



Projeto de Introdução à Arquitetura de Computadores

Segunda parte

A segunda parte do projeto consiste no desenvolvimento de um programa, em Assembly do P4, que simula o movimento de uma bola num plano inclinado. Para simular o movimento, devem ser utilizadas as funções desenvolvidas na primeira parte do projeto, com possível exceção da função que calcula a aceleração: na primeira parte do projeto a aceleração provinha apenas da gravidade, ao passo que na segunda parte a aceleração provém do acelerómetro da placa utilizada. Para simplificar, o movimento será apenas unidimensional, ou seja, no eixo dos x.

O projeto deverá implementar a leitura periódica da aceleração e consequente utilização para o movimento e deverá mostrar o movimento da bola num ecrã do terminal.

Simulação do movimento

O acelerómetro da placa permite obter as acelerações segundo os eixos x, y e z. Neste projeto apenas será considerada a aceleração no eixo x. Para mais informações sobre a leitura da aceleração deve ser consultado o manual do P4.

A bola deverá mover-se até encontrar uma parede. Ao colidir, deverá ressaltar na direcção oposta.

A janela do terminal, de dimensão máxima 80, deverá mostrar os limites, ou paredes, como na Figura 1. As paredes devem ser representadas pelo carácter “*”. A bola deverá ser representada pelo carácter ‘o’ (minúsculo).



Figura 1 Terminal com a vista inicial do jogo. As paredes delimitam a região de jogo, entre as posições 0 e 79.

A posição inicial da bola deverá ser a primeira posição disponível, ou seja, a seguir à parede, que corresponde ao 1 (de 0 a 79).

Ao inclinar a placa, a bola deverá mover-se na direção da inclinação. Dependendo da velocidade da bola, poderá ressaltar ou não ao colidir com a parede. Este movimento está ilustrado na Figura 2.

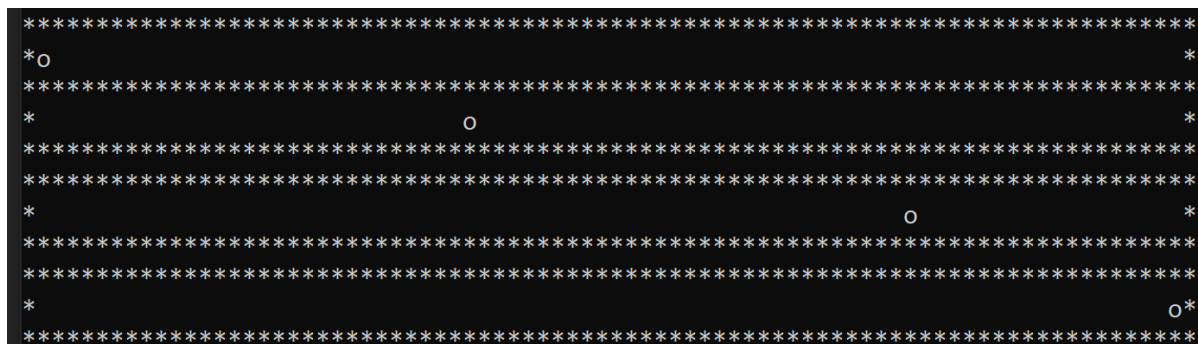


Figura 2 Movimento da bola ao inclinar a placa para a direita. Dependendo da velocidade, a bola poderá ainda ressaltar na direção oposta após colidir com a parede.

Para simplificar o cálculo do ressalto, deverá assumir que a velocidade após colisão é a simétrica da velocidade antes da colisão, ou seja:

$$v_{\text{após colisão}} = -v_{\text{antes da colisão}}$$

A leitura da aceleração e correspondente atualização da velocidade e posição deve ser feita de forma periódica, com um intervalo de tempo configurável através de uma constante. A rotina que trata a interrupção do temporizador (que desencadeia esta atualização) deve ser muito curta, limitando-se a ajustar o valor de uma ou mais variáveis, que depois serão utilizadas para controlar a execução do programa principal. Note que a qualidade do código faz parte da avaliação do projeto, o que inclui modularidade (e.g., uso de funções reutilizáveis e fáceis de manter), a passagem de parâmetros pela pilha de forma que obedeça às convenções ensinadas na cadeira, a utilidade dos comentários, ou ausência de rotinas de tratamento de interrupções longas.

Entrega

Dia 23 de Novembro, 23h59

Deverão ser entregues os ficheiros de código criados (ficheiros com formato .as) e o relatório do projeto (máximo duas páginas, 11pt, em formato .pdf).