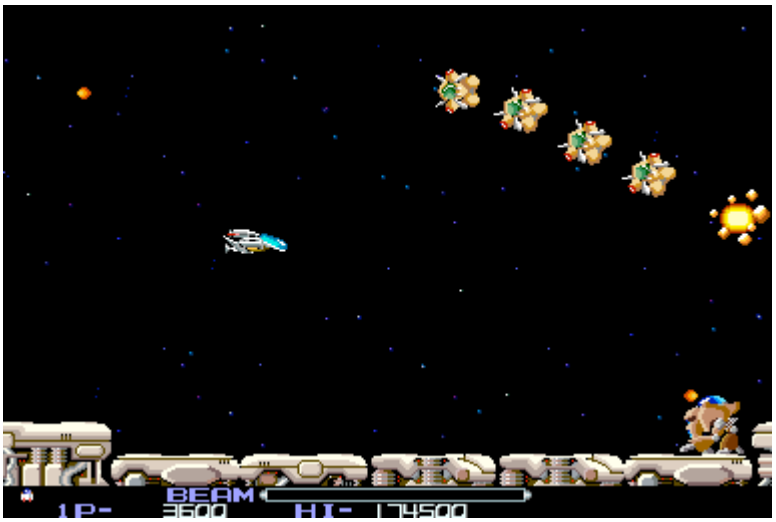
INSTITUTO SUPERIOR TÉCNICO

LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES

PROJETO DE INTRODUÇÃO À ARQUITETURA DE COMPUTADORES

R-TYPE



2016 / 2017

ÍNDICE

OBJETIVO

O projeto consiste no desenvolvimento de uma versão minimalista do jogo 'de arcada' R-Type. Nesta versão o objectivo é ter uma nave espacial a percorrer a maior distância possível sem chocar contra asteroides ou buracos negros. O jogo decorre numa janela de texto. A nave poderá estar em qualquer zona da janela, deslocando-se na vertical e na horizontal. No lado direito do ecrã vão surgindo asteroides e buracos negros que se deslocam a uma velocidade constante em direção à nave, e que devem ser evitados e/ou destruídos.

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.

DESCRIÇÃO DO JOGO

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 caráter ASCII.

Na janela de texto estão desenhadas duas linhas horizontais (a primeira e a última linha do ecrã) que delimitam o terreno de jogo. Estas linhas estão desenhadas recorrendo a uma sequência de carateres '#'.

A nave é formada por quatro carateres [\ ) > /], em 3 linhas:

```
\
) >
/
```

O canhão laser está no centro da nave e dispara tiros que se deslocam da esquerda para a direita na horizontal.

Os asteroides são representados por um ou mais carateres asterisco "*", e devem surgir na coluna mais à direita do ecrã, e deslocar-se na direção da nave, numa velocidade controlada pelo jogo, criando assim a ilusão do avanço dos asteroides ao longo do percurso. O espaçamento entre asteroides deve ser sempre de 5 carateres.

A Figura 1 apresenta o espaço de jogo em execução, incluindo todos os elementos.

```
################################################################################




                                  *                  O



                                                           *                O

                    \                                            *
                    |>     -            *
                    /


                                               *



                                                                       *


################################################################################
```

Figura 1 – Disposição do espaço de jogo na Janela de Texto do P3.

NAVE ESPACIAL

A nave deve começar com o canhão na coluna 3 e na linha 5. Usando os botões de interrupção, o jogador pode deslocar a nave para: baixo (I0), cima (I1), esquerda (I2) e direita (I3). A nave mantém-se sempre visível e não deve tocar nos asteroides nem nos buracos negros que vão passando, sob pena de terminar o jogo. A nave também não pode ultrapassar os limites do espaço de jogo, podendo escolher se nesse caso termina o jogo ou apenas impossibilita o movimento da nave.

ASTEROIDES & BURACOS NEGROS

Os asteroides, simbolizados por '*'s, surgem na coluna da direita da Janela de Texto, e correspondem a formações rochosas que devem ser evitadas e/ou destruídas usando o canhão laser da nave. Se parte da nave colidir com um asteroide o jogo termina e deve ser impressa no centro de ecrã uma mensagem de fim de jogo. Os asteroides surgem espaçados de 5 caracteres (5 espaços depois do '*' do último asteroide gerado, mesmo sendo destruído). Os asteroides, quando chegam à parte esquerda da janela de texto, desaparecem.

Os buracos negros, simbolizados por 'O's, não podem ser destruídos pelo canhão laser, e a nave espacial não lhes pode tocar sob pena de terminar o jogo. Resta ao utilizador desviar-se dos buracos negros.

Após serem gerados 3 asteroides (ou asteriscos) surge um buraco negro no lugar de um novo asteroide (seria o 4º). As regras de geração (5 espaços de intervalo e linha aleatória mantém-se).

TIROS LASER

Os tiros do canhão laser, simbolizados por um carater '-', surgem no canhão da nave espacial e seguem uma trajetória horizontal. Os tiros movem-se ao dobro da velocidade dos asteroides, mas em sentido contrário. Quando um tiro colide com um asteroide, deve suprimir o asteroide. O disparo de um tiro laser é desencadeado pelo botão I4.

Numa primeira fase, para uma cotação parcial, comece por colocar asteroides com apenas um '*' na parte inferior da janela (na linha 20), que a nave terá de evitar colidir. Só após ter esta funcionalidade completa deverá, numa fase seguinte, finalmente mudar para asteroides que aparecem numa posição aleatória do ecrã. Só após esta funcionalidade estar completa deve implementar a funcionalidade dos tiros laser.

PERIFÉRICOS PARA INTERACÇÃO COM O UTILIZADOR

Para além da funcionalidade principal do jogo na janela de texto, são usados também os seguintes dispositivos periféricos:

- O LCD deverá mostrar, na primeira linha, as coordenadas do canhão da nave: XX colunas e YY linhas.

- Os displays de 7 segmentos deverão indicar o número de asteroides (*) destruídos (equivalente a pontuação);

- Os LEDs são usados para indicar a ocorrência de um asteroide destruído: os leds acendem quando um asteroide é destruído.

- Os botões I0-I3 deslocam a nave para: baixo, cima, esquerda e direita, respetivamente.

- O botão I4 dispara um tiro laser.

- O botão IE (re)inicia o jogo.

INÍCIO DO JOGO

Quando arranca, o programa deverá escrever, centrado respetivamente nas linhas 12 e 14, as mensagens "Prepare-se" e "Prima o botão IE". Quando IE for premido:

- a janela de texto deverá ficar como indicado na Figura 1;

- a pontuação atual deve ser colocada a 0;

- os leds apagados;

- os displays de 7 segmentos devem mostrar o valor 0.

- é gerado um primeiro asteroide, que depois terá o comportamento indicado;

FIM DO JOGO

O jogo termina quando a nave chocar com um asteroide ou buraco negro. Nesta altura, o programa deverá escrever, centrado respetivamente nas linhas 12 e 14, as mensagens “Fim do Jogo” e a pontuação final (em asteroides destruídos). Quando qualquer botão ou interruptor for premido o programa deve reiniciar.

IMPLEMENTAÇÃO

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

MOVIMENTAÇÃO DA NAVE

A posição da nave é atualizada aquando de movimentos dados pelo utilizador. Este movimento implica a atualização dos quatro carateres que definem a nave.
Naturalmente, é necessário verificar os limites da zona de jogo. Numa primeira fase, para uma cotação parcial, comece por considerar que a nave apenas se desloca para cima e para baixo e ignore os limites de jogo.
Numa segunda fase, e para a cotação completa, deverá considerar que a nave se desloca nas quatro direções e considerar os limites (podendo optar se termina o jogo ou impossibilita o movimento para além dos limites).
A deslocação da nave nas quatro direções é adicional ao movimento contínuo do terreno de jogo, que é simulado com o avançar dos asteroides da parte direita para a parte esquerda do écran, conforme explicado de seguida.

MOVIMENTAÇÃO DOS ASTEROIDES E DOS BURACOS NEGROS

A nova posição do asteroide, ou do buraco negro, é sempre numa posição à esquerda da última posição. O ritmo de atualização da sua posição é determinado pela sua velocidade, que é controlada variando as temporizações. Quando um asteroide, ou um buraco negro, atinge a coluna da esquerda da Janela de Texto, ele desaparece.

Se da movimentação de um asteroide ou de um buraco negro ocorrer uma colisão com a nave espacial, o jogo termina.

TEMPORIZAÇÕES

O ritmo de atualização das posições dos asteroides, dos buracos negros e dos tiros é controlado através do temporizador disponível no simulador do P3. O temporizador terá por isso que ser programado por forma a ter uma velocidade adequada a uma boa experiência de jogo. No entanto, como a velocidade dos tiros laser é o dobro da dos asteroides, tal requer duas contagens de tempo diferentes. Assim, sugere-se configurar o temporizador para um intervalo de tempo fixo, e utilizar contadores individuais (cujas variáveis devem ter inicializações diferentes) para controlar a movimentação dos asteroides e do laser. Desta forma, sempre que ocorre um interrupção do temporizador, pode decrementar cada uma destas variáveis. Sempre que uma destas variáveis atinge o valor zero, devem atualizar os objetos correspondentes. Quando o utilizador movimenta a nave, essa atualização deve ocorrer de imediato.

Note que em geral as interrupções devem ser rotinas muito curtas, normalmente limitando-se a ajustar o valor de uma ou mais variáveis, que depois serão utilizadas para controlar a execução do programa principal.

VALORES ALEATÓRIOS

Quando aparece um novo asteroide é necessário obter um valor aleatório para saber que linhas ele vai ocupar. O Anexo A descreve um algoritmo para a geração de valores aleatórios de 16 bits.

DÚVIDAS NA ESPECIFICAÇÃO DO ENUNCIADO

Sempre que o enunciado não defina completamente como deve implementar alguma parte da solução, deve tomar a decisão que achar mais apropriada, tendo sempre em mente o objetivo de melhorar a experiência de jogo. Estas decisões devem ser descritas e justificadas no relatório final.

PLANO DE DESENVOLVIMENTO

DESENVOLVIMENTO DO TRABALHO

Sugere-se o seguinte plano de desenvolvimento:

- O primeiro passo é entender a estrutura geral da aplicação. Em particular, esta deverá ter, após o início do jogo, um ciclo de jogo cujas mudanças no estado do jogo ocorrem pela mudança de valor de uma dada variável, que por sua vez é modificada pela rotina de tratamento da interrupção do temporizador. Esta estrutura geral deve ser discutida entre os elementos do grupo, e se necessário também com o docente dos laboratórios, para ser bem entendida.

- 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 primeira aula de projeto (conforme a calendarização no final do enunciado).

- Para o conjunto de procedimentos que definiu, identifique claramente os parâmetros de entrada, as saídas (valores de retorno) e os registos modificados na sua execução.

- 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.

- Configure o temporizador disponibilizado pelo simulador e associe o vetor de interrupção respetivo com a rotina a executar periodicamente. Configure o ciclo de jogo com a estrutura atrás indicada, em que a mudança de valor de uma variável desencadeia uma atualização no estado do jogo.

- Realize a ligação entre os vários procedimentos, de forma a obter o comportamento desejado e especificado.

- Comente e indente devidamente o código desenvolvido. Inclua nos comentários referências aos fluxogramas que irá entregar para auxiliar a leitura e compreensão do programa.

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.

PLANO DE ENTREGA

SEMANA DE 14 DE NOVEMBRO, NA AULA DE LABORATÓRIO

Entrega em papel:

- 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 cenário de jogo, bem como a colocação da nave e o seu movimento nas quatro direções (N-S, E-W). A nave pode não disparar tiros laser.

DIA 5 DE DEZEMBRO, ATÉ ÀS 10H00, NA SALA DE ESTUDO DO DEI

Entrega em papel:

- Código desenvolvido devidamente comentado e indentado (impresso frente e verso a duas páginas por face).

DIA 2 DE DEZEMBRO, ATÉ ÀS 23H59, NO FÉNIX

Entrega electrónica:

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:

- Breve relatório em PDF (máximo 2 páginas em 11pt) 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 5 DE DEZEMBRO, NA AULA DE LABORATÓRIO

Demonstração do 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.

Haverá uma discussão final, com perguntas individuais a cada elemento do grupo, e que podem incluir perguntas sobre o projeto e sobre os laboratórios. O calendário das discussões será acordado com o docente do turno respectivo. A discussão terá lugar, preferencialmente, na semana de 12 de Dezembro.

MATRIZ DE AVALIAÇÃO DO PROJETO

Elemento de avaliação	Valor
Primeira entrega	3
Funcionalidade básica (movimentos dos vários componentes no espaço de jogo, colisões, tiros)	7
Funcionalidade avançada (níveis, uso de LCD, LEDs e display de sete segmentos, versões avançadas)	3
Relatório	2.5
Código	4.5
Total projeto	20

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 Ni e gera-se um novo valor pseudoaleatório, Ni+1. A raiz desta sequência (N0) 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). Não deve modificar o valor de Ni entre invocações. Para obter um valor aleatório entre 0 e M pode dividir o valor de Ni por M e retirar o resto, mas sempre sem modificar o valor de Ni.

*** Fim ***