

Dec 05, 13 18:12	jogo_final.as	Page 1/21
<pre> //////////////////// ; CONSTANTES //////////////////// ;CONSTANTES DE CONTROLO IO_DISPLAY EQU FFF0h IO_READ EQU FFFFh IO_WRITE EQU FFFEh IO_CONTROL EQU FFFCh SP_INICIAL EQU FDFFh POS_INI EQU 001Ch POS_LIME EQU 001Eh POS_LIMD EQU 0035h POS_BIC EQU 152Ah ;Guardada em R5 POS_INI_BIC EQU 152Ah INT_MASK_ADDR EQU FFFAh INT_MASK EQU 1000000000000010b INT_MASKCICLO EQU 10001100000000101b ;Máscara de interrupções usada d urante o ciclo de jogo INT_MASKPAUSA EQU 00000100000000000b ;Máscara de interrupções usada d urante a pausa MASK_COLUNA EQU 00FFh ;Selecciona apenas a coluna MASK_LINHA EQU FF00h ;Selecciona apenas a linha MASK_NIB_ESQ EQU F000h MASK_ALEAT EQU 1000000000010110b LCD_CONTROL EQU FFF4h LCD_WRITE EQU FFF5h DURACAO EQU FFF6h ACTIVA_TEMP EQU FFF7h LED_CONTROL EQU FFF8h POS_BEMVINDO EQU 0918h POS_BEMVINDO2 EQU 0925h POS_OBS_INI EQU 002Fh APAGA_LCD EQU 8020h LCD_LINHA1 EQU 8000h LCD_LINHA2 EQU 8010h UM_NIBBLE EQU 4 DOIS_NIBBLES EQU 8 VINTE_QUATR_COL EQU 24 ;CONSTANTES AUXILIARES FIM_TEXTO EQU '@' ;Caracter que indica fim de texto LINHA_SEIS EQU 0600h LINHA_FIM EQU 0A00h LINHA_BIKE EQU 1500h NUM_NIBBLES EQU 4 ;Numero de nibbles de cada palavra de 16 bits QUATRO_LEDS EQU F000h ;F000h aponta para os 4 leds mais a esquerda OITO_LEDS EQU FF00h ;FF00h aponta para os 8 leds mais a esquerda DOZE_LEDS EQU FFF0h ;FFF0h aponta para os 12 leds mais a esquerda DEZASSEIS_LEDS EQU FFFFh ;FFFFh aponta para os 16 leds mais a esquerda LCD_L2_COL8 EQU 8018h LCD_L1_COL11 EQU 800Bh DUAS_LINHAS EQU 0200h UMA_LINHA EQU 0100h VINTE_OITO_COL EQU 001Ch NUM_ESPACO_APAG EQU 38 ;Numero de espacos necessario para apagar as men sagens. LARGURA_PISTA EQU 21 DENTRO_LIM EQU 30 //////////////////// ; STRINGS //////////////////// </pre>		

Dec 05, 13 18:12	jogo_final.as	Page 2/21
<pre> //////////////////// ; VARIAVEIS //////////////////// MSG1 STR 8000h MSG2 STR 'Bem-Vindo a Corrida de Bicicleta!',FIM_TEXTO MSG2_1 STR 'Prima o interruptor I1 para comecar.',FIM_TEXTO MSG2_2 STR 'Fim do Jogo',FIM_TEXTO MSG2_2 STR 'Prima o interruptor I1 para recomecar.',FIM_TEXTO LIM_ESQ STR '+ ',FIM_TEXTO LIM_DIR STR ' +',FIM_TEXTO ESPACO STR ' ',FIM_TEXTO BICICLETA STR 'O O',FIM_TEXTO OBSTACULO STR '***',FIM_TEXTO ESPACO2 STR ' ',FIM_TEXTO LCD_1 STR 'Distancia:00000m',FIM_TEXTO LCD_2 STR 'Maximo:00000m',FIM_TEXTO LCD_PAUSA STR 'PAUSA',FIM_TEXTO ; NIVEL NIVEL WORD 1 POS_OBS1 WORD 0000h POS_OBS2 WORD 0000h POS_OBS3 WORD 0000h POS_OBS4 WORD 0000h ESPACO_ENTREOBS WORD 5 MOVE_OBS WORD 0 ANDA_DIR WORD 0 ANDA_ESQ WORD 0 RANDOM WORD 0000h POS_RANDOM WORD 0000h N_OBS WORD 4 N_PARTES WORD 3 N_PARTES_OBS WORD 3 VELOCIDADE WORD 5 VEZES_JOGADAS WORD 0 DISTANCIA WORD 0 MAXIMO WORD 0 ESTA_PAUSA WORD 0 ESTA_TURBO WORD 0 PAUSA WORD 0 TURBO WORD 0 CONTADOR WORD 0000h LCD_TEMPOR WORD 0000h LCD_CONTROL_NUM WORD 800Bh LCD_CONTROL_REC WORD 8018h ; TABELA DE INTERRUPCOES ; ; TABELA DE INTERRUPCOES ; INT0 ORIG FE00h WORD MovEsquerda INT1 ORIG FE01h WORD CicloJogo INTB ORIG FE0Bh WORD MovDireita </pre>		

Dec 05, 13 18:12	jogo_final.as	Page 3/21
INT16	ORIG FE0Fh WORD Temp	
INTA	ORIG FE0Ah WORD Poe_Pausa	
INT2	ORIG FE02h WORD Poe_Turbo	
<pre> //////////////////// ; CÓDIGO ; //////////////////// //////////////////// ; INICIO DO JOGO ; //////////////////// Inicio: ORIG 0000h DSI MOV R7,SP_INICIAL MOV SP,R7 MOV R7,INT_MASK MOV M[INT_MASK_ADDR],R7 MOV M[ACTIVA_TEMP],R0 CMP M[VEZES_JOGADAS],R0 ;Se ja se tiver jogado, JMP.NZ BemVindoAgain ;escreve-se uma mensagem diferen te CALL BemVindo BR SaltaInicio BemVindoAgain: CALL BemVindo2 SaltaInicio: CALL Apaga_bic ENI MOV R5,POS_BIC MOV M[ACTIVA_TEMP],R0 CALL ApagaObs ;Loop que verifica os pedidos de interrupcao, isto é, ;os movimentos da bicicleta e dos obstaculos, ;e o estado de turbo e pausa Continua: INC M[RANDOM] CMP M[PAUSA],R0 CALL.NZ Pausa CMP M[TURBO],R0 CALL.NZ Turbo CMP M[MOVE_OBS],R0 CALL.NZ RotinaObstac CMP M[ANDA_DIR],R0 CALL.NZ MovDir CMP M[ANDA_ESQ],R0 CALL.NZ MovEsq BR Continua //////////////////// ; CICLO DO JOGO ; //////////////////// ;Rotina que prepara o jogo, incluindo a pista, o display ;e o LCD, a bicicleta, a posicao dos obstaculos, e a </pre>		

Dec 05, 13 18:12	jogo_final.as	Page 4/21
;velocidade inicial. Inicia a pilha.		
CicloJogo:	DSI	
	PUSH R1	
	PUSH R2	
	MOV R1,M[VELOCIDADE]	
	MOV M[DURACAO],R1	
	MOV R1,1	
	MOV M[ACTIVA_TEMP],R1	
	CALL ApagaMsg	
	MOV R5,POS_INI_BIC	
	CALL Escreve_espaco	
	CALL Escreve_bic	
	CALL RandomizePos ;Aleatoriza a posição dos obstáculos	
	MOV R2,QUATRO_LEDS	
	MOV M[LED_CONTROL],R2	
	CALL EscCont ;Inicia o contador de obstáculos	
	PUSH R1	
	PUSH R2	
	MOV R1,LCD_1	
	MOV R2,LCD_LINHA1	
	CALL EscreveLcd	
	MOV R1,LCD_2	
	MOV R2,LCD_LINHA2	
	CALL EscreveLcd	
	CALL RecordeLcd	
	POP R2	
	POP R1	
	ENI	
	MOV R7,INT_MASKCICLO	
	MOV M[INT_MASK_ADDR],R7	
	POP R2	
	POP R1	
	RTI	
<pre> //////////////////// ; TEMPORIZADOR ; //////////////////// ;Rotina que, a cada ciclo do temporizador, pede o avanco dos obstaculos. Temp: PUSH R1 MOV R1,1 MOV M[MOVE_OBS],R1 MOV R1,M[VELOCIDADE] MOV M[DURACAO],R1 MOV R1,1 MOV M[ACTIVA_TEMP],R1 POP R1 RTI ;Rotina que escreve a mensagem de boas vindas BemVindo: PUSH R1 PUSH R2 MOV R1,Msg1 MOV R6,IO_READ MOV M[IO_CONTROL],R6 MOV R6,POS_BEMVINDO MOV M[IO_CONTROL],R6 BemVindoCiclo1: MOV R2,M[R1] CMP R2,FIM_TEXTO BR.Z MudaDeLinha </pre>		

Dec 05, 13 18:12	jogo_final.as	Page 5/21
	<pre> MOV M[IO_WRITE],R2 INC R6 MOV M[IO_CONTROL],R6 INC R1 BR BemVindoCiclo1 MudaDeLinha: MOV R6,POS_BEMVINDO ADD R6,DUAS_LINHAS ;Avanca 2 linhas para escrever a segunda <i>parte da mensagem de boas vindas</i> SUB R6,1 MOV M[IO_CONTROL],R6 MOV R1,Msg2 BemVindoCiclo2: MOV R2,M[R1] CMP R2,FIM_TEXTO BR.Z AcabaMsg MOV M[IO_WRITE],R2 INC R6 MOV M[IO_CONTROL],R6 INC R1 BR BemVindoCiclo2 AcabaMsg: POP R2 POP R1 RET ;Rotina que apaga a mensagem de boas vindas ApagaMsg: PUSH R7 PUSH R3 PUSH R2 MOV R7,Espaco MOV R3,POS_BEMVINDO MOV M[IO_CONTROL],R3 MOV R2,NUM_ESPACO_APAG CALL ProxApagaBV BR MudaLinhaApaga ProxApagaBV: MOV R6,M[R7] MOV M[IO_WRITE],R6 INC R3 MOV M[IO_CONTROL],R3 DEC R2 BR.NZ ProxApagaBV RET MudaLinhaApaga: MOV R3,POS_BEMVINDO ADD R3,DUAS_LINHAS SUB R3,1 ;Recua o cursor uma coluna MOV M[IO_CONTROL],R3 MOV R7,Espaco MOV R2,NUM_ESPACO_APAG CALL ProxApagaBV SaiApagaBV: POP R2 POP R3 POP R7 RET ;Rotina que escreve a mensagem de fim de jogo BemVindo2: PUSH R1 PUSH R2 MOV R1,Msg2_1 MOV R6,IO_READ </pre>	

Dec 05, 13 18:12	jogo_final.as	Page 6/21
	<pre> MOV M[IO_CONTROL],R6 MOV R6,POS_BEMVINDO2 MOV M[IO_CONTROL],R6 BemVindoCic1: MOV R2,M[R1] CMP R2,FIM_TEXTO BR.Z MudaDeLinha1_1 MOV M[IO_WRITE],R2 INC R6 MOV M[IO_CONTROL],R6 INC R1 BR BemVindoCic1 MudaDeLinha1_1: MOV R6,POS_BEMVINDO ADD R6,DUAS_LINHAS SUB R6,1 ;Recua o cursor uma coluna MOV M[IO_CONTROL],R6 MOV R1,Msg2_2 BemVindoCic2: MOV R2,M[R1] CMP R2,FIM_TEXTO BR.Z AcabaMsg2 MOV M[IO_WRITE],R2 INC R6 MOV M[IO_CONTROL],R6 INC R1 BR BemVindoCic2 AcabaMsg2: POP R2 POP R1 RET ;Rotina que apaga as mensagens de fim de jogo ApagaMsg2: PUSH R7 PUSH R3 PUSH R2 MOV R7,Espaco MOV R3,POS_BEMVINDO MOV M[IO_CONTROL],R3 MOV R2,NUM_ESPACO_APAG CALL ProxApagaBV2 BR MudaLinhaApaga2 ProxApagaBV2: MOV R6,M[R7] MOV M[IO_WRITE],R6 INC R3 MOV M[IO_CONTROL],R3 DEC R2 BR.NZ ProxApagaBV2 RET MudaLinhaApaga2: MOV R3,POS_BEMVINDO ADD R3,DUAS_LINHAS SUB R3,1 MOV M[IO_CONTROL],R3 MOV R7,Espaco MOV R2,NUM_ESPACO_APAG CALL ProxApagaBV2 SaiApagaBV2: POP R2 POP R3 POP R7 RET ;ESCREVE O ESPAÇO DO JOGO Escreve_espaco: PUSH R7 </pre>	

Dec 05, 13 18:12	jogo_final.as	Page 7/21
Escreve_linha:	<pre> MOV R2,IO_READ MOV M[IO_CONTROL],R2 MOV R2,POS_INI MOV M[IO_CONTROL],R2 MOV R3,VINTE_QUATR_COL CALL Escr_lim_esq CALL Muda_coluna CALL Escr_lim_dir CALL Muda_linha DEC R3 BR.NZ Escreve_linha POP R7 RET </pre>	
Acaba:		
;	APAGA O ESPAÇO DO JOGO	
Apaga_espaco:	<pre> PUSH R7 MOV R2,IO_READ MOV M[IO_CONTROL],R2 MOV R2,POS_INI MOV M[IO_CONTROL],R2 MOV R3,VINTE_QUATR_COL CALL Apaga_lim_esq CALL Muda_coluna CALL Apaga_lim_dir CALL Muda_linha DEC R3 BR.NZ Apaga_linha POP R7 RET </pre>	
Apaga_linha:		
Acaba_Apaga:		
;	Rotina que apaga o limite esquerdo da pista	
Apaga_lim_esq:	<pre> PUSH R6 PUSH R7 MOV R7,Espaco2 MOV R6,M[R7] CMP R6,FIM_TEXTO BR.Z SaiApagaE MOV M[IO_WRITE],R6 INC R2 MOV M[IO_CONTROL],R2 INC R7 BR ProxApagaEsq POP R7 POP R6 RET </pre>	
ProxApagaEsq:		
SaiApagaE:		
;	Rotina que apaga o limite direito da pista	
Apaga_lim_dir:	<pre> PUSH R6 PUSH R7 MOV R7,Espaco2 MOV R6,M[R7] CMP R6,FIM_TEXTO BR.Z SaiApagaD MOV M[IO_WRITE],R6 INC R2 MOV M[IO_CONTROL],R2 INC R7 BR ProxApagaDir POP R7 POP R6 RET </pre>	
ProxApagaDir:		
SaiApagaD:		

Dec 05, 13 18:12	jogo_final.as	Page 8/21
Muda_linha:	<pre> ;Rotina que muda de linha, em que 001Ch (28d) é o numero de colunas a recuar, ;e 0100h é o numero de linhas a avançar (significa adicionar 1 linha) SUB R2,VINTE_OITO_COL ADD R2,UMA_LINHA MOV M[IO_CONTROL],R2 RET </pre>	
Muda_coluna:	<pre> ADD R2,VINTE_QUATR_COL MOV M[IO_CONTROL],R2 RET </pre>	
;	Rotina que escreve o limite esquerdo da pista	
;	Recebe em R2 a posição do cursor	
Escr_lim_esq:	<pre> PUSH R6 PUSH R7 MOV R7,Lim_esq MOV R6,M[R7] CMP R6,FIM_TEXTO BR.Z SaiE MOV M[IO_WRITE],R6 INC R2 MOV M[IO_CONTROL],R2 INC R7 BR ProxEsq POP R7 POP R6 RET </pre>	
ProxEsq:		
SaiE:		
;	Rotina que escreve o limite direito da pista	
;	Recebe em R2 a posição do cursor	
Escr_lim_dir:	<pre> PUSH R6 PUSH R7 MOV R7,Lim_dir MOV R6,M[R7] CMP R6,FIM_TEXTO BR.Z Said MOV M[IO_WRITE],R6 INC R2 MOV M[IO_CONTROL],R2 INC R7 BR ProxDir POP R7 POP R6 RET </pre>	
ProxDir:		
Said:		
;	//////////	
;	BICICLETA	
;	//////////	
;	Rotina que coloca a variável ANDA_ESQ a 1 quando o botão de	
;	interrupção IO é premido	
MovEsquerda:	<pre> PUSH R1 MOV R1,1 MOV M[ANDA_ESQ],R1 POP R1 RTI </pre>	
;	Rotina que coloca a variável ANDA_DIR a 1 quando o botão de	
;	interrupção IB é premido	

Dec 05, 13 18:12	jogo_final.as	Page 9/21
MovDireita:	PUSH R1 MOV R1,1 MOV M[ANDA_DIR],R1 POP R1 RTI	
;Rotina que escreve a bicicleta		
;Recebe em R5 a posição do cursor		
Escreve_bic:	PUSH R7 PUSH R5 MOV R7,Bicicleta MOV M[IO_CONTROL],R5 MOV R6,M[R7] CMP R6,FIM_TEXTO BR.Z SaiBic MOV M[IO_WRITE],R6 ADD R5,UMA_LINHA MOV M[IO_CONTROL],R5 INC R7 BR ProxBic SaiBic: POP R5 POP R7 RET	
;Rotina que apaga a bicicleta		
;Recebe em R5 a posição do cursor		
Apaga_bic:	PUSH R7 PUSH R5 MOV R7,Espaco MOV M[IO_CONTROL],R5 MOV R6,M[R7] CMP R6,FIM_TEXTO BR.Z SaiApag MOV M[IO_WRITE],R6 ADD R5,UMA_LINHA MOV M[IO_CONTROL],R5 INC R7 BR ProxApag SaiApag: POP R5 POP R7 RET	
;CONTROLO DA BICICLETA		
;Rotina que move a bicicleta para a esquerda		
MovEsq:	PUSH R5 AND R5,MASK_COLUNA CMP R5,POS_LIME POP R5 BR.Z FimEsq CALL Apaga_bic SUB R5,1 CALL Colisao CALL Escreve_bic FimEsq: MOV M[ANDA_ESQ],R0 RET	
;Rotina que move a bicicleta para a direita		

Dec 05, 13 18:12	jogo_final.as	Page 10/21
MovDir:	PUSH R5 AND R5,MASK_COLUNA CMP R5,POS_LIMD POP R5 BR.Z FimDir CALL Apaga_bic ADD R5,1 CALL Colisao CALL Escreve_bic FimDir: MOV M[ANDA_DIR],R0 RET	
/////////////////////////////////// ; OBSTÁCULOS ; ///////////////////////////////////		
;Rotina que apaga todos os obstáculos, move-os, verifica se há colisão, e escreve-os caso não haja. É incrementada a distância e essa distância é actualizada no LCD		
RotinaObstac:	CALL ApagaObs CALL MoveObs CALL Colisao CALL EscreveObs INC M[DISTANCIA] CALL IncrementaLcd MOV M[MOVE_OBS],R0 RET	
;Rotina que gera um número aleatório de acordo com o algoritmo dado no enunciado.		
;Recebe um número gerado pela constante incrementação		
;da variável RANDOM		
;Retorna uma variável POS_RANDOM		
Aleat:	PUSH R4 MOV R4,M[RANDOM] TEST R4,1 BR.NZ Aleat_XOR ROR R4,1 BR FimAleat	
Aleat_XOR:	XOR R4,MASK_ALEAT ROR R4,1	
FimAleat:	PUSH R5 MOV M[RANDOM],R4 MOV R5,R4 MOV R4,LARGURA_PISTA DIV R5,R4 ADD R4,DENTRO_LIM MOV M[POS_RANDOM],R4 POP R5 POP R4 RET	;Divide-se o número por 21, para que a posição aleatória que fica em R4 seja um número compreendido ent re 21 (largura da pista) ;Adiciona-se 30 para que a posiç ão esteja entre a posição dos limites da pista
;Rotina que gera uma posição aleatória para todos os obstáculos,		

Dec 05, 13 18:12	jogo_final.as	Page 11/21
;a partir da variavel POS_RANDOM que é recebida		
RandomizePos:	PUSH R1	
	PUSH R2	
	PUSH R3	
LoopRandom:	MOV R2,M[N_OBS]	
	MOV R3,POS_OBS1	
	CALL Aleat	
	MOV R1,M[POS_RANDOM]	
	MOV M[R3],R1	
	INC R3	
	DEC R2	
	BR.NZ LoopRandom	;0 ciclo é repetido para todos os 4 obst
áculos		
	POP R3	
	POP R2	
	POP R1	
	RET	
;Rotina que gera uma posição aleatória para apenas um obstáculo,		
;a partir da variável POS_RANDOM		
RandomlObs:	PUSH R1	
	PUSH R2	
	MOV R1,POS_OBS4	
	CALL Aleat	
	MOV R2,M[POS_RANDOM]	
	MOV M[R1],R2	
	POP R2	
	POP R1	
	RET	
;Rotina que escreve um obstáculo		
;Recebe em R4 a posição do obstáculo		
EscreveObstac:	PUSH R7	
	PUSH R6	
	PUSH R4	
	MOV R7,Obstaculo	
	MOV M[IO_CONTROL],R4	
ProxObs:	MOV R6,M[R7]	
	CMP R6,FIM_TEXTO	
	BR.Z SaiObs	
	MOV M[IO_WRITE],R6	
	INC R4	
	MOV M[IO_CONTROL],R4	
	INC R7	
	BR ProxObs	
SaiObs:	POP R4	
	POP R6	
	POP R7	
	RET	
;Rotina que apaga um obstáculo		
;Recebe em R4 a posição do obstáculo		
ApagaObstac:	DSI	
	PUSH R7	
	PUSH R6	
	PUSH R4	
	MOV R7,Espaco	
	MOV M[IO_CONTROL],R4	

Dec 05, 13 18:12	jogo_final.as	Page 12/21
ProxApagaObs:	MOV R6,M[R7]	
	CMP R6,FIM_TEXTO	
	BR.Z SaiApagaObs	
	MOV M[IO_WRITE],R6	
	INC R4	
	MOV M[IO_CONTROL],R4	
	INC R7	
	BR ProxApagaObs	
SaiApagaObs:	POP R4	
	POP R6	
	POP R7	
	ENI	
	RET	
;Rotina que move todos os 4 obstáculos		
MoveObs:	PUSH R4	
	PUSH R2	
	PUSH R1	
	MOV R4,POS_OBS1	
	MOV R1,M[N_OBS]	
LoopMovObs:	MOV R2,UMA_LINHA	
	ADD M[R4],R2	
	PUSH R4	
	PUSH R3	
	MOV R3,M[R4]	
	AND R3,MASK_LINHA;	
	SHR R3,DOIS_NIBBLES ;Faz shift de modo a que os nibble da li	
nha estejam à direita	CMP R3,7	;Compara a posição do obstáculo da frent
e com a 7ª linha	POP R3	
	POP R4	
	BR.N SaiMovOb	
	PUSH R3	
	PUSH R4	
	MOV R3,M[R4]	
	AND R3,MASK_LINHA	
	SHR R3,DOIS_NIBBLES ;Faz shift de modo a que os nibble da li	
nha estejam à direita	CMP R3,24	;Compara a posição do obstáculo da frent
e com a última linha	POP R4	
	POP R3	
	BR.NZ SaltaGera	
	CALL GeraNova	
	BR LoopMovObs	
SaltaGera:	INC R4	
	DEC R1	
	BR.NZ LoopMovObs	
SaiMovOb:	POP R1	
	POP R2	
	POP R4	
	RET	
;Rotina que gera uma nova posição para o obstáculo que desapareceu		
;São feitas trocas para que o obstáculo que desapareceu seja agora representado		
pelo		
;último obstáculo, e que o obstáculo que vinha atrás deste seja agora o obstácul		
o 1		
GeraNova:	PUSH R1	

Dec 05, 13 18:12	jogo_final.as	Page 13/21
	<pre> PUSH R2 PUSH R3 PUSH R4 MOV R1,M[POS_OBS1] MOV R2,M[POS_OBS2] MOV R3,M[POS_OBS3] MOV R4,M[POS_OBS4] XCH R1,R2 XCH R2,R3 XCH R3,R4 MOV M[POS_OBS1],R1 MOV M[POS_OBS2],R2 MOV M[POS_OBS3],R3 MOV M[POS_OBS4],R4 CALL Random1Obs INC M[CONTADOR] CALL EscCont CALL Nivel POP R4 POP R3 POP R2 POP R1 RET ;Rotina que escreve os obstáculos por ordem, isto é, ;OBS1, OBS2, OBS3 e OBS4, que são, respectivamente, ;do obstáculo da frente até ao de trás ;Apenas aparecem novos obstáculos quando o da frente tiver ;passado a 7ª linha, de modo a que haja um intervalo de 5 espaços ;entre obstáculos consecutivos EscreveObs: PUSH R4 PUSH R2 PUSH R1 MOV R2,POS_OBS1 MOV R1,M[N_OBS] LoopEscObs: MOV R4,M[R2] PUSH R4 AND R4,MASK_LINHA SHR R4,DOIS_NIBBLES CMP R4,23 ;Compara a posição do obstáculo com a linha da b icicleta POP R4 BR.P NEscreveObs CALL EscreveObstac PUSH R4 AND R4,MASK_LINHA SHR R4,DOIS_NIBBLES CMP R4,7 ;Compara a posição do obstáculo com a linha 7 POP R4 BR.N SaiEscObs NEscreveObs: INC R2 DEC R1 BR.NZ LoopEscObs SaiEscObs: POP R1 POP R2 POP R4 RET ;Rotina que apaga todos os 4 obstáculos, por ordem ApagaObs: PUSH R4 PUSH R2 PUSH R1 </pre>	

Dec 05, 13 18:12	jogo_final.as	Page 14/21
	<pre> MOV R2,POS_OBS1 MOV R1,M[N_OBS] LoopApagaObs: MOV R4,M[R2] CALL ApagaObstac INC R2 DEC R1 BR.NZ LoopApagaObs SaiApagaOb: POP R1 POP R2 POP R4 RET //////////////////// ; COLISÃO //////////////////// ;Rotina que verifica a colisão do obstáculo que está na linha da bicicleta ;com a bicicleta Colisao: PUSH R1 PUSH R2 PUSH R3 PUSH R5 MOV R3,M[N_OBS] MOV R2,3 MOV R1,POS_OBS1 LoopColisao1: PUSH R1 PUSH R2 PUSH R3 PUSH R4 MOV R4,M[R1] MOV R2,MASK_LINHA MOV R3,LINHA_BIKE AND R4,R2 CMP R4,R3 POP R4 POP R3 POP R2 POP R1 BR.N SaiColisao ;É verificada a colisão com todas a 3 partes do obstáculo, e para cada uma dessa s ;partes são verificadas todas as partes da bicicleta VerificaCol: CMP M[R1],R5 JMP.Z FimJogo ADD R5,UMA_LINHA DEC M[N_PARTES] BR.NZ VerificaCol ;Verifica-se a colisao na parte esquerda do obstaculo com todas as 3 partes da bicicleta o obstaculo INC M[R1] ;Verificam-se as outras partes d MOV M[N_PARTES],R2 MOV R5,M[SP+1] DEC M[N_PARTES_OBS] JMP.NZ VerificaCol MOV M[N_PARTES],R2 SUB M[R1],R2 INC R1 ;Verifica-se o proximo obstaculo </pre>	

Dec 05, 13 18:12	jogo_final.as	Page 17/21
	<pre> PUSH R1 PUSH R2 PUSH R3 PUSH R6 PUSH R7 MOV R1,M[DISTANCIA] CALL JuntaNum MOV R6,4 MOV R7,R3 AND R7,MASK_NIB_ESQ ROL R7,4 ADD R7,'0' ;Transforma o numero em ascii MOV M[LCD_TEMPOR],R7 MOV R1,LCD_TEMPOR MOV R2,M[LCD_CONTROL_NUM] CALL EscreveLcdNum INC M[LCD_CONTROL_NUM] SHL R3,4 DEC R6 BR.NZ LoopIncLcd MOV R7,LCD_L1_COL11 MOV M[LCD_CONTROL_NUM],R7 POP R7 POP R6 POP R3 POP R2 POP R1 ENI RET ;Rotina que junta os números da distância recebida em R1, ;colocando-os em decimal ;Retorna em R3 os números juntos JuntaNum: PUSH R6 MOV R6,3 LoopJuntaNum: CMP R6,R0 BR.Z SaiJuntaNum MOV R2,10 DIV R1,R2 SHL R2,12 ADD R3,R2 SHR R3,4 DEC R6 BR LoopJuntaNum SaiJuntaNum: POP R6 RET ;Rotina que escreve no LCD o número que está na ;posição recebida em R1 EscreveLcdNum: PUSH R3 MOV R3,M[R1] MOV M[LCD_CONTROL],R2 MOV M[LCD_WRITE],R3 POP R3 RET ;Rotina que coloca a distância máxima percorrida pelo jogador ;no LCD (caso esta tenha sido ultrapassada) NovoRecorde: PUSH R1 </pre>	

Dec 05, 13 18:12	jogo_final.as	Page 18/21
	<pre> MOV R1,M[DISTANCIA] MOV M[MAXIMO],R1 POP R1 RET RecordeLcd: PUSH R1 PUSH R2 PUSH R3 PUSH R6 PUSH R7 MOV R1,M[MAXIMO] CALL JuntaNum MOV R6,4 MOV R7,R3 AND R7,MASK_NIB_ESQ ROL R7,UM_NIBBLE ADD R7,'0' ;Transforma o numero em ascii MOV M[LCD_TEMPOR],R7 MOV R1,LCD_TEMPOR MOV R2,M[LCD_CONTROL_REC] CALL EscreveLcdNum INC M[LCD_CONTROL_REC] SHL R3,UM_NIBBLE DEC R6 BR.NZ LoopRecLcd MOV R7,LCD_L2_COL8 MOV M[LCD_CONTROL_REC],R7 POP R7 POP R6 POP R3 POP R2 POP R1 RET ;;;;;;;;;;;;; ; PAUSA ; ;;;;;;;;;;;;; ;Rotina que coloca a variável PAUSA a 1, quando o botão ;de interrupção IA é premido Poe_Pausa: PUSH R1 MOV R1,1 MOV M[PAUSA],R1 POP R1 RTI ;Rotina que pausa o jogo, caso este esteja a decorrer ;e que retoma o jogo, caso este esteja em pausa Pausa: PUSH R1 PUSH R2 MOV R1,M[ACTIVA_TEMP] XOR R1,1 MOV M[ACTIVA_TEMP],R1 CMP M[ESTA_PAUSA],R0 BR.NZ RetomaJogo CALL LimpaLcd MOV R1,LCD_PAUSA MOV R2,LCD_LINHA1 CALL EscreveLcd COM M[ESTA_PAUSA] </pre>	

Dec 05, 13 18:12	jogo_final.as	Page 19/21
RetomaJogo:	<pre> MOV R1,INT_MASKPAUSA MOV M[INT_MASK_ADDR],R1 BR FimPausa CALL LimpaLcd MOV R1,LCD_1 MOV R2,LCD_LINHA1 CALL EscreveLcd MOV R1,LCD_2 MOV R2,LCD_LINHA2 CALL EscreveLcd CALL IncrementaLcd CALL RecordeLcd COM M[ESTA_PAUSA] MOV R1,1 MOV M[ACTIVA_TEMP],R1 MOV R1,INT_MASKCICLO MOV M[INT_MASK_ADDR],R1 FimPausa: MOV M[PAUSA],R0 POP R2 POP R1 RET </pre>	
	<pre> //////////////////// ; TURBO ; //////////////////// ;Rotina que coloca a variável TURBO a 1, quando o botão ;de interrupção I2 é premido Poe_Turbo: PUSH R1 MOV R1,1 MOV M[TURBO],R1 POP R1 RTI ;Rotina que coloca o jogo em modo turbo caso este esteja normal, ;e retoma ao estado normal caso esteja em turbo ;A velocidade retomada é aquela que corresponde ao nível actual Turbo: PUSH R1 PUSH R2 CMP M[ESTA_TURBO],R0 BR.NZ TiraTurbo MOV R1,2 MOV M[VELOCIDADE],R1 MOV R2,DEZASSEIS_LEDS MOV M[LED_CONTROL],R2 MOV M[DURACAO],R1 MOV R1,1 MOV M[ESTA_TURBO],R1 BR SaiTurbo TiraTurbo: MOV M[ESTA_TURBO],R0 MOV R1,1 CMP M[NIVEL],R1 CALL.Z MudaPara1 MOV R1,2 CMP M[NIVEL],R1 CALL.Z MudaPara2 MOV R1,3 CMP M[NIVEL],R1 CALL.Z MudaPara3 SaiTurbo: MOV M[TURBO],R0 MOV R1,1 MOV M[ACTIVA_TEMP],R1 </pre>	

Dec 05, 13 18:12	jogo_final.as	Page 20/21
	<pre> POP R2 POP R1 RET //////////////////// ; DISPLAY ; //////////////////// ;Rotina que escreve no display o número de obstáculos já ;ultrapassados pela bicicleta EscCont: PUSH R1 PUSH R2 PUSH R3 PUSH R4 DSI MOV R1,M[CONTADOR] MOV R2,NUM_NIBBLES MOV R3,IO_DISPLAY Ciclo: MOV R4,10 DIV R1,R4 MOV M[R3],R4 INC R3 DEC R2 BR.NZ Ciclo ENI POP R4 POP R3 POP R2 POP R1 RET //////////////////// ; FIM DO JOGO ; //////////////////// ;Rotina que coloca todas variáveis de controlo a 0, faz reset na distância percorrida, ;no contador de obstáculos, no nível e na velocidade, actualiza a distância máxima percorrida, ;e apaga os obstáculos, a bicicleta e a pista. Incrementa também o número de vezes já jogadas. FimJogo: DSI PUSH R4 MOV M[TURBO],R0 MOV M[ESTA_TURBO],R0 MOV M[MOVE_OBS],R0 MOV M[ACTIVA_TEMP],R0 MOV M[ANDA_DIR],R0 MOV M[ANDA_ESQ],R0 MOV M[CONTADOR],R0 MOV R4,M[DISTANCIA] CMP R4,M[MAXIMO] CALL.P NovoRecorde CALL EscCont MOV M[DISTANCIA],R0 MOV M[LED_CONTROL],R0 CALL ApagaObs CALL Apaga_bic CALL Apaga_espaco MOV R4,5 </pre>	

Dec 05, 13 18:12

jogo_final.as

Page 21/21

```
MOV    M[VELOCIDADE],R4
INC    M[VEZES_JOGADAS]
MOV    R4,1
MOV    M[NIVEL],R4
POP    R4
ENI
JMP    Inicio
```