

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 1/17
<pre>;/Projeto de IAC - LEIC-A - IST 2015/2016 ;/Grupo 4 ;/83559 - Rodrigo Joao Fraga Lima ;/83567 - Tiago Miguel Calhanas Goncalves ;/-----/ ;/ ;/ ;/ ;/ ;/-----/ IO_READ EQU FFFFh IO_WRITE EQU FFEHh IO_STATUS EQU FFFDh IO_CONTR EQU SP_INICIAL EQU ;Auxiliares de Linha: LINHA_FIM EQU 0018h MUD_LINHA EQU 000Ah ;Caracteres: Limit EQU '-' Obsta EQU 'X' Clear EQU ',' Fim_Linha EQU '@' ;Coordenadas do campo de jogo: Titu_X EQU 0024h Titu_X2 EQU 001Eh Titu_Y EQU 000Ch Comp_Lin EQU 004Eh Col_Ini EQU 0014h Lin_Ini EQU 000Ch ;Mascaras de interrupcoes: InterruptMask EQU FFFAh INT_Mask01 EQU 8007h INT_Mask00 EQU ;Valores do Temporizador: TimerValue EQU FFF6h TimerControl EQU FFF7h TimeLong EQU 0001h EnableTimer EQU 0001h ;Constantes Auxiliares de Temporizador C_TEMP_GRAV EQU 0002h ;Limite inferior do jogo Lim_Inf EQU 0017h ;Enderecos de 7 segmentos: IO_DISPLAY EQU FFF0h ;Endereco dos LEDs IO_LED EQU FFF8h ;Enderecos do LCD IO_LCD_CTRL EQU FFF4h</pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 2/17
<pre>IO_LCD_CAR EQU FFF5h ;Velocidade Vel_Final EQU 0005h ;Contantes de Obstaculos Gap EQU 0010h RANDOMMASK EQU ;/-----/ ;/ ;/ ;/ ;/-----/ ;/ ;/ ;/ ;/-----/ ;Strings: Bird STR 'O>' Line2 STR 'Prima o Interruptor II@' Line1 STR 'Prepare-se@' Str1 STR 'Distancia:@' Str2 STR 'colunas@' EndLine1 STR 'Fim do Jogo@' ;Coordenadas Passaro: BirdX WORD 0014h BirdY WORD 000Ch Bird_Y2 WORD 000Ch ;Coordenadas Auxiliares do Escreve CoordX WORD 0000h CoordY WORD 0000h ;Flags: Verifica WORD 0001h Start_S WORD 0001h Gravi WORD 0000h Estadescendo WORD 0000h DEC_Nivel WORD 0001h INC_Nivel WORD 0001h FlagGameOver WORD 0000h ;LED LEVELS: XORLED WORD 8000h LED_LEVEL WORD 8000h LVL WORD 0001h ;Auxiliares temporais: TEMP_OBSTA WORD 0000h TEMP_GRAVIDADE WORD 0000h ;Pontuacoes: COLUNAS WORD 0000h PONTOS WORD 0000h ;Variacao de velocidade dos obstaculos C_TEMP_OBSTA WORD 000Ah Velocidade WORD 0000h ;Variaveis dos Obstaculos: XClear WORD 0000h EspacoAtual WORD 0000h</pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as		Page 3/17
Obstaculos	TAB	6	
TABRNG	TAB	6	
Obts_Atuais	WORD	0000h	
NUMRANDOM	WORD	FFFFh	
Jamovido	WORD	0000h	
Xy	WORD	0000h	
Indice	WORD	0000h	
DiminuiObj	WORD	0000h	
FlagCol	WORD	0000h	
IndiceColisao	WORD	0000h	
;/-----/			
;/			
;/ Tabela de			
;/ Interupcoes			
;/-----/			
INT0	WORD	ORIG	FE00h
		Botao	
INT01	WORD	ORIG	FE01h
		Start	
INT02	WORD	ORIG	FE02h
		Inc_dif	
INT15	WORD	ORIG	FE0Fh
		TimerSub	
;/-----/			
;/ Botao 0			
;/-----/			
Botao:	MOV	M[Verifica], R0	
		RTI	
;/-----/			
;/ Botao 1			
;/-----/			
Start:	MOV	M[Start_S], R0	
		RTI	
;/-----/			
;/ Botao 1 - Funcao 2			
;/-----/			
Dec_dif:	MOV	M[DEC_Nivel], R0	
		RTI	
;/-----/			
;/ Botao FAZ NADA			
;/-----/			
Faz_nada:	RTI		
;/-----/			
;/ Botao 2			
;/-----/			
Inc_dif:	MOV	M[INC_Nivel], R0	
		RTI	
;/-----/			
;/ Rotina			
;/ de apoio ao temporizador			
;/-----/			
TimerSub:	PUSH	R7	
		PUSH	R4

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as		Page 4/17
	PUSH	R2	
	PUSH	R1	
	MOV	R7, INT_Mask01	
	MOV	M[InterruptMask], R7	
	MOV	R7, TimeLong	
	MOV	M[TimerValue], R7	
	MOV	R7, EnableTimer	
	MOV	M[TimerControl], R7	
	DEC	M[TEMP_GRAVIDADE]	
	MOV	R7, M[Gravi]	
	CMP	R7, R0	
	BR.Z	TS1	
	INC	M[Estadescendo]	
TS1:	DEC	M[TEMP_OBSTA]	
	POP	R1	
	POP	R2	
	POP	R4	
	POP	R7	
	RTI		
;Rotina que faz o controlo de todas as flags e responsavel pela execucao			
;do jogo.			
;/-----/			
;/ Programa			
;/ Principal			
;/-----/			
	ORIG	0000h	
	MOV	R1, SP_INICIAL	
	MOV	SP, R1	
	MOV	R1, FFFFh	
	MOV	M[IO_CONTR], R1	
	MOV	R7, INT_Mask00	
	MOV	M[InterruptMask], R7	
	END		
	CALL	SplashScreen	
		M[NUMRANDOM]	
			;Impede o mesmo
NotStart:	DEC		
padrao de aparecer nos obstaculos			
	CMP	M[Start_S], R0	
	BR.NZ	NotStart	
	MOV	R1, FFFFh	
	MOV	M[IO_CONTR], R1	
	CALL	ClrScr	
	CALL	E_Inicial	
	CALL	LCDINI	
	MOV	R7, INT_Mask01	
	MOV	M[InterruptMask], R7	
	MOV	R7, Faz_nada	
	MOV	M[INT01], R7	
	MOV	R7, TimeLong	
	MOV	M[TimerValue], R7	
	MOV	R7, EnableTimer	
	MOV	M[TimerControl], R7	
	MOV	R7, 8000h	
	MOV	M[IO_LED], R7	
	MOV	R7, M[C_TEMP_OBSTA]	
			;Carrega
a constante de velocidade dos obstaculos, controla a velocidade com que estes a			
vançam.			

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 5/17
<pre>MOV M[TEMP_OBSTA], R7 MOV R7, C_TEMP_GRAV ;Carrega a constante de gravidade, que controla a força da mesma MOV M[TEMP_GRAVIDADE], R7 Sobe: CMP M[Verifical], R0 Dificuldadel1: CMP M[Dificuldadel] CALL Subir M[DEC_Nivel], R0 BR.NZ Dificuldade2 CALL Diminui_Dif INC M[DEC_Nivel] Dificuldade2: CMP M[INC_Nivel], R0 BR.NZ Temp_Obsta CALL Aument_Dif ;FLAGS DE CONTROLO QUE ACCIONAM AS ROTINAS INC M[INC_Nivel] Temp_Obsta: CMP M[TEMP_OBSTA], R0 CALL Temp_Grav Temp_Grav: CMP M[TEMP_GRAVIDADE], R0 CALL Descendo Descendo: CMP M[Estadescendo], R0 BR.Z Esc_gap CALL Descer Esc_gap: CMP M[EspacoAtual], R0 BR.NZ UP CALL Cria_Obsta JMP Sobe ;As rotinas aumenta e diminui dificuldade mudam a contante de velocidade ;dos obstaculos (esta controla a velocidade dos obstasculos), conforme o ;interruptor premido e faz tambem a atualizacao da variavel LVL que indica o ;nivel de dificuldade atual nos LEDs da placa. ;/------/ ;/- Aumenta dificuldade / ;/------/ Aument_Dif: PUSH R1 MOV R2 PUSH R7 SHR R1, M[XORLED] MOV R1, 1 MOV R2, M[LED_LEVEL] XOR R2, R1 INC M[LVL] MOV R7, 0008h CMP M[LVL], R7 BR.NZ AD MOV R7, Faz_nada MOV M[INT02], R7 MOV M[LED_LEVEL], R2 MOV M[IO_LED], R2 MOV R7, Dec_dif MOV M[INT01], R7 DEC R7 POP R2 POP R1</pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 6/17
<pre>RET ;/------/ ;/- Diminui dificuldade / ;/------/ Diminui_Dif: PUSH R1 PUSH R2 PUSH R7 MOV R7, INT_Mask01 MOV M[InterruptMask], R7 MOV R1, M[XORLED] MOV R2, M[LED_LEVEL] XOR R2, R1 DEC M[LVL] MOV R7, 0001h CMP R7, M[LVL] BR.NZ DD MOV R7, Faz_nada MOV M[INT01], R7 MOV M[LED_LEVEL], R2 MOV M[IO_LED], R2 SHL R1, 1 MOV M[XORLED], R1 MOV R7, Inc_dif MOV M[INT02], R7 INC M[C_TEMP_OBSTA] POP R7 POP R2 POP R1 RET ;Rotina responsavel por escrever todo o ecrã inicial, com limites ;o passaro inicial, e o primeiro obstaculo ;/------/ ;/- Preenchimento / ;/- inicial do Ecrã / ;/------/ E_Inicial: PUSH R1 MOV M[CoordX], R0 MOV M[CoordY], R0 CALL Pos_Curs CALL Escr_Linha CALL M[CoordX], R0 MOV R1, 0017h MOV M[CoordY], R1 CALL Pos_Curs CALL Escr_Linha CALL Bird_Ini CALL Cria_Obst POP R1 RET ;Rotina responsavel por escrever os limites superior e inferior do jogo. ;/------/ ;/- Escreve Linha / ;/------/ Escr_Linha: PUSH R1</pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 7/17
Repete:	<pre>MOV R1, Comp_Lin Limit CALL Esc_Car INC M[CoordX] CALL Pos_Curs CMP M[CoordX], R1 BR.NZ Repete POP R1 RET</pre>	
<i>;Esta funcao escreve qualquer caracter numa dada posicao dada pelo</i>		
<i>;usando a rotina pos_curs como referencial para a posicao</i>		
<pre>;/-----/ ;/ Escreve Character / ;/-----/ Esc_Car: R2 MOV R2, M[SP+3] MOV M[IO_WRITE], R2 POP R2 RETN 1</pre>		
<i>;Reecebe uma coordenada X e Y, e' feito um shift ao Y left para que fiquem</i>		
<i>;todos os valores nos bits mais significativos nos bits menos significaticos</i>		
<i>;fica o valor de X. Esse valor e' injetado no cursor da janela de texto que fica</i>		
<i>;pronto a receber um caracter.</i>		
<pre>;/-----/ ;/ Posicao / ;/ do Cursor / ;/-----/ Pos_Curs: R1 PUSH R2 PUSH R3 MOV R1, M[CoordX] MOV R2, M[CoordY] SHL R3, R2 MVBL R3, R1 MOV M[IO_CONTR], R3 POP R3 POP R2 POP R1 RET</pre>		
<i>;Coloca o passaro nas suas coordenadas iniciais, para que se possa dar inicio ao</i>		
<pre>;/jogo ;/-----/ ;/ Posicao / ;/ do Passaro inicial / ;/-----/ Bird_Ini: R1 PUSH R4 PUSH R5 MOV R5, Bird MOV R4, M[R5] PUSH R4 MOV R1, Col_Ini MOV M[CoordX], R1 MOV R1, Lin_Ini MOV M[CoordY], R1</pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 8/17
<pre>CALL Pos_Curs CALL Esc_Car INC M[CoordX] CALL Pos_Curs MOV R4, M[R5+1] PUSH R4 CALL Esc_Car DEC M[CoordX] POP R5 POP R4 POP R1 RET</pre>		
<p><i>;Esta rotina comeca por apagar o passaro seguida a incrementacao do Y</i></p> <p><i>;do mesmo e depois desenha o passaro com as novas coordenadas</i></p> <p><i>;a rotina tambem verifica se existe uma colisao com algum objecto durante</i></p> <p><i>;o movimento do passaro.</i></p> <pre>;/-----/ ;/ Escreve / ;/ Passaro / ;/-----/ Bird_Pos: R1 PUSH R4 PUSH R5 PUSH Clear MOV R1, M[Bird_Y2] MOV M[CoordY], R1 MOV R1, Col_Ini MOV M[CoordX], R1 CALL Pos_Curs CALL Esc_Car DEC M[CoordX] MOV R5, Bird MOV R4, M[R5] PUSH R4 MOV R1, M[BirdY] MOV M[CoordY], R1 CALL Pos_Curs CALL Esc_Car INC M[CoordX] CALL Pos_Curs MOV R4, M[R5+1] PUSH R4 CALL Esc_Car DEC M[CoordX] CALL Busca_col POP R5 POP R4 POP R1 RET</pre>		
<p><i>;A gravidade 'e feita atraves do temporizador e de um contador(que aumenta ou di</i></p> <p><i>minui</i></p> <p><i>;a velocidade da gravidade), que a cada tick do temporizador o contador 'e decre</i></p> <p><i>mentado</i></p> <p><i>;e quando este chega a 0 a velocidade de gravidade (caso nao tenha sido dada a o</i></p> <p><i>rdem de subir)</i></p>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 9/17
<pre>;e' aumentada ;/-----GRAVIDADE/ ;/-----/ Gravidade: PUSH R1 MOV R1, M[Velocidade] CMP R1, Vel_Final BR.Z Retorno INC M[Velocidade] MOV R1, M[Velocidade] MOV R1, C_TEMP_GRAV MOV R1, M[TEMP_GRAVIDADE], R1 POP R1 RET Descer: PUSH R1 MOV R1, M[BirdY] CMP R1, Lim_Inf BR.NZ NotGameOver JMP EndScreen MOV M[Bird_Y2], R1 INC M[BirdY] CALL Bird_Pos MOV R1, 0003h MOV M[TEMP_GRAVIDADE], R1 DEC M[Gravil] DEC M[Estadescendo] POP R1 RET NotGameOver: MOV R1 MOV R1, 0003h MOV M[TEMP_GRAVIDADE], R1 DEC M[Gravil] DEC M[Estadescendo] POP R1 RET ;Rotina que incrementa o valor Y do passaro e atualiza a sua posicao ;verificando tambem se houve algum tipo de colisao com o limite superior ;/-----/ ;/ Subir / ;/-----/ Subir: PUSH R1 MOV R1, M[BirdY] CMP R1, R0 BR.NZ fim_jogo CALL fim_jogo MOV M[Bird_Y2], R1 DEC M[BirdY] CALL Bird_Pos INC M[Verifica] MOV M[Estadescendo], R0 MOV M[Velocidade], R0 MOV R1, C_TEMP_GRAV MOV M[TEMP_GRAVIDADE], R1 MOV M[Gravil], R0 POP R1 RET ;Funcao responsavel pelo movimento das colunas ao longo do jogo, faz o acesso ;da mesma maneira que a funcao anterior as posicoes das colunas usando uma ;variavel de controlo para que seja correspondido corretamente o valor random a tabela. ;Faz depois um movimento para a esquerda de todas as colunas. Quando a coluna at inge a ;coordenada x 0, o valor na tabela obstaculos e colocado a 0 de modo a permitir a</pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 10/17
<pre>;criacao de um novo obstaculo. ;E'tambem feito um teste de colisao com o passaro a cada movimento das colunas. ;/-----/ ;/ OBSTACULOS / ;/-----/ Move_Obst: PUSH R1 MOV R1, M[Jamovido], R0 CALL Clear_Obsta MOV R1, M[C_TEMP_OBSTA] CALL Bird_Pos MOV R1, M[TEMP_OBSTA], R1 INC M[Jamovido] MOV R1, M[Jamovido] CMP R1, M[Obts_Atuais] BR.NZ ContinuaMov DEC M[EspacoAtual] INC M[COLUMNAS] CALL LcDesc POP R1 RET Clear_Obsta: PUSH R1 MOV R2, Obstaculos MOV R2, M[Jamovido] ADD R1, M[R2] MOV R1, M[CoordX], R1 MOV R1, 0001h MOV R1, M[CoordY], R1 MOV R1, 0017h Clear_Again: PUSH Clear CALL Pos_Curs CALL Esc_Gar INC M[CoordY] CMP R1, M[CoordY], R1 BR.NZ Clear_Again POP R2 POP R1 RET Esc_Obs: R1 PUSH R2 PUSH R3 PUSH R4 MOV R4, Obstaculos ADD R4, M[Jamovido] MOV R1, M[R4] CMP R1, 0014h BR.NZ NoPoint INC M[PONTOS] CALL EscPONT MOV R1, R0 BR.NZ NotZero INC M[DiminuiObj] JMP Pops MOV R1 DEC R1 MOV M[FlagGameOver], R0 MOV M[CoordX], R1 CMP R1, M[BirdX] BR.NZ Nao_verifique INC M[FlagGameOver]</pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 11/17
Nao_verifique:	DEC	R1
	CMP	R1, M[BirdX]
	BR.NZ	Nao_verifique2
	INC	M[FlagGameOver]
Nao_verifique2: MOV	R1, 0000h	
	MOV	M[CoordY], R1
	MOV	R1, 0016h
Esc_Again:	PUSH	
	INC	M[CoordY]
	CALL	Pos_Curs
	CALL	Esc_Car
	MOV	R3, M[CoordY]
	CMP	M[FlagGameOver], R0
	BR.Z	NFimJogo
	CMP	R3, M[BirdY]
	BR.NZ	NFimJogo
	CALL	EndScreen
NFimJogo:	MOV	R2, TABRNG
	ADD	R2, M[Jamovido]
	CMP	R3, M[R2]
	BR.NZ	Esc_M
	ADD	R3, 0005h
	MOV	M[CoordY], R3
Esc_M:	CMP	M[CoordY], R1
	DEC	Esc_Again
	R4	M[R4]
Pops:	POP	R4
	POP	R3
	POP	R2
	POP	R1
	RET	
<pre> ;Rotina que cria uma nova fila de obstaculos. Esta faz uso de um gerador pseudo aleatorio ;para poder criar um buraco de 5 posicoes na coluna. ;As posicoes de todas as colunas sao guardadas numa tabela e o seu valor aleator io de onde ;se encontra o buraco noutra mas os valores (estando ambos no mesmo endereço rel ativamente a tabela) ;A rotina coloca as posicoes das colunas no primeiro endereco da tabela com ende reco 0. ; /-----/ ; / Cria Obstaculos ; /-----/ Cria_Obst: PUSH R1 PUSH R2 PUSH R3 CALL Busca_0 MOV R2, Obstaculos ADD R2, M[Indice] MOV R1, 004Ch MOV M[CoordX], R1 MOV M[R2], R1 MOV R1, 0000h MOV M[CoordY], R1 MOV R1, 0016h CALL GenRndPos Obsta INC M[CoordY] CALL Pos_Curs CALL Esc_Car MOV R3, M[CoordY] </pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 12/17
	<pre> MOV R2, TABRNG ADD R2, M[Indice] CMP R3, M[R2] BR.NZ n_esc ADD R3, 0005h MOV M[CoordY], R3 CMP M[CoordY], R1 BR.NZ Salta CMP M[DiminuiObj], R0 BR.NZ NoINC INC M[Obts_Atuais] MOV R1, Gap MOV M[DiminuiObj], R0 MOV M[EspacoAtual], R1 POP R3 POP R2 POP R1 RET </pre>	
n_esc:	<pre> CMP </pre>	
NoINC:	<pre> MOV </pre>	
	<pre> ;Rotina que preenche o ecrã que prepara o jogador para o jogo, ;faz uso do cursor de janela. Após a execucao desta rotina o programa ;fica a espera que seja premido o botao II. ;-----/ ;/ Splash Screen ;/-----/ Splash_Screen: PUSH R5 </pre>	
Continua:	<pre> CALL Pos_Curs INC Esc_Car INC M[CoordX] INC R5 MOV R4, M[R5] MOV R1, Titu_X MOV M[CoordX], R1 MOV R1, Titu_Y MOV M[CoordY], R1 PUSH R4 Pos_Curs CALL Esc_Car INC M[CoordX] INC R5 MOV R4, M[R5] CMP R4, Fim_Linha BR.Z Segunda_Linha PUSH R4 Segunda_Linha BR Continua M[CoordY] INC MOV R5, Line2 MOV R4, M[R5] MOV R1, Titu_X2 MOV M[CoordX], R1 PUSH R4 Pos_Curs CALL Esc_Car INC M[CoordX] INC R5 MOV R4, M[R5] CMP R4, Fim_Linha PUSH R4 Continua2 BR.NZ POP R4 POP R4 POP R5 RET </pre>	
Continua2:	<pre> CALL </pre>	

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.as	Page 13/17
<pre> ;Rotina que gera um numero pseudo aleatorio para criar o buraco ;na coluna de obstaculos ; /-----/ ; / Generate Random ; /-----/ GenRndPos: PUSH R1 PUSH R2 CALL GenRndNum MOV R2, M[NUMRANDOM] MOV R1, 0011h DIV R2, R1 INC R1 MOV R2, TABRNG ADD R2, M[Indice] MOV M[R2], R1 POP R2 POP R1 RET GenRndNum: PUSH R1 MOV R1, M[NUMRANDOM] TEST R1, 0001h BR.NZ NoXor XOR R1, RANDOMASK ROR R1, 0001h MOV M[NUMRANDOM], R1 POP R1 RET NoXor: ROR R1, 0001h MOV M[NUMRANDOM], R1 POP R1 RET ;Esta rotina vai procurar a primeira posicao numa tabela que tenha como valor ;0000h e devolver a sua posicao relativa a primeira posicao/ ; /-----/ ; / Procura zeros ; /-----/ Busca_0: PUSH R1 MOV R2, M[Indice], R0 MOV R1, M[Indice] MOV M[R2], R0 BR.Z Acaba INC R1 INC R2 BR Repeat MOV M[Indice], R1 POP R1 RET Acaba: Repeat: CMP R1, R2 BR.Z Acaba INC R1 INC R2 BR Repeat MOV M[Indice], R1 POP R1 RET ;Escreve a pontuacao no display de 7 segmentos, para tal devemos dividir a pontu acao em ;4 algarismo diferentes, divindo o numero por 10 e ficando com o resto, esse va lor e' ;injetado na posicao correspondente do display. ; /-----/ ; / Escreve Pontuacao ; /-----/ ESCPONT: PUSH R1 PUSH R2 PUSH R3 PUSH R4 </pre>		

dez 02, 15 19:56	C:\Users\tiago\Desktop\p3print_win\1.a.s	Page 14/17
Refaz:	<pre>MOV R4, 0004h MOV R1, M[PONTOS] MOV R2, IO_DISPLAY MOV R3, 000Ah DIV R1, R3 MOV M[R2], R3 INC R2 DEC R4 BR.NZ Refaz POP R4 POP R3 POP R2 POP R1 RET</pre>	
	<pre>;Rotina que coloca a distancia (que e' contada a cada mudance de posicao do pass aro), ; LCD 'e praticamente da mesma maneira que a janela de texto, colocando apenas no bit mais significativo ; o valor 1 para que seja tivada essa possicao. ; /-----/ ; / Escreve LCD INI / ; /-----/ LCDINI: PUSH R1 PUSH R2 PUSH R3 PUSH R4 PUSH R5 MOV R1, IO_LCD_CTRL MOV R2, 8000h MOV M[R1], R2 MOV R5, IO_LCD_CAR MOV R3, Str1 MOV R4, M[R3] MOV R4, Fim_Linha FimEscreveCol MOV M[R5], R4 INC R3 INC R2 MOV M[R1], R2 Cicle R5 POP R4 POP R3 POP R2 POP R1 RET</pre>	
	<pre>FimEscreveCol: POP R5 ;Rotina que coloca a distancia (que e' contada a cada mudance de posicao do pass aro), ; LCD 'e praticamente da mesma maneira que a janela de texto, colocando apenas no bit mais significativo ; o valor 1 para que seja tivada essa possicao. Depois tem-se de dividir o numero de colunas em 4 caracteres ;para isso faz-se a divisao por 10, e guardamos o resto, adicionamos 0003h que converte qualquer alagarismo ;num caracter ASCII e fazemos inject para o LCD, faz-se isto para todos os alaga rimos do numero de colunas. ; /-----/ ; / Escreve LCD Colunas / ; /-----/ LCDEsc: PUSH R1</pre>	

dez 02, 15 19:56	C:\Users\Tiago\Desktop\p3print_win\1.as	Page 15/17
<pre> PUSH R2 PUSH R3 PUSH R4 PUSH R5 MOV R1, IO_LCD_CTRL MOV R2, 800Fh MOV R5, IO_LCD_CAR MOV R3, M[COLUMNAS] MOV M[R1], R2 MOV R4, 000Ah MOV R3, R4 ADD R4, 0030h ;Converte qualquer alg arismo para codificacao ASCII MOV M[R5], R4 DEC R2 CMP R2, 800Ah BR.NZ LCDCic1 POP R5 POP R4 POP R3 POP R2 POP R1 RET ;Rotina que apaga todo o conteudo que se encontra na janela de texto. ; /-----/ ; / Apaga Ecra / ; /-----/ ClrScr: PUSH R1 PUSH R2 PUSH R3 MOV R3,Comp_Lin MOV M[CoordX], R0 MOV M[CoordY], R0 Clear CALL Pos_Curs CALL Esc_Car INC M[CoordX], R3 BR.NZ Ciclo INC M[CoordY] MOV M[CoordX], R0 MOV R1, LINHA_FIM CMP M[CoordY], R1 BR.NZ Ciclo POP R3 POP R2 POP R1 RET ;Rotina chamada sempre que o passaro atingir algum obstaculo, faz um clear scree n ;seguida da colocacao de uma mensagem de fim de jogo e a respetiva pontuacao. ; /-----/ ; / END SCREEN / ; /-----/ EndScreen: PUSH R1 PUSH R5 PUSH R4 DSI ClrScr CALL</pre>		

dez 02, 15 19:56	C:\Users\Tiago\Desktop\p3print_win\1.as	Page 16/17
<pre> MOV M[TimerControl], R0 MOV R5, EndLine1 MOV R4, M[R5] PUSH R4 MOV R1, Titu_X MOV M[CoordX], R1 MOV R1, Titu_Y MOV M[CoordY], R1 CALL Pos_Curs CALL Esc_Car INC M[CoordX] INC R5 MOV R4, M[R5] CMP R4, Fim_Linha BR.Z Pont PUSH R4 BR Continua3 INC M[CoordY] INC M[CoordY] MOV R1, 002Ah MOV M[CoordX], R1 MOV R1, 0004h MOV R5, M[PONTOS] MOV R4, 000Ah DIV R5, R4 ADD R4, 0030h ; converte qualquer al agarismo para codificacao ASCII PUSH R4 CALL Pos_Curs CALL Esc_Car DEC M[CoordX] DEC R1 BR.NZ Continua4 POP R4 POP R4 POP R5 POP R1 BR Fim Fim: BR ;Verifica apos cada mnovimentacao do passaro se este esta a colidir com algum ob jecto. ;Faz uso da tabela que contem os valores random de cada obstaculo e da tabela qu e contem a posicao de cada obstaculo ;verifica se ha algum obstaculo na coluna do passaro e verifica se o passaro se encontra no buraco criado pelo ;numero aleatorio se nao se encontrar o jogo acaba. ; /-----/ ; / Procura colisao (bird) / ; /-----/ Busca_col: PUSH R1 PUSH R2 PUSH R3 PUSH R4 MOV M[FlagCol], R0 MOV R1, TABRNG MOV R2, Obstaculos MOV R3, 0014h MOV M[IndiceColisao], R0 MOV R4, M[Obts_Atuais] CMP M[R2], R3 BR.Z VerificaY INC R2</pre>		

dez 02, 15 19:56	C:\Users\Tiago\Desktop\p3print_win\1.as	Page 17/17
	<div>INC M[IndexColisao] CMP M[IndexColisao], R4 BR.NZ VerificaX MOV M[IndexColisao], R0 MOV R2, Obstaculos INC R3 CMP R3, 0016h BR.NZ VerificaX BR NtGameOver MOV R3, M[BirdY] MOV R4, M[IndexColisao] MOV R2, TABRNG ADD R2, R4 MOV R4, M[R2] INC R4 MOV R1, 0005h CMP R3, R4 Verificanext BR NtGameOver DEC R1 BR.NZ Reverifica CALL EndScreen R4 POP R3 POP R2 POP R1 RET</div>	
VerificaY:	MOV	
Reverifica:	CMP	
Verificanext:	INC R4	
NtGameOver:	POP	