 6), e prossegue a voltar para o menu principal.

## Editor

Permite ao jogador a criação de um labirinto à sua escolha (Figura 7), o jogador move-se para apagar o ecrã, botão “DEL” constrói paredes, ao ser pressionado novamente volta ao modo apagar ecrã, o botão “PGUP” mete letra em ordem crescente no labirinto e o botão “HOME” guarda o labirinto para um ficheiro de texto.

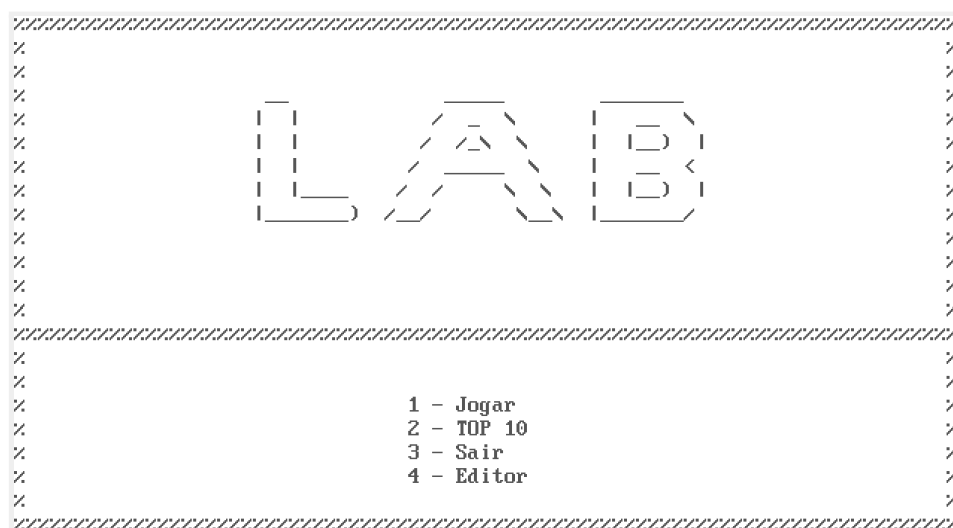


Figura 1.

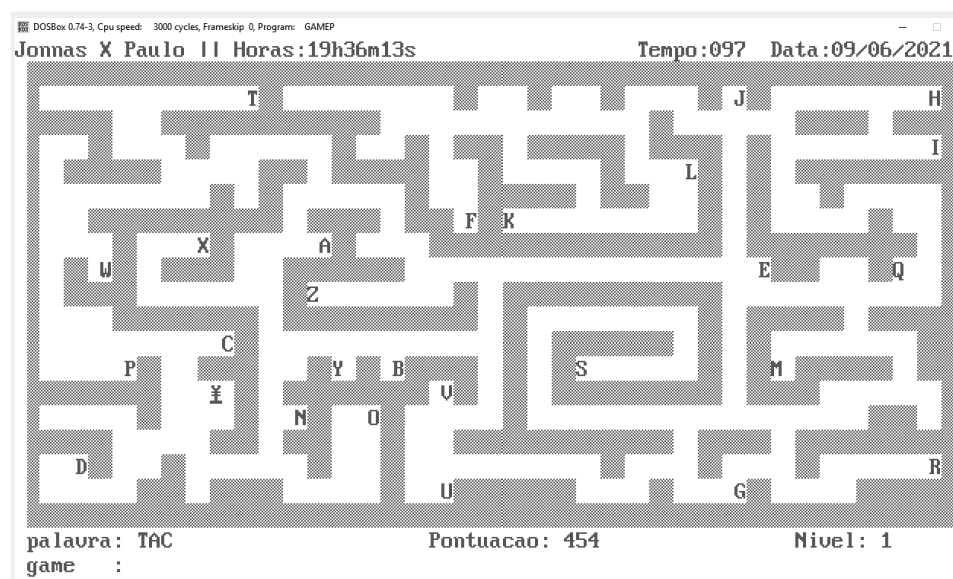


Figura 2.



Figura 3.

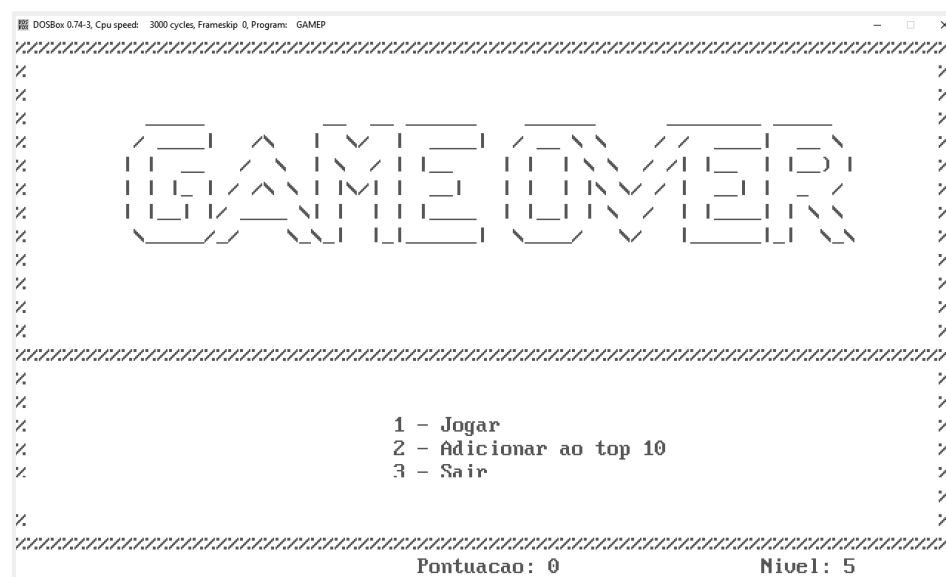


Figura 4.

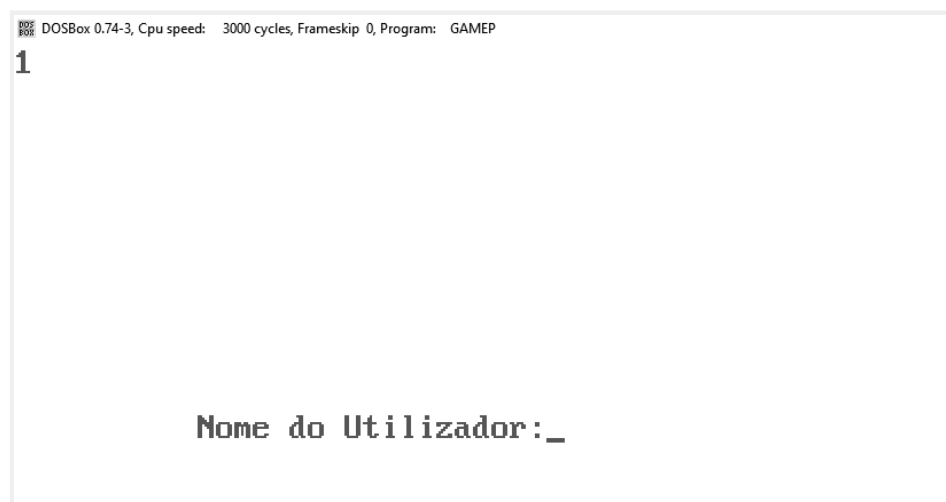


Figura 5.

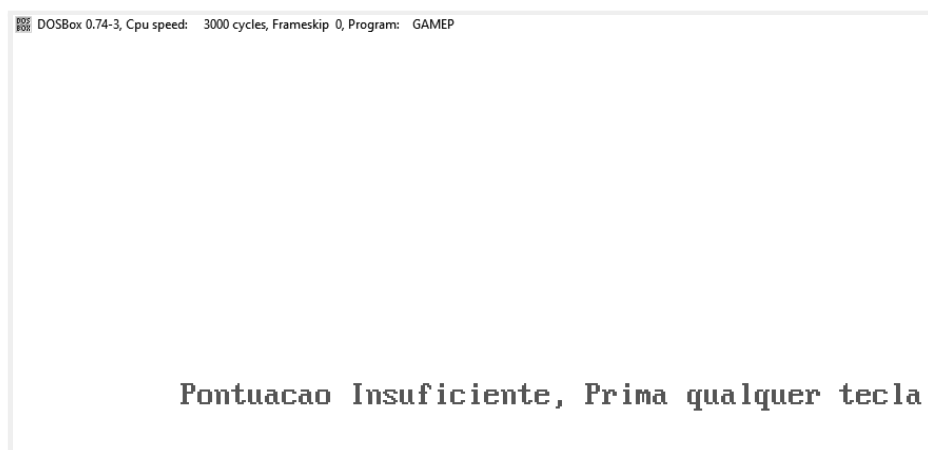


Figura 6.

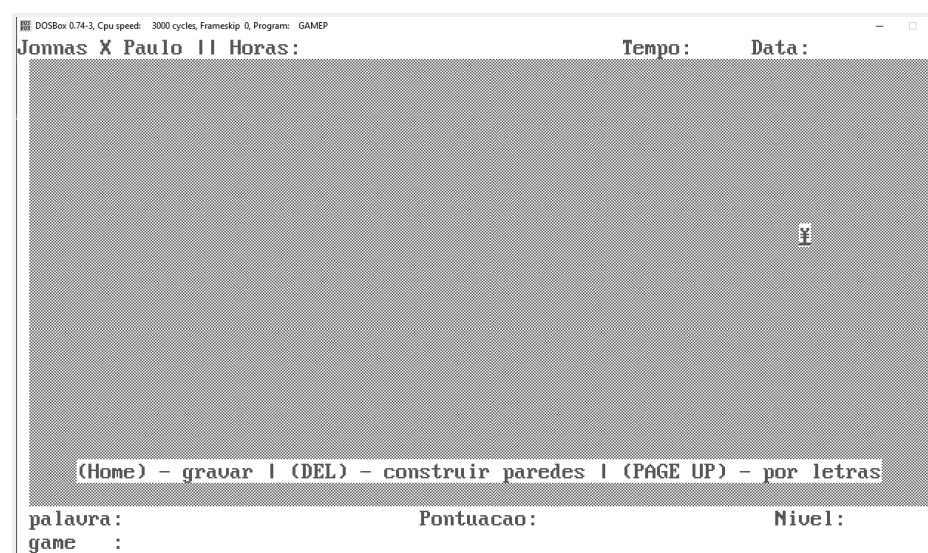


Figura 7.

## Funções Usadas

**Main** – Gera a execução do jogo aproveitando-se das outras funções criadas. Mostra o menu, lê as suas opções, corre os vários níveis com as devidas diferenças e condições de passagem. Corre o editor, lê o ecrã para uma matriz para sua posterior exportação, imprime o ficheiro top10 ou sai do programa.

**imp\_fich** – procedimento para imprimir o que está num ficheiro de texto;

**apaga\_ecran** – limpa o ecrã de display;

**Ler\_Tempo** – guarda as horas/minutos/segundos do computador em variáveis;

**tam\_str** – extrai o tamanho da string da palavra fornecida;

**resetvars** – é usada para retornar as variáveis ao seu valor original ou aleatório às variáveis da nova posição do avatar;

**prosseguir** – trata de colocar o carácter do avatar no ecrã, tanto como imprimir o carácter da posição do avatar no canto superior direito do ecrã. Possibilita também o movimento da avatar;

**prosseguir2** – É idêntica à função “prosseguir”, a diferença é que usada para o editor de labirinto, incluindo a construção de paredes, input de letras, e a gravação do labirinto criado;

**exportamatriz** – exporta uma matriz para um ficheiro de texto;

**Verificar\_letra** – função para o funcionamento da detecção das letras que o jogador apanhou, verifica se foi apanhada uma letra e mostra a string da palavra a ser construída se a letra apanhada é a correta.

**PRINTDIG** – Imprime um número de 16bits para a posição do cursor através de uma série de divisões por 10 e conversões de número para carácter. Coloca um espaço no final do número

**PRINTDIGSTR** – Imprime o número para uma string com o mesmo algoritmo da PRINTDIG

**LER\_FICH** – encarregue de abrir ficheiro, com verificações caso a haja erros ao lê-lo, e grava-o numa matriz;

**HOJE** – mete a data do calendário atual numa string;

**LE\_TECLA** – interpreta os inputs do teclado do computador para a movimentação do avatar e construção do labirinto, inicia também o timer, data atual e horas do computador;

**Trata\_Horas** – atualização das horas do computador e do timer, display de ambas tanto como display da data atual;

**winner** – display do ecrã de vitória;

**GAME\_OVER** – display do ecrã de derrota;

**addtop10** – verifica se a pontuação obtida pelo jogador é suficiente para entrar no top 10, se não, avisa o jogador que a pontuação foi suficiente, se sim, substitui o primeiro valor menor que encontra;

**extrainumero** – com o displacement a apontar para o início da linha da matriz, extrai a pontuação desse jogador para a variável extrairpontuacao;

**nplayer** – Lê e grava um nome de jogador fornecido.

**novorandNUM** – modifica o número aleatório da pilha para uma gama do tamanho do ecrã em display;

**resetplayer** – Preenche a string do nome de utilizador com espaços;

