Relatório do Trabalho Prático de TAC  
 Jogo do Labirinto

Bruno Sousa [a2020132971@isec.pt](mailto:a2020132971@isec.pt)

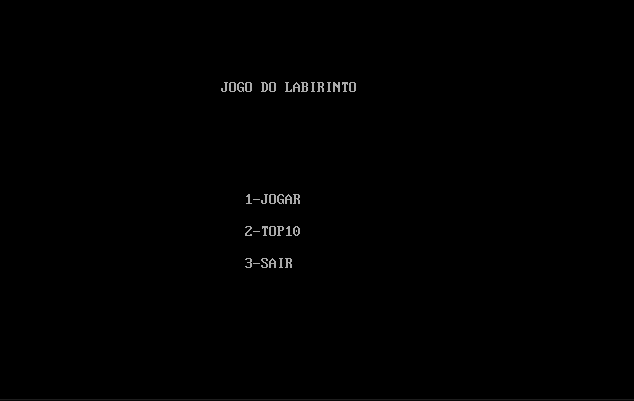
Jorge Santos [a2020133143@isec.pt](mailto:a2020133143@isec.pt)

Turma: P9

Docente Responsável: Francisco Duarte

Funcionamento do jogo:

Quando o jogo é iniciado, o jogador encontra o seguinte menu:



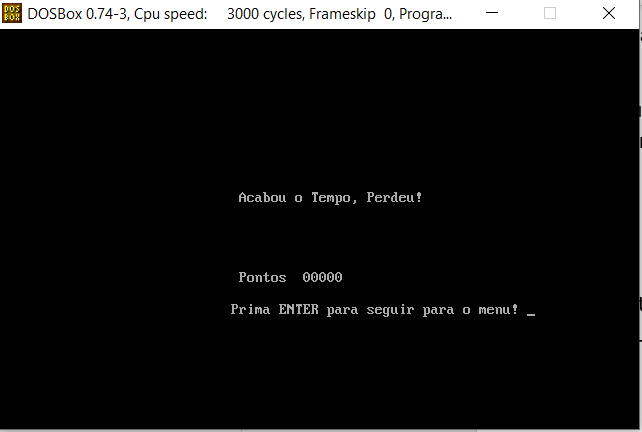
No qual, se o jogador clicar na tecla 1, irá começar a jogar o jogo do Labirinto.



Neste jogo, o jogador terá que percorrer o labirinto à procura das letras que correspondem à palavra indicada no canto inferior esquerdo.

No enunciado deste trabalho prático, tínhamos de escolher entre 2 variantes. O nosso grupo escolheu fazer a 1º variante, na qual o labirinto irá ser sempre o mesmo, mas as palavras e o tempo se alteram. No entanto, o jogador não é obrigado a ir às letras por ordem, podendo começar pela que entender que seja mais rápida para poder passar o nível. São apresentados no total 6 níveis, onde as palavras aumentam e o tempo diminui. Apesar de o labirinto ser igual, o jogador não irá começar o nível sempre no mesmo sítio, tendo que encontrar a posição inicial do avatar para começar a procurar as letras.

Se o jogador ficar sem tempo, irá ser mostrado uma mensagem a dizer que perdeu e quantos pontos obteve e depois de clicar no ENTER, este irá ser retornar ao menu inicial.



Se ganhar todos os níveis irá aparecer que ganhou e a sua pontuação retornando para o menu inicial.

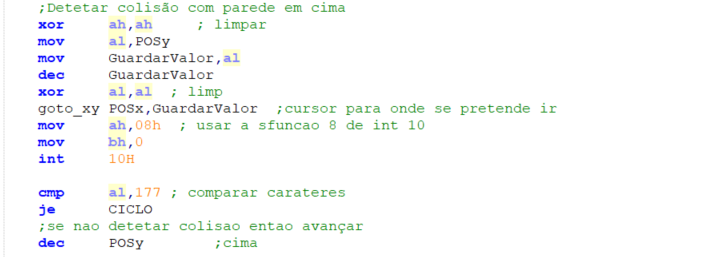
ATALHOS: Para sair durante o jogo, basta apertar ESCAPE.

No menu inicial, o jogador pode simplesmente clicar na tecla 3.

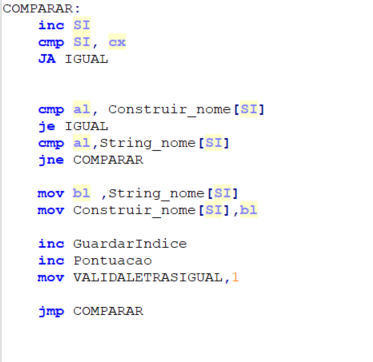
Código do jogo:

O jogo contém algum código fornecido pelo professor, assim como as procs Avatar, apaga\_ecran, IMP\_FICH,Ler\_Tempo,LE\_TECLA,Hoje e Trata\_Horas e as macros GOTO\_XY e MOSTRA.

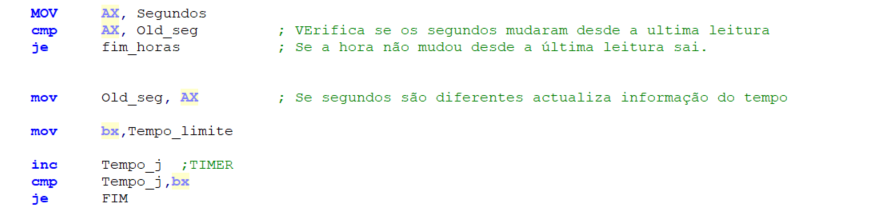
Para o avatar detetar as paredes, fizemos com que o programa comparasse o carácter da posição onde o avatar é suposto ir com o carácter das paredes (177 ASCII TABLE) como demonstra a seguinte figura:



Para o avatar detetar as letras, usamos quase o mesmo mecanismo, ou seja, detetamos o carácter da posição do avatar e fizemos um loop através da palavra definida e comparamos para ver se o carácter se encontrava na palavra, se sim então detetamos a posição da letra na palavra e adicionámos a uma string vazia para imprimir no ecrã as letras à medida que vamos encontrando:

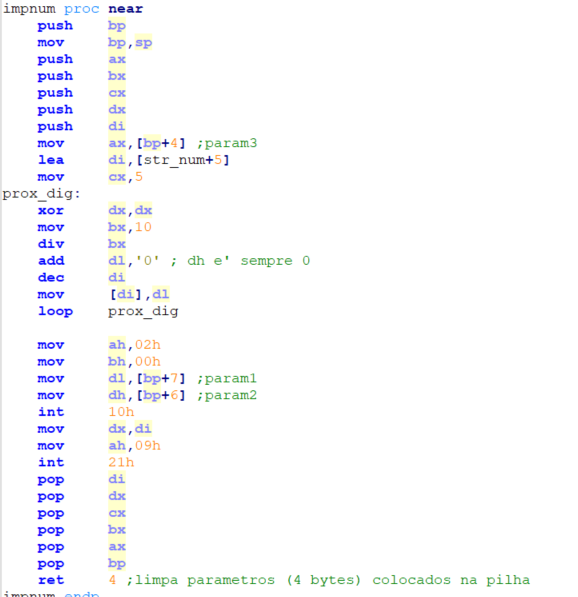


Para o tempo, usamos o tempo do sistema obtido nas procs de horas fornecidas pelo professor e como estas guardam a cada momento que a função é chamada os novos segundos, tudo o que tínhamos de fazer era comparar a variável Old\_Segundos com Segundos , se estas forem diferentes significa que 1 segundo passou e devemos incrementar 1 ao tempo de jogo:

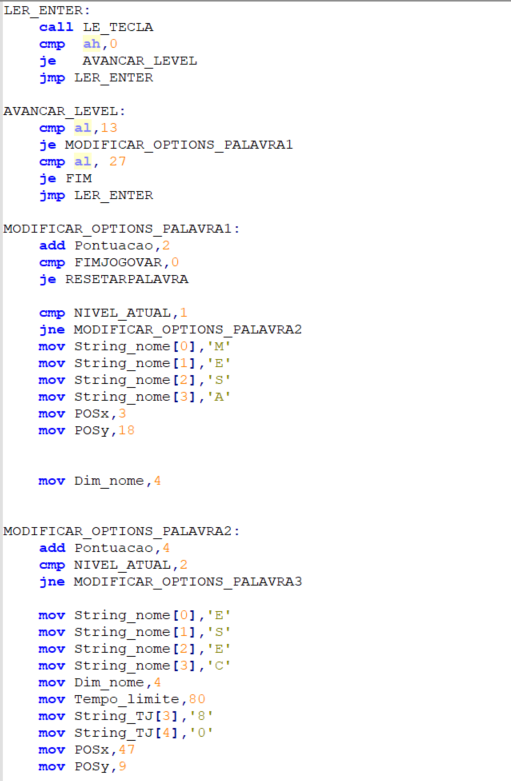
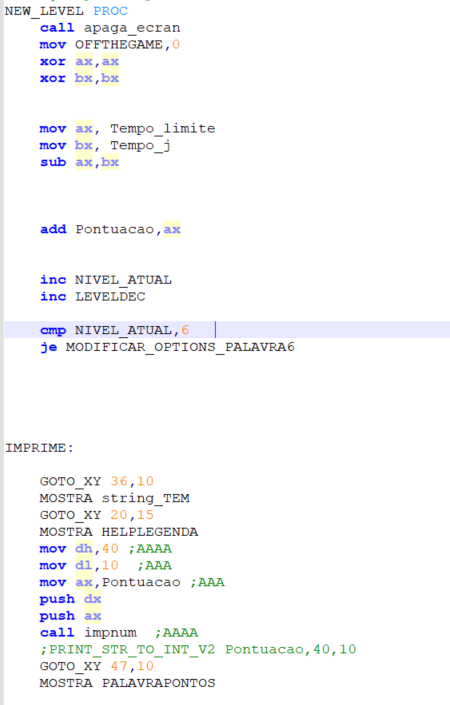


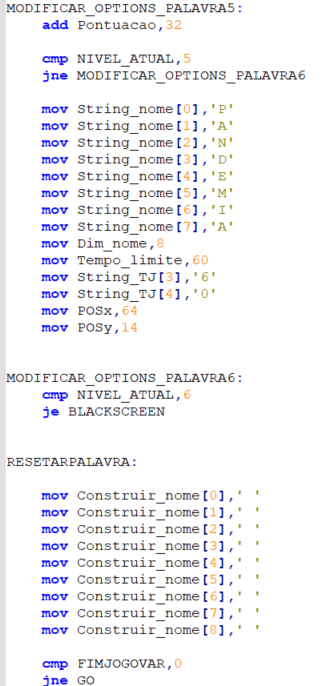
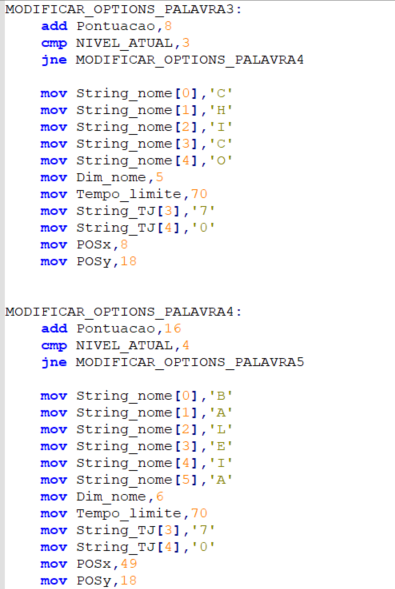
Para fazer com que o tempo seja atualizado no ecrã só temos que chamar a proc Trata\_Horas na proc LE\_TECLA, pois esta está constantemente a ser chamada no CICLO do AVATAR, atualizando assim o tempo no jogo.

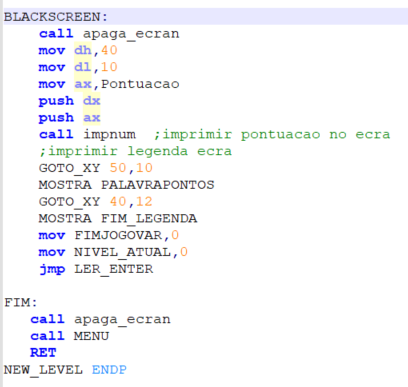
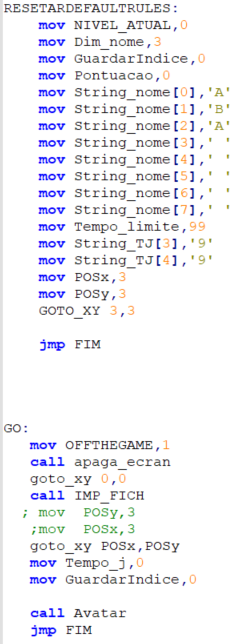
Para a pontuação, simplesmente fomos adicionando a uma variável de números inteiros pontos ao longo do jogo, como por exemplo a diferença do Tempo\_limite – Tempo\_jogo adicionando mais pontos se o jogador demorou menos tempo naquele nível e também é adicionado bónus no fim de cada nível. Para imprimir a pontuação no ecrã, usamos uma proc fornecida pelo professor (impnum).



Para organizar os novos níveis e definir os novos parâmetros foi criada uma nova proc chamada NEW\_LEVEL que controla os níveis e suas definições, assim como o fim de jogo e os SplashScreens durante os níveis que são imprimidos antes do próximo nível começar, sendo necessário que o jogador aperte ENTER para continuar para o próximo nível.







As zonas MODIFICAR\_OPTIONS\_PALAVRAX é onde são modificadas o tempo e as palavras do próximo nível assim como é acrescentado à pontuação o bónus de cada nível.

Nesta proc também lida com o fim do jogo(vitória) mostrando um splashscreen de vitória no final e faz reset aos valores do jogo colocando-os para o primeiro nível para o caso de o jogador decidir jogar de novo.

No fim, é apresentado o menu, que simplesmente permite a inicialização do jogo e também apresenta um TOP10 que não está funcional, pois a sua implementação acabou por não ser conseguida.

Com isto, os diferentes algoritmos foram explicados, havendo algumas variáveis ou outros elementos que foram deixados de fora pois acreditamos que não haja necessidade de os mencionar.

CONCLUSÃO:

A implementação deste jogo não foi uma tarefa fácil, tendo ainda em conta que não foi conseguido a implementação de um top10. Apesar disto, acreditamos que a funcionalidade do jogo se encontra em bom estado e é permitido ao utilizador poder jogar durantes os diferentes níveis sem apresentação de erros graves. Com isto, apesar de ter sido difícil não deixou de ser divertido programar este projeto, assim como este permitiu aprender bastante sobre assembly devidos aos diferentes algoritmos e elementos usados nele.