83417, Afonso Samora.

83480, João Moniz.

83576, Vitor Nunes.

## Introdução

O projeto consiste na conceção de um jogo utilizando a linguagem *assembly* do processador P3. O objetivo do jogo é levar o pássaro a percorrer a maior distância possível sem colidir contra obstáculos que se deslocam a uma velocidade constante. Cada obstáculo possui uma zona de passagem (número fixo) por onde o pássaro deverá efetuar o seu trajeto de forma a sobreviver a mais um obstáculo.

O utilizador apenas pode subir o pássaro. A descida será realizada por efeito da gravidade.

O nível do jogo pode ser alterado se o jogador premir os botões I1 e I2. Essa alteração provoca um aumento da velocidade a que os objetos se deslocam sem alterar a velocidade do pássaro em relação à gravidade.

## Implementação

### Aspetos Gerais

De seguida apresentam-se algumas características genéricas do jogo:

* O pássaro pode deslocar-se para cima até ao limite superior do ecrã e poderá aterrar no limite inferior;
* O pássaro colide quando o seu corpo (representado por “o”) colidir com um obstáculo;

### Temporizador

O temporizador é composto por 3 contadores distintos e o seu tempo de *delay* é de 100 ms. Os contadores “Cont\_Ix” representam os contadores das respetivas interrupções.

Cont\_Timer representa um contador auxiliar nos quais todos os outros estão contidos. Foi necessário utilizá-lo uma vez que era necessário reiniciar as definições do temporizador em cada instante.

### Obstáculos

Os obstáculos foram implementados com recurso a uma tabela de posições contíguas em memória. O tamanho da tabela corresponder ao número máximo de obstáculos em simultâneo no ecrã.

Nessa tabela os elementos seguem o seguinte exemplo:

|  |  |  |  |
| --- | --- | --- | --- |
| 024D | 094A | … | 1635 |

* Os octetos mais significativos representam a primeira linha da zona de passagem (ex: linha 2);
* Os octetos menos significativos representam a coluna onde o objeto será colocado. (ex: coluna 77 = 4D)

Sempre que um elemento se encontrar na primeira coluna o respetivo elemento na tabela é colocado a zero. Assim, quando a tabela estiver totalmente preenchida o programa volta ao primeiro índice armazenando o valor do próximo obstáculo.

A função “**movobst**” apenas decrementa os elementos não nulos da tabela. Esta é responsável por colocar a zero os elementos cuja coluna é zero. Ou seja, se um elemento é, por exemplo, 1200h, então esse elemento passará a ser 0000h porque significa que esse elemento atingiu o limite do ecrã.

A função “**escreveObst**” percorre a tabela e escreve os obstáculos cujas coordenadas não nulas são representadas pelos elementos da tabela. Esta função invoca instâncias de “**putObs**”, função que recebe as coordenadas e escreve um obstáculo de acordo com as mesmas.

## Funcionalidades em falta

De salientar algumas funcionalidades/aspetos que ficaram por melhorar, por motivos de diversos (falta de tempo, implementação):

* A utilização de variáveis globais ao invés dos registos R6 e R7 para armazenar as coordenadas do pássaro e do ponteiro do ecrã, respetivamente;
* A ação gravítica através da utilização de vírgula fixa no contador (Cont\_Pass);

## Conclusão

Na sequência do acima exposto, verifica-se que o trabalho que apresentamos se traduz na conceção e desenvolvimento de um jogo com recurso a linguagem assembly.

Embora não tenhamos concretizado todas as funcionalidades, pelas razões que referimos acima, o objetivo a que nos propusemos - conceção de um jogo utilizando a linguagem assembly do processador P3, em que conseguíssemos levar o pássaro a percorrer a maior distância possível sem colidir contra obstáculos que se deslocam a uma velocidade constante, foi conseguido, tendo deste trabalho resultado uma aprendizagem e aprofundamento de conhecimentos das matérias abordadas nas aulas, que consideramos muito relevantes para a área que estamos a estudar.