

RELATÓRIO DO PROJETO DE INTRODUÇÃO À ARQUITETURA DE COMPUTADORES: DINO

Intro:

O jogo simplificado do Dino no browser Chrome foi criado usando os recursos oferecidos pelo assembler e simulador do P4. Utilizamos o terminal para servir de base gráfica do jogo, após ter sido feita a base lógica do terreno na primeira parte do projeto. Inicialmente, o terminal encontra-se vazio, aguardando que seja pressionado o botão de pressão 0. Neste momento, o jogo começa. É dada a permissão ao jogador para controlar o dinossauro, mas apenas pode saltar para se desviar dos cactos. Durante o jogo irá aparecer no display de 7 segmentos o “score” do jogo atual. Após colisão do dinossauro com um cacto, o jogo termina, sendo apresentado no ecrã a mensagem: “Game over” e “Play Again: prima 0”. Assim, o programa fica a aguardar que seja pressionado o botão de pressão 0, sendo, nesse instante, reiniciado o jogo. De salientar que, após cada jogo, o display de sete segmentos mostra o “highscore” de cada sessão (a sessão termina ao voltar a assembler o programa).

Funções:

Primeira Parte:

- atualizajogo: rotina que gera um novo vetor cada vez que é executada, formando, com o auxílio do “geracacto” e do “salto” a parte lógica do programa.
- geracacto: função auxiliar que gera um número aleatório entre 0 e 4, retorno esse valor mesmo valor à função atualizajogo. Este número representa a altura do cacto a ser gerado, sendo que o valor 0 indica que não é gerado cacto, dando-se print no terminal apenas do terreno de jogo.

Segunda Parte:

- print_bloco: rotina auxiliar que recebe uma cadeia de caracteres (com cinco de comprimento) e imprime-a no terminal. Esta função pode imprimir partes do cacto, dinossauro, terreno de jogo e até limpar os cinco caracteres seguintes, no caso de ser oferecida uma cadeia de caracteres com cinco espaços vazios (“ ”).
- print_terreno: rotina que recebe um determinado valor, n, do vetor que representa o terreno e imprime um cacto de altura n ou um troço de terreno de jogo, se n for zero.
- print_dino: rotina que imprime o dinossauro a uma determinada altura dada com argumento após ter sido impresso o terreno de jogo na sua totalidade
- atualiza_term: rotina que permite uma transição suave entre cada bloco de cinco colunas e o controlo da velocidade do jogo com o auxílio do temporizador.
- salto: função que implementa o salto do dinossauro. Durante o salto, em movimento ascendente, o dinossauro sobe duas linhas por frame. Pelo contrário, no movimento descendente, o dinossauro desce uma linha por frame. De salientar que a variável de controlo “isJumping” toma o valor 1, quando está a ser efetuado um salto, 0 caso contrário.
- colisão: função que implementa a deteção de colisão do dinossauro com um obstáculo / cacto. Recebe como argumentos de entrada: em R1, a posição dos pés do dinossauro relativa ao eixo y (Dino_posy); em R2, o endereço do vetor relativo ao espaço (colunas) onde o dinossauro é imprimido no terminal; e em R4, o valor atual da iteração do loop em questão. A função retorna 1, caso haja colisão, 0 caso contrário.
- game_over: caso o valor retornado pela função anterior seja 1, esta função é invocada, comparando o “score” do jogo atual com o “highscore”. Caso o primeiro seja maior, o display de sete segmentos é atualizado com esse valor. Para além disso, imprime no terminal a indicação de que o jogo terminou, bem

como que botão é necessário pressionar, caso o utilizador queira jogar novamente.

Interface:

Decidimos, apesar de oferecer uma solução mais complexa, continuar com a nossa ideia inicial de desenhar todos os elementos do jogo com cinco colunas de largura. Deu origem a que o terminal fosse impresso por indução e não por consulta direta do vetor originado.

Trabalho realizado por:

Grupo nº88

Bruno Campos, 99187

Joao Henriques Sereno, 9924