



INSTITUTO
SUPERIOR
TÉCNICO

Relatório do Projecto “Pac-Man”

Licenciatura em Engenharia Informática
e de Computadores
Arquitectura de Computadores
2011/2012



GRUPO 33

André Silva	68707
Mariana Azevedo	72595



Índice

INTRODUÇÃO	2
ORGANIZAÇÃO DO PROGRAMA.....	3
LÓGICA DO PROGRAMA PRINCIPAL.....	3
DESCRIÇÃO DO JOGO	4
ESTRUTURAS DE DADOS	4
PRINCIPAIS ROTINAS	5
<i>EscreveNivel</i>	5
<i>EscreveMapa</i>	5
<i>EscrevePont</i>	5
<i>ActMapa</i>	6
<i>Rotinas de Movimento do Pac-Man</i>	6
<i>Rotinas de Movimento dos Monstros</i>	6
<i>RandomMon</i>	7
<i>ActDific</i>	7
<i>MudaNivel</i>	7
<i>RotGameOver</i>	8
<i>Rotinas de Tratamento da Interrupção</i>	8
Interrupções 0 a 3	8
Interrupção 15 (Temporizador)	8
INSTRUÇÃO “RANDOM” (MICROPROGRAMAÇÃO).....	9
FUNCIONALIDADES EXTRA	10
PREVENÇÃO DO OVERFLOW DA PONTUAÇÃO.....	10
MENSAGEM NO DISPLAY LCD	10
ANIMAÇÃO “GAME OVER”	11
CONCLUSÃO.....	12

Introdução

Este projecto foi realizado no âmbito da disciplina de Arquitectura de Computadores, da Licenciatura em Engenharia Informática e de Computadores no Instituto Superior Técnico, com o objectivo de desenvolver um jogo, aplicando os nossos conhecimentos adquiridos ao longo das aulas teóricas e de laboratórios.

O jogo desenvolvido é uma versão simplificada do popular jogo “Pac-Man”, desenvolvido em linguagem Assembly do processador P3. Utiliza caracteres ASCII de forma a simular o ambiente de jogo.

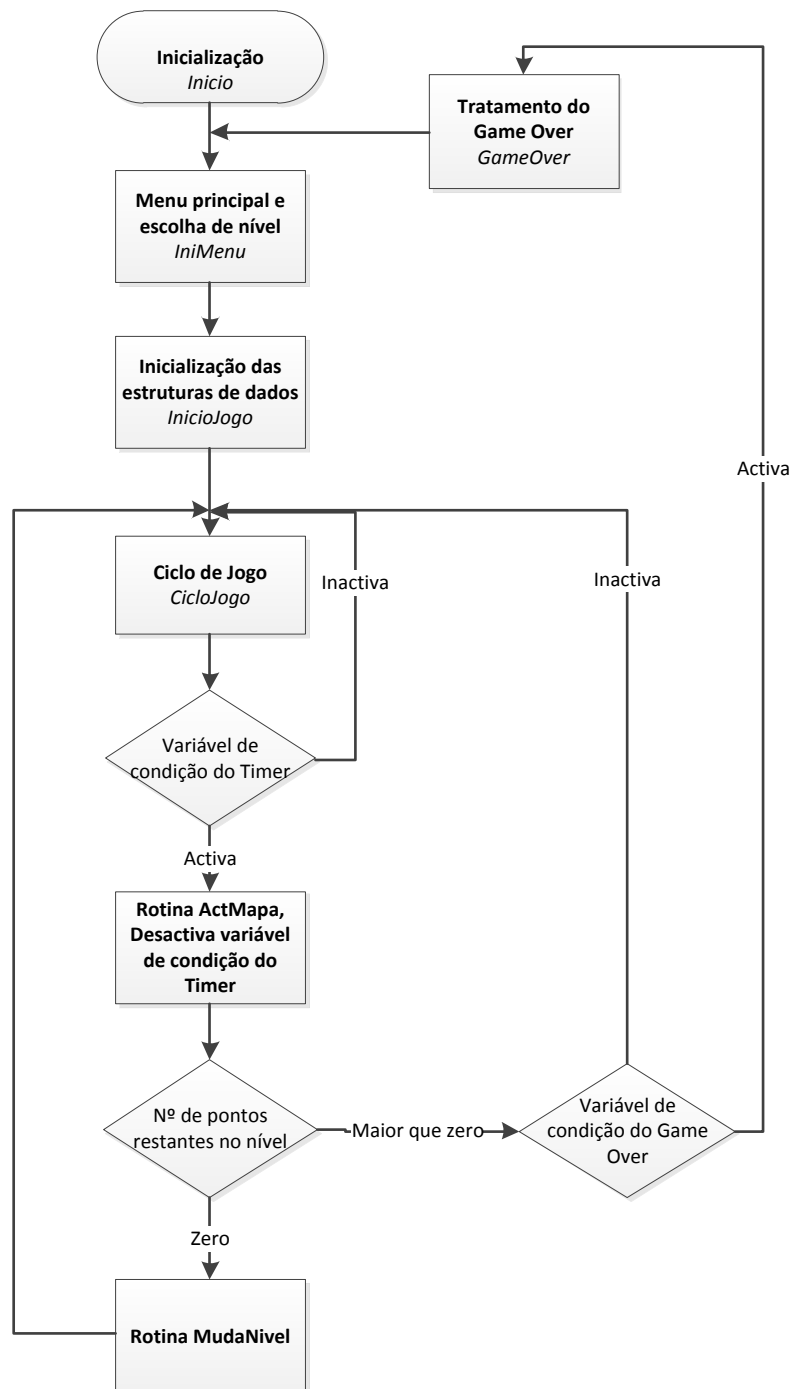
O jogo comporta três mapas distintos, cada um deles representando um nível. Cada nível contém um número diferente de monstros (representados pelo carácter ‘M’) que tentam capturar o Pac-Man (carácter ‘@’). O objectivo do Pac-Man em cada nível é capturar todos os pontos no mapa (carácter ‘.’), evitando tocar nos monstros. A velocidade de jogo (e por efeito, a sua dificuldade), aumenta à medida que o jogador captura um elevado número de pontos. O jogo é considerado ganho quando todos os pontos no mapa são recolhidos, sendo de seguida carregado o nível seguinte. Se um monstro alcançar o Pac-Man, o jogo termina, regressando ao menu principal.

Para este projecto todas as funcionalidades requisitadas foram implementadas, tendo sido adicionadas algumas funcionalidades extra, que serão discriminadas neste relatório.

Organização do Programa

Lógica do programa principal

O programa principal corresponde a um ciclo onde são testadas várias variáveis. Dependendo do estado dessas variáveis, são chamadas rotinas e efectuados saltos para instruções. De seguida apresenta-se um fluxograma minimalista representando o funcionamento do programa principal.



O comportamento das rotinas aqui referidas pode ser visto na secção Principais Rotinas.

Descrição do Jogo

Estruturas de Dados

Designação	Descrição	Nome das variáveis	Valores possíveis
Direcção	Variáveis que contêm um valor correspondente a uma direcção de movimento. Existe uma para o Pac-Man e para cada monstro.	Direccao DirMonstro1 DirMonstro2 DirMonstro3 DirMonstro4 DirMonstro5	0 (Cima) 1 (Baixo) 2 (Esquerda) 3 (Direita) 4 (Indefinida)
Posição em memória	Variáveis que contêm o endereço de memória onde o Pac-Man e os monstros se encontram no espaço reservado para o jogo.	PosPacManM PosMonstro1M PosMonstro2M PosMonstro3M PosMonstro4M PosMonstro5M	Endereço de memória
Posição na janela de texto	Variáveis que contêm as coordenadas da posição do Pac-Man e dos monstros na janela de texto do simulador.	PosPacManT PosMonstro1T PosMonstro2T PosMonstro3T PosMonstro4T PosMonstro5T	Byte mais significativo: 0 a 13 (nº de linhas) Byte menos significativo: 0 a 20 (nº de colunas)
O que está por debaixo de um monstro	Os monstros não capturam pontos nem bónus. Estas variáveis preservam o carácter sobre qual um monstro se encontra.	BaixoMon1 BaixoMon2 BaixoMon3 BaixoMon4 BaixoMon5	‘ ’ (Espaço) ‘.’ (Ponto) ‘)’ (Banana) ‘&’ (Pêra) ‘%’ (Gelado)
Contador do nº de elementos no mapa	Quando o mapa é carregado em memória e na janela de texto, estes contadores são incrementados sempre que um elemento correspondente é encontrado. O contador de pontos é decrementado sempre que o Pac-Man captura um.	NumPontos NumMonstros	Pontos: Entre 0 e o número de pontos no mapa actual Monstros: 1 (Mapa 1) 3 (Mapa 2) 5 (Mapa 3)
Pontuação	Variáveis que guardam a pontuação do jogo actual e a pontuação máxima obtida em todos os jogos.	Pontuacao PontMaxima	0 a 9999
Valor de tempo a contar pelo temporizador	Variável que guarda um inteiro a colocar no endereço do temporizador. Este valor flutua de acordo com a velocidade de jogo.	TimeLong	10 a 3 (vezes 100 milissegundos)
Mapa Actual	Variável que guarda um código correspondente ao mapa em que o jogador se encontra no momento.	MapaActual	0000000000000000b (Nível 1) 0000000000000001b (Nível 2) 0000000000000010b (Nível 3)

Principais Rotinas

EscreveNivel

Esta rotina lê o estado das alavancas da janela de placa do simulador do P3. Este estado é passado como valor a um registo. De seguida, é aplicada uma máscara a esse registo (ALA_Mask), que atribui indiferença a qualquer interruptor que não seja um dos dois da direita. O valor no registo é comparado com um código atribuído a cada nível (COD_Niv1, COD_Niv2 ou COD_Niv3), de forma a escolher qual o mapa a carregar. O endereço onde a primeira célula do mapa escolhido está armazenado (NivelX_L0) é passado como parâmetro pela pilha à rotina EscreveMapa.

EscreveMapa

A rotina EscreveMapa recebe como parâmetro pela pilha o primeiro endereço onde um mapa se encontra armazenado em memória (acessível em M[SP+7]). Utiliza o número de colunas e de linhas (definidos em constante) de forma a escrever propriamente o mapa na janela de texto. Ao mesmo tempo, cria uma cópia do mapa em memória, usando um espaço para isso reservado (EspacoJogo). A rotina efectua, também, uma série de procedimentos (através de sub-rotinas respectivas) essenciais ao bom funcionamento do jogo, nomeadamente:

- Regista o número de pontos inicial numa variável
- Regista o número de monstros no mapa numa variável
- Regista as posições em memória e na janela de texto do Pac-Man e dos monstros (apenas para os que estão no mapa) em variáveis. Um monstro que não esteja no mapa é considerado que se encontra na posição 0, facto que é levado em conta pela lógica do programa.

EscrevePont

Esta é a rotina que escreve a pontuação máxima obtida em todos os jogos no menu principal (à frente da mensagem "Pontuacao Maxima: "). A pontuação máxima é carregada para um registo. Os dígitos da pontuação são obtidos através de divisões sucessivas por 10 (máximo de quatro dígitos), sendo cada dígito guardado num registo diferente. Como o porto de escrita espera receber caracteres ASCII, é necessário converter os inteiros nos registos. Para esse efeito, é somado o valor 30h (ASCII_Int) ao inteiro. De seguida, basta colocar no porto de escrita os dígitos pela ordem oposta de que foram adquiridos.

ActMapa

Esta é a principal rotina do jogo. É chamada pelo programa principal quando a variável de condição do temporizador (CondTimer) está activada (ver Rotinas de Interrupção). É esta a rotina que desencadeia a chamada de outras rotinas que tratam da lógica do jogo. ActMapa verifica a direcção actual do Pac-Man e dos monstros (mas apenas dos monstros que estiverem no mapa). Consoante as direcções obtidas, as rotinas de movimento apropriadas são chamadas.

Rotinas de Movimento do Pac-Man

(PacCima, PacBaixo, PacEsq, PacDir)

Estas rotinas são dependentes da direcção de movimento do Pac-Man e funcionam prevendo o carácter à frente do Pac-Man, reagindo apropriadamente a cada situação. A movimentação do jogador será equivalente para todas as ocasiões, excepto para quando se encontra um monstro ou uma parede, casos que estão previstos no início da rotina.

- Quando encontrado um monstro, é chamada a sub-rotina PacCapMonstro, que activa uma variável de condição para o caso "Game Over" (fim do jogo).
- Quando é encontrada uma parede, a rotina simplesmente retorna, resultando assim que o Pac-Man se mantém na mesma posição.
- Em todas as outras situações, é chamada a sub-rotina PacXGeral (com X = Cima, Baixo, Esq, Dir) que move o Pac-Man uma posição na respectiva direcção (actualizando essa posição em memória e na janela de texto). Caso a posição destino contenha um ponto ou bónus, é também chamada uma sub-rotina que trata esse acontecimento (PacCapX, com X = Ponto, Banana, Pera, Gelado).

Qualquer posição deixada pelo Pac-Man é preenchida por um espaço em branco.

Rotinas de Movimento dos Monstros

(MonCima, MonBaixo, MonEsq, MonDir)

Estas rotinas tratam do movimento de QUALQUER monstro, bastando para isso passar como parâmetro pela pilha a direcção do monstro (DirMonstroX), o carácter sobre o qual este se encontra (BaixoMonX) e a posição do monstro em memória (PosMonstroXM) e na janela de texto (PosMonstroXT) (para todos os casos, X = 1, 2, 3, 4, 5). A lógica das rotinas é semelhante à da movimentação do Pac-Man, com as seguintes diferenças:

- Quando encontrado um monstro ou uma parede, (ou seja, um obstáculo), é chamada a rotina RandomMon (ver abaixo), passando como parâmetro a direcção actual do monstro.
- Quando encontrado o Pac-Man, a rotina activa a variável de condição para o "Game Over" (o monstro é escrito em cima do Pac-Man, simbolizando assim a sua captura).
- Para todos os outros casos, a rotina move o monstro na respectiva direcção (actualizando essa posição em memória e na janela de texto). O conteúdo da posição destino é copiado para a variável BaixoMonX.

Qualquer posição deixada por um monstro é restaurada com o conteúdo da variável BaixoMonX correspondente. Isto permite que o monstro não afecte os pontos e os bónus presente no mapa de jogo.

RandomMon

Esta é a rotina que atribui uma nova direcção de movimento a um monstro (recebendo como parâmetro pela pilha a variável de direcção desse monstro). Funciona usando a instrução Random (IIOP), previamente microprogramada. Esta instrução gera um valor semi-aleatório utilizando um valor semente ou o último valor gerado pela instrução. Este valor é posteriormente dividido por 4, ou seja, o resto da operação será um valor entre 0 e 3 (correspondente a uma das quatro direcções possíveis de movimento). Este valor é comparado com o valor da direcção anterior – se forem iguais, é gerada uma nova direcção, de forma a minimizar o tempo que um monstro fica parado a decidir qual a próxima direcção. A nova direcção é guardada na variável de direcção do monstro desejado.

ActDific

ActDific é chamada sempre que a pontuação do jogador é incrementada. A nova pontuação é então passada pela pilha a esta rotina. A pontuação é testada para um conjunto de múltiplos da constante LIM_DifPts (correspondente ao número 20), com os possíveis efeitos:

Pontuacao	Valor a colocar em M[TimeLong]	LEDs a acender
$< 1 * LIM_DifPts$	DIF_Niv1 (10)	1 LED (LEDs1)
$\geq 1 * LIM_DifPts$ e $< 2 * LIM_DifPts$	DIF_Niv2 (9)	2 LEDs (LEDs2)
$\geq 2 * LIM_DifPts$ e $< 3 * LIM_DifPts$	DIF_Niv3 (8)	3 LEDs (LEDs3)
$\geq 3 * LIM_DifPts$ e $< 4 * LIM_DifPts$	DIF_Niv4 (7)	4 LEDs (LEDs4)
$\geq 4 * LIM_DifPts$ e $< 5 * LIM_DifPts$	DIF_Niv5 (6)	5 LEDs (LEDs5)
$\geq 5 * LIM_DifPts$ e $< 6 * LIM_DifPts$	DIF_Niv6 (5)	6 LEDs (LEDs6)
$\geq 6 * LIM_DifPts$ e $< 7 * LIM_DifPts$	DIF_Niv7 (4)	7 LEDs (LEDs7)
$\geq 7 * LIM_DifPts$	DIF_Niv8 (3)	8 LEDs (LEDs8)

Como a variável TimeLong corresponde ao valor a colocar no endereço do temporizador, o aumento da pontuação do jogador resultará no aumento da velocidade do jogo.

MudaNivel

Esta rotina é activada quando o jogador captura todos os pontos no mapa. Serve para decidir qual o próximo mapa a carregar. Utilizando a informação disponível na variável MapaActual, irá mudar para o Nível 2 caso se tenha vencido o Nível 1, e para o Nível 3 no caso contrário (este nível sucede tanto o Nível 2 como o 3). Passa o endereço do mapa escolhido como parâmetro pela pilha à rotina EscreveMapa.

RotGameOver

Esta importante rotina é chamada pelo programa principal no caso de a variável de condição do “Game Over” estar activada. A sua função é preparar as estruturas relevantes de forma a estarem prontas para um novo jogo. Nomeadamente, a rotina:

- Actualiza a pontuação máxima caso a nova pontuação seja superior.
- Apresenta uma pequena animação em ASCII (ver Funcionalidades Extra)
- Reinicializa as variáveis necessárias (posição e direcção dos monstros e do Pac-Man, velocidade do jogo, pontuação)

Antes de terminar, a rotina coloca a variável de condição do “Game Over” a zero.

Rotinas de Tratamento da Interrupção

Interrupções 0 a 3

Estas interrupções correspondem às teclas de mudança de direcção do Pac-Man. A variável Direccao é carregada com o valor correspondente à interrupção activada.

Interrupção 15 (Temporizador)

Esta interrupção é activada quando o temporizador termina uma contagem. Activa a variável de condição do relógio para ser usada pelo programa principal e reinicia a contagem.

Instrução “Random” (microprogramação)

De seguida apresenta-se a codificação da instrução Random em RTL (Register Transfer Language), e a sua respectiva codificação hexadecimal.

Endereço	Codificação (RTL)	Codificação (Hexadecimal)
112	R8<-02Dh, SBR<-CAR+1, CAR<-F1	E4002DF8
113	R8<-ror R8	00280098
114	shr RD	0020001D
115	!c? CAR<-117h	81411700
116	RD<-RD xor R8	0016309D
117	RD<-ror R8, CAR<-WB	7028309D

Esta instrução é associada à instrução Assembly IIOP (17) pela ROM A.

O valor inicial da semente do Random é A5A5h. Este valor é incrementado constantemente na rotina EsperaTecla enquanto o utilizador não carrega numa tecla, de forma a garantir que raramente a semente tenha o mesmo valor para cada execução de jogo. Esta acção, em conjunto com o algoritmo da instrução Random, simula o funcionamento de um Pseudorandom Number Generator (PRNG), um algoritmo para gerar números semi-aleatórios.

Funcionalidades Extra

As seguintes funcionalidades foram introduzidas por iniciativa do grupo de trabalho, não estando elas previstas no enunciado do projecto.

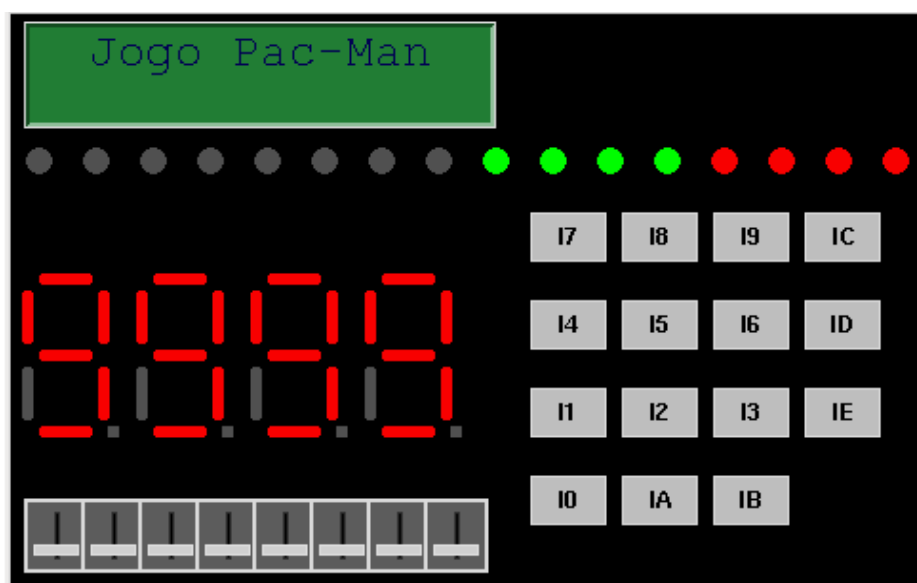
Prevenção do overflow da pontuação

A pontuação actual do jogador é mostrada em tempo real no display de sete segmentos da janela de placa do simulador do P3. No entanto, este display apenas suporta quatro dígitos, pelo que a partir de uma certa pontuação, o valor da pontuação mostrado ao jogador seria incorrecto. O que foi feito foi introduzir um limite de pontuação de forma que o jogador nunca ultrapasse os 9999 pontos (valor máximo representável com quatro dígitos). Esta lógica pode ser observada na rotina AumPontos.

A funcionalidade pode ser facilmente testada alterando o valor da constante PtsPonto para um valor alto (por exemplo, 500).

Mensagem no display LCD

O display LCD na janela de placa era uma funcionalidade do simulador que não havia sido explorada tanto nas aulas de laboratório como no projecto. O grupo, por curiosidade, decidiu estudar o funcionamento da mesma, e implementou uma pequena mensagem no display, escrita quando o jogo é inicializado pela primeira vez. Este procedimento pode ser encontrado na rotina EscreveLCD.



Janela da placa mostrando a pontuação máxima e a mensagem de jogo

Animação “Game Over”

Quando o Pac-Man é capturado por um monstro, o jogo termina e regressa ao menu principal. O grupo considerou esta transição bastante repentina, e decidiu implementar uma pequena animação usando caracteres ASCII, representando um monstro e a mensagem GAME OVER (como ilustra a imagem), pedindo de seguida ao jogador para pressionar uma tecla, voltando então ao menu.

Esta animação é escrita por cima do mapa actual. Usou-se, então, por conveniência, a rotina `EscreveMapa`, recebendo como parâmetro o endereço de memória onde a animação está alojada.



Ecrã personalizado com animação “Game Over”

Conclusão

Finalizado o projecto, consideramo-nos extremamente satisfeitos com o resultado final. Todas as funcionalidades obrigatórias foram implementadas atempadamente com sucesso, dando-nos tempo para a implementação dos extras.

Tivemos o especial cuidado de efectuar um armazenamento organizado das estruturas em memória, ou seja, acima do endereço 8000h e até à última posição utilizada não há uma única localização que tenha ficado por usar.

Até à data da entrega do código, a lógica do programa foi constantemente optimizada, tendo sido eliminadas cerca de 400 linhas de código desnecessárias desde o primeiro ficheiro estável até à versão final.

Finalizando, o grupo gostaria de agradecer a todos os docentes da cadeira de Arquitectura de Computadores por todas as sugestões, dicas, e esclarecimentos que nos foram dados ao longo da realização do projecto.