

O Código tem como função principal **Process_pontos**, onde maior parte do jogo é atualizado, desde a atualização do terminal até aos mostradores dos pontos.

Começamos por definir várias constantes e variáveis que serão usadas ao longo do código, por exemplo para ditar se loops serão ou não executados, e ajustadas de forma a refletir o estado do jogo para que todas as instruções sejam executadas conforme esse estado (no caso das variáveis). Definimos a atualização do cronómetro a cada 200 ms, logo o jogo correrá a cerca de 5 Frames por Segundo. A altura máxima do dinossauro é de 6 linhas, a dos cactos é de 4 linhas, isto tendo os seus cursores correspondentes.

No **Main** guardamos em tabelas os caracteres usados para desenhar o dinossauro, os cactos, e a frase GAME OVER, e iniciamos a Stackbase e a máscara de interrupções.

De seguida, entramos no **Loop**, que verifica o valor da variável **Estado_jogo**, caso o valor seja 0 ficamos à espera que o valor passe a 1, caso contrário, significa que o jogo está a decorrer e portanto vamos executar as próximas instruções, ou seja, vamos verificar se a variável **Timer_tick** nos indica que temos uma interrupção do cronómetro por tratar e se sim executamos o **Process_pontos**.

Na função **Process_pontos**, como já referido, é onde maior parte do código do jogo é tratado, começamos por ver se há colisão do dinossauro com um cacto, em **.show_time**, se existir colisão escrevemos GAME OVER no centro do ecrã, em **.eh_colisao_2** e damos reset no terreno de modo ao jogo começar sempre sem cactos no ecrã, que é feito em **.Reset_terreno**

Se não existir colisão devemos incrementar 1 ponto no display de 7 dígitos. De modo a que o score fosse representado em decimal decidimos que, quando o display mais à direita apresenta um nove, devemos colocar esse mesmo a zero, e avançar para o display à esquerda desse mesmo e incrementar 1 valor. Este processo é feito em **.ADD_DISPLAY_0**, **.ADD_DISPLAY_1**, **.ADD_DISPLAY_2**, **.ADD_DISPLAY_3** e **.MAKE0**. O valor máximo mostrado no display é 9999, dado que é muito complicado chegarmos a 9999 pontos.

De notar que ao longo deste processo guardamos e damos load várias vezes aos valores dos registos, de modo a respeitar a convenção estabelecida no P4.

Em **.cacto_atualiza**, é visto se desenhámos um cacto, recorrendo à tabela **valorterreno**, criada pelas funções **atualizajogo** e **geracacto**. Se não estiver um cacto numa coluna apagamos o 'possível' cacto escrevendo o carácter espaço na extensão da altura máxima que o cacto pode ter. Caso exista um cacto, ele é desenhado em **.desenha_cacto** e **.altura**, onde é escrito no terminal recorrendo à tabela **Cacto_Tab**, onde se encontram os caracteres do cacto, e desenhamo-lo até chegarmos à altura à qual ele corresponde. Depois de percorrermos a tabela toda acabamos o ciclo e prosseguimos a desenhar o dinossauro.

Em **.DES_DINO** é comparado o valor da variável **Estado_Dino**, que dita se o dinossauro está a subir, a descer ou no chão. Caso o **Estado_Dino** esteja a zero desenhámos o dinossauro no chão, o que é feito em **.Dino_chao_loop**, caso ele esteja a subir ou descer, (se a variável **Estado_Dino** esteja a 1 ou -1), saltamos para **.SALTA_DESCE**, e vemos se o cursor dos pés do dinossauro se encontra na altura máxima, se isso se verificar, ele deve passar a descer, senão desenhámos o dino a saltar nesta etiqueta, caso ele deva descer, saltamos para **.START_DESCE** e **.DESCE**, onde é desenhado o dinossauro quando ele se encontra a descer.

No fim de tratarmos do terminal e do display de 7 segmentos somos dirigidos à etiqueta **.FIM_reset_cron** onde damos Pop dos registos que estavam antes de executarmos as instruções de atualização do terreno.

Quando ocorre a interrupção da Key0, que nos direciona ao endereço 7F00h, e por conseguinte à função auxiliar **Aux_Inicia_jogo**, fazemos Push dos registos R1 a R5, colocamos os valores do display, os valores das variáveis e os que se encontram no mostrador a 0, depois apagamos o Game Over, que pode ou não estar no centro do ecrã, no loop **.loop_del_game_over**, e depois colocamos o chão todo a zero, usando um loop também. Acabamos a função auxiliar dando Pop aos valores guardados no início da função e voltando ao endereço 7F00h, depois dando RTI.

Já quando ocorre a interrupção da Seta para Cima saltamos para o endereço 7F30h, onde comparamos se o **Estado_Dino**, se encontra a 0, se isso se verificar, o dinossauro irá saltar, para isto acontecer, a única coisa que a função **Salta_Dino** faz é mudar a variável **Estado_Dino** para 1. Já tendo sido referido que usaremos esta variável como ponto crucial no programa principal.

Por fim, ao ocorrer uma interrupção do cronómetro, indo depois para o endereço 7FF0h, chamaremos duas funções. A primeira função, **Aux_Reset_Cron**, simplesmente reinicia o cronómetro, e a segunda função, **Chamafuncao**, irá chamar as funções estabelecidas na 1ª parte do projeto, (**atualizajogo** e **geracacto**), de modo a atualizar a tabela **valorterreno**, que é usada para definir o terreno.

A função **atualizajogo** modifica o valor da tabela **valorterreno**, movendo todos os valores da tabela uma posição para a esquerda, dando delete ao valor que se encontra mais à esquerda da tabela, e adicionando no valor mais à direita da tabela o valor gerado pela função **geracacto**, esta função tem a estrutura do pseudocódigo fornecido no pdf da primeira parte do projeto, esta função retorna um número, sendo 0 com cerca de 95% de certeza ou um valor entre 1 e 4.

Tivemos como pontos em consideração a eficiência do código, como a sua simplicidade e ter as interrupções as mais pequenas possíveis com o objetivo de fazer o jogo o mais fluído possível.

Escolhemos os caracteres dos cactos, do dinossauro e do chão de forma a ter uma simplicidade ao implementá-lo, mas também atribuir um nível de detalhe que satisfaz o jogador.