Relatório do Projeto de IAC

(Jogo baseado no chrome://dino/)

Projeto realizado por:
João Silva
Tiago Vieira da Silva

INTRODUÇÃO

Este projeto consiste em replicar o jogo do dinossauro do browser "Chrome" da Google utilizando a linguagem "Assembly" para o processador P4. O jogo consiste em ter um boneco (dinossauro) que percorre um terreno de jogo e salta sobre obstáculos (catos). No nosso jogo, o dinossauro está representado pela letra "D", e sempre que salta sobre um cato, a pontuação aumenta em 1. Em seguida iremos descrever de forma geral o que as principais funções do programa fazem e justificar algumas escolhas que foram feitas no projeto.

ATUALIZAJOGO:

Irá receber uma tabela e o tamanho da mesma como argumentos. Irá movimentar todos os elementos da tabela para o endereço anterior ao endereço onde estão guardados, eliminando o primeiro elemento e gerando um número aleatório que será inserido no último endereço.

Esse número irá ser gerado pela função GERACACTO, que é a aplicação em Assembly do "pseudo-código" que foi fornecido no primeiro enunciado.

ATUALIZATERRENO:

Recebe como argumento a tabela de jogo e atualiza o terminal para representar graficamente o jogo. Invoca as seguintes funções:

- TERRENOJOGO: Insere o solo do terreno de jogo no terminal.
- INSERIR_SPACE: Insere espaços nos sítios onde não é suposto haver catos, tendo como base a tabela. Deste modo apagamos o terminal antigo.
- INSERIR_CACTO: Insere o cato dependendo do sítio onde houver catos na tabela. O tamanho do cato depende do valor guardado na tabela.

Estas funções não modificam o terminal todo, apenas partes do mesmo. Esta escolha foi simplesmente para não ter que estar a apagar o terminal todo constantemente e estar a inserir continuamente o título "DINO". No entanto, outra opção seria apagar o terminal todo com a função "LIMPAR_TER" e invocar a função "WRITE_DINO" juntamente com as outras funções.

COLISÕES:

A função "COLISAO" trata da verificação de colisões. É invocada no início da função "INSERIR_CACTO" e vai verificar que no sítio onde vamos inserir o cato no terminal não se encontra ocupado pelo dinossauro. Para isso, verifica a altura em que se encontra o dinossauro e que altura terá o cato que irá ser inserido. Se a altura do dinossauro for menor ou igual que a do cato, irá haver uma colisão e o programa salta para a função "GAME_OVER"
GAME_OVER:

Esta função irá invocar a função “WRITE_GAME_OVER” que irá inserir o texto “GAME OVER” no ecrã. Após isso, irá esperar que o utilizador clique no botão 0, mudando a variável “GAME_RESTART” para 1. Após isso ocorrer, através da invocação de funções auxiliares repõe os valore iniciais das variáveis e limpa o display de 7 dígitos e volta para o início do programa principal.

INTERRUPÇÕES:

TIMER_ISR: Incrementa a variável TIMER_TICK (esta função foi reutilizada do exercício 3 da preparação para o laboratório 4).

SALTO_DINO: Sempre que o botão “seta para cima” for premido, muda o valor da variável “SALTO” para 1 (se já não estiver a 1). Essa variável, quando for igual a 1, indica ao programa que deve executar o salto.

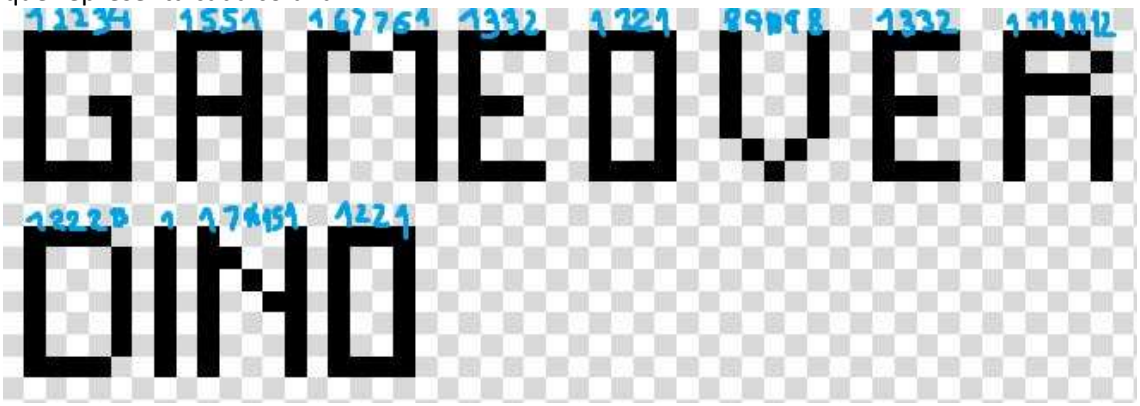
RESTART_GAME: Muda o valor da variável “GAME_RESTART” para 1. Essa variável, quando for igual a 1, indica ao programa que deve reiniciar o jogo.

PONTUAÇÃO:

Por cada cato que o dinossauro salta, a pontuação aumenta 1 valor. A pontuação é armazenada na variável “PONTUACAO”, e apresentada no display de 7 dígitos em base decimal. Para converter de base hexadecimal para decimal a pontuação, aplicamos a técnica das divisões sucessivas, sendo o cada dígito o número de vezes que decrementamos cada a pontuação por 10^n (sendo n o índice do dígito).

ESCRITA DO TEXTO GRÁFICO “DINO” e “GAMEOVER”:

Para escrever ambos os textos gráficos, escreveu-se cada letra coluna a coluna, empregando uso das funções “COLUNA”, visto haver letras que teriam colunas repetidas. Está indicado em comentário no programa quais são as posições da coluna que cada uma das funções “COLUNA” preenche. Na imagem seguinte está indicado qual é o número da função que representa cada coluna.



Não respeitamos a convenção do P4 para o salto de funções visto que isso faria o código desnecessariamente mais complexo e porque os valores de R4, R5 e R7 já são guardados nas funções “WRITE_DINO” e “WRITE_GAME_OVER”, tornando-se redundante guardar os valores destes registos na pilha de novo.