

LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES

**Projeto de
Introdução à Arquitetura de Computadores**

Corridas de Bicicleta

2013 / 2014

INSTITUTO SUPERIOR TÉCNICO

Índice

1. Objetivo	3
2. Descrição do Jogo	3
2.1. Espaço de Jogo	3
2.2. Bicicleta	4
2.3. Obstáculos	4
2.4. Dispositivos.....	5
2.5. Nível de Jogo	5
2.6. Início do Jogo.....	6
2.7. Fim do Jogo.....	6
2.8. Versões avançadas	6
3. Implementação	6
3.1. Movimentação da bicicleta.....	6
3.2. Movimentação do obstáculo	7
3.3. Temporizações	7
3.4. Valores aleatórios.....	7
4. Plano de Desenvolvimento	8
4.1. Desenvolvimento do trabalho.....	8
4.2. Faseamento da codificação	8
5. Plano de Entrega	9
Anexo A – Geração de sequência pseudoaleatória	10

1. Objetivo

A projeto consiste no desenvolvimento de um jogo de corridas de bicicleta, em que o objectivo é percorrer a maior distância possível sem chocar contra obstáculos. O jogo decorre numa janela de texto. Na janela de texto deve ser desenhada uma pista onde se desloca a bicicleta. Esta estará sempre no fundo da janela, deslocando-se para a esquerda ou para a direita, mas dentro dos limites da pista. No topo da pista vão surgindo obstáculos que se deslocam, descendo na na direção da bicicleta, e que devem ser evitados.

Neste documento são descritos os detalhes de funcionamento pretendidos para o jogo.

O jogo será programado usando a linguagem *assembly* do P3. O desenvolvimento e teste do programa serão realizados usando o simulador do P3 (p3sim), sendo usados os diversos recursos disponibilizados, de que se destacam: a janela de texto, os *displays* de 7 segmentos, os LEDs, o *display* LCD e os interruptores. O projeto deverá depois ser demonstrado no laboratório na placa com a versão hardware do P3.

2. Descrição do Jogo

2.1. Espaço de Jogo

O jogo irá desenrolar-se na Janela de Texto do simulador do P3. Esta janela corresponde a uma matriz de 80 colunas por 24 linhas em que, em cada posição, pode ser escrito um carácter ASCII. Apenas parte da janela será utilizada da forma que se descreve de seguida.

A pista incluindo as paredes tem uma largura de 26 caracteres, e encontra-se centrada no ecrã, com início na coluna 30 (primeira coluna da Janela de Texto é a coluna 0). As paredes são formadas pelos caracteres “+|” tal como representado na figura. A bicicleta é formada pelos caracteres, “O”-“|”-“O” dispostos verticalmente. Os obstáculos são representados por três asteriscos, “***”. Os obstáculos devem aparecer em posições aleatórias dentro da pista e deslocar-se na direção da bicicleta, numa velocidade controlada pelo jogo, criando assim a ilusão do avanço da bicicleta ao longo da estrada. O espaçamento entre obstáculos deve ser sempre de 5 caracteres.

A Figura 1 apresenta o espaço de jogo no início do jogo.

2.4. Dispositivos

O jogo propriamente dito decorre na Janela de Texto. No entanto, são usados também os seguintes dispositivos:

- o LCD deverá mostrar, na primeira linha, a distância atual percorrida (XXXXX), e na segunda linha a distância máxima percorrida (YYYYY):
Distancia: XXXXX m
Maximo: YYYYY m
- os displays de 7 segmentos deverão indicar o número de obstáculos ultrapassados;
- os leds são usados para indicar o nível do jogo:
 - 4 leds mais à esquerda acessos corresponde ao nível 1
 - 8 leds mais à esquerda acessos corresponde ao nível 2
 - 12 leds mais à esquerda acessos corresponde ao nível 3
 - todos os leds acessos corresponde ao Turbo (versões avançadas)

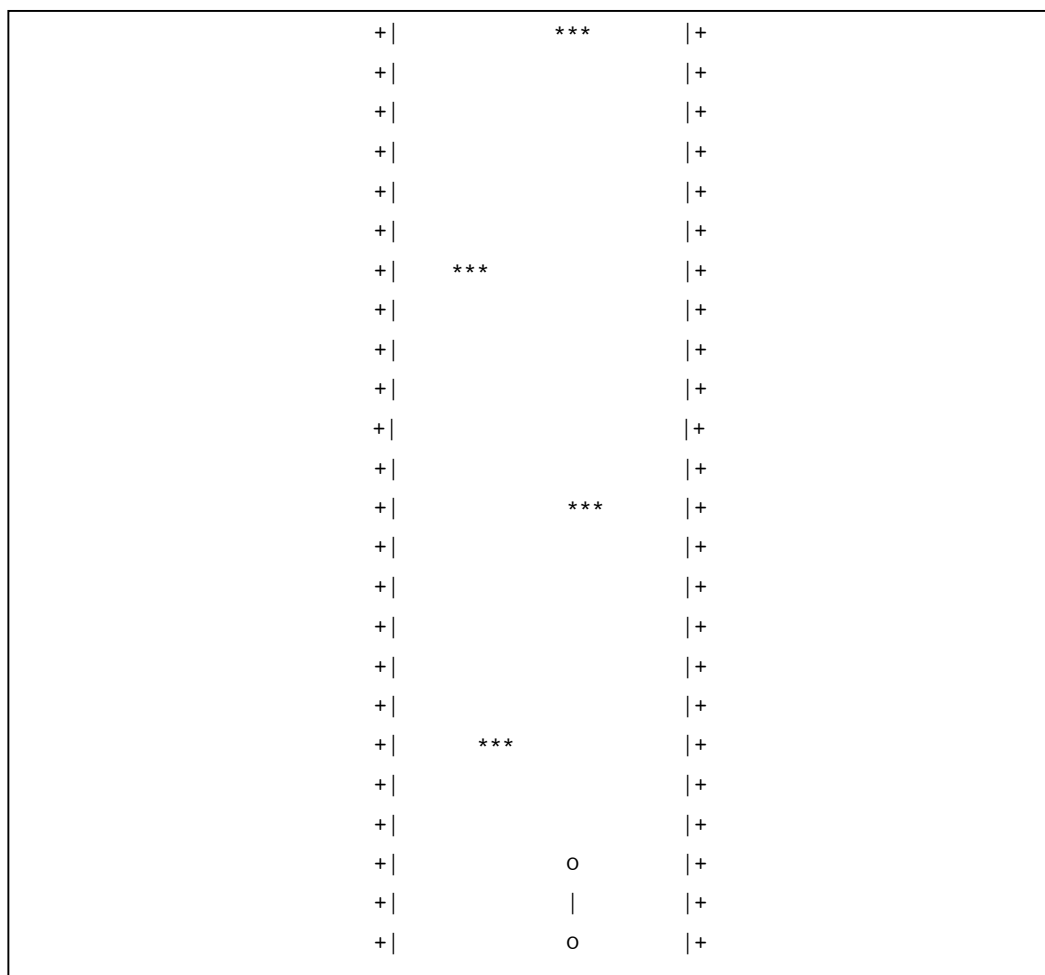


Figura 2 – Situação de jogo.

2.5. Nível de Jogo

O nível muda depois de ultrapassados 4 obstáculos. Deverão ser implementados 3 níveis. No primeiro nível a velocidade é de 2m/s, tal como já referido, no segundo será de 2,5m/s e no terceiro será de 3,3m/s.

2.6. Início do Jogo

Quando arranca, o programa deverá escrever, centrado respetivamente nas linhas 12 e 14, as mensagens “Bem-vindo à Corrida de Bicicleta!” e “Prima o interruptor I1 para começar”.

Quando I1 for premido:

- a janela de texto deverá ficar como indicado na Figura 1;
- a pontuação atual deve ser colocada a 0;
- apenas os 4 leds mais à esquerda se devem acender;
- os displays de 7 segmentos devem mostrar o valor 0.
- é gerado um primeiro obstáculo, que depois terá o comportamento indicado na Secção 2.3;

2.7. Fim do Jogo

O jogo termina quando a bicicleta chocar com um obstáculo. Nesta altura, o programa deverá escrever, centrado respetivamente nas linhas 12 e 14, as mensagens “Fim do Jogo” e “Prima o interruptor I1 para recomeçar”. A pontuação máxima deve ser também atualizada. Quando I1 for premido o jogo deve recomeçar.

2.8. Versões avançadas

Na versão base, a velocidade da bicicleta é apenas dependente do nível de jogo.

Esta versão pode ser estendida de duas formas, independentes entre elas:

- Implementar um modo de turbo. Neste caso, quando se carrega no botão de interrupção I2, a velocidade passará a ser igual a 5m/s e deverá manter-se assim até se pressionar novamente o botão I2. Quando em modo turbo, todos os leds devem estar acesos.
- Implementar um modo de pausa. Neste caso quando se carrega no botão de interrupção IA a execução do programa é suspensa e a mensagem “PAUSA” deve aparecer no LCD. Carregando novamente em IA, a execução do programa é continuada.

Naturalmente, estas variantes apenas devem ser consideradas depois da versão base estar afinada.

3. Implementação

Uma vez iniciado, o jogo consistirá num ciclo em que se atualizam as posições da bicicleta e dos obstáculos. Os movimentos são conseguidos apagando o símbolo da posição atual (i.e., escrevendo um espaço por cima) e escrevendo-o na nova posição.

3.1. Movimentação da bicicleta

A posição da bicicleta é atualizada por cada vez que o jogador premir os interruptores IO e IB, movendo uma posição para a esquerda ou para a direita, respetivamente. Este movimento implica a atualização dos três caracteres que definem a bicicleta. Naturalmente, é necessário verificar os limites da pista de jogo.

3.2. Movimentação do obstáculo

A nova posição do obstáculo é sempre a posição abaixo da atual. O ritmo de atualização da sua posição é determinado pela sua velocidade, com as temporizações indicadas na Secção 3.3. Quando o obstáculo atinge a linha de baixo da Janela de Texto o obstáculo desaparece.

3.3. Temporizações

O ritmo de atualização das posições dos obstáculos é controlado através do temporizador disponível no simulador do P3. O temporizador terá por isso que ser programado por forma a permitir as seguintes velocidades para a bicicleta:

- Nível 1: 2m/s
- Nível 2: 2,5m/s
- Nível 3: 3,3m/s
- Turbo (versões avançadas): 5m/s

De notar que apesar de estar referido que estas são as velocidades da bicicleta na verdade o que será temporizado será o movimento dos obstáculos.

3.4. Valores aleatórios

Quando aparece um obstáculo é necessário obter um valor aleatório para saber qual a coluna onde ele irá aparecer. O Anexo A descreve um algoritmo para a geração de valores aleatórios de 16 bits.

4. Plano de Desenvolvimento

4.1. Desenvolvimento do trabalho

Sugere-se o seguinte plano de desenvolvimento:

1. Desenhe o fluxograma que descreve cada um dos procedimentos da aplicação, com especial atenção à relação entre o fluxo do programa principal e as várias rotinas de tratamento de interrupção. Estes fluxogramas e a lógica funcional do programa principal deverão ser apresentados na 1ª aula de projeto.
2. Para o conjunto de procedimentos que definiu, identifique claramente as entradas, as saídas e os registos modificados na sua execução.
3. Programe e teste minuciosamente cada uma das rotinas que efetuam a interface com os dispositivos de entrada (interruptores) e os dispositivos de saída (janela de texto, LEDs, display 7 segmentos e LCD), com especial atenção à passagem de parâmetros entre estas rotinas e o programa principal.
4. Configure o temporizador disponibilizado pelo simulador e associe o vector de interrupção respectivo com a rotina a executar periodicamente. Configure o ciclo de jogo para que este seja sincronizado pela mudança de valor de uma dada variável, modificada pela rotina de tratamento da interrupção do temporizador.
5. Realize a ligação entre os vários procedimentos, de forma a obter o comportamento desejado e especificado.
6. Comente e indente devidamente o código desenvolvido. Inclua nos comentários referências aos fluxogramas que irá entregar para auxiliar a leitura e compreensão do programa.

4.2. Faseamento da codificação

Não deve tentar codificar todo o programa de uma só vez. Implemente as várias funcionalidades do programa de uma forma faseada e efetue os testes necessários para verificar o seu correto funcionamento. Não prossiga para a implementação de funcionalidades mais avançadas sem ter garantido que as que lhe servem de base estão corretamente implementadas.

Estando o sistema a funcionar corretamente, pode incluir eventuais funções opcionais que entretanto tenha desenvolvido.

5. Plano de Entrega

Semana de 11 a 15 de Novembro, na aula de laboratório	
Entrega: (em papel)	<ul style="list-style-type: none"> Fluxogramas da aplicação e dos principais procedimentos que lhe servem de base (mesmo que ainda não estejam implementados), com especial atenção à relação entre o fluxo do programa principal, os vários módulos funcionais e as rotinas de tratamento de interrupção.
Demonstração:	<ul style="list-style-type: none"> O programa a apresentar deverá fazer o desenho do cenário de jogo, a colocação da bicicleta e o seu movimento.
Dia 6 de Dezembro, até às 17h00, na Sala de Estudo do DEI	
Entrega: (em papel)	<ul style="list-style-type: none"> Código desenvolvido devidamente comentado e indentado (impresso frente e verso a duas páginas por face).
Dia 6 de Dezembro, até às 23h59, no Fénix	
Entrega: (electrónica)	<p>A entrega final deverá ser submetida num ficheiro zip com o nome no formato tAaBBgC.zip, em que: A é o dia da semana (2 a 6); BB é a hora de início (basta a hora, com 2 dígitos); C é o número do grupo. Este ficheiro deverá conter:</p> <ul style="list-style-type: none"> Breve relatório em PDF, com a descrição do projeto realizado, organização do programa e explicação dos aspectos mais relevantes da implementação. Na conclusão deverá ser feito um balanço do que foi realizado, com indicação dos aspectos nos quais o projeto tenha divergido do enunciado base (funcionalidades adicionais implementadas, funcionalidades não implementadas, outras variações ou divergências, etc.). Fluxogramas finais da aplicação e dos principais procedimentos que lhe servem de base. Código, quer ficheiro fonte, quer um PDF gerado com a aplicação p3print fornecida na página da cadeira.
Semana de 9 a 13 de Dezembro, na aula de laboratório	
Demonstração:	<ul style="list-style-type: none"> Funcionamento do jogo concebido.

As duas entregas em papel devem ser feitas num envelope identificado com o dia e hora do turno e número do grupo.

O calendário das discussões será acordado com o docente do turno respectivo. A discussão terá lugar, preferencialmente, na semana de 16 a 20 de Dezembro.

Anexo A – Geração de sequência pseudoaleatória

O seguinte algoritmo gera uma sequência aparentemente aleatória de números de 16 bits, com distribuição uniforme (isto é, os números são equiprováveis), com um passo de repetição elevado:

```
Máscara = 1000 0000 0001 0110 b  
  
if(n0 =0) /*Testa o bit de menor peso*/  
    Ni+1 = rotate_right (Ni);  
else  
    Ni+1 = rotate_right (XOR (Ni, Mascara));
```

Em cada invocação desta função lê-se o valor anterior N_i e gera-se um novo valor pseudoaleatório, N_{i+1} . A raiz desta sequência ($N_0 \neq 0$) pode ser obtida a partir de um parâmetro que varie com o decorrer do jogo (por exemplo, o número de ciclos de programa ou ciclos de espera entre a inicialização do programa e o início efetivo do jogo).