



## Projeto de Introdução à Arquitetura de Computadores

### Segunda parte

A segunda parte do projeto consiste no desenvolvimento de um programa, em Assembly do P4, que implementa o jogo "dino" utilizando as funções desenvolvidas na primeira parte do projeto.

#### Saída ("Output") no terminal

No terminal do P4 deverá desenhar, e periodicamente atualizar, o terreno de jogo (ver secção 5.1 do Manual do P4). Têm liberdade total de utilizar os caracteres ASCII que quiserem para representar o dinossauro, os cactos, e o solo. O dinossauro será sempre desenhado na mesma coluna do écran, e ocupando (quando não estiver a saltar) algumas linhas imediatamente acima da linha onde está representado o solo.

#### Salto do dinossauro

O salto é desencadeado quando o utilizador carrega na tecla "seta para cima", mas apenas no caso de não estar nenhum salto em curso. A trajetória do dinossauro deve ser implementada de uma forma muito simples, através de um movimento a velocidade constante (por exemplo, uma linha por cada atualização da saída no terminal), inicialmente para cima até atingir uma altura máxima (determinada por uma constante no código), seguido de uma descida também a velocidade constante até atingir novamente a linha imediatamente acima do solo.

#### Início e final do jogo

O jogo inicia quando o utilizador carrega no botão "0", estando o terreno de jogo inicialmente sem cactos. Quando o dinossauro colide com um cacto deve aparecer a frase "G A M E O V E R" no centro do écran e o programa fica a aguardar que o utilizador volte a carregar em "0".

#### Pontuação

Tal como no jogo original, à medida que o jogo progride a pontuação vai sendo aumentada, sendo mostrada, em decimal, nos mostradores de 7 segmentos. (Ao contrário do original, não é necessário manter o "high score".)

## Implementação

- Antes de começarem a implementação devem planejar a estrutura geral do código. Se for útil devem discutir esta estrutura com o docente de Laboratório (nas aulas de dúvidas, na aula de Laboratório, ou por e-mail).
- A informação que descreve o estado do jogo deve ser mantida em variáveis mantidas em memória. Naturalmente, do ponto de vista de eficiência da execução do código, deverá trazer as variáveis da memória para os registos antes destas serem manipuladas, se possível mantê-las em registos durante um trecho de código significativo (por exemplo, na execução de uma função), e voltar a escrevê-las em memória após a sua utilização.
- A passagem do tempo e consequente atualização do terreno de jogo é controlada através do temporizador, cujas interrupções devem ser programadas adequadamente. Notem que em geral as interrupções devem ser rotinas muito curtas, normalmente limitando-se a ajustar o valor de uma ou mais variáveis, que depois serão utilizadas para controlar a execução do programa principal.
- Irá ser avaliada a qualidade do código, nomeadamente no que diz respeito à sua modularidade (e.g., uso de funções reutilizáveis e fáceis de manter), à passagem de parâmetros de forma que obedeça às convenções ensinadas na cadeira, ou à utilidade dos comentários.

## Entrega

### **Dia 4 de dezembro, 23h59**

Deverão ser entregues os ficheiros de código criados (ficheiros com formato .as) e o relatório do projeto (máximo duas páginas, 11pt, em formato .pdf). O relatório deve descrever a estrutura geral do código, um sumário das funções implementadas, e justificar as decisões principais no desenho da aplicação.