

Arquitectura de Computadores

4º Trabalho de Laboratório

Programação Assembly

Objetivo: Pretende-se que os alunos compreendam a metodologia usada no desenvolvimento de programas em assembly, incluindo o uso de periféricos e de rotinas de interrupção. A programação será desenvolvida para o processador P3, sendo utilizado um assembler e um simulador disponibilizados para o efeito. **O trabalho terá a duração de duas semanas, devendo o projeto ser demonstrado na aula na segunda semana do laboratório.** O relatório final será entregue até à sexta-feira da segunda semana de laboratório, ou seja, **até dia 29 de Maio**, via Fénix. O trabalho deverá ser preparado fora do horário de laboratório, destinando-se as 3 horas de aula à resolução de eventuais dúvidas e demonstração do trabalho realizado.

1 JOGO PONG

Pretende-se implementar neste trabalho uma versão do jogo clássico da Pong. Pode ver o jogo no link:

<http://www.ponggame.org/>

É um jogo de dois jogadores em que o objetivo é pontuar. Cada jogador controla uma raquete que se desloca verticalmente junto a uma das paredes do jogo. Além das raquetes há uma bola que se desloca livremente pelo jogo, refletindo-se nas paredes horizontais, superiores e inferiores e nas raquetes. Sempre que um jogador deixa passar a bola e esta atinge a parede vertical o jogador adversário pontua.

2 IMPLEMENTAÇÃO DO JOGO

O jogo deve iniciar-se surgindo no ecrã a mensagem:

“Prima I0 para iniciar o jogo”,

ao que o jogo deverá ficar parado, esperando que o utilizador pressione o botão de pressão I0. Quando o botão for pressionado o jogo deve se iniciar.

2.1 Espaço de jogo

O espaço de jogo consiste em toda a janela de texto. Este deve estar limitado horizontalmente por duas paredes formadas pelo carácter ‘-’ e verticalmente por duas paredes formadas pelo carácter ‘|’.

2.2 Raquetes

As raquetes devem se movimentar nas colunas 5 e 74 da janela de texto, e não devem ultrapassar os limites da área de jogo. Cada raquete é representada por 5 caracteres ‘#’ na vertical. As teclas que controlam o movimento das raquetes são o ‘q’ e o ‘a’ para a raquete da esquerda e o ‘o’ e o ‘l’ para a raquete da direita. O movimento das raquetes deve ser independente do movimento da bola.

2.3 A bola

A bola deve-se deslocar em linha reta, refletindo-se sempre que chocar com uma parede horizontal ou com as raquetes. A reflexão deve ser perfeita, invertendo a componente horizontal ou vertical do movimento da bola. A bola deve ter apenas 4 direções possíveis, nomeadamente: para cima e para a direita, para cima e para a esquerda, para baixo e para a esquerda, para baixo e para a direita. A bola deve se deslocar um carácter de cada vez com temporizações controladas pelo timer do P3, com um período de 0.1s. Esta deve ser representada pelo carácter ‘O’.

2.4 Pontuação

Sempre que a bola atingir uma parede vertical o jogador adversário recebe um ponto. A bola deve ser recolocada numa posição aleatória do centro do espaço de jogo.

2.5 Posicionamento aleatório da bola

Depois de uma colisão com uma parede vertical, ou no início do jogo a bola deve ser posicionada aleatoriamente no centro do espaço de jogo. Nomeadamente a bola deve aparecer entre as linhas 6 e 17 inclusive e entre as colunas 30 49 inclusive. A direção da bola deve ser também aleatória.

Para gerar um inteiro de 16 bits pseudoaleatório deve utilizar o seguinte algoritmo:

```
Mascara = 1001 1100 0001 0110b
if (Ni0==0) /* Testa o bit de menor peso de Ni */
    Ni+1 = rotate_right(Ni);
else
    Ni+1 = rotate_right (XOR (Ni, Mascara))
```

Este algoritmo baseia na simulação de um registo de deslocamento modificado com realimentação, que permite gerar uma sequência pseudoaleatória de números de 16 bits, com um passo de repetição longo e com uma distribuição uniforme (i.e., os números são equiprováveis). Em cada invocação desta função lê-se o valor anterior N_i e gera-se um novo valor pseudoaleatório, N_{i+1} . O valor de N_i deve ser inicializado um valor diferente de zero (seed).

No lab vai ser necessário obter um número pseudoaleatório entre zero e $M-1$. Para obter este pode dividir N_i por M e obter o resto. Note que não deve modificar o valor de N_i , mas calcular um novo número, por exemplo Z_i , com,

$Z_i = \text{resto da divisão de } N_i \text{ por } M.$

2.6 Fim do Jogo

Quando um dos jogadores atingir os 5 pontos o jogo termina. Ganha o jogador com mais pontos. Deve ser exibida a mensagem

“Ganhou o jogador X”,

seguido de

“Prima I0 para iniciar o jogo”,

ao que o jogo deverá ficar parado, esperando que o utilizador pressione o botão de pressão I0. Quando o botão for pressionado o jogo deve se reiniciar.

3 JANELA DE TEXTO

A Janela de texto deverá ser semelhante ao representado na Ilustração 1 durante o decorrer do jogo.



Ilustração 1 – Janela de texto durante o decorrer do jogo

Esta é formada pelas paredes da área de jogo, representadas pelos caracteres ‘-’ e ‘|’, pelas raquetes formadas pelos caracteres ‘#’ e pela bola formada pelo carácter ‘O’.

4 JANELA DA PLACA

A Janela da placa está representada na Ilustração 2. Os periféricos da placa têm diferentes funções definidas de seguida.

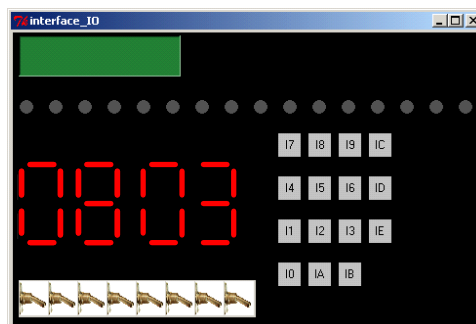


Ilustração 2 – Janela da placa

Display LCD – No display LCD deve estar indicada a pontuação atual de cada jogador, na forma:

J1: 3 J2: 5

Em que J1 indica a pontuação do jogador 1 e J2 indica a pontuação do jogador 2.

LEDs – Os LEDs devem se acender momentaneamente sempre que a bola choque com uma parede vertical.

Display 7 Segmentos – No display de sete segmentos deve estar representado o tempo decorrido desde o início do jogo em minutos e segundos.

Botões de pressão – O botão de pressão IA deve colocar o jogo em modo de pausa. Pressionando novamente o botão retira o jogo do modo de pausa. Enquanto o jogo está em pausa, a bola deve manter-se imóvel no ecrã e o tempo decorrido deixa de contar. O botão de pressão IO serve para dar início ao jogo, como já referido.

Nos diferentes periféricos vão necessitar de converter números inteiros para decimal e posteriormente para ASCII. A conversão para decimal deve ser efetuada por divisões sucessivas por 10, onde o resto é o dígito decimal que se vai retirando. Para converter para ASCII basta somar ‘0’ ou 48 ao dígito decimal.

Todas as questões adicionais, sobre o funcionamento e implementação do jogo, não definidas no enunciado ficam ao critério de cada grupo.

5 IMPLEMENTAÇÃO EM ASSEMBLER

Deve utilizar duas variáveis globais para a direção e para a posição da bola. Nomeadamente DIRECAO e POSICAO. A variável POSICAO deve corresponder às coordenadas na janela de texto tais como interpretadas pelo seu porto de controlo. Deve igualmente utilizar duas variáveis globais para a posição das raquetes.

6 FASEAMENTO

O trabalho decorrerá durante duas semanas, devendo ser discutido e apresentado na 2ª aula de laboratório. O trabalho deverá ser planeado e desenvolvido, tanto quanto possível, fora do horário de laboratório, sendo impossível realiza-lo sem qualquer preparação prévia e apenas durante as 3 horas de laboratório.

Na 1ª aula de laboratório devem ser tiradas todas e quaisquer dúvidas sobre o projeto. No 2º laboratório apenas será efetuada a visualização dos projetos.

Por fim, essencial a consulta do texto “Manual do Simulador do Processador P3”, e aconselha-se o estudo do programa de demonstração disponibilizado. Este material está disponível na página da cadeira.

7 RELATÓRIO

O relatório e o código comentado devem ser entregues via Fénix até à sexta-feira da 2ª semana do laboratório, ou seja até às 23h59 do dia **29 de Maio**. Entregas fora de prazo devem ser submetidas por email para o docente responsável pelo laboratório, que decidirá sobre a penalização a aplicar.

O relatório não incluindo o código não deve exceder as 3 páginas. O relatório deve ser fornecido no formato “.pdf” e o código em formato de texto, “.as” e em formato “.pdf”, tudo num ficheiro zip submetido via Fénix. Caso sejam submetidos múltiplos ficheiros, apenas será tomado em consideração o último ficheiro submetido.

8 AVALIAÇÃO

O trabalho será avaliado da seguinte forma:

1. Relatório a entregar via Fénix (**4 valores**), que consiste no seguinte:

- Fluxograma simplificado com a estrutura do programa desenvolvido. (**1 val.**);
- Listagem do código **DEVIDAMENTE COMENTADO**, onde serão avaliados a qualidade do código (estruturação, eficiência dos algoritmos, etc.) e a qualidade dos comentários; (**2 val.**);
- Apresentação do relatório (**1 val.**);

2. Funcionamento do programa (**16 valores**), avaliado nos seguintes pontos:

Janela de Interface

- DISPLAY 7 Segmentos (**1 val.**)
- LCD (**1 val.**)
- LEDs (**1 val.**)
- Interrupções (**1 val.**)
- Velocidade do jogo (**2 val.**)

Janela de Texto

- Representação estática do jogo (**2 val.**)
- Movimento da bola (**2 val.**)
- Movimento das raquetes (**2 val.**)
- Colisões (**2 val.**)
- A bola surge em posições aleatórias (**2 val.**)

9 BIBLIOGRAFIA

- [1] N. Horta, “Arquitetura de Computadores”, Aulas Teóricas.
- [2] M. Morris Mano, Charles R. Kime, “Logic and Computer Design Fundamentals”, 4th Edition Updated, Prentice-Hall International, 2008.
- [3] G. Arroz, J. Monteiro, A. Oliveira, “Arquitectura de Computadores: dos Sistemas Digitais aos Microprocessadores”, IST Press, 2007.
- [4] G. Arroz, J.C. Monteiro, A. Oliveira, “Manual do Simulador do P3”, IST, 2005