

Projeto de Introdução à Arquitetura de Computadores (Primeira Parte)

De seguida encontra-se um pequeno relatório do programa que simula o estado do terreno de jogo. É apresentada uma breve descrição de todas as constantes e variáveis usadas, sub-rotinas implementadas e do programa de teste para esta primeira entrega do projeto.

Constantes e Variáveis

Para o funcionamento do programa declarou-se uma tabela **JOGO** mantida em memória a partir do endereço 0000h e de tamanho **N** (= 80). Esta tabela vai representar o terreno de jogo. Declarou-se também a constante **altura** que representa a altura máxima que os cactos podem ter (obrigatoriamente uma potência de 2) e a variável **x** que é usada como “semente” para gerar um número aleatório.

Sub-rotinas

principal: função principal onde se vai colocar em R1 o endereço de início da tabela JOGO (inicialmente com as colunas todas a 0), em R2 a dimensão da tabela e se inicializa a pilha (R6) que vai ser usada ao longo do programa. De seguida, realiza um salto para a função **atualizajogo**. Ao sair dessa função, volta a executar a função principal em ciclo infinito.

atualizajogo: função que recebe o endereço de memória do início da tabela JOGO (R1) e a dimensão da tabela (R2) como argumentos e atualiza o terreno de jogo a cada iteração. Esta função inicializa R5 a 1 que vai funcionar como variável auxiliar para ajudar a percorrer a tabela JOGO até ao fim durante o deslocamento dos seus valores, o que vai acontecer a seguir ao entrar na etiqueta **ciclo**. Não retorna nada.

ciclo: vai deslocar todas os elementos da tabela JOGO uma posição para a esquerda. Começa por incrementar R1 e guardar o seu valor em R4 (este vai ser o valor da coluna $n + 1$). Volta ao endereço onde estava (coluna n) e coloca lá o valor que está em R4. Incrementa R1 para ir avançando na tabela e R5 para atualizar a posição onde está. Ao incrementar R5, compara o novo valor à dimensão da tabela. Se for diferente de 0, isto significa que ainda não chegou ao fim, por isso volta a executar **ciclo** até esta condição se verificar. Quando chega ao fim prepara-se para chamar a função **geracacto** que recebe a altura, passada para R1 como argumento. Assim, tem de se guardar o valor de R1 (endereço da ultima coluna da tabela) em R4. Realiza o salto para **geracacto** e depois de sair desta função devolve a R1 o seu valor e guarda nesta posição o valor retornado por **geracacto**. Volta à função principal.

geracacto: função que recebe como argumento a altura máxima e executa o código fornecido no enunciado para geração de um valor aleatório para altura do próximo cacto. Esta função pode retornar 0 através da etiqueta **semcacto** ou um valor entre 1 e altura passado por R3, dependendo do valor do x . Atualiza x a cada iteração.

alterabit: função auxiliar que altera potencialmente qualquer bit de x e volta à função **geracacto**.

semcacto: retorna 0 por R3, ou seja não há cacto.

Programa de teste

Para testar o programa apenas realizamos um ciclo infinito na função principal, que futuramente acabará quando o dinossauro colidir com um cacto. Desta forma conseguimos observar várias iterações da função **atualizajogo** e verificar que o terreno de jogo é de facto atualizado podendo ou não ser gerado um cacto na ultima coluna a cada iteração.