

João Palet de Almeida – 86447

Tomás Vieira – 86523

Introdução:

O projeto consiste num jogo programado em Assembly para ser executado no processador P3 onde o objetivo é controlar uma nave que deve percorrer a maior distância possível, matando o maior número de asteroides possível. Enquanto isso, deve evitando colidir com os obstáculos, sendo estes asteroides e buracos negros, sob pena de *Game Over*.

Desenvolvimento:

O programa contém um ciclo principal de jogo, que é precedido de uma rotina de inicialização, onde há um período de espera, até que o utilizador prima o botão IE para iniciar o jogo, e é complementado por rotinas auxiliares.

A rotina de inicialização tem como objetivos:

- Inicializar o *Stack Pointer*;
- Ativar a máscara das interrupções;
- Inicializar o cursor;
- Criar a janela inicial, onde se aguarda que o utilizador prima o botão IE para começar o jogo;
- Limpar a janela de início e começar o jogo.

Durante o ciclo principal, o programa vai executando as rotinas que são chamadas e vai testando se foi ativada alguma *Flag* associada às interrupções. Se alguma for executada, o programa irá executar a respetiva rotina de tratamento, repondo a *Flag* ao seu valor inicial.

À medida que o jogo vai sendo jogado, vai sendo ainda atualizado o LCD com a posição ocupada pela nave, o *Display* de 7 segmentos com o Score no momento e os LEDs sempre que um asteroide é atingido.

Exemplos de rotinas auxiliares do ciclo principal importantes são:

- Movimento da nave;
- Disparo
- Criação de um valor aleatório para o *spawn* dos obstáculos
- Verificação de colisões tiro/obstáculo e nave/obstáculo
- Reset

Foram implementadas todas as funções propostas, e foi criada a opção de fazer um *Restart* ao jogo a qualquer ponto do *Gameplay*, bastando para isso ao utilizador premir o botão IE a qualquer momento. Se esta opção for tomada, o utilizador será levado para um ecrã onde apareça escrito que o jogo terminou, assim como a pontuação obtida até ao momento.

Ex:



Deste ponto, como diz na janela, o utilizador poderá premir qualquer interrupção de modo a reiniciar o jogo.

Conclusão:

Em suma, seguimos o enunciado na sua maioria à exceção da já mencionada opção de fazer *Restart* a qualquer momento do jogo. O método de passagem de parâmetros mais utilizado foi a passagem pela *Stack*, pois muitas das vezes permite deixar registos livres que podem ser úteis mais tarde, para além de reduzir o número de linhas. Criámos várias rotinas auxiliares para facilitar não só a escrita, mas também a compreensão do código, pois deste modo o tamanho das funções principais também é reduzido. Por fim, as maiores dificuldades surgiram na implementação das funções que testavam as colisões entre obstáculos e tiro, nomeadamente na parte em que quando este colidia com um asteróide ambos desapareciam e quando colidia com um buraco negro era “sugado”, mantendo-se o buraco na janela.