Mai 18, 12 15:28	3	pacman33	3.as	Page 1/31
;========= ; ; ; ; ;	Grupo 3 André S Mariana	stura de Computadores - Silva - 68707 A Azevedo - 72595		
; Definição de	constant	ces		
Topo_Pilha Int_Mask	EQU EQU	FDFFh FFFAh		
TimerValue	EQU	FFF6h	;Recebe o valor a "c ;em intervalos de 10	
TimerControl	EQU	FFF7h	;1 = conta, 0 = pára	
INI_Int_Mask	EQU	100000000001111b	;Permite ints. 0,1,2	,3,15
N_LIN N_COL	EQU EQU	14 21	;Altura do mapa de j ;Largura do mapa de	
N_LIN_m1 N_COL_m1	EQU EQU	13 20	;N_LIN - 1 ;N_COL - 1	
ENABLE	EQU	1	;Constante de uso ge	ral
COD_Niv1 COD_Niv2 COD_Niv3	EQU EQU EQU	00000000000000000b 00000000000000001b 00000000	;Código associado ao ;Código associado ao ;Código associado ao	Nível 2
ALA_Estado ALA_Mask	EQU EQU	FFF9h 0000000000000011b	;Estado actual das a ;Máscara do valor re ;do estado das alava	levante
PORTO_Estado PORTO_Ctrl PORTO_Esc INI_Index	EQU EQU EQU	FFFDh FFFCh FFFEh FFFFh	;Porto de estado ;Porto de controlo ;Porto de escrita ;M[INI_Index] funcio ;porto de leitura	na como
LEDs Display	EQU EQU	FFF8h FFF0h	;Endereço dos LEDs d ;Endereço do display ;7 segmentos	
LCD_Ctrl LCD_Esc LCD_Ini	EQU EQU EQU	FFF4h FFF5h 1000000000000010b	;Porto de controlo (;Porto de escrita (D ;Estado inicial do p ;de controlo (LCD)	isplay LCD)
INC_Col	EQU	0001h	;Somando à posição a	
INC_Lin	EQU	0100h	;para a coluna segui ;Somando à posição a ;para a linha seguin	ctual passa
CIMA BAIXO ESQ DIR INDEF	EQU EQU EQU EQU	0 1 2 3 4	;Códigos corresponde ;direcção de movimen	
DIF_Niv1 DIF_Niv2	EQU EQU	10 9	;Níveis de dificulda ;Correspondem aos va	

Mai 18, 12 15:28		pacman	33.as	Page 2/31
DIF_Niv3 DIF_Niv4 DIF_Niv5 DIF_Niv6 DIF_Niv7 DIF_Niv8	EQU EQU EQU EQU EQU	8 7 6 5 4 3	;colocar na variá	vel TimeLong
LIM_DifPts	EQU	20	;Dificuldade aume ;'LIM_DifPts' seg	
LIM_Pontuacao	EQU	9999	;Define como pont ;9999 (para evita ;pontuação na jan	r overflow da
LEDs1	EQU	0000000000000001b	;Valores cujo núm	ero na etiquet
LEDs2	EQU	000000000000011b	;indica o número	de LEDs ligado:
LEDs3 LEDs4 LEDs5 LEDs6 LEDs7 LEDs8	EQU EQU EQU EQU EQU	0000000000000111b 0000000000001111b 00000000	;na placa. ;Correspondem aos ;colocar no ender. ;pela constante L	eço definido
PtsPonto PtsBanana PtsPera PtsGelado	EQU EQU EQU EQU	2 10 20 30	;Número de pontos ;bónus atribui	que cada
ASCII_Int	EQU	0030h	;Somado a um inte ;o código ASCII d	
Fim_Menu_L3 Fim_Menu_L5 Fim_Menu_L7 Fim_Menu_L8	EQU EQU EQU	800Bh 801Dh 804Bh 8068h	;Posição em memór ;strings, para se ;parar de escreve ;Utilizar constan	saber quando r. tes em vez de
Fim_MJogo_L7 Fim_MJogo_L10 Fim_MJogo_L11	EQU EQU EQU	806Eh 8083h 8092h	;guardar em memór ;indicadores de f ;evita a ocupação ;de posições de m	inal de texto desnecessária
; Tabela de Int	errupco	es		
INTO INT1 INT2 INT3	ORIG WORD WORD WORD WORD	FE00h IntCima IntBaixo IntEsq IntDir		
INT 15	ORIG WORD	FE0Fh IntTimer		
; Início do arm	nazename	nto em memória		
	ORIG	8000h		

```
pacman33.as
 Mai 18, 12 15:28
                                                               Page 3/31
Menu L3
              STR
                      'Jogo Pac-Man'
                                           ;Usado também nas mens. de jogo
Menu_L5
              STR
                      'Pontuacao maxima: '
Menu L7
              STR
                      'Use as alavancas para escolher o nivel inicial'
Menu L8
              STR
                     'e em seguida prima uma tecla.'
; Mensagens dentro do jogo
MJogo L7
              STR
                      'Nivel'
MJogo L10
              STR
                     'Premir uma tecla para'
MJogo L11
                     'iniciar o jogo.'
              STR
                     111
MJogo N1
              STR
                     121
MJogo N2
              STR
                     131
MJogo N3
              STR
: Armazenamento dos mapas em memória
; Mapa correspondente ao nível 1
                      012345678901234567890
Nivel1 L0
              STR
                      Nivell L1
                      '#&....%#'
              STR
                      '#.#####. #.# .#####.#'
Nivel1 L2
              STR
Nivell L3
              STR
                      '#.# #. #.# .# #.#'
Nivell L4
              STR
                      '#.# # #. #.# .# # #.#'
Nivell L5
              STR
                      ′#....#
Nivell L6
              STR
                     '##.#### # ) # #### ##'
Nivell L7
              STR
                      '##.#### ##### #### ### ##'
              STR
Nivell L8
Nivell L9
              STR
                     '# # # # # # # # # #.#'
              STR
                     '##
                           # # # # #.#'
Nivel1_L10
Nivell L11
              STR
                     '# ##### # # #####.#'
Nivell L12
              STR
                               @....#
Nivell L13
              STR
                     ; Mapa correspondente ao nível 2
                      012345678901234567890
Nivel2 L0
              STR
                      Nivel2 L1
              STR
                     '#....#'
Nivel2 L2
              STR
                     '#.#####..###..######.#'
Nivel2 L3
              STR
                     '#.#...#..#.#..#..#..#."
Nivel2 L4
              STR
                     '#.#.#.#.#.#.#.#.#.#.#.#'
Nivel2 L5
              STR
                     '#.....M#'
Nivel2 L6
              STR
                     '##.####.####.###.##"
Nivel2 L7
              STR
                     '##.#.##.#.).#.##.#.#'
Nivel2 L8
              STR
                     ' #M....# '
Nivel2 L9
              STR
                     '#.#######.###########.#'
Nivel2 L10
              STR
                     Nivel2 L11
                     '#.##.##..#.#..##.##.#
              STR
Nivel2 L12
              STR
                     '#....#'
Nivel2 L13
              STR
                     ; Mapa correspondente ao nível 3
                      012345678901234567890
                     Nivel3 L0
              STR
Nivel3 L1
              STR
                      Nivel3_L2
                     '#.#####..#.#..#####.#'
              STR
Nivel3 L3
              STR
                      '#.#....#.#.....#.#'
Nivel3 L4
              STR
                      '#.#####..###..#####.#'
Nivel3 L5
              STR
                      '#M....#'
                      '##.#..####@####..#.##'
Nivel3 L6
              STR
```

'##.####..###..###.##'

Nivel3 L7

STR

```
pacman33.as
 Mai 18. 12 15:28
                                                                   Page 4/31
Nivel3 L8
               STR
                       Nivel3 L9
               STR
                       Nivel3 L10
                       '#&#...#.#.).#.#...#%#'
               STR
Nivel3 L11
                       '#.##.##.##.##.##.##.#
               STR
Nivel3 L12
               STR
                       '#....#
Nivel3 L13
               STR
                       : Mapa correspondente à animação "Game Over" (funcionalidade extra)
                        012345678901234567890
               STR
GameOver L0
                       4####
                                       #####
                                       ##### '
GameOver L1
               STR
                       1#####
                                 00
GameOver L2
               STR
                       /#####
                                       #####
GameOver L3
               STR
                       4####
                                       #####
GameOver L4
               STR
                       /#####
                                       #####
GameOver L5
               STR
                       '##### GAME OVER #####
GameOver L6
               STR
                       /#####
                                       #####
GameOver L7
               STR
                       GameOver L8
               STR
GameOver L9
               STR
                       '#Pressione uma tecla#
GameOver L10
               STR
                       '#para voltar ao menu#
GameOver L11
               STR
GameOver L12
               STR
                       GameOver_L13
               STR
                       : Definição de variáveis
RandomVar
               WORD
                       A5A5h
                                              ;Conterá número semi-aleatório
               WORD
                       COD Niv1
                                              ;Conterá COD_NivX
MapaActual
NumPontos
               WORD
                       0000h
                                              ;Núm. de pontos no mapa actual
NumMonstros
               WORD
                       0000h
                                              ;Núm. de monstros no mapa actual
PosPacManT
                       0000h
               WORD
                                              :Pos. actual Pac-Man
                                              ;na janela de texto
                       0000h
                                              ;Pos. actual Pac-Man em memória
PosPacManM
               WORD
PosMonstro1T
               WORD
                       0000h
                                              :Pos. actual Monstro 1
                                              na ianela de texto:
PosMonstro1M
               WORD
                       0000h
                                              ;Pos. actual Monstro 1
                                              ;em memória
PosMonstro2T
               WORD
                       0000h
                                              :Pos. actual Monstro 2
                                              ;na janela de texto
PosMonstro2M
               WORD
                       0000h
                                              :Pos. actual Monstro 2
                                              em memória:
PosMonstro3T
               WORD
                       0000h
                                              :Pos. actual Monstro 3
                                              :na janela de texto
PosMonstro3M
               WORD
                       0000h
                                              :Pos. actual Monstro 3
                                              em memória:
                                              :Pos. actual Monstro 4
PosMonstro4T
               WORD
                       0000h
                                              na ianela de texto
                       0000h
PosMonstro4M
               WORD
                                              ;Pos. actual Monstro 4
                                              em memória:
                                              ;Pos. actual Monstro 5
PosMonstro5T
               WORD
                       0000h
                                              ;na janela de texto
                       0000h
                                              ;Pos. actual Monstro 5
PosMonstro5M
               WORD
                                              em memória:
                       0000h
               WORD
                                              :Pontuacao actual do jogador
Pontuacao
PontMaxima
               WORD
                       0000h
                                              ;High score
Direccao
                       INDEF
                                              ;Direcção de movim. do Pac-Man
               WORD
DirMonstro1
               WORD
                       INDEF
                                              ;Direcção do movim. do Monstro 1
DirMonstro2
               WORD
                       INDEF
                                              ;Direcção do movim. do Monstro 2
DirMonstro3
               WORD
                       INDEF
                                              :Direcção do movim. do Monstro 3
DirMonstro4
                                              ;Direcção do movim. do Monstro 4
               WORD
                       INDEF
DirMonstro5
               WORD
                       INDEF
                                              ;Direcção do movim. do Monstro 5
```

Mai 18, 12 15:28	3	pacman3	3.as	Page 5/31
BaixoMon1	WORD	1 1	;0 que está por baixo	do Mons. 1
BaixoMon2	WORD	<i>i i</i>	;0 que está por baixo	
BaixoMon3	WORD	<i>i i</i>	;0 que está por baixo	
BaixoMon4		, ,		
	WORD		;0 que está por baixo	
BaixoMon5	WORD	, ,	;0 que está por baixo	do Mons. 5
CondTimer	WORD	0	;1 após INT15.	
CondGameOver	WORD	0	;1 após o Pac-Man ser	capturado
TimeLong	WORD	DIF_Niv1	;Valor a colocar no	-
			;endereço TimerValue	
; Reserva de es	spaço em	n memória para o jogo ac	tual	
EspacoJogo	TAB	294	;Cada mapa contém 294	células
; Início do pro	ograma			
	ORIG	0000h		
	JMP	Inicio		
;========				
; EsperaTecla:	Rotina	que espera que o utiliz	ador carregue numa	
;	tecla a	ntes de prosseguir.		
;		nta também M[RandomVar]	. para um valor	
<i>`.</i>		ais aleatório	, , , , , , , , , , , , , , , , , , , ,	
, ;========		=======================================		
EsperaTecla:	PUSH	R1		
EsperaTecla2:	INC	M[RandomVar]		
Esperarectaz.				7 . 2
		RU MIPORTO ESTADOI		
	CMP		;Foi pressionada uma t	
	BR.Z	EsperaTecla2	;Se não, verifica outr	a vez
	-	EsperaTecla2		a vez
	BR.Z	EsperaTecla2	;Se não, verifica outr	a vez
	BR.Z MOV	EsperaTecla2 R1, M[INI_Index]	;Se não, verifica outr	a vez
;======	BR.Z MOV POP RET	EsperaTecla2 R1, M[INI_Index]	;Se não, verifica outr ;Recoloca porto de est	a vez
,	BR.Z MOV POP RET	EsperaTecla2 R1, M[INI_Index] R1	;Se não, verifica outr ;Recoloca porto de est =======	a vez
; EscolheNivel: ;	BR.Z MOV POP RET ===================================	EsperaTecla2 R1, M[INI_Index] R1	;Se não, verifica outr ;Recoloca porto de est ====================================	a vez
; EscolheNivel: ;	BR.Z MOV POP RET ===================================	EsperaTecla2 R1, M[INI_Index] R1 que lê o estado das alder um nível	;Se não, verifica outr ;Recoloca porto de est ====================================	a vez
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escoli	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das ala er um nível R1	;Se não, verifica outr ;Recoloca porto de est ====================================	a vez
; EscolheNivel: ; ;	BR.Z MOV POP RET : Rotina escolh :	EsperaTecla2 R1, M[INI_Index] R1 que lê o estado das alder um nível R1 R1, M[ALA_Estado]	;Se não, verifica outr ;Recoloca porto de est ======== avancas para ========	ra vez rado a O
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escoli	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das ala er um nível R1	;Se não, verifica outr ;Recoloca porto de est ====================================	ra vez cado a 0 cenas aos
; EscolheNivel: ; ;	BR.Z MOV POP RET : Rotina escoli escoli PUSH MOV AND	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcher um nível R1 R1, M[ALA_Estado] R1, ALA_Mask	;Se não, verifica outr ;Recoloca porto de est ====================================	ra vez rado a 0 penas aos n direita
; EscolheNivel: ; ;	BR.Z MOV POP RET : Rotina escolh :======= PUSH MOV AND	EsperaTecla2 R1, M[INI_Index] R1 eque lê o estado das alcer um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1	;Se não, verifica outr ;Recoloca porto de est ====================================	ra vez rado a 0 penas aos n direita
; EscolheNivel: ; ;	BR.Z MOV POP RET : Rotina escoli escoli PUSH MOV AND	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das al. er um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1	;Se não, verifica outr ;Recoloca porto de est ====================================	ra vez rado a 0 penas aos ra direita ralavancas
; EscolheNivel: ; ;	BR.Z MOV POP RET : Rotina escolh :======= PUSH MOV AND	EsperaTecla2 R1, M[INI_Index] R1 eque lê o estado das alcer um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1	;Se não, verifica outr ;Recoloca porto de est ====================================	ra vez rado a 0 penas aos ra direita ralavancas
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escolk PUSH MOV AND MOV CMP BR.Z	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcer um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1	;Se não, verifica outr ;Recoloca porto de est ====================================	enas aos direita alavancas
; EscolheNivel: ; ;	BR.Z MOV POP RET : Rotina escoli ENDEMINION OF AND MOV AND MOV CMP BR.Z CMP	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcher um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2	;Se não, verifica outr ;Recoloca porto de est ====================================	enas aos direita alavancas
; EscolheNivel: ; ;	BR.Z MOV POP RET : Rotina escolk ESCOL PUSH MOV AND MOV CMP BR.Z CMP BR.Z	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das ala her um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2	;Se não, verifica outr ;Recoloca porto de est ====================================	enas aos direita alavancas
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escolk PUSH MOV AND MOV CMP BR.Z CMP BR.Z CMP	EsperaTecla2 R1, M[INI_Index] R1 Rque lê o estado das alcher um nível R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3	;Se não, verifica outr ;Recoloca porto de est ====================================	enas aos direita alavancas
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escolk BR.Z MOV AND MOV CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcer um nível R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3	;Se não, verifica outr ;Recoloca porto de est estante estante estante ;Atribui relevância ap ;dois interruptores da ;Regista o mapa actual ;Compara o estado das ;com o código dos níve ;de forma a selecciona ;o mapa apropriado	enas aos direita alavancas is,
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escolk escolk MOV AND MOV CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z MOV	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcher um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3 R1, M[INI_Index]	;Se não, verifica outr ;Recoloca porto de est ====================================	enas aos direita alavancas is,
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escolk MOV AND MOV CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcer um nível R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3	;Se não, verifica outr ;Recoloca porto de est estante estante estante ;Atribui relevância ap ;dois interruptores da ;Regista o mapa actual ;Compara o estado das ;com o código dos níve ;de forma a selecciona ;o mapa apropriado	enas aos direita alavancas is,
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escolk escolk MOV AND MOV CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z MOV	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcher um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3 R1, M[INI_Index]	;Se não, verifica outr ;Recoloca porto de est estante estante estante ;Atribui relevância ap ;dois interruptores da ;Regista o mapa actual ;Compara o estado das ;com o código dos níve ;de forma a selecciona ;o mapa apropriado	enas aos direita alavancas is,
; EscolheNivel;; ;;======== EscolheNivel:	BR.Z MOV POP RET Rotina escolk BR.Z CMP	EsperaTecla2 R1, M[INI_Index] R1 R1 Rque lê o estado das alcher um nível R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3 R1, M[INI_Index] EsperaTecla EscolheNivel	;Se não, verifica outr ;Recoloca porto de est estante estante estante ;Atribui relevância ap ;dois interruptores da ;Regista o mapa actual ;Compara o estado das ;com o código dos níve ;de forma a selecciona ;o mapa apropriado ;Recoloca porto de est	enas aos direita alavancas is, ar
; EscolheNivel;; ;;======== EscolheNivel:	BR.Z MOV POP RET Rotina escolk PUSH MOV AND MOV CMP BR.Z CMP	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das ala er um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3 R1, M[INI_Index] EsperaTecla EscolheNivel R1	;Se não, verifica outr ;Recoloca porto de est ====================================	penas aos direita alavancas sis, ar ado a 0
; EscolheNivel;; ;;======== EscolheNivel:	BR.Z MOV POP RET Rotina escolk escolk MOV AND MOV CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z MOV CALL BR POP PUSH	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcer um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3 R1, M[INI_Index] EsperaTecla EscolheNivel R1 Nivel1_L0	;Se não, verifica outr ;Recoloca porto de est ====================================	penas aos direita alavancas sis, ar ado a 0
; EscolheNivel;; ;;======== EscolheNivel:	BR.Z MOV POP RET Rotina escolk escolk MOV AND MOV CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z MOV CALL BR POP PUSH CALL	EsperaTecla2 R1, M[INI_Index] R1 Rque lê o estado das alcher um nível R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3 R1, M[INI_Index] EsperaTecla EscolheNivel R1 Nivel1_L0 EscreveMapa	;Se não, verifica outr ;Recoloca porto de est ===================================	penas aos direita alavancas sis, ar ado a 0
; EscolheNivel: ; ;	BR.Z MOV POP RET Rotina escolk escolk MOV AND MOV CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z MOV CALL BR POP PUSH	EsperaTecla2 R1, M[INI_Index] R1 a que lê o estado das alcer um nível R1 R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3 R1, M[INI_Index] EsperaTecla EscolheNivel R1 Nivel1_L0	;Se não, verifica outr ;Recoloca porto de est este de la compara estado das ;compara o estado das ;compara o estado das ;com o código dos níve ;de forma a selecciona ;o mapa apropriado ;Recoloca porto de est ;Dependendo do nível e ;passa como parâmetro ;a rotina EscreveMapa ;o primeiro endereço	cenas aos direita alavancas is, ur ado a 0
; EscolheNivel;; ;;======== EscolheNivel:	BR.Z MOV POP RET Rotina escolk escolk MOV AND MOV CMP BR.Z CMP BR.Z CMP BR.Z CMP BR.Z MOV CALL BR POP PUSH CALL	EsperaTecla2 R1, M[INI_Index] R1 Rque lê o estado das alcher um nível R1, M[ALA_Estado] R1, ALA_Mask M[MapaActual], R1 R1, COD_Niv1 SelecNiv1 R1, COD_Niv2 SelecNiv2 R1, COD_Niv3 SelecNiv3 R1, M[INI_Index] EsperaTecla EscolheNivel R1 Nivel1_L0 EscreveMapa	;Se não, verifica outr ;Recoloca porto de est ===================================	cenas aos direita alavancas is, ur ado a 0

Mai 18, 12 15:28	3	pacma	an33.as	Page 6/31
SelecNiv3: FimEscNivel:	CALL BR POP PUSH CALL RET	EscreveMapa FimEscNivel R1 Nivel3_L0 EscreveMapa	;memória	
; EscreveMapa: ; ; ; Parâmetro pa	Rotina q copia-o quando e ssado a e M[SP+7]	ue escreve um mapa para memória, e cha ncontra determinado sta rotina pela pil <- NivelX_L0 (X=1,2	ma rotinas relevantes s caracteres. ha:	
EscreveMapa: EscreveMais:	DSI PUSH PUSH PUSH PUSH PUSH MOV	R1 R2 R3 R4 R5 R1, INI_Index M[PORTO_Ctrl], R1 R4, R0 M[PORTO_Ctrl], R4 R5, EspacoJogo M[NumPontos], R0 M[NumMonstros], R0 R3, M[SP+7] R1, M[R3] M[PORTO_Esc], R1 M[R5], R1 R1, '@' RegPosPacMan R1, '.' IncNumPontos R1, 'M' IncNumMonstros R3 R5	;Limpa a janela de ;Coloca cursor na ;Num. pontos inic. ;Num. monstros in: ;Pos. da primeira ;Escreve o caracte. ;Coloca o caracte. ;Se for escrito ou: ;Se for escrito u: ;Aponta para próx. ;Aponta para próx.	posição (0,0) ial = 0 icial = 0 célula do mapa er na janela r no EspacoJogo Pac-Man m ponto m monstro
IncLinha:	MOV MVBL CMP BR.Z MOV ADD MOV JMP MOV MVBH SHR CMP BR.Z MOV ADD MOV ADD MOV MVBL MOV JMP	R2, R0 R2, R4 R2, N_COL_ml IncLinha R2, INC_Col R4, R2 M[PORTO_Ctrl], R4 EscreveMais R2, R0 R2, R4 R2, 8 R2, N_LIN_ml FimEscrita R2, INC_Lin R4, R2 M[PORTO_Ctrl], R4 R4, R0 M[PORTO_Ctrl], R4 EscreveMais	;no EspacoJogo ;Se já estiver na ;Se escrita a últ. ;Incrementa linha ;Volta à coluna 0	

Mai 18, 12 15:28		pacman33.	as	Page 7/31
FimEscrita:	POP POP POP POP CALL MOV CALL CALL ENI RETN	R5 R4 R3 R2 R1 EscMensJogo R1, M[INI_Index] EsperaTecla ApagMensJogo		stado a 0
; RegPosPacMan: ; ; Parâmetros pa: ;	Rotina do Pac- ssados a R5 <- P R4 <- P	que guarda em variáveis a Man na janela de texto e esta rotina por registo: osição do Pac-Man em memo osição do Pac-Man na jana	a posição inicial em memória : ória ela de texto	
RegPosPacMan:	MOV MOV RET	M[PosPacManM], R5 M[PosPacManT], R4		
; IncNumPontos: ;	Rotina pontos	que incrementa o contador '.' inicial num mapa	r de número de	
IncNumPontos:	INC RET	M[NumPontos]		
; DecNumPontos: ;	Rotina durante	que decrementa o número d o jogo, após um ponto te	de pontos no mapa er sido capturado	
DecNumPontos:	DEC RET	M[NumPontos]		
; IncNumMonstros: Rotina que incrementa o contador do número de ; monstros 'M' inicial num mapa, regista a ; posição deles na janela de texto e em memória ; e atribui-lhes uma direcção de movimento ; inicial. Também declara que os monstros se ; encontram em cima de uma casa vazia ; Parâmetros passados a esta rotina por registo: ; R5 <- Posição de um monstro em memória ; R4 <- Posição de um monstro na janela de texto				
IncNumMonstros:	PUSH PUSH INC MOV CMP BR.Z CMP BR.Z CMP	R1, M[NumMonstros] R1, 1 RegPosMon1	;+1 monstro no mapa ;Verifica, dependend ;número de monstros ;qual vai ser o mons ;tratar neste chamam ;rotina	já tratados, tro a

Mai 18, 12 15:28		pacn	nan33.	as Page 8/3	31
	JMP.Z CMP JMP.Z JMP	RegPosMon3 R1, 4 RegPosMon4 RegPosMon5		· ·	
RegPosMon1:	MOV MOV MOV MOV PUSH CALL JMP	M[PosMonstrolM], M[PosMonstrolT], R2, ' ' M[BaixoMon1], R2 DirMonstrol RandomMon FimIncMons			
RegPosMon2:	MOV MOV MOV MOV PUSH CALL JMP	M[PosMonstro2M], M[PosMonstro2T], R2, ' ' M[BaixoMon2], R2 DirMonstro2 RandomMon FimIncMons			
RegPosMon3:	MOV MOV MOV MOV PUSH CALL BR	M[PosMonstro3M], M[PosMonstro3T], R2, ' ' M[BaixoMon3], R2 DirMonstro3 RandomMon FimIncMons			
RegPosMon4:	MOV MOV MOV PUSH CALL BR	M[PosMonstro4M], M[PosMonstro4T], R2, ' ' M[BaixoMon4], R2 DirMonstro4 RandomMon FimIncMons			
RegPosMon5:	MOV MOV MOV MOV PUSH CALL	M[PosMonstro5M], M[PosMonstro5T], R2, ',' M[BaixoMon5], R2 DirMonstro5 RandomMon			
FimIncMons:	POP POP RET	R2 R1			
; EscMensJogo: ;	Rotina q de jogo	que escreve as mens	agens :	iniciais no mapa	
EscMensJogo:	PUSH PUSH PUSH MOV CMP	R1 R3 R4 R1, M[CondGameOve		;Se estiver a escrever a	
EscMJogoL1:	JMP.Z MOV MOV	FimEscMenJog R4, 011Dh M[PORTO_Ctrl], R4	ŀ	;animação "Game Over" ;não escreve a mensagem de jo ;Começamos a escrever em (1,2	

Mai 18, 12 15:28		Page 9/31		
	MOV	R3, Menu_L3		
CicloMJogoL1:	MOV	R1, M[R3]	Tarana a samat	
	MOV CMP	M[PORTO_Esc], R1 R3, Fim Menu L3	;Escreve o caract	er na janeia
	BR.Z	EscMJogoL7		
	INC	R3		
	INC	R4		
	MOV	M[PORTO_Ctrl], R4	;Põe o cursor na	coluna seguinte
	BR	CicloMJogoL1	~	(5.04)
EscMJogoL7:	MOV	R4, 071Fh M[PORTO_Ctrl], R4	;Começamos a escr	ever em (7,31)
	MOV	R3, MJogo L7		
CicloMJogoL7:	MOV	R1, M[R3]		
-	MOV	M[PORTO_Esc], R1	;Escreve o caract	er na janela
	CMP	R3, Fim_MJogo_L7		
	BR.Z	EscNumNivel	;Para escrever o	num. do nível
	INC	R3 R4		
	MOV	M[PORTO_Ctrl], R4	;Põe ocursor na c	oluna seguinte
	BR	CicloMJogoL7	·	3
EscNumNivel:	INC	R4	;Põe o cursor a s	
	MOV	M[PORTO_Ctrl], R4	;Põe o cursor na	coluna seguinte
	MOV CMP	R1, COD_Niv1 M[MapaActual], R1	;Verifica qual o	mana em oue
	BR.Z	EscNumNivel1	;se vai jogar, de	
	MOV	R1, COD_Niv2	;escrever o númer	
	CMP	M[MapaActual], R1	;na mensagem de j	ogo
	BR.Z	EscNumNivel2		
	MOV CMP	R1, COD_Niv3 M[MapaActual], R1		
	BR.Z	EscNumNivel3		
EscNumNivel1:	MOV	R1, M[MJogo_N1]		
	MOV	M[PORTO_Esc], R1	;Escreve 1 a segu	ir a "Nivel "
1 30	BR	EscMJogoL10		
EscNumNivel2:	MOV	R1, M[MJogo_N2]		i "NTi1 "
	MOV BR	M[PORTO_Esc], R1 EscMJogoL10	;Escreve 2 a segu	ir a "Nivei "
EscNumNivel3:	MOV	R1, M[MJogo_N3]		
	MOV	M[PORTO_Esc], R1	;Escreve 3 a segu	ir a "Nivel "
EscMJogoL10:	MOV	R4, 0A1Ah	;Começamos a escr	ever em (10,26)
	MOV	M[PORTO_Ctrl], R4		
CicloMJogoL10:	MOV MOV	R3, MJogo_L10 R1, M[R3]		
CICIONO OGOLIO:	MOV	M[PORTO_Esc], R1	;Escreve o caract	er na janela
	CMP	R3, Fim_MJogo_L10	,	J
	BR.Z	EscMJogoL11		
	INC	R3		
	INC MOV	R4 M[PORTO_Ctrl], R4	;Põe o cursor na	coluna seguinte
	BR	CicloMJogoL10	, roc o cursor na	coruna begunnte
EscMJogoL11:	MOV	R4, 0B1Ah	;Começamos a escr	ever em (11,26)
-	MOV	M[PORTO_Ctrl], R4	-	
		R3, MJogo_L11		
Oi al ante estate	MOV			
CicloMJogoL11:	MOV	R1, M[R3]	·Fagreyo o garast	er na janola
CicloMJogoL11:	MOV MOV	R1, M[R3] M[PORTO_Esc], R1	;Escreve o caract	er na janela
CicloMJogoL11:	MOV	R1, M[R3]	;Escreve o caract	er na janela
CicloMJogoLl1:	MOV MOV CMP BR.Z INC	R1, M[R3] M[PORTO_Esc], R1 R3, Fim_MJogo_L11 FimEscMenJog R3	;Escreve o caract	er na janela
CicloMJogoL11:	MOV MOV CMP BR.Z INC INC	R1, M[R3] M[PORTO_Esc], R1 R3, Fim_MJogo_L11 FimEscMenJog R3 R4		
CicloMJogoL11:	MOV MOV CMP BR.Z INC	R1, M[R3] M[PORTO_Esc], R1 R3, Fim_MJogo_L11 FimEscMenJog R3	;Escreve o caract ;Põe cursor na co	

Mai 18, 12 15:28	3	pacman3	3.as	Page 10/31
	POP POP RET	R3 R1		
•		======================================		
;	"Premi	ir uma tecla para inicia 		
ApagMensJogo:	PUSH	R1		
	PUSH PUSH	R3 R4		
	MOV	R1, ''	;R1 conterá caract	
ApaMJogoL10:	MOV MOV	R4, 0A1Ah M[PORTO_Ctrl], R4	;Começamos a apaga	r em (10,26)
	MOV	R3, MJogo_L10		
CApaMJogoL10:	MOV	M[PORTO_Esc], R1	;Escreve um espaço	
	CMP BR.Z	R3, Fim_MJogo_L10 ApaMJogoL11	;Já apagámos a lin	ha toda?
	INC	R3		
	INC	R4	-~	
	MOV BR	M[PORTO_Ctrl], R4 CApaMJogoL10	;Põe o cursor na c	oluna seguinte
ApaMJogoL11:	MOV	R4, 0B1Ah	;Começamos a apaga	r em (11,26)
	MOV	M[PORTO_Ctrl], R4		
CApaMJogoL11:	MOV	R3, MJogo_L11 M[PORTO_Esc], R1	;Escreve um espaço	na ianela
01150110 030777	CMP	R3, Fim_MJogo_L11	;Já apagámos a lin	
	BR.Z	FimApaMenJog		
	INC	R3 R4		
	MOV	M[PORTO_Ctrl], R4	;Põe o cursor na c	oluna seguinte
FimApaMenJog:	BR POP	CApaMJogoL11 R4		
r mapanenoog.	POP	R3		
	POP	R1		
	RET			
;========				
· FagreyeMenu.	Potina	que escreve o menu prim	aginal do jogo	
;		ela de texto	icipai do jogo	
;=======				
EscreveMenu:	PUSH	R1		
	PUSH	R3		
	PUSH	R4		
	MOV	R1, INI_Index M[PORTO_Ctrl], R1	;Limpa a janela de	texto
EscMenuL3:	MOV	R4, 0300h	;Começamos a escre	
	MOV	M[PORTO_Ctrl], R4		
CicloMenuL3:	MOV MOV	R3, Menu_L3 R1, M[R3]		
	MOV	M[PORTO_Esc], R1	;Escreve o caracte	
	CMP BR.Z	R3, Fim_Menu_L3 EscMenuL5	;Já escrevemos a l	inha toda?
	INC	R3		
	INC	R4	-~	
	MOV BR	M[PORTO_Ctrl], R4 CicloMenuL3	;Põe o cursor na c	oluna seguinte
EscMenuL5:	MOV	R4, 0500h	;Começamos a escre	ver na linha 5
	MOV	M[PORTO_Ctrl], R4		

Mai 18, 12 15:28	3	pacman3	3.as	Page 11/31
CicloMenuL5:	MOV MOV MOV CMP BR.Z INC	R3, Menu_L5 R1, M[R3] M[PORTO_Esc], R1 R3, Fim_Menu_L5 EscMenuL7 R3	;Escreve o caracto ;Já escrevemos a	
EscMenuL7:	INC MOV BR CALL MOV	R4 M[PORTO_Ctrl], R4 CicloMenuL5 EscrevePont R4, 0700h	;Põe o cursor na o ;Escreve o high s ;Começamos a escr	core na janela
CicloMenuL7:	MOV MOV MOV CMP	M[PORTO_Ctrl], R4 R3, Menu_L7 R1, M[R3] M[PORTO_Esc], R1 R3, Fim_Menu_L7	;Escreve o caracto ;Já escrevemos a	
EscMenuL8:	BR.Z INC INC MOV BR MOV	EscMenuL8 R3 R4 M[PORTO_Ctrl], R4 CicloMenuL7 R4, 0800h	;Põe o cursor na o	
CicloMenuL8:	MOV MOV MOV CMP	M[PORTO_Ctrl], R4 R3, Menu_L8 R1, M[R3] M[PORTO_Esc], R1 R3, Fim_Menu_L8 FimEscMenu	;Escreve o caracto ;Já escrevemos a	er na janela
FimEscMenu:	BR.Z INC INC MOV BR POP POP	R3 R4 M[PORTO_Ctrl], R4 CicloMenuL8 R4 R3	;Põe o cursor na o	coluna seguinte
;=====================================	POP RET	R1 R1 que escreve a pontuação		
; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	mensage A pontu e escri	m "Pontuacao maxima: " ação é transformada em ta na janela de texto	no menu principal.	
EscrevePont:	PUSH PUSH PUSH PUSH PUSH PUSH	R1 R2 R3 R4 R5 R6		
	MOV MOV DIV	R1, M[PontMaxima] R2, 10 R1, R2	;Obtém a pontuação ;Primeiro dígito o ;significativo em	menos
	MOV ADD MOV DIV	R3, R2 R3, ASCII_Int R2, 10 R1, R2	;Código ASCII des.	se dígito em R3
	MOV ADD	R4, R2 R4, ASCII_Int	;significativo em	R2

			•		
Mai 18, 12 15:28		pacman33.a	S	Page 12/31	
	OV R2, 10 IV R1, R2		;Terceiro dígito meno ;significativo em R2	ps	
Al		CII_Int	;Código ASCII desse c	lígito em R5	
D	OV R2, 10 IV R1, R2		;Dígito mais signific	ativo em R2	
EscritaPont: Mo	ov R1, 05	12h	;Código ASCII desse o ;Escrever na linha 5,		
Mo			;Escrevemos o dígito ;mais significativo ;Saltamos para a colu	una seguinte	
Mo	OV M[PORTO	O_Ctrl], R1 O_Esc], R5	;Escrevemos o terceir ;menos significativo		
Mo		O_Ctrl], R1	;Saltamos para a colu		
I	NC R1		;Escrevemos o segundo ;menos significativo ;Saltamos para a colu		
М	OV M[PORTO		;Escrevemos o primeir ;menos significativo	ro dígito	
P(P(P(P(P(OP R6 OP R5 OP R4 OP R3 OP R2 OP R2 OP R1				
; EscreveLCD: Rot. ; no (ina que escre display LCD da ncionalidade e	ve a mensagem "Jog a janela de placa extra)	o Pac-Man"		
Pi Pi	USH R1 USH R2 USH R3	T2			
M	OV R2, Mei OV R1, LC1 OV M[LCD (D_Ini	;Endereço da mensagen ;Liga display, posici		
M(C) B) I)	OV R3, M[I OV M[LCD_I MP R2, Fin R.Z FimEscl NC R2	R2] Esc], R3 m_Menu_L3 LCD	;Escreve caracter no ;Já escreveu tudo? ;Aponta para próximo	display caracter	
FimEscLCD: PC PC	OP R3 OP R2 OP R1		;Escreve-se na coluna	n seguinte	
; ActMapa: Rotina ; movime:	Barrier de la companya della company				

Mai 18, 12 15:28		pacman33.as		Page 13/31	
; Mon	Dir pela	-			
;=======	=======	:===========	=======================================		
ActMapa:	DSI				
	PUSH	R1			
TrataMovPacMan		R1, M[Direccao]			
	CMP	R1, CIMA	;Verifica a direcç		
		PacCima	;do Pac-Man para s		
	CMP	R1, BAIXO PacBaixo	;a sub-rotina a cl	namar	
	CMP	R1, ESQ			
		PacEsq			
	CMP	R1, DIR			
	CALL.Z	PacDir			
TrataMovMons1:	MOV	R1, M[DirMonstrol]			
	PUSH	DirMonstro1	;Passa os parâmetı	ros referidos	
	PUSH	BaixoMon1	;acima		
	PUSH PUSH	PosMonstro1M PosMonstro1T			
	CMP	R1, CIMA	;para a sub-rot	-ina	
	-	MonCima	;correspondente à		
	CMP	R1, BAIXO	;do monstro 1	,	
	CALL.Z	MonBaixo			
	CMP	R1, ESQ			
		MonEsq			
	CMP	R1, DIR			
	MOV	MonDir			
	CMP	R1, M[PosMonstro2T] R1, R0			
	JMP.Z	FimActMapa			
TrataMovMons2:		R1, M[DirMonstro2]			
	PUSH	DirMonstro2	;Passa os parâmetı	ros referidos	
	PUSH	BaixoMon2	;acima		
	PUSH	PosMonstro2M			
	PUSH	PosMonstro2T			
	CMP	R1, CIMA	;para a sub-rot		
	CMP	MonCima R1, BAIXO	;correspondente à ;do monstro 2	direcção	
	-	MonBaixo	, do monscio z		
	CMP	R1, ESQ			
	_	MonEsq			
	CMP	R1, DIR			
	CALL.Z	MonDir			
	MOV	R1, M[PosMonstro3T]			
	CMP	R1, R0 FimActMapa			
TrataMovMons3:	JMP.Z MOV	R1, M[DirMonstro3]			
TT a carlo vironiss.	PUSH	DirMonstro3	;Passa os parâmetı	ros referidos	
	PUSH	BaixoMon3	;acima		
	PUSH	PosMonstro3M			
	PUSH	PosMonstro3T			
	CMP	R1, CIMA	;para a sub-rot		
		MonCima	;correspondente à	aırecção	
	CMP	R1, BAIXO	;do monstro 3		
	CALL.Z	MonBaixo R1, ESQ			
	-	MonEsq			
	CMP	R1, DIR			
	CALL.Z	MonDir			
	MOV	R1, M[PosMonstro4T]			
	CMP	R1, R0			

Moi 10 10 15:00		nacman??	26	Dogo 14/21
Mai 18, 12 15:28		pacman33.	as	Page 14/31
TrataMovMons4:	JMP.Z MOV PUSH PUSH PUSH PUSH	FimActMapa R1, M[DirMonstro4] DirMonstro4 BaixoMon4 PosMonstro4M PosMonstro4T	;Passa os parâmetros ;acima	referidos
TrataMovMons5:	CMP CALL.Z CMP CALL.Z CMP CALL.Z CMP CALL.Z CMP CALL.Z MOV CMP BR.Z MOV PUSH	R1, CIMA MonCima R1, BAIXO MonBaixo R1, ESQ MonEsq R1, DIR MonDir R1, M[PosMonstro5T] R1, R0 FimActMapa R1, M[DirMonstro5] DirMonstro5	;para a sub-rotina ;correspondente à dir ;do monstro 4 ;Passa os parâmetros	recção
	PUSH PUSH PUSH CMP CALL.Z CMP	BaixoMon5 PosMonstro5M PosMonstro5T R1, CIMA	;acima ;para a sub-rotina ;correspondente à dir ;do monstro 5	
FimActMapa:	POP ENI RET	R1		
; MOVIMENTO PAC ;====================================	======			
PacCima:	PUSH PUSH PUSH PUSH PUSH PUSH MOV SUB MOV	R2 R3 R4 R5 R6 R7 R2, M[PosPacManM] R2, N_COL R3, M[R2]	;R3 contém agora a po	sição
	CMP CALL.Z CMP CALL.NZ POP POP POP POP POP POP POP POP RET	R3, 'M' PacCapMonstro R3, '#' PacCimaGeral R7 R6 R5 R4 R3 R2	;acima do Pac-Man;Se essa posição for;chama a interacção o;Caso contrário, e se;uma parede, chama ca	om monstro não for

Mai 18, 12 15:28	}	pacman3	33.as Page 15/31
PacCimaGeral:	MOV MOV	R4, M[PosPacManT]	
	MOV MOV	R6, M[PosPacManM] R7, '@'	
	MOV MOV	M[PORTO_Ctrl], R4 M[PORTO_Esc], R5	;Escreve um espaço na posição ;que o Pac-man deixa livre
	MOV	M[R6], R5	;Escreve um espaço na posição ;de memória que o pacman
	SUB	R4, INC_Lin	
	SUB MOV	R6, N_COL M[PosPacManT], R4	;Actualiza a posição do
	MOV	M[PosPacManM], R6	;Pac-Man na janela ;Actualiza a posição do ;Pac-Man em memória
	MOV	M[PORTO_Ctrl], R4	, rac man em memoria
	MOV	M[PORTO_Esc], R7	;Escreve Pac-Man na nova ;posição na Janela
	MOV	M[R6], R7	;Escreve Pac-Man na nova ;posição em Memória
	CMP	R3, '.'	- •
		PacCapPonto	
	CMP CALL. 7	R3, ')' PacCapBanana	
	CMP	R3, '&'	
	CALL.Z	PacCapPera	
	CMP	R3, '%'	
. MONTHENED DAG	CALL.Z RET	PacCapGelado	
•	CALL.Z RET C-MAN PAR cotinas q do Pac-Ma	PacCapGelado A BAIXO 	
;=======; ; Conjunto de 1 ; descendente d	CALL.Z RET C-MAN PAR cotinas q do Pac-Ma	PacCapGelado A BAIXO 	
;======== ; Conjunto de 1 ; descendente c ;========	CALL.Z RET C-MAN PAR C-MAN PAR Cotinas q No Pac-Ma C-MAN PUSH PUSH PUSH PUSH PUSH PUSH MOV ADD MOV CMP CALL.Z CMP	PacCapGelado A BAIXO Lue prevêem e tratam o n R2 R3 R4 R5 R6 R7 R2, M[PosPacManM] R2, N_COL	movimento

Mai 18, 12 15:28		pacman33.	as	Page 16/31
	MOV MOV MOV MOV	R5, '' R6, M[PosPacManM] R7, '@' M[PORTO_Ctrl], R4 M[PORTO_Esc], R5	;Escreve um espaço na ;que o Pac-man deixa	
; Conjunto de re	e====== otinas qu	M[R6], R5 R4, INC_Lin R6, N_COL M[POSPACMANT], R4 M[POSPACMANM], R6 M[PORTO_Ctrl], R4 M[PORTO_Esc], R7 M[R6], R7 R3, '.' PacCapPonto R3, ')' PacCapBanana R3, '&' PacCapPera R3, '&' PacCapGelado A A ESQUERDA Le prevêem e tratam o mo	;Escreve um espaço na ;de memória que o pac ;Actualiza a posição ;Pac-Man na janela ;Actualiza a posição ;Pac-Man em memória ;Escreve Pac-Man na na ;posição na Janela ;Escreve Pac-Man na na ;posição em Memória ;	posição man do do
; esquerdo do Pa ;======= PacEsq:	PUSH PUSH PUSH PUSH PUSH PUSH MOV DEC MOV CMP CALL.Z	R2 R3 R4 R5 R6 R7 R2, M[PosPacManM] R2 R3, M[R2] R3, 'M' PacCapMonstro R3, '#' PacEsqGeral R7 R6 R5 R4 R3 R2	;R3 contém agora a po ;à esquerda do Pac-Ma ;Se essa posição for ;chama a interacção c ;Caso contrário, e se ;uma parede, chama ca	n um monstro om monstro não for
PacEsqGeral:	MOV MOV MOV MOV	R4, M[PosPacManT] R5, '' R6, M[PosPacManM] R7, '@' M[PORTO_Ctrl], R4	;Saltamos para a colu	na seguinte

Mai 18, 12 15:28		pacman3	3.as	Page 17/31
	MOV	M[PORTO_Esc], R5	;Escreve um espaço ;que o Pac-man dei	
	MOV	M[R6], R5	;Escreve um espaço ;de memória que o	na posição
	SUB	R4, INC_Col	,	2
	DEC MOV	R6 M[PosPacManT], R4	;Actualiza a posiç	
	MOV	M[PosPacManM], R6	;Pac-Man na janela ;Actualiza a posiç	ão do
	MOV	M[PORTO_Ctrl], R4	;Pac-Man em memóri	
	MOV	M[PORTO_Esc], R7	;Escreve Pac-Man r. ;posição na Janela	1
	MOV	M[R6], R7	;Escreve Pac-Man r ;posição em Memóri	
	CMP CALL.Z CMP CALL.Z CMP CALL.Z CMP CALL.Z CMP CALL.Z CMP	R3, '.' PacCapPonto R3, ')' PacCapBanana R3, '&' PacCapPera R3, '%' PacCapGelado		
; MOVIMENTO PAC		A A DIREITA 		
; direito do Pa	c-Man -	ue prevêem e tratam o		
;=====================================	PUSH	R2	=========	
	PUSH PUSH PUSH PUSH PUSH MOV INC	R3 R4 R5 R6 R7 R2, M[PosPacManM] R2		
	MOV	R3, M[R2]	;R3 contém agora a ;à direita do Pac-	-Man
	CMP CALL.Z CMP CALL.NZ POP POP POP POP POP POP POP POP POP PO	R3, 'M' PacCapMonstro R3, '#' PacDirGeral R7 R6 R5 R4 R3 R2	;Se essa posição f ;chama a interacçã ;Caso contrário, e ;uma parede, chama	ío com monstro e se não for
PacDirGeral:	MOV	R4, M[PosPacManT] R5, ''		
	MOV MOV	R6, M[PosPacManM]		
			;Escreve um espaço	na posição

Mai 18, 12 15:28		pacman3	3.as	Page 18/31
	ADD	R4, INC_Col		
	MOV	R6 M[PosPacManT], R4	;Actualiza a posição ;Pac-Man na janela	do
	MOV	M[PosPacManM], R6	;Actualiza a posição ;Pac-Man em memória	do
	MOV MOV	M[PORTO_Ctrl], R4 M[PORTO_Esc], R7	;Escreve Pac-Man na ;posição na Janela	nova
	MOV	M[R6], R7	;Escreve Pac-Man na ;posição em Memória	nova
	CMP CALL.Z CMP CALL.Z CMP	R3, '.' PacCapPonto R3, ')' PacCapBanana R3, '&' PacCapPera R3, '%' PacCapGelado	,, , , , , , , , , , , , , , , , , , , ,	
; ROTINAS GERAIS DE MOVIMENTO DO PAC-MAN ;====================================				
PacCapPonto:	PUSH CALL CALL RET	PtsPonto AumPontos DecNumPontos	;Passa núm. pontos p	ela pilha
PacCapBanana:	PUSH CALL RET	PtsBanana AumPontos	;Passa núm. pontos p	ela pilha
PacCapPera:	PUSH CALL RET	PtsPera AumPontos	;Passa núm. pontos p	ela pilha
PacCapGelado:	PUSH CALL RET	PtsGelado AumPontos	;Passa núm. pontos p	ela pilha
; Quando à fren	te do Pa	c-Man há um monstro		
PacCapMonstro:	MOV MOV MOV	R4, M[PosPacManT] R5, '' R6, M[PosPacManM] M[PORTO_Ctrl], R4	Anaga o Pag Man ad	mboliganda
	MOV	M[PORTO_Esc], R5 R5, ENABLE	;Apaga o Pac-Man, si ;a sua morte	.III)O11ZANQO
	MOV	M[CondGameOver], R5	;Activa a var. condi ;o "Game Over"	ção para
	RET		, o dame over	
; MOVIMENTO MON				
•		======================================		

Mai 18, 12 15:28		pacman3	33.as	Page 19/31
; Parâmetros pa		a estas rotinas pela p	ilha:	
<i>;</i>		1] <- DirMonstroX		
<i>i</i>		0] <- BaixoMonX		
<i>i</i>	M[SP+9 M[SP+8] <- PosMonstroXM] <- PosMonstroXT		
; ;=======		=======================================		
MonCima:	PUSH PUSH	R2 R3		
	PUSH	R4		
	PUSH	R5		
	PUSH	R6		
	PUSH	R7		
	MOV	R2, M[SP+9]		
	MOV	R2, M[R2]		
	SUB	R2, N_COL		
	MOV	R3, M[R2]	;R3 contém agora ; ;acima do Monstro	a posição
	CMP	R3, 'M'	;Se nessa posição	estiver um
	JMP.Z	MonCimaObstac	;monstro ou uma p	
	CMP	R3, '#'	-	
	JMP.Z	MonCimaObstac		
	CMP	R3, '@'	;Se nessa posição	estiver o
	JMP.Z	MonCimaPacMan	;Pac-Man	
MonCimaGeral:	MOV	R4, M[SP+8]	;caso contrário	o, é executado
	MOV	R4, M[R4]	;o caso geral	
	MOV	R5, M[SP+10]		
	MOV	R5, M[R5]		
	MOV MOV	R6, M[SP+9]		
	MOV	R6, M[R6] M[PORTO_Ctrl], R4		
	MOV	M[PORTO_Esc], R5	;Restora o conteú	do da posição
		11(101110_2201), 113	;que o Monstro de	
	MOV	M[R6], R5	;Restora a posiçã	
			;que o Monstro de	ixa livre
	MOV	R7, R4		
	SUB	R7, INC_Lin		
	MOV	M[PORTO_Ctrl], R7		
	MOV	R4, M[SP+8] M[R4], R7	;Actualiza a posi	cão do
	110 4	LINT), KI	;Monstro na janel	
	MOV	R7, R6	, i i i i i i i i i i i i i i i i i i i	
	SUB	R7, N_COL		
	MOV	R6, M[SP+9]		
	MOV	M[R6], R7	;Actualiza a posi ;Monstro em memór:	
	MOV	R5, M[SP+10]	, rougeto em memor	<u>.</u> .
	MOV	M[R5], R3	;Actualiza o que	
			;baixo do Monstro	
	MOV	R5, R7		
	MOV	R7, 'M'		
	MOV	M[PORTO_Esc], R7	;Escreve Monstro	na nova
			;posição na Janel	
	MOV	M[R5], R7	;Escreve Monstro	
	TMD	FimMonGimo	;posição em Memór	ia
	JMP	FimMonCima		
MonCimaObstac:	MOV	R4, M[SP+11]		
	PUSH	R4		
	CALL	RandomMon		

Moi 10, 10, 15,00		naamanaa	200	Dogo 00/04
Mai 18, 12 15:28		pacman33	.a5	Page 20/31
	JMP	FimMonCima		
MonCimaPacMan:	MOV	R4, M[SP+8]		
	MOV	R4, M[R4]		
	MOV	R5, M[SP+10]		
	MOV	R5, M[R5] R6, M[SP+9]		
	MOV MOV	R6, M[SP+9] R6, M[R6]		
	MOV	M[PORTO_Ctrl], R4		
	MOV	M[PORTO_Esc], R5	;Restora o conteúdo ;que o Monstro deix	
	MOV	M[R6], R5	;Restora a posição ;que o Monstro deix	
	MOV	R7, R4	· -	
	SUB	R7, INC_Lin		
	MOV	M[PORTO_Ctrl], R7		
	MOV MOV	R4, M[SP+8]	·Agtualiza a nogicâ	io do
		M[R4], R7	;Actualiza a posiçâ ;Monstro na janela	io uo
	MOV SUB	R7, R6 R7, N_COL		
	MOV	R6, M[SP+9]		
	MOV	M[R6], R7	;Actualiza a posiçã ;Monstro em memória	
	MOV	R5, R7	,	
	MOV	R7, 'M'		
	MOV	M[PORTO_Esc], R7	;Escreve Monstro em ;Pac-Man na Janela	
	MOV	M[R5], R7	;Escreve Monstro em ;Pac-Man em Memória	
	MOV MOV	R5, ENABLE M[CondGameOver], R5	;Activa a var. cond ;o "Game Over"	lição para
FimMonCima:	POP	R7		
	POP	R6		
	POP	R5		
	POP	R4		
	POP	R3		
	POP RETN	R2 4		
		_		
; MOVIMENTO MON	STRO PAF	RA BAIXO 	========	
; Conjunto de r ; descendente d		que prevêem e tratam o m	 novimento	
		a estas rotinas pela pil	ha:	
;		l] <- DirMonstroX		
;	-)] <- BaixoMonX		
<i>i</i>	M[SP+9]			
; ;========	M[SP+8] ======	<- PosMonstroXT =============	=======================================	
Manapatan	D	D.0		
MonBaixo:	PUSH	R2		
	PUSH PUSH	R3 R4		
	PUSH	R5		
	PUSH	R6		
	PUSH	R7		
	MOV	R2, M[SP+9]		
	MOV ADD	R2, M[R2] R2, N_COL		
	עעא	NZ, N_COL		

Mai 18, 12 15:28		pacman	33.as	Page 21/31
	MOV CMP JMP.Z CMP JMP.Z CMP JMP.Z	R3, M[R2] R3, 'M' MonBaixoObstac R3, '#' MonBaixoObstac R3, '@' MonBaixoPacMan	;R3 contém agora ;abaixo do Monstro;Se nessa posição ;monstro ou uma posição ;Se nessa posição ;Pac-Man	o estiver um arede
MonBaixoGeral:	MOV MOV MOV MOV MOV MOV	R4, M[SP+8] R4, M[R4] R5, M[SP+10] R5, M[R5] R6, M[SP+9] R6, M[R6] M[PORTO_Ctrl], R4	;caso contrário ;o caso geral	
	MOV	M[PORTO_Esc], R5 M[R6], R5	;Restora o conteú ;que o Monstro de ;Restora a posiçã ;que o Monstro de	ixa livre o de memória
	MOV ADD MOV MOV MOV	R7, R4 R7, INC_ Lin M[PORTO_Ctr1], R7 R4, M[SP+8] M[R4], R7	;Actualiza a posi	
	MOV ADD MOV MOV	R7, R6 R7, N_COL R6, M[SP+9] M[R6], R7	;na janela ;Actualiza a posi ;Monstro em memór	
	MOV MOV	R5, M[SP+10] M[R5], R3	;Actualiza o que : ;baixo do Monstro	
	MOV MOV MOV	R5, R7 R7, 'M' M[PORTO_Esc], R7	;Escreve Monstro ; ;posição na Janel	
	MOV	M[R5], R7	;Escreve Monstro ;posição em Memór	na nova
MonBaixoObstac:	JMP MOV PUSH CALL JMP	FimMonBaixo R4, M[SP+11] R4 RandomMon FimMonBaixo		
MonBaixoPacMan:	MOV MOV MOV MOV MOV MOV MOV	R4, M[SP+8] R4, M[R4] R5, M[SP+10] R5, M[R5] R6, M[SP+9] R6, M[R6] M[PORTO_Ctr1], R4 M[PORTO_ESC], R5	;Restora o conteú ;que o Monstro de	
	MOV MOV ADD	M[R6], R5 R7, R4 R7, INC_ Lin	;Restora a posiçã ;que o Monstro de	o de memória

Mai 18, 12 15:28		pacman33.a	as	Page 22/31	
	MOV MOV	R4, M[SP+8] M[R4], R7	;Actualiza a posição ;Monstro na janela	do	
	MOV ADD MOV MOV	R7, R6 R7, N_COL R6, M[SP+9] M[R6], R7	;Actualiza a posição ;Monstro em memória	do	
	MOV MOV MOV	R5, R7 R7, 'M' M[PORTO_Esc], R7 M[R5], R7	;Escreve Monstro em c ;do Pac-Man na Janela ;Escreve Monstro em c		
	MOV MOV	R5, ENABLE M[CondGameOver], R5	;do Pac-Man em Memóri ;Activa a var. condiç ;para o "Game Over"	а	
FimMonBaixo:	POP POP POP POP POP POP RETN	R7 R6 R5 R4 R3 R2 4			
; MOVIMENTO MONSTRO PARA A ESQUERDA					
; esquerdo dum I ; Parâmetros pa: ; ; ; ;	Monstro. ssados a M[SP+11 M[SP+10 M[SP+9] M[SP+8]	ue prevêem e tratam o mov estas rotinas pela pilha] <- DirMonstroX] <- BaixoMonX <- PosMonstroXM <- PosMonstroXT	a:		
MonEsq:	PUSH PUSH PUSH PUSH PUSH MOV MOV DEC	R2 R3 R4 R5 R6 R7 R2, M[SP+9] R2, M[R2] R2 R3, M[R2]	;R3 contém agora a po ;à esquerda do Monstr	0	
	CMP JMP.Z CMP JMP.Z CMP JMP.Z	R3, 'M' MonEsqObstac R3, '#' MonEsqObstac R3, '@' MonEsqPacMan	;Se nessa posição est ;monstro ou uma pared ;Se nessa posição est ;Pac-Man	le	
MonEsqGeral:	MOV MOV MOV MOV MOV MOV	R4, M[SP+8] R4, M[R4] R5, M[SP+10] R5, M[R5] R6, M[SP+9] R6, M[R6] M[PORTO_Ctrl], R4	;caso contrário, é;o caso geral	executado	

Mai 18, 12 15:28		pacman3	3.as	Page 23/31
	MOV	M[PORTO_Esc], R5	;Restora o conteúd ;que o Monstro dei	
	MOV	M[R6], R5	;Restora a posição ;que o Monstro dei	de memória
	MOV	R7, R4	, 1	
	SUB	R7, INC_ Col		
	MOV	M[PORTO_Ctrl], R7		
	MOV	R4, M[SP+8]		
	MOV	M[R4], R7	;Actualiza a posiç ;Monstro na janela	
	MOV	R7, R6		
	DEC	R7		
	MOV	R6, M[SP+9]		
	MOV	M[R6], R7	;Actualiza a posiç ;Monstro em memóri	
	MOV	R5, M[SP+10]		
	MOV	M[R5], R3	;Actualiza o que f ;baixo do Monstro	ica por
	MOV	R5, R7		
	MOV MOV	R7, 'M' M[PORTO_Esc], R7	;Escreve Monstro n	a nova
	MOV	M[PORIO_ESC], K/	;posição na Janela	
	MOV	M[R5], R7	;Escreve Monstro n	
	JMP	FimMonEsq	;posição em Memóri	
		_		
MonEsqObstac:	MOV	R4, M[SP+11]		
	PUSH	R4		
	CALL	RandomMon		
	JMP	FimMonEsq		
MonEsqPacMan:	MOV	R4, M[SP+8]		
	MOV	R4, M[R4]		
	MOV	R5, M[SP+10]		
	MOV	R5, M[R5]		
	MOV	R6, M[SP+9]		
	MOV	R6, M[R6]		
	MOV	M[PORTO_Ctrl], R4		
	MOV	M[PORTO_Esc], R5	;Restora o conteúd	o da posição
			;que o Monstro dei	
	MOV	M[R6], R5	;Restora a posição ;que o Monstro dei	
	MOV	R7, R4	-	
	SUB	R7, INC_ Col		
	MOV	M[PORTO_Ctrl], R7		
	MOV	R4, M[SP+8]		
	VOM	M[R4], R7	;Actualiza a posiç ;Monstro na janela	
	MOV	R7, R6	-	
	DEC	R7		
	MOV	R6, M[SP+9]		
	MOV	M[R6], R7	;Actualiza a posiç ;Monstro em memóri	
	MOV	R5, M[SP+10]		
	MOV	R7, 'M'		
	VOM	M[PORTO_Esc], R7	;Escreve Monstro e ;do Pac-Man na Jan	
	MOV	M[R5], R7	;Escreve Monstro e ;do Pac-Man em Mem	
	MOV	R5, ENABLE		
		M[CondGameOver], R5		

Mai 18, 12 15:28		pacman33.a	as	Page 24/31
			;o "Game Over"	
FimMonEsq:	POP POP POP POP POP RETN	R7 R6 R5 R4 R3 R2		
; Conjunto de r ; direito dum M ; Parâmetros pa ; ; ; ; ;	====== otinas q onstro. ssados a M[SP+11 M[SP+10	ue prevêem e tratam o movestas rotinas pela pilha] <- DirMonstroX] <- BaixoMonX <- PosMonstroXM <- PosMonstroXT	vimento lateral	
MonDir:	PUSH PUSH PUSH PUSH PUSH MOV MOV INC MOV CMP JMP.Z CMP JMP.Z CMP JMP.Z	R2 R3 R4 R5 R6 R7 R2, M[SP+9] R2, M[R2] R2 R3, M[R2] R3, 'M' MonDirObstac R3, '#' MonDirObstac R3, '@' MonDirPacMan	;R3 contém agora a po ;à direita do Monstro ;Se nessa posição est ;monstro ou uma pareo ;Se nessa posição est ;Pac-Man	civer um
MonDirGeral:	MOV	R4, M[SP+8] R4, M[R4] R5, M[SP+10] R5, M[R5] R6, M[SP+9] R6, M[R6] M[PORTO_Ctrl], R4 M[PORTO_Esc], R5 M[R6], R5 R7, R4 R7, INC_Col M[PORTO_Ctrl], R7 R4, M[SP+8]	;caso contrário, é; o caso geral ;Restora o conteúdo o; que o Monstro deixa; Restora a posição de; que o Monstro deixa	da posição livre e memória
	MOV MOV INC MOV MOV	R4, M[SP+8] M[R4], R7 R7, R6 R7 R6, M[SP+9] M[R6], R7 R5, M[SP+10]	;Actualiza a posição ;Monstro na janela ;Actualiza a posição ;Monstro em memória	

M[R5], R3 R5, R7 R7, 'M' M[PORTO_Esc], R7 M[R5], R7 FimMonDir R4, M[SP+11] R4 RandomMon FimMonDir R4, M[SP+8] R4, M[R4] R5, M[SP+10] R5, M[R5] R6, M[SP+9] R6, M[R6]	;Actualiza o que fica por ;baixo do Monstro ;Escreve Monstro na nova ;posição na Janela ;Escreve Monstro na nova ;posição em Memória
R7, 'M' M[PORTO_Esc], R7 M[R5], R7 FimMonDir R4, M[SP+11] R4 RandomMon FimMonDir R4, M[SP+8] R4, M[SP+8] R4, M[R4] R5, M[SP+10] R5, M[SP+10] R6, M[SP+9] R6, M[R6]	;posição na Janela ;Escreve Monstro na nova
FimMonDir R4, M[SP+11] R4 RandomMon FimMonDir R4, M[SP+8] R4, M[SP+8] R5, M[SP+10] R5, M[SP+10] R6, M[SP+9] R6, M[R6]	;posição na Janela ;Escreve Monstro na nova
R4, M[SP+11] R4 RandomMon FimMonDir R4, M[SP+8] R4, M[R4] R5, M[SP+10] R5, M[SP+10] R6, M[SP+9] R6, M[R6]	;posição em Memoria
R4 RandomMon FimMonDir R4, M[SP+8] R4, M[R4] R5, M[SP+10] R5, M[SP+9] R6, M[SP+9]	
R4, M[R4] R5, M[SP+10] R5, M[R5] R6, M[SP+9] R6, M[R6]	
M[PORTO_Ctrl], R4 M[PORTO_Esc], R5	;Restora o conteúdo da posição ;que o Monstro deixa livre
M[R6], R5	;que o monstro deixa livre ;Restora a posição de memória ;que o Monstro deixa livre
R7, R4 R7, INC_ Col M[PORTO_Ctrl], R7	
R4, M[SP+8] M[R4], R7	;Actualiza a posição do ;Monstro na janela
R7, R6 R7 R6. M[SP+9]	,
M[R6], R7	;Actualiza a posição do ;Monstro em memória
R7, 'M'	;Escreve Monstro em cima do
M[R5], R7	;Pac-Man na Janela ;Escreve Monstro em cima do
R5, ENABLE M[CondGameOver], R5	;Pac-Man em Memória ;Activa a var. condição para ;o "Game Over"
R7 R6 R5 R4 R3 R2	
	R7, INC_Col M[PORTO_Ctrl], R7 R4, M[SP+8] M[R4], R7 R7, R6 R7 R6, M[SP+9] M[R6], R7 R5, R7 R7, 'M' M[PORTO_Esc], R7 M[R5], R7 R5, ENABLE M[CondGameOver], R5 R7 R6 R5 R4 R3 R2

```
pacman33.as
 Mai 18, 12 15:28
                                                          Page 26/31
           parâmetro pela pilha
; Parâmetro passado a esta rotina pela pilha:
          M[SP+4] \leftarrow DirMonstroX (X=1,2,3,4,5)
RandomMon:
             PUSH
                    R6
             PUSH
                    R7
Random:
             I10P
                    M[RandomVar]
             MOV
                    R6, M[RandomVar]
             MOV
                    R7.4
                                        ;Ouatro direcções possíveis
             DIV
                    R6, R7
                                        :0 resto, valor entre 0 e 3,
                                        ;ficará em R7
             MOV
                    R6, M[SP+4]
                                        ; Passa end. DirMonstroX para R6
             CMP
                    M[R6], R7
                                        :Se a nova direcção for iqual à
             BR.Z
                    Random
                                        ;anterior repete a geração de
                                        ;número aleatório
             MOV
                    M[R6], R7
                                        ;Guarda a nova direcção em M[R6]
             POP
                    R7
             POP
                    R6
             RETN
                    1
._____
; AumPontos: Rotina que soma pontos obtidos à pontuação actual
          do jogador
; Parâmetro passado a esta rotina pela pilha:
          M[SP+3] <- PtsPonto, PtsBanana, PtsPera ou PtsGelado
AumPontos:
             PUSH
                    R7
                    R7, M[SP+3]
             MOV
             ADD
                    M[Pontuacao], R7
             MOV
                    R7, M[Pontuacao]
             CMP
                    R7, LIM_Pontuacao
                                        ;Se pontuação for maior que 9999
             BR.NP
                    ContAumPontos
                    R7, LIM_Pontuacao
             MOV
                                        ;força-se pontuação = 9999
             MOV
                    M[Pontuacaol, R7
ContAumPontos:
             MOV
                    R7, M[Pontuação]
             PUSH
             CALL
                    ActDisplay
             PUSH
             CALL
                    ActDific
             POP
                    R7
             RETN
                    1
; ActDisplay: Rotina que, recebendo como parâmetro pela
           pilha a pontuação actual do jogador, coloca
           esse valor no display de 7 segmentos
; Parâmetro passado a esta rotina pela pilha:
           M[SP+5] <- M[Pontuação] em registo
ActDisplay:
             PUSH
                    R1
             PUSH
                    R2
                    R3
             PUSH
             MOV
                    R1, Display
                                        ;Passa endereço Display para R1
             MOV
                    R2, M[SP+5]
                                        ;Passa pontuação para R2
             MOV
                    R3, 10
             DIV
                    R2, R3
                                        ;Obtém e escreve dígito menos
             MOV
                    M[R1], R3
                                        ;significativo no display
             MOV
                    R3, 10
             DIV
                    R2, R3
                                        ;Obtém e escreve segundo dígito
```

Mai 18, 12 15:28		pacman33.	as	Page 27/31
	MOTZ	•		
	MOV DIV MOV MOV	M[R1+1], R3 R3, 10 R2, R3 M[R1+2], R3 R3, 10	;menos significativo ;Obtém e escreve ter ;menos significativo	ceiro dígito
	DIV MOV POP POP POP RETN	R2, R3 M[R1+3], R3 R3 R2 R1	;Obtém e escreve díg ;significativo no di	_
; pilk ; a di ; Parâmetro pass	ina que, na a pon ificulda sado a e	recebendo como parâmetr tuação actual do jogador de (velocidade) do jogo sta rotina pela pilha: M[Pontuacao] em registo	o pela	
ActDific:	PUSH PUSH PUSH	R1 R2 R3		
	MOV MOV MUL	R2, M[SP+5] R3, LIM_DifPts R1, 1 R1, R3	;Passa pontuação par ;Passa margens de di	fic. para R3
	CMP JMP.N MOV MOV MUL CMP	R2, R3 ActDifNiv1 R3, LIM_DifPts R1, 2 R1, R3 R2, R3	;As comparações aqui ;destinam-se a verif ;a pontuação actual ;dentro das margens ;pontuação que defin ;dificuldade.	icar onde se encaixa de
	JMP.N MOV MOV MUL CMP	ActDifNiv2 R3, LIM_DifPts R1, 3 R1, R3 R2, R3	;Estas margens podem ;conferidas na págin ;do relatório do pro	a 7
	JMP.N MOV MUL CMP	ActDifNiv3 R3, LIM_DifPts R1, 4 R1, R3 R2, R3		
	JMP.N MOV MUL CMP	ActDifNiv4 R3, LIM_DifPts R1, 5 R1, R3 R2, R3		
	JMP.N MOV MOV MUL CMP	ActDifNiv5 R3, LIM_DifPts R1, 6 R1, R3 R2, R3		
	JMP.N MOV MOV MUL CMP JMP.N	ActolfNiv6 R3, LIM_DifPts R1, 7 R1, R3 R2, R3 ActolfNiv7		
ActDifNiv1:	JMP MOV	ActDifNiv8 R1, DIF_Niv1	;Dependendo da pontu	ação,

Mai 18, 12 15:	28	pacman	33.as	Page 28/31
	MOV MOV MOV JMP	M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 FimActDific	;a variável Tim ;(correspondent ;de jogo) e o n ;acesos são alt	e à velocidade úmero de LEDs
ActDifNiv2:	MOV MOV MOV JMP	R1, DIF_Niv2 M[TimeLong], R1 R1, LEDs2 M[LEDs], R1 FimActDific		
ActDifNiv3:	MOV MOV MOV JMP	R1, DIF_Niv3 M[TimeLong], R1 R1, LEDs3 M[LEDs], R1 FimActDific		
ActDifNiv4:	MOV MOV MOV JMP	R1, DIF_Niv4 M[TimeLong], R1 R1, LEDs4 M[LEDs], R1 FimActDific		
ActDifNiv5:	MOV MOV MOV BR	R1, DIF_Niv5 M[TimeLong], R1 R1, LEDs5 M[LEDs], R1 FimActDific		
ActDifNiv6:	MOV MOV MOV MOV BR	R1, DIF_Niv6 M[TimeLong], R1 R1, LEDs6 M[LEDs], R1 FimActDific		
ActDifNiv7:	MOV MOV MOV BR	R1, DIF_Niv7 M[TimeLong], R1 R1, LEDs7 M[LEDs], R1 FimActDific		
ActDifNiv8:	MOV MOV MOV	R1, DIF_Niv8 M[TimeLong], R1 R1, LEDs8 M[LEDs], R1		
FimActDific:	POP POP POP RETN	R3 R2 R1 1		
; MudaNivel: ;	Rotina que come os po	e trata a mudança de n ontos todos do mapa		or
MudaNivel:	DSI PUSH MOV CMP BR.Z BR	R2 R1, M[MapaActual] R1, COD_Niv1 MudaNiv2 MudaNiv3	;Se estiver no . ;Muda para o Ní ;Caso contrário	

•	3	pacman33	3.as Page 29/31
MudaNiv2:	MOV MOV PUSH	R1, COD_Niv2 M[MapaActual], R1 Nivel2_L0	;Passa o endereço do mapa ;do Nível 2
	CALL BR	EscreveMapa FimMudaNivel	;para a rotina EscreveMapa
MudaNiv3:	MOV MOV PUSH	R1, COD_Niv3 M[MapaActual], R1 Nivel3_L0	;Passa o endereço do mapa ;do Nível 3
	CALL	EscreveMapa	;para a rotina EscreveMapa
FimMudaNivel:	POP ENI MOV MOV	R2 R1, INDEF M[Direccao], R1	;Colocar estas duas instruções ;aqui evita que o Pac-Man se ;mexa devido a interrupções ;pendentes
	RET		
; ; ; ;=======			
RotGameOver:	PUSH	R2	
	MOV MOV CMP BR.NN	R1, M[Pontuacao] R2, M[PontMaxima] R2, R1 ContGameOver	;Se pontuação não for maior que
	MOV MOV CMP	R1, M[Pontuacao] R2, M[PontMaxima] R2, R1	;Se pontuação não for maior que ;o high score ;Caso contrário, actualiza o ;high score
ContGameOver:	MOV MOV CMP BR.NN MOV	R1, M[Pontuacao] R2, M[PontMaxima] R2, R1 ContGameOver M[PontMaxima], R1 GameOver_L0 EscreveMapa	;o high score ;Caso contrário, actualiza o
ContGameOver:	MOV MOV CMP BR.NN MOV PUSH CALL MOV MOV	R1, M[Pontuacao] R2, M[PontMaxima] R2, R1 ContGameOver M[PontMaxima], R1 GameOver_L0 EscreveMapa R1, INDEF M[PosMonstrolT], R0 M[PosMonstrolM], R0	;o high score ;Caso contrário, actualiza o ;high score ;Passa o endereço da animação ;"Game Over"para a rotina ;EscreveMapa (func. opcional) ;Reinicializa a posição dos ;monstros em memória e na ;janela de texto (escondidos)
ContGameOver:	MOV MOV CMP BR.NN MOV PUSH CALL MOV MOV	R1, M[Pontuacao] R2, M[PontMaxima] R2, R1 ContGameOver M[PontMaxima], R1 GameOver_L0 EscreveMapa R1, INDEF M[PosMonstrolT], R0	;o high score ;Caso contrário, actualiza o ;high score ;Passa o endereço da animação ;"Game Over"para a rotina ;EscreveMapa (func. opcional) ;Reinicializa a posição dos ;monstros em memória e na
ContGameOver:	MOV MOV CMP BR.NN MOV PUSH CALL MOV MOV MOV MOV	R1, M[Pontuacao] R2, M[PontMaxima] R2, R1 ContGameOver M[PontMaxima], R1 GameOver_L0 EscreveMapa R1, INDEF M[PosMonstrolT], R0 M[PosMonstrolM], R0 M[DirMonstrol], R1 M[PosMonstrol], R1	;o high score ;Caso contrário, actualiza o ;high score ;Passa o endereço da animação ;"Game Over"para a rotina ;EscreveMapa (func. opcional) ;Reinicializa a posição dos ;monstros em memória e na ; janela de texto (escondidos) ;Coloca igualmente a direcção ;inicial de todos eles como ;indefinida, de forma a estarem

Mai 18, 12 15:28		pacman33.a	IS	Page 30/31
		;do Pac-Man		
	MOV MOV MOV MOV	R1, DIF_Niv1 M[TimeLong], R1 M[Pontuacao], R0 M[CondGameOver], R0 R2	;Reinicializa a dif. ;Reinicializa a pontu ;Coloca a var. condiç ;o "Game Over" a 0	ação
	RET	KZ		
	====== spondente iável de			
IntCima:	PUSH MOV MOV POP RTI	R1 R1, CIMA M[Direccao], R1 R1		
IntBaixo:	PUSH MOV MOV POP RTI	R1 R1, BAIXO M[Direccao], R1 R1		
IntEsq:	PUSH MOV MOV POP RTI	R1 R1, ESQ M[Direccao], R1 R1		
IntDir:	PUSH MOV MOV POP RTI	R1 R1, DIR M[Direccao], R1 R1		
; IntTimer: Act	iva varia	ável de condição do tempo a contagem		
IntTimer:	PUSH MOV MOV MOV MOV MOV POP RTI	R1 R1, M[TimeLong] M[TimerValue], R1 R1, ENABLE M[CondTimer], R1 M[TimerControl], R1 R1		
; Programa princ	cipal			
Inicio:	MOV MOV	R1, Topo_Pilha SP, R1	;Inicialização da pil	
	MOV	R1, INI_Int_Mask	;a posição mais alta	Possivei:

MOV CALL BR DSI CALL CALL MOV MOV MOV MOV MOV MOV PUSH CALL MOV MOV MOV	pacman33 M[Int_Mask],R1 EscreveLCD IniMenu RotGameOver EscreveMenu R1, M[INI_Index] EsperaTecla EscolheNivel R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay R1, M[TimeLong]	;Definição da máscara de ;interrupções ;Reinicializar os parâmetros ;Mostrar menu principal ;Recoloca porto de estado a 0 ;Inicializa velocidade de jogo ;Acende o primeiro LED ;Coloca pontuação inicial a 0
BR DSI CALL CALL MOV CALL MOV MOV MOV MOV MOV PUSH CALL MOV MOV MOV	IniMenu RotGameOver EscreveMenu R1, M[INI_Index] EsperaTecla EscolheNivel R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Reinicializar os parâmetros ;Mostrar menu principal ;Recoloca porto de estado a 0 ;Inicializa velocidade de jogo ;Acende o primeiro LED
DSI CALL CALL MOV	RotGameOver EscreveMenu R1, M[INI_Index] EsperaTecla EscolheNivel R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Mostrar menu principal ;Recoloca porto de estado a 0 ;Inicializa velocidade de jogo ;Acende o primeiro LED
CALL CALL MOV CALL CALL MOV	EscreveMenu R1, M[INI_Index] EsperaTecla EscolheNivel R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Mostrar menu principal ;Recoloca porto de estado a 0 ;Inicializa velocidade de jogo ;Acende o primeiro LED
CALL MOV CALL CALL MOV MOV MOV MOV MOV MOV MOV MOV CALL MOV MOV MOV	EscreveMenu R1, M[INI_Index] EsperaTecla EscolheNivel R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Mostrar menu principal ;Recoloca porto de estado a 0 ;Inicializa velocidade de jogo ;Acende o primeiro LED
MOV CALL CALL MOV MOV MOV MOV MOV MOV MOV CALL MOV MOV MOV MOV MOV MOV	R1, M[INI_Index] EsperaTecla EscolheNivel R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Recoloca porto de estado a 0 ;Inicializa velocidade de jogo ;Acende o primeiro LED
CALL CALL MOV MOV MOV MOV MOV MOV MOV PUSH CALL MOV MOV	EsperaTecla EscolheNivel R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Inicializa velocidade de jogo ;Acende o primeiro LED
CALL MOV MOV MOV MOV MOV PUSH CALL MOV MOV	EscolheNivel R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Acende o primeiro LED
MOV MOV MOV MOV MOV PUSH CALL MOV MOV	R1, DIF_Niv1 M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Acende o primeiro LED
MOV MOV MOV PUSH CALL MOV MOV	M[TimeLong], R1 R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Acende o primeiro LED
MOV MOV MOV PUSH CALL MOV MOV	R1, LEDs1 M[LEDs], R1 R1, R0 R1 ActDisplay	;Acende o primeiro LED
MOV MOV PUSH CALL MOV MOV	M[LEDs], R1 R1, R0 R1 ActDisplay	· ·
MOV PUSH CALL MOV MOV	R1, R0 R1 ActDisplay	· ·
PUSH CALL MOV MOV	R1 ActDisplay	:Coloca pontuação inicial a O
CALL MOV MOV	ActDisplay	:Coloca pontuação inicial a 0
VOM VOM		:COLOCA DONLUAÇÃO INICIAL A U
VOM		,
	M[TimerValue], R1	;Passa velocidade para o Timer
MOV	R1, ENABLE	, rassa velocidade para o limer
MOV	M[TimerControl], R1	;Activa o temporizador
	M[IIMCICONCIOI], KI	,Activa o temporizador
	R1. INDEF	;Colocar estas duas instruções
MOV	M[Direccao], R1	;aqui evita que o Pac-Man se ;mexa devido a interrupções ;pendentes
MOV	R1, M[CondTimer]	,1
CMP	R1, ENABLE	;Verifica se a var. de condição
BR.NZ	CicloJogo	;do temporizador foi activada
CALL	ActMapa	_
VOM	M[CondTimer], R0	;Desactiva variável condição do ;temporizador
MOV	R1, M[NumPontos]	· -
CMP	R1, R0	;Se tiver comido os pontos todos
CALL.Z	MudaNivel	
MOV	<pre>R1, M[CondGameOver]</pre>	
CMP	R1, ENABLE	;Se tiver sido apanhado por ;um monstro
JMP.Z	GameOver	;trata do Game Over e ;reinicializa
BR	CicloJogo	
	ENI MOV MOV MOV EMP ER.NZ CALL MOV MOV EMP CALL.Z MOV EMP CALL.Z	MOV R1, INDEF MOV M[Direccao], R1 MOV R1, M[CondTimer] R1, ENABLE CicloJogo CALL ActMapa MOV M[CondTimer], R0 MOV R1, M[NumPontos] R1, R0 CALL.Z MudaNivel MOV R1, M[CondGameOver] R1, ENABLE MOV R1, ENABLE MOV R1, M[CondGameOver] R1, ENABLE