Descrição das funções da parte 1 do projeto de Introdução à Arquitetura de Computadores

Função *atualizajogo*: adiciona os parâmetros iniciais do nosso jogo. No sentido de eliminar o valor contido na posição mais à esquerda do vetor, deslocámos o elemento para o endereço anterior (fora do vetor).

***.loop*:** subdivisão da função *atualizajogo* que têm como objetivo atualizar o vetor (mover os elementos todos do vetor uma posição para a esquerda (n = n-1)) até ao momento em que podemos saltar para a função *geracacto.*

Função *geracacto*: determina a altura dos cactos do nosso jogo. há uma chance de 95% da altura ser 0. A altura máxima do nosso código é 4, logo o valor de altura está entre [0, 4].

***returnzero*:** subdivisão de *geracacto*, gera um valor de altura 0, uma vez que após o XOR entre x e B400, x é menor que 62258. Demos o nome de *returnzero* pois é a subdivisão que faz a altura do cacto ser zero.

***xorf*:** subdivisão de *geracacto*, gera um valor de altura entre 1 a 4, uma vez que após o XOR entre x e B400, x é maior que 62258. Demos o nome de *xorf* pois usámos o XOR.

return: subdivisão de *geracacto*, para facilitar a leitura do código. após a execução de *xorf*, voltamos para esta parte.

***teste*:** em sentido do programa de teste, cria um loop infinito do nosso código para ser testada múltiplas vezes. Ao verificar o endereço 2000h da memória de dados, podemos visualizar a deslocação para a esquerda dos valores do vetor. Ao verificar o endereço 6000h da memória de dados, podemos visualizar a alteração do valor x.

`x & (altura -1) = mod(x, altura): a disjunção entre x e (altura -1) é igual ao resto da divisão entre x e altura, se altura = 2^n`

Isto é verdade porque:

2^n em decimal vai ser 1 seguido de n 0's em binário [2^2 = 4 | 100 = 4 em binário = 1 seguido de 2 zeros].
2^n -1 em decimal vai ser representado por n-1 1's em binário [$2^2 - 1 = 3$ | 11 = 3 em binário = 2 uns].
Logo, AND de um número 2^n vai resultar nos n bits menos significativos desse mesmo número (ou seja a potência de 2^n – número). Deste modo, podemos afirmar que as afirmações são de facto iguais.

EX: número = 14, 01110 | potência de 2 = 16, 10000 | 16 – 1 = 15, 01111

AND 01110, 01111 = 01110

mod(01110, 10000) = 01110 -> uma vez que 14/16 dá 0 e o resto é 14

Inês Ye Ji - **99238**
Raquel Filipa Marques Cardoso - **99314**