



RELATÓRIO
Trabalho Prático
Tecnologias de Arquitetura e Computadores

CARLOS SANTOS - a2003035578@isec.pt

DIOGO COELHO – a2019143273@isec.pt

2020-2021

RELATÓRIO

Trabalho Prático

Tecnologias de Arquitetura e Computadores



ALUNO N.º 2003035578

Carlos Santos

ALUNO N.º 2019143273

DIOGO COELHO

ENTIDADE

Instituto Superior de Engenharia de Coimbra

PROFESSORES

Francisco Duarte

Cristiana Areias

José Nunes



INSTITUTO
SUPERIOR DE
ENGENHARIA
DE COIMBRA

COIMBRA – JUNHO – 2021

Notas iniciais do trabalho

O objetivo principal deste trabalho é preparar os alunos da Unidade Curricular (UC) de TAC para o desenvolvimento de programas linguagem em assembly.

Cada arquitetura de computador tem seu próprio código de máquina, isto é, cada família de processadores possui a sua própria linguagem assembly, já que cada processador possui o seu próprio conjunto de instruções embutidas. Como proposto para o desenvolvimento do jogo utilizaremos o Assembly 8086.

Utilizamos como base os Ficheiros de apoio ao Trabalho Prático fornecidos pelos docentes e neste relatório irão ser apresentadas as alterações ao código de base explicando sucintamente o desenvolvimento feito pelos alunos.

O código de base que nos foi fornecido:

- “avatar.asm” - A estrutura de base do programa com algumas rotinas de base e variáveis. Esta estrutura de base verifica-se o movimento do avatar por um mapa sem parar nas paredes em conjunto com o ficheiro, “labi.txt”, mapa do jogo.

- “CriaFich.asm” - Grava no ficheiro de texto “pergunta.txt” o conteúdo da matriz “buffer”;

- hms_dma.asm - Lê uma string e escreve noutra zona do ecrã e simultaneamente actualiza a hora e data no ecrã;

- Random.asm - Demonstração duma rotina de cálculo de números aleatórios;

ÍNDICE

1 - INTRODUÇÃO	5
2 - Levantamento de requisitos principais	6
3 – Secções do Jogo.....	7
3.1. Menu	7
3.1.1 Jogar	7
3.1.2 TOP10	7
3.1.3 Info /credits	8
3.1.3 Sair	8
4 – Jogar.....	8
4.1. Avatar	8
4.2. Labirinto	9
4.3. Data e Tempo	9
4.4. Níveis	9
4.5 Variante de Jogo	10
4.6 Ganhar – Encontrar palavras	10
5 – Conclusão	10

1 - INTRODUÇÃO

O Trabalho prático consiste no desenvolvimento de um jogo de labirinto fechado que além das paredes tem também no seu interior todas as letras do alfabeto. O objetivo é encontrar todas as letras numa palavra antes de acabar o tempo limite.

No enunciado é apresentada uma lista de funcionalidades que estão enumeradas neste relatório na parte de levantamento de requisitos. Desenvolvemos soluções para a maioria das funcionalidades mas por falta de tempo não conseguimos completar tudo.

Como pontos principais de avaliação e estado de desenvolvimento:

- Completado - o Avatar a movimentar-se pelo labirinto base, com detecção de final de jogo;
- Completado - a gestão dos vários níveis do Jogo;
- Completado - a Contabilização do tempo de jogo;
- Por completar 60% - a apresentação do "Top 10". No entanto, iniciamos parte do seu desenvolvimento (60% do desenvolvimento – écran top 10, pontos com tempo, pedido de nome de jogador – ficheiro guardado com informação);
- As funcionalidades extra não conseguimos desenvolver embora o seu algoritmo não seja complexo. (Pensámos em desenhar o labirinto com o avatar e carregando em teclas diferentes para escrever paredes ou letras. Depois opção numérica para gravar no fim todo o écran para um ficheiro de texto através da memória de video).

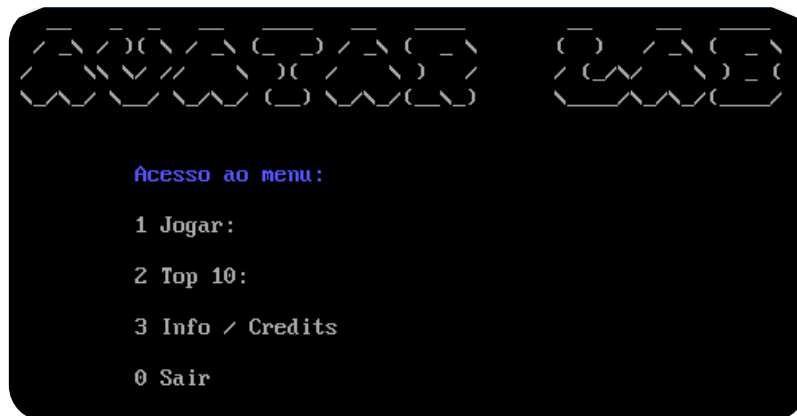
2 - Levantamento de requisitos principais

Interface	<p>Acesso a menu com, pelo menos, as seguintes opções:</p> <ul style="list-style-type: none">- Jogar- Top 10- Sair
Movimentação do avatar	<ul style="list-style-type: none">- Avançar pelos corredores do labirinto com setas (←, ↑, →, ↓);- Não é permitido atravessar as paredes;
Posição aleatória	<ul style="list-style-type: none">- Dentro do labirinto deve aparecer um avatar numa posição aleatória;- O avatar deve percorrer as sucessivas letras da palavra apresentada e devem ser mostradas a letras que já foram encontradas;- Verificar palavra no final;
Deteção de letras e palavra	<ul style="list-style-type: none">- Primeira variante: Passar pelas palavras sem uma ordem específica, se a palavra apresentar letras repetidas, basta passar apenas uma vez por cima dessa letra no labirinto.- Segunda variante: Respeitar a ordem de letras se o jogador se enganar na ordem o nível deve recomeçar, mas o tempo não deve voltar a zero.
Níveis	<ul style="list-style-type: none">- Quando são encontradas todas as letras da palavra dentro do tempo limite, termina o nível do jogo e passa para o próximo.- Vários níveis de dificuldade – mínimo 5;- Primeira variante: Dificuldade por tempo reduzido e tamanho de palavras;
Tempo	<ul style="list-style-type: none">- Contabilização do tempo limite;- Criar tempo limite para cada nível;- Reset de tempo com passagem de nível;- Mostrador de tempo limite para o jogador;
Mapas	<ul style="list-style-type: none">- Primeira variante: Labirinto é sempre igual para todos os níveis do jogo apresentando todas as letras do alfabeto;- Segunda variante: Mapas diferentes para diferentes níveis;- Segunda variante: Cada nível apresenta um labirinto diferente onde estão apenas as letras da palavra a encontrar. Conforme o avatar vai passando por cima da letra ela vai desaparecendo e letras repetidas na palavra, devem também estar repetidas no labirinto;
TOP 10	<ul style="list-style-type: none">- Deve ser verificado se a pontuação permite a entrada no top. Em caso afirmativo, deverá ser pedido o nome do jogador;- O top 10 deverá ser guardado num ficheiro de texto para ser utilizado na próxima execução;
Extra – Editor de labirinto	<ul style="list-style-type: none">- Editor do Labirinto, com criação, edição de vários labirintos. Nesta opção será possível desenhar um labirinto do zero e também abrir um labirinto já existente podendo alterar e gravar com novo nome. Existindo vários labirintos pode ser definido qual o labirinto que irá ser utilizado no jogo.
Extra – Edição de palavras	<p>possível a edição e configuração das palavras que serão apresentadas e tempo de jogo a utilizar em cada nível do jogo.</p>

3 – Secções do Jogo

3.1. Menu

No menu principal, o utilizador pode optar por jogar, ver o Top10, ver Info/Créditos ou sair. O jogo retorna sempre a este menu.

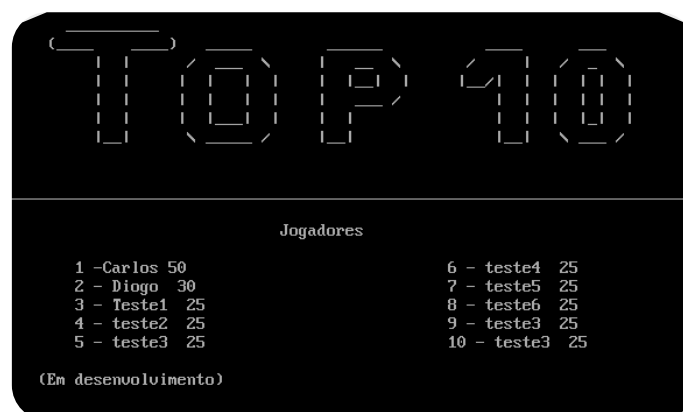


3.1.1 Jogar

Permite entrar na interface principal de jogo e dar início ao primeiro nível de jogo. Esta parte do programa será explicada em mais detalhe numa secção à frente neste relatório.

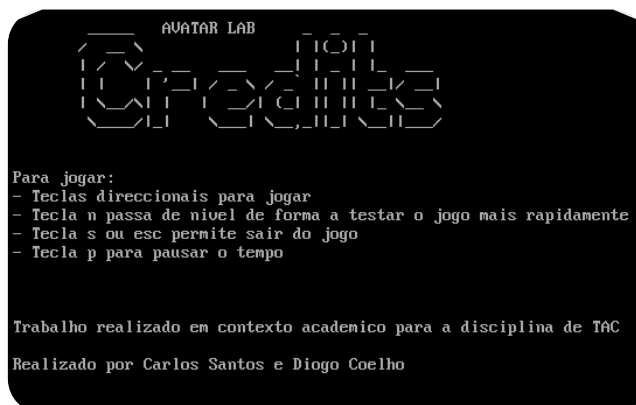
3.1.2 TOP10

Entrada no écran do Top 10, funcionalidade que não ficou completa por uma questão de tempo.



3.1.3 Info /credits

Nos créditos, aparecem informações sobre o jogo e dos autores do jogo. Além disso, colocámos ainda a informação para teclas a utilizar durante o jogo para se mover, saltar níveis (tecla especial) e pausar o tempo.



3.1.3 Sair

Para finalizar o programa e sair do jogo.

4 – Jogar

Neste jogo de labirinto o avatar é colocado aleatoriamente no labirinto, usámos a função CalcAleat, impnum proc e reset_position para que quando começar um novo jogo ou quando se passa de nível o avatar não ser colocado no mesmo sítio.

Depois de ser colocado o avatar, o jogador terá que percorrer o labirinto até encontrar neste caso as 4 letras necessárias para passar de nível (ISEC), mas funciona para outras palavras se necessário. Caso este não consiga encontrar a palavra a tempo irá ocorrer uma limpeza do ecrã, através da função apaga_ecran, em que este vai receber uma mensagem a informar que perdeu e será redirecionado para o menu onde pode voltar a jogar outra vez, caso assim o queira.

Ao passar de nível o jogador irá ser redirecionado para um novo nível com maior grau de dificuldade, quer seja por tempo reduzido ou por um labirinto diferente ou mais complexo. O objetivo é passar os 5 níveis nos tempos que são propostos.

4.1. Avatar

Alterámos o character do avatar, mudamos a cor e eliminámos o piscar do cursor. Além disso, colocamos o avatar a aparecer numa posição do mapa de forma aleatória, sem ser na parede ou numa letra, mantêm-se dentro dos limites do labirinto pois este está limitado a x e y espaços do mapa.

O avatar usa as direções pedidas no trabalho, usando as setas do teclado ele move-se facilmente pelo labirinto sem qualquer problema, se se usarem outras teclas sem ser as setas ele não se move.

4.2. Labirinto

O labirinto foi construído com a base do labirinto fornecido.

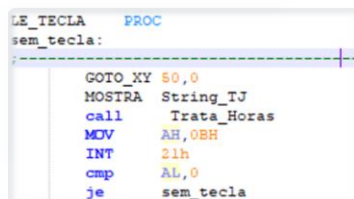
Quando o avatar bate numa parede este não a atravessa, ficando parado na casa anterior em que estava inserido. Isto é, na percepção do jogador é o que acontece no entanto, na realidade de cada vez que ele entra na parede o avatar retorna à posição imediatamente anterior.

Existem várias letras do alfabeto no labirinto, quando o avatar passa por cima delas um espaço é escrito no lugar, ou seja, cria um efeito de “comer” as letras. Se a letra encontrada pertencer à palavra é adicionada à string “Construir_nome”, senão segue em frente, apagando-a a letra do labirinto e continuando à procura pelo resto das letras.

4.3. Data e Tempo

Para metermos a data e o tempo correspondente no nosso jogo adaptámos parte do código disponibilizado em “hms_dma.asm” e implementámos um contador (Timer_Dec e Timer) para quando o relógio chegar ao 0 o jogador perca e o jogo reinicie.

Também utilizamos a função recuperada de “hms_dma.asm” para que o tempo corra mesmo se o avatar não se esteja a movimentar (Sem_Tecla).



```
LE_TECLA      PROC
sem_tecla:
    GOTO_XY 50,0
    MOSTRA  String_IJ
    call    Trata_Horas
    MOV     AH,0BH
    INT     21h
    cmp     AL,0
    je      sem_tecla
```

4.4. Níveis

Como pedido no enunciado do trabalho, este jogo de labirinto tem 5 níveis e cada nível é mais difícil do que o anterior, quer seja devido a um labirinto diferente ou por uma redução de tempo. Para isso, usámos uma função para limitar o tempo de cada nível.

Com a macro “Mostra” o labirinto é escrito e quando a palavra pedida é encontrada o nível é superado. No entanto, se a palavra não for encontrada a tempo aparecerá no ecrã a informação de que jogador perdeu.

O final do jogo é após a passagem pelos cinco níveis, aí o jogador é congratulado com uma mensagem de parabéns onde lhe é pedido o seu nome para o inserir no top 10, caso atinja um bom resultado. O nome do jogador este está limitado em 10 letras e utilizámos a rotina “PLAYER_NAME”.

4.5 Variante de Jogo

No enunciado, foram dadas como opção duas variantes, nós “optámos” por fazer uma mistura das duas, pois a ideia era implementar-mos as duas mas que acabamos por não ter tempo de o fazer. Nesta opção usamos sempre a mesma palavra para todos os níveis, usamos diferentes mapas de labirinto por nível, quer estes tenham mais ou menos tempo limite. As letras podem ser encontradas independentemente da ordem até o tamanho da palavra ser atingido sendo que esta apenas vai contabilizar as letras pedidas uma única vez, sem que estas sejam repetidas para a string.

4.6 Ganhar – Encontrar palavras

É necessário encontrar as letras para completar a palavra antes que tempo no labirinto termine para passar de nível.

Quando é encontrada uma das letras necessárias para a palavra, esta vai ser introduzida no ecrã em baixo da palavra pedida, colorizando a letra da String_Nome. Quando as quatro letras são preenchidas, o nível termina e é sinalizado com um sorriso.

Nota: Mesmo que se encontre a mesma letra duas vezes, apenas se vai escrever uma.

5 – Conclusão

Em conclusão, a programação em Assembly é muito interessante e facilmente podemos desenvolver programas e jogos numa linguagem de baixo nível próxima da linguagem máquina. Particularmente, achámos que o uso das interrupções ajuda bastante e são como uma caixa de ferramentas do Assembly.

Este trabalho ajudou a desenvolver bastante a nossa lógica de programação e a estruturámos melhor a nossa lógica de programação mesmo para outras linguagens.

Muito obrigado pela atenção dispensada na correcção dos nossos trabalhos e na passagem de conhecimento durante as aulas.