Relatório do Projeto de IAC

**Estrutura geral do código:**

De uma forma sucinta, o nosso programa começa através de um loop infinito que “espera” que o jogo comece, ou seja, apenas sai desse loop quando a tecla 0 (Key0) do simulador é premida pelo jogador.

Primeiramente, o nosso programa começa por limpar o terreno, a pontuação, e, de uma forma geral, o terminal, de forma a que fique apenas o chão implementado, e que o jogo possa começar.

De seguida, implementamos um temporizador (através de uma função que aguarda para que um determinado evento aconteça) e sempre que esse o temporizador acaba (a cada 100 ms), o jogo atualiza. Para este efeito, desenvolvemos uma rotina que chama uma série de funções que permitem o desenvolvimento do jogo (mencionadas abaixo).

Agora que o jogo começou, o programa começa por atualizar o vetor que contém o terreno. Posteriormente, averigua-se se nos encontramos em situação de salto. Caso seja, é tratado e interpretada a posição do dinossauro; caso contrário, o dinossauro é implementado normalmente, sem problemas.

Depois, escrevemos o terreno que foi atualizado, e verificamos se nos deparamos numa situação de gameover (explicada abaixo). Caso se verifique, a rotina volta ao início (à espera que a key0 seja premida para que o jogo recomece); senão, o programa continua para a fase seguinte.

A próxima fase consiste em adicionar 1 (decimal) à pontuação geral do jogo (escrita no LCD), e regressamos à fase de espera do temporizador, mencionada no 3º parágrafo.

**Descrição de funções:**

**limpa\_terreno**: É invocada quando o jogo começa (ou recomeça) e limpa o terreno, de forma a que se gere um jogo novo e diferente.

**Limpa\_pontuacao**: É invocada quando o jogo começa (ou recomeça) e limpa a pontuação, para que uma nova pontuação possa ser escrita

**atualizajogo**: A função vai deslocar todos os elementos do vetor uma posição para a esquerda (isto é, do endereço n para o endereço n-1). Como tal, o valor contido na posição mais à esquerda vai ser eliminado do vetor

**geracato**: A função gera um número aleatório, que pode ser zero com uma probabilidade predefinida de 95%, ou um valor uniformemente distribuído entre 1 e o valor máximo passado como parâmetro.

**desenha\_chao**: Através de alguns caracteres especiais, esta função desenha o chão no qual o dinossauro andará por cima

**escreveterreno**: Esta função escreve o vetor do terreno atualizado através da função atualizajogo. A função vai percorrer o vetor e sempre que encontrar um número maior que 0 vai desenhar no terminal um cato, apagando o a posição do cato no momento anterior.

**escrevedino**: Esta função vai desenha o dinossauro

**saltodino**: Esta função permite ao dinossauro executar o salto, sem que o terreno se mova com ele. Inicialmente a função começa por verificar se o dinossauro se encontra a meio da descida. Para isto vê se uma variável (DOWN\_TICK) se encontra a 1. Se for esse o caso, o dinossauro é desenhado uma linha abaixo, apagando a sua posição anterior, no final, verificando se já tocou no chão. Se o dinossauro estiver a subir, desenhamo-lo uma liha acima verificando também se já atingiu a altura máxima do salto (ALT\_SALTO). Se antigiu a altura máxima, define-se a variável DOWN\_TICK como 1. Se tocou no chão define-se a variável UP\_TICK a 0.

**gameover**: Função que verifica se o jogador perdeu o jogo, ou seja, se tocou num cato. Para averiguar o gameover, verifica-se se, no momento em que o jogador está na mesma coluna que um certo cato, se a sua altura é menor ou igual à do mesmo (se for esse o caso, significa que perdeu). Caso o jogador perca, o sinal 'GAME OVER' deverá aparecer. As funções que se seguem são obtidas do editor de texto da placa gráfica de forma a que o sinal 'GAME OVER' seja corretamente desenhado e centrado no ecrã do programa.

**PONTUACAO**: Esta função é usada para reproduzir a pontuação que o jogador obteve no jogo. Deste modo, em cada atualização do terreno, o jogador recebe 1 valor de score, que será reproduzido e atualizado no ecrã LCD. Como o valor reproduzido no ecrã seria em hexadecimal, tem de se passar este valor para decimal, ou seja, sempre que algum dos dígitos atingir o número Ah (10 em decimal) significa que esse dígito do display deverá passar a ter o valor 0, e o dígito do LCD seguinte o valor 1 (isto apenas é válido porque os 9

primeiros valores em hexadecimal e decimal são iguais). Quando as necessárias condições se verificarem, escreve-se então o número no display LCD. Após escrever o valor da pontuação, regressa para o endereço de retorno. Caso o valor seja Ah, tem de se converter esta nomenclatura para decimal.

**Observações:**

Por vezes, devido à estrutura das interrupções do P4, quando se pressiona o botão de seta para cima (UpKey) o temporizador não é inicializado pelo que o jogo para. Do mesmo modo, também é frequente que, quando se reinicia o jogo, este não é apresentado desde o início - pelo que aparece de imediato a tela de Game Over - ou o jogo só aparece a meio (apesar de ele estar a correr sem ser visualizado no terminal).