



ARQUITECTURA DE COMPUTADORES

LEIC-A

[illegible]

Versão 1.0

2009/2010

Nuno.Roma@ist.utl.pt

Conteúdo

1	Objectivo	3
2	Descrição do Jogo	3
2.1	Espaço de jogo	3
2.2	Entidades presentes	3
2.2.1	Comboios	4
2.2.2	Linha	4
2.2.3	Agulhas	4
2.2.4	Passageiros	4
2.2.5	Estações	5
2.3	Lógica de funcionamento	5
2.3.1	Movimentação dos comboios	5
2.3.2	Fases do jogo	5
3	Implementação	6
3.1	Deslocação dos comboios	7
3.2	Temporização	7
3.3	Colocação dos passageiros	7
3.4	Estruturas de dados e simulação	7
4	Microprogramação	7
5	Plano de Desenvolvimento	8
5.1	Desenvolvimento do trabalho	8
5.2	Faseamento da codificação	8
6	Plano de Entrega	9
A	Geração de sequência pseudo-aleatória	10

1 Objectivo

O objectivo do projecto é a concepção e implementação de um jogo interactivo usando o simulador do processador P3, como forma de desenvolver as capacidades de programação em assembly e de micro-programação.

O jogo que irá ser implementado será o “Jogo do Comboio”, análogo ao conhecido “*Train Game*”. Neste jogo existe uma linha férrea, onde se deslocam comboios. No espaço de jogo existe também um conjunto de agulhas, que determinam o percurso por onde circulam os comboios.

Neste documento são descritos os detalhes de funcionamento pretendidos para o jogo. O jogo será programado usando a linguagem assembly do P3 e será também microprogramada uma nova instrução. O teste e execução 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.

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 caracter ASCII.

O espaço de jogo, ilustrado na Figura 1, ocupa 64 colunas (0 a 63) e 24 linhas (0 a 23). Os números presentes na Figura 1 servem apenas para ajudar a identificar as várias linhas e colunas e não devem ser representados. O canto superior esquerdo corresponde à linha 0 e coluna 0.

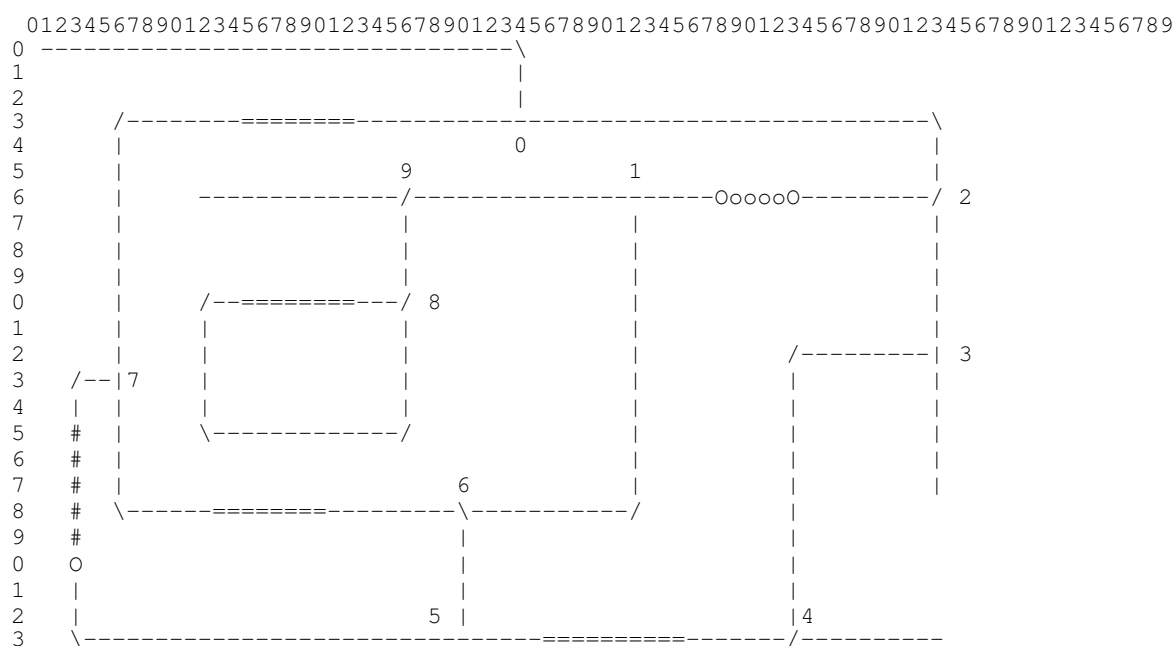


Figura 1: Espaço de jogo na Janela de Texto.

2.2 Entidades presentes

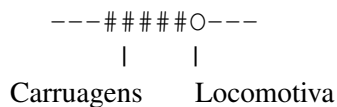
As entidades relevantes para o jogo são a linha férrea, as agulhas, as estações, os passageiros e dois comboios que circulam ao longo da via:

- **comboio do utilizador**, cujos pontos alcançados revertem para o jogador;
- **automotora “fantasma”**, com o objectivo de dificultar o jogo ao utilizador.

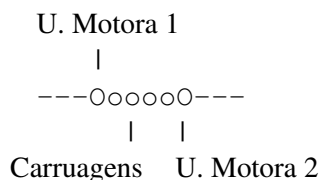
De seguida apresenta-se a representação de cada uma destas entidades.

2.2.1 Comboios

O **comboio do utilizador** é constituído por 5 carruagens e por 1 locomotiva situada à frente do comboio, onde se encontra sentado o maquinista que a conduz. A sua representação gráfica é a seguinte:



A **automotora “fantasma”** é constituída por 4 carruagens e por 2 unidades motoras (uma à frente e outra a trás), que lhe permite circular em ambos os sentidos da via férrea. A sua representação gráfica é a seguinte:



Na figura seguinte encontram-se representadas algumas posições possíveis de cada um dos comboios:



2.2.2 Linha

A linha férrea deverá ser representada pelos caracteres ‘-’ e ‘|’, correspondentes às direcções vertical e horizontal, respectivamente. As curvas deverão ser representadas com os caracteres ‘/’ e ‘\’. Ilustra-se, abaixo, a utilização destes caracteres para representar um pequeno troço da via:



2.2.3 Agulhas

As agulhas representam pontos de bifurcação da via e permitem escolher o troço da via férrea por onde o comboio irá circular. Estes elementos deverão ser representados por um identificador, correspondente a um dado número de ‘0’ a ‘9’, que indica o botão que permite ao utilizador trocar o estado actual de bifurcação da via.

Cada agulha deverá ser representada com os mesmos símbolos utilizados para representar a via férrea (‘-’, ‘|’, ‘/’ e ‘\’), devendo estes reflectir a direcção de comutação correspondente ao seu estado actual. Ilustra-se, abaixo, dois exemplos de representação de agulha (agulha ‘3’ e ‘7’):



2.2.4 Passageiros

Os passageiros deverão ser colocados junto das estações localizadas ao longo da via e deverão ser representados através do símbolo: &.

2.2.5 Estações

As estações correspondem a pontos de entrada e saída de passageiros e deverão ser representadas através do símbolo: ===== (ver Figura 1). Ilustra-se, abaixo, duas situações de representação das estações (com e sem passageiros):

Nenhum passageiro à espera:

3 passageiros à espera:

-----&&&=====

2.3 Lógica de funcionamento

As únicas entidades activas no cenário de jogo serão o **comboio do utilizador** e a **automotora fantasma**, que se deslocam ao longo da via férrea. O objectivo do jogo consiste em controlar as agulhas existentes ao longo da via de maneira a permitir ao **comboio do utilizador** apanhar o maior número possível de passageiros. O jogo termina assim que ocorrer uma das três situações seguintes:

- A locomotiva do comboio do utilizador atingiu um troço inválido, nomeadamente, o final de uma via férrea sem saída;
- A locomotiva do comboio do utilizador passou por cima de uma agulha indevidamente direccionada, fazendo-o descarrilar;
- A locomotiva ou uma carruagem do comboio do utilizador colidiu com a automotora fantasma.

No decorrer do jogo irão ser colocados passageiros nas várias estações existentes ao longo da linha. Sempre que qualquer um dos comboios passe pela estação, os passageiros que se encontram à espera deverão ser apagados, assumindo-se que entraram para o comboio.

2.3.1 Movimentação dos comboios

Os comboios deverão deslocar-se ao longo da via férrea e segundo o trajecto definido pelas várias agulhas por onde eles passam. O estado de cada uma destas agulhas deverá ser controlado pelo utilizador.

Contudo, enquanto que o comboio do utilizador apenas se desloca segundo o sentido para o qual está direccionada a sua locomotiva e de acordo com a direcção das agulhas por onde esta passa, a automotora fantasma rege-se segundo um princípio diferente:

- Sempre que uma das suas unidades motoras atinge o final de uma via férrea sem saída, o sentido do movimento *inverte-se*, passando a ser a outra unidade motora a comandar a automotora;
- Sempre que a unidade motora que comanda a automotora atinge uma agulha, esta pode ignorar o seu estado de bifurcação actual (de forma a evitar o descarrilamento), podendo assim escolher a direcção que lhe permite continuar por uma via férrea válida. Cabe ao programador definir e implementar o algoritmo que melhor permite concretizar esta funcionalidade.

2.3.2 Fases do jogo

O jogo desenvolve-se ao longo das seguintes fases:

1. Inicialização;
2. Jogo - movimentação dos comboios;
3. Apresentação dos resultados.

Início do jogo

O jogo inicia-se com o desenho no ecrã (janela de I/O) do cenário do jogo constituído pela linha férrea, as agulhas, os comboios e as estações.

É parte essencial deste trabalho a microprogramação de uma nova instrução assembly que deverá ser utilizada nesta fase do programa. Só após todo o cenário de jogo ter sido desenhado o jogo pode iniciar-se.

O jogo começa quando o jogador carregar numa tecla qualquer. A partir desse momento os comboios começam a deslocar-se segundo o trajecto definido pela linha e pelas agulhas por onde passam.

No caso do comboio do utilizador, o sentido de deslocação é definido pela posição da locomotiva no cenário inicial. No caso da automotora fantasma, este deverá ser definido de uma forma aleatória, utilizando para isso a rotina de extração de números aleatórios fornecida no Anexo A.

A velocidade do jogo (nível de dificuldade) é definida pela posição dos interruptores no início do jogo, de acordo com o indicado na Secção 3.2.

Pontuação

O objectivo do jogo consiste em controlar as agulhas existentes ao longo da via de forma a permitir ao **comboio do utilizador** apanhar o maior número possível de passageiros e fazer com que este percorra a maior distância possível sem que descarrile ou colida com a automotora fantasma.

A pontuação é atribuída em função do número de passageiros apanhados e da distância percorrida:

- Por cada passageiro apanhado, devem ser somados 100 pontos;
- Por cada segmento (=1 caracter) de via percorrido, deve ser somado 1 ponto.

A pontuação actual do jogador deve ser visualizada nos displays de 7 segmentos do simulador.

Fim do jogo

O jogo termina sempre que uma das três situações descritas no início desta secção ocorre, fazendo com que o comboio descarrile.

Nesse instante, o LCD deverá ser actualizado com a indicação da pontuação máxima conseguida até ao momento nos diversos jogos já jogados (*record* máximo actual).

Quando o jogo termina, o cenário no plano de jogo deverá permanecer estático e na zona lateral direita deverá ser apresentada a seguinte mensagem:

```
Prima tecla  
'N' para  
novo jogo
```

Quando o jogador pressionar na tecla "N", inicia-se um novo jogo.

3 Implementação

Uma vez iniciado, o jogo consistirá num ciclo em que se actualiza a posição dos comboios e se desencadeia a geração dos passageiros, a colocar nas várias estações.

3.1 Deslocação dos comboios

A nova posição de cada um dos comboios é determinada pela linha férrea pela qual ele circula. Para além dos segmentos rectilíneos, esta linha inclui pontos de mudança de direcção (curvas) e pontos de bifurcação (agulhas). Independentemente do tipo de troço, a locomotiva do comboio apenas deve inspeccionar o carácter que se encontra na direcção actual do deslocamento:

- Caso o carácter seja '|' ou '—', e esteja alinhado com o sentido do movimento, a locomotiva deverá continuar a direcção actual;
- Caso o carácter seja '\ ' ou '/ ', a locomotiva deverá realizar uma mudança de direcção segundo uma rotação de 90° determinada por esse mesmo carácter, pelo que a posição que deverá ser inspeccionada no próximo passo de jogo situa-se à esquerda ou à direita, face à direcção actual.

3.2 Temporização

O ritmo de actualização da posição dos comboios deverá ser controlado através do temporizador disponível no simulador do P3. Sempre que o temporizador expirar, a interrupção associada deverá assinalar e desencadear a execução de um novo ciclo.

O período a considerar pelo temporizador será determinado pela posição dos interruptores. Caso estejam todos desligados (valor 0), o período a carregar será de 300ms (velocidade mais rápida). Caso contrário, a leitura dos interruptores será interpretada como o número de incrementos de 100ms a serem adicionados ao valor base. Por exemplo, a combinação em que apenas estão ligados o primeiro e o terceiro interruptores é interpretada como o valor 5, pelo que o período do temporizador será de $5 \times 100 + 300 = 800 \text{ ms}$.

O número de LEDs acessos em cada instante deverá reflectir o nível de dificuldade seleccionado.

3.3 Colocação dos passageiros

Deverá ser colocado um novo passageiro numa das estações existentes ao longo da via a cada **50 ciclos** de jogo. A escolha da estação onde o novo passageiro deverá ser colocado deverá ser definida de uma forma aleatória, utilizando-se para isso a rotina de extração de números aleatórios fornecida. A colocação do novo passageiro deverá ser **abortada** caso o número máximo de passageiros em espera (**8**) na estação seleccionada já tenha sido atingido.

3.4 Estruturas de dados e simulação

Deverá ser concebida uma estrutura de dados residente na memória do processador que represente fielmente o cenário de jogo. Será através desta estrutura que será gerido todo o conjunto de eventos do jogo, nomeadamente, a detecção de mudança de direcção da locomotiva, a posição das estações, etc.

Contudo, embora seja naturalmente aceite a representação nesta estrutura de algumas das entidades presentes no jogo (ex: passageiros, etc.) várias alternativas existem quanto à representação das entidades móveis do jogo. Por conseguinte, deixa-se ao critério dos alunos a definição da(s) estrutura(s) de dados que sirva(m) este propósito o mais eficientemente possível.

4 Microprogramação

No âmbito do projecto deverá ser microprogramada uma nova instrução assembly que será usada para facilitar o desenho do cenário de jogo, durante a fase inicialização. Esta nova instrução deverá realizar a escrita na *Janela de Texto* do simulador de uma sequência de 64 caracteres de uma determinada linha do cenário de jogo. A sua utilização será:

```
I2OP op1, op2
```

em que:

- I2OP é o nome da nova instrução. É um nome genérico reservado no compilador/simulador para a implementação de uma nova instrução de 2 operandos, e tem o código de instrução (opcode) 101111 (2Fh);
- op1 é o endereço da cadeia de 64 caracteres que deverá ser escrita na *Janela de Texto*;
- op2 é a posição, na *Janela de Texto*, a partir da qual será escrita a linha de 64 caracteres endereçada por op1. O byte mais significativo (MSB) deste operando de 16 bits indica o número da linha (0-23), enquanto que o byte menos significativo (LSB) indica o número da coluna (0-79).

5 Plano de Desenvolvimento

5.1 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 projecto.
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 efectuem a interface com os dispositivos de entrada (teclado e 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. Associe as rotinas que realizam a interface com os botões de pressão do teclado com o vector de interrupção respectivo.
5. 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.
6. Realize a ligação entre os vários procedimentos, de forma a obter o comportamento desejado e especificado.
7. Embora o desenho inicial do cenário de jogo vá ser realizado pela nova instrução I2OP, comece por realizar uma implementação provisória deste procedimento utilizando uma rotina escrita em linguagem assembly. Deste modo poderá iniciar, o mais cedo possível, o desenvolvimento dos vários procedimentos envolvidos no jogo. Esta rotina deverá ser substituída pela invocação da nova instrução logo após a 2ª aula de projecto.
8. Comente e indente devidamente o código desenvolvido. Inclua nos comentários referências aos fluxogramas que irá entregar para auxiliar a leitura e compreensão do programa.

5.2 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 efectue os testes necessários para verificar o seu correcto funcionamento. Não prossiga para a implementação de funcionalidades mais avançadas sem ter garantido que as que lhe servem de base estão correctamente implementadas.

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

6 Plano de Entrega

Todas as entregas devem ser feitas num envelope identificado com o dia e hora do turno e número do grupo.

A entrega final deve conter ainda:

- Relatório.
- Código impresso. Cada ficheiro deverá ser impresso utilizando 2 páginas por face, frente e verso, numeradas; sem linhas cortadas; devidamente comentado e indentado; agrafado; com a identificação do grupo como comentário na primeira página. Para tal deverá utilizar a aplicação **p3print** fornecida na página da cadeira.
- CD-ROM com o código e relatório (devidamente identificado com o número do grupo)

Os projectos que sejam entregues após a data limite serão aceites com uma penalização de 2 valores por cada dia de atraso (incluindo sábados, domingos e outros dias não úteis).

O calendário das discussões será acordado com o docente do turno respectivo. Para os projectos entregues atempadamente a discussão terá lugar, preferencialmente, entre 31 de Maio e 9 de Junho. As discussões dos trabalhos entregues fora de prazo poderão ser adiadas.

1ª Aula (19 a 23 de Abril)	
Entrega:	- 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:	O programa a apresentar deverá fazer o desenho do troço completo da linha férrea, bem como a movimentação do comboio num segmento da linha.
2ª Aula (10 a 14 de Maio)	
Entrega:	- Microcódigo da instrução I2OP; - Código assembly do programa de teste da instrução I2OP realizada.
Demonstração:	Execução da instrução I2OP utilizando um programa de teste que demonstre todos os casos possíveis.
3ª Aula (17 de Maio a 21 de Maio)	
Entrega:	- Breve relatório com a descrição do projecto 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 projecto 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 desenvolvido devidamente comentado e indentado (impresso frente e verso a duas páginas por face). - CD-ROM com o código e relatório.
Demonstração:	Funcionamento do jogo concebido.

A Geração de sequência pseudo-aleatória

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 ( $n_0 == 0$ ) /* Testa o bit de menor peso de  $N_i$  */
     $N_{i+1} = \text{rotate\_right} (N_i);$ 
else
     $N_{i+1} = \text{rotate\_right} (\text{XOR} (N_i, \text{Mascara}));$ 
```

Em cada invocação desta função lê-se o valor anterior N_i e gera-se um novo valor pseudo-aleatório, 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 efectivo do jogo).