

Dec 02, 16 21:34		Rtype_g10.as		Page 1/18
;Ficheiro do projeto IAC 16/17 LEIC-AL				
;Versão final de entrega				
;Rafael PAndrade (86503) & Diogo Ramalho (86407)				
;-----				
;Constantes do sistema;				
;-----				
;_____Janela de texto_____				
CURSOR	EQU	FFFCh		
ESCRITA	EQU	FFFEh		
SP_INIT	EQU	FDFFh		
;_____Placa_____				
;				
MASCARA	EQU	FFFFh	;mascara de interrupcoes	
RECETOR_MAS	EQU	FFFAh	;respetivo recetor	
DISPLAY	EQU	FFF0h	;endereco de memoria do display de 7 segmentos	
LEDS	EQU	FFF8h	;endereco de memoria dos leds	
LCD_CONTROLO	EQU	FFF4h	;endereco de controlo-Axxx xxxx xxxL CCCC	
LCD_ESCRITA	EQU	FFF5h	;endereco de recetor do caractere	
;_____Temporizador_____				
;				
TEMPO_UN	EQU	0002h	;N. de 100's de milisegundos por interrupt	
TEMPO_EN	EQU	FFF6h	;Endereco do relógio	
TEMPO_ON	EQU	FFF7h	;Relógio on se bit - significativo =1	
;-----				
;Constantes "Graficas"				
;-----				
RAND_MASK	EQU	1000000000010110b		
NUM_CAR_JAN_L	EQU	1700h	;numero de linhas na janela de texto	
NUM_CAR_JAN_C	EQU	004Eh	;numero de colunas na janela de texto	
POS_TXT_PREP	EQU	0C23h		
POS_TXT_PRIMA	EQU	0E20h		
POS_TXT_POINTS	EQU	0E26h		
CANTO	EQU	174Eh	;coordenadas do canto do display	
POS_NAVE_INIT	EQU	0503h	;coordenadas da posicao inicial da nave	
BORDA	EQU	'#'		
VAZIO	EQU	' '		
ASA_CIMA	EQU	'\'		
CAB	EQU	')'		
CANHAO	EQU	'>'		
ASA_BAIXO	EQU	'/'		
TIROS	EQU	'-'		
ASTER	EQU	'*'		
B_HOLE	EQU	'O'		

Dec 02, 16 21:34

Rtype_g10.as

Page

```
;-----
;Arquivo de dados
;Glossario:
;      -P:posicao
;      -F:flag
;      -D:Dados
;      -TXT:texto
;-----

P_NAVE      ORIG      8000h      ;Arquivo dos dados:
WORD      POS_NAVE_INIT      ;-Posicao em linha da nave
P_TIROS     TAB       15      ;-15 tiros
P_ASTER     TAB       12      ;-12 asteroides
P_B_HOLE    TAB       4       ;-4 buracos negros
F_INTERRUPT WORD      0000h    ;-ativada por qualquer botao
F_BAIXO     WORD      0000h    ;-ativada por I0
F_CIMA      WORD      0000h    ;-ativada por I1
F_TRAS      WORD      0000h    ;-ativada por I2
F_FRENTE    WORD      0000h    ;-ativada por I3
F_SHOT      WORD      0000h    ;-ativada por I4
F_REINIT    WORD      0000h    ;-ativada por IE
F_PAUSE     WORD      0000h    ;-ativada por ID
F_TICKTACK  WORD      0000h    ;-ativado por temporizador
F_COUNT_2   WORD      0001h    ;-guarda periodos de 2 intervalos
F_COUNT_3   WORD      0000h    ;-conta o numero de asteroides seguidos
F_COUNT_6   WORD      0005h    ;-conta o espaco entre objeto celeste
F_GAME_OVER WORD      0000h    ;-ativada por colisoes contra a nave
D_POINTS    TAB       4       ;-4 digitos da pontuacao
TXT_PREP    STR       'Prepare-se#'
TXT_PRIMA_IE STR       'Prima o botao IE#'
TXT_G_OVER  STR       'Fim do jogo#'
D_RANDOM_N  WORD      AA35h    ;valor aleatorio inicial
P_INIT_CEL  WORD      0000h

;-----
;Tabela de Interrupções (e rotinas)
;-----

INT0      ORIG      FE00h
WORD      I_Baixo
INT1      WORD      I_Cima
INT2      WORD      I_Tras
INT3      WORD      I_Frente
INT4      WORD      I_Shot
INT5      WORD      I_Init
INT6      WORD      I_Init
INT7      WORD      I_Init
INT8      WORD      I_Init
INT9      WORD      I_Init
INTA      WORD      I_Init
INTB      WORD      I_Init
INTC      WORD      I_Init
INTD      WORD      I_Pause
INTE      WORD      I_Reinit
INTF      WORD      I_Tempo
```

Dec 02, 16 21:34	Rtype_g10.as	Page 3/18
I_Baixo:	INC M[F_BAIKO] INC M[F_INTERRUPT] RTI	
I_Cima:	INC M[F_CIMA] INC M[F_INTERRUPT] RTI	
I_Tras:	INC M[F_TRAS] INC M[F_INTERRUPT] RTI	
I_Frente:	INC M[F_FRENTE] INC M[F_INTERRUPT] RTI	
I_Shot:	INC M[F_SHOT] INC M[F_INTERRUPT] RTI	
I_Init:	INC M[F_INTERRUPT] RTI	
I_Pause:	COM M[F_PAUSE] INC M[F_INTERRUPT] RTI	
I_Reinit:	INC M[F_REINIT] INC M[F_INTERRUPT] RTI	
I_Tempo:	COM M[F_TICKTACK] RTI	
;-----		
;Rotinas		
;-----		
	ORIG 4000h	
;-----		
;_____Escrita_____		
;Rotina que escreve caracteres no ecran		
;Argumentos: R1-Posicao; R2-Caractere (passagem por registo)		
Escrita:	MOV M[CURSOR], R1 MOV M[ESCRITA], R2 RET	
;-----		
;_____Recontagem_____		
;Reinicia o temporizador		
;Registos usados:R7		
Recontagem:	MOV R7, TEMPO_UN ;set do timer a MOV M[TEMPO_EN], R7 ;TEMPO_UN MOV R7,0001h MOV M[TEMPO_ON], R7 ;comeca o relógio RET	

Dec 02, 16 21:34	Rtype_g10.as	Page
;_____INTRO_____		
;Ecran inicial, escreve e apaga essas mensagens		
;Registos usados: R1, R2, R3		
Intro:	PUSH R1 PUSH R2 PUSH R3	
texto1:	MOV R1, POS_TXT_PREP MOV R3, TXT_PREP	
intro1:	MOV R2,M[R3] CMP R2,BORDA ;caractere delimitador BR.Z texto2 CALL Escrita INC R1 INC R3 BR intro1	
texto2:	MOV R1, POS_TXT_PRIMA MOV R3, TXT_PRIMA_IE	
intro2:	MOV R2,M[R3] CMP R2,BORDA BR.Z espera_ie CALL Escrita INC R1 INC R3 BR intro2	
espera_ie:	CMP M[F_REINIT],R0 BR.Z espera_ie MOV R2, VAZIO MOV R1, POS_TXT_PREP MOV R3,10 ;Comprimento de TXT_PREP CMP R3,R0 BR.Z prep_apaga2 CALL Escrita INC R1 DEC R3 BR apaga1	
prep_apaga2:	MOV R1, POS_TXT_PRIMA MOV R3,16 ;Comprimento de TXT_PRIMA CMP R3,R0 BR.Z introed CALL Escrita INC R1 DEC R3 BR apaga2	
apaga2:	MOV M[F_REINIT],R0 MOV M[F_INTERRUPT],R0 POP R3 POP R2 POP R1 RET	
introed:		
;_____Apaga tudo_____		
;Escreve espacos no ecran, 0s no display de 7 segmentos, apaga leds		
Apaga_tudo:	PUSH R1 PUSH R2 PUSH R3 ;linhas PUSH R4 ;colunas MOV R2, VAZIO MOV R3, R0 MOV R4, R0	

Dec 02, 16 21:34	Rtype_g10.as	Page 5/18
apaga_ciclo:	MOV M[TEMPO_ON],R0 ;para o relógio MVBH R1,R3 MVBL R1,R4 CALL Escrita INC R4 ;vai andando nas colunas CMP R4, NUM_CAR_JAN_C BR.NP apaga_ciclo MOV R4, R0 ADD R3, 0100h ;vai recomeçando a linha CMP R3, NUM_CAR_JAN_L BR.NP apaga_ciclo MOV M[LEDS],R0 ;apaga leds MOV R4,DISPLAY CMP R4,FFF4h ;apaga 7 segmentos BR.NN saida_apaga MOV M[R4],R0 INC R4 BR disp_7_ciclo POP R4 POP R3 POP R2 POP R1 RET	
; _____ Reset _____		
;Reset da memória: 0's nas flags e nas posicoes dos tiros e obj celestes		
;POS_NAVE_INIT em M[P_NAVE], 0001h em M[F_COUNT_2] e 0005h em M[F_COUNT_6]		
;(para criar um asteroide logo que inicie o jogo)		
Reset:	PUSH R3 PUSH R4 MOV R3,P_NAVE ;M[P_NAVE]<-POS_NAVE_INIT MOV R4, POS_NAVE_INIT MOV M[R3],R4 INC R3 CMP R3,TXT_PREP ;ate este endereco, BR.NN reset_saida ;por a 0s MOV M[R3],R0 INC R3 BR reset_ciclo MOV R4,0001h MOV M[F_COUNT_2],R4 MOV R4,0005h MOV M[F_COUNT_6],R4 POP R4 POP R3 RET	
reset_ciclo:		
reset_saida:		
; _____ Game_over _____		
;Mostra a mensagem de fim de jogo+pontuacao		
;fica esperando M[F_INTERRUPT] antes de reiniciar o jogo		
;vide fluxograma n. 4		
Game_over:	CALL Apaga_tudo PUSH R1 PUSH R2 PUSH R6 MOV R1, POS_TXT_PREP ;1a linha-'Fim de jogo' MOV R6,TXT_G_OVER MOV R2,M[R6]	
g_o_t_ciclo:		

Dec 02, 16 21:34	Rtype_g10.as	Page
	CMP R2,BORDA ;caractere delimitador BR.Z g_o_pontos CALL Escrita INC R1 INC R6 BR g_o_t_ciclo MOV R1, POS_TXT_POINTS ;escreve 4 digitos dos pontos MOV R6,D_POINTS ADD R6,3 MOV R2,M[R6] ADD R2,'0' CALL Escrita INC R1 DEC R6 CMP R6,F_GAME_OVER BR.P g_o_p_ciclo MOV M[F_INTERRUPT],R0 ENI CMP M[F_INTERRUPT],R0 BR.Z g_o_espera POP R6 POP R2 POP R1 CALL Apaga_tudo ;limpa o ecran CALL Reset ;reset de valores CALL Desenha_nave ;desenha a nave CALL Wall ;desenha paredes de jogo CALL Recontagem ;recomeca o temporizador RET	
g_o_pontos:		
g_o_p_ciclo:		
g_o_espera:		
; _____ Paredes _____		
;Registos usados: R1,R2		
Wall:	PUSH R1 PUSH R2 MOV R2,BORDA MOV R1,R0 CALL Escrita ;escreve numa coluna, linha de cima MVBH R1,CANTO ;escreve na coluna, na linha de baixo CALL Escrita CMP R1,CANTO BR.NN emparedado INC R1 MVBH R1,R0 BR up CALL Inicia_LCD ;legenda no lcd, com coordenadas POP R2 ;da nave POP R1 RET	
up:		
down:		
emparedado:		
; _____ Desenha_nave _____		
;Desenha a nave nas coordenadas actuais		
;Registos: R1,R2		
Desenha_nave:	PUSH R1 PUSH R2 MOV R1, M[P_NAVE] ;posicao do canhao MOV R2, CANHAO	
coord_init:		
d_canhao:		

Dec 02, 16 21:34	Rtype_g10.as	Page 7/18
d_cabine:	CALL Escrita DEC R1 ;posicao da cabine MOV R2, CAB CALL Escrita ADD R1, 0100h ;posicao da asa de baixo MOV R2, ASA_BAIXO CALL Escrita SUB R1, 0200h ;posicao da asa de cima MOV R2, ASA_CIMA CALL Escrita POP R2 POP R1 RET	
;_____Rotina Apaga_nave_____		
;Apaga a nave nas coordenadas atuais		
;Registos: R1,R2		
Apaga_nave:	PUSH R1 PUSH R2 MOV R2, VAZIO MOV R1, M[P_NAVE] ;coord. canhao CALL Escrita DEC R1 ;coord. cabine CALL Escrita ADD R1, 0100h ;coord. asa baixo CALL Escrita SUB R1, 0200h ;coord. asa cima CALL Escrita POP R2 POP R1 RET	
;_____Rotina Interrupt_____		
;Se qualquer botão foi carregado		
;chama rotinas apropriadas a flags enabled		
;vide fluxograma n. 2		
Interrupt:	CMP M[F_BAIXO],R0 CALL.NZ Baixo CMP M[F_CIMA], R0 CALL.NZ Cima CMP M[F_TRAS], R0 CALL.NZ Tras CMP M[F_FRENTE],R0 CALL.NZ Frente CMP M[F_SHOT],R0 CALL.NZ Cria_tiro CMP M[F_PAUSE],R0 CALL.NZ Pausa CMP M[F_REINIT],R0 CALL.NZ Reinit CALL Colisoes_nave CMP M[F_GAME_OVER],R0 CALL.NZ Game_over DEC M[F_INTERRUPT] ;Menos uma interrupcao RET	
cima:		
tras:		
frente:		
tiro:		
pause:		
reinit:		
end:		

Dec 02, 16 21:34	Rtype_g10.as	Page
;_____Rotina Baixo_____		
;Verifica se pode ir para baixo; se puder, vai		
Baixo:	PUSH R1 ;Posicao da nave PUSH R3 MOV R3, 144Fh ;se a nave nao pode ir a 15XXh MOV R1, M[P_NAVE] CMP R3,R1 BR.N salta_b CALL Apaga_nave ADD R1, 0100h MOV M[P_NAVE], R1 CALL Desenha_nave POP R3 POP R1 DEC M[F_BAIXO] ;Menos uma interrupcao para lidar CALL Escreve_L_LCD ;escreve a nova linha no lcd RET	
salta_b:		
;_____Rotina Cima_____		
;Verifica se pode ir para cima; se puder, vai		
Cima:	PUSH R1 ;Posicao da nave PUSH R3 MOV R3, 0300h ;nave nao pode ir a 01XXh MOV R1, M[P_NAVE] CMP R1,R3 BR.N salta_c CALL Apaga_nave SUB R1, 0100h MOV M[P_NAVE], R1 CALL Desenha_nave POP R3 POP R1 DEC M[F_CIMA] ;Menos uma interrupcao para lidar CALL Escreve_L_LCD ;escreve a nova linha no lcd RET	
salta_c:		
;_____Rotina Tras_____		
;Verifica se pode ir para Tras; se puder, vai		
Tras:	PUSH R1 ;Posicao da nave PUSH R3 MVBL R3, 0002h ;nave não pode ir a XX01h MVBL R1, M[P_NAVE] CMP R1,R3 BR.N salta_t CALL Apaga_nave DEC M[P_NAVE] CALL Desenha_nave POP R3 POP R1 DEC M[F_TRAS] ;Menos uma interrupcao para lidar CALL Escreve_C_LCD ;escreve a nova coluna no lcd RET	
salta_t:		

Dec 02, 16 21:34	Rtype_g10.as	Page 9/18
;_____Rotina Frente_____		
;Verifica se pode ir para a frente; se puder, vai		
Frente:	PUSH R1 ;Posicao da nave PUSH R3 MVBL R3, 004Dh ;nave não pode ir a XX4Fh MVBL R1, M[P_NAVE] CMP R3,R1 BR.N salta_f CALL Apaga_nave INC M[P_NAVE] CALL Desenha_nave POP R3 POP R1 DEC M[F_FRENTE] ;Menos uma interrupcao para lidar CALL Escreve_C_LCD ;escreve a nova coluna no lcd RET	
salta_f:		
;_____Rotina Cria_tiro_____		
;Cria um tiro na frente do canhão		
;Registos usados:1-4		
Cria_tiro:	PUSH R1 PUSH R2 PUSH R4 MOV R4, P_TIROS MOV R2, TIROS MOV R1, M[P_NAVE] ;posicao do canhao INC R1 ;coloca tiro na ponta do canhao 	
ciclo_tiro:	CMP R4,800Fh BR.P saida_tiro CMP M[R4],R0 BR.Z salta_p_tiro INC R4 ;procura por um espaco livre na lista BR ciclo_tiro ;de tiros, 	
salta_p_tiro:	MOV M[R4], R1 ;coloca a coord do tiro no espaco livre CALL Escrita ;escreve no ecran 	
saida_tiro:	POP R4 POP R2 POP R1 DEC M[F_SHOT] ;Menos uma interrupcao para lidar RET	
;_____Rotina Pausa_____		
;Pára o jogo momentaneamente		
Pausa:	PUSH R1 COM M[TEMPO_ON] 	
ciclo_pausa:	CMP M[F_PAUSE],R0 BR.NZ ciclo_pausa MOV R1,F_INTERRUPT 	
anula_i_pausa:	CMP R1,F_PAUSE ;anula botoes carregados durante BR.NN fim_pausa ;a pausa MOV M[R1],R0 INC R1 	
fim_pausa:	BR anula_i_pausa CALL Recontagem POP R1 RET	

Dec 02, 16 21:34	Rtype_g10.as	Page 1
;_____Rotina Reinit_____		
;Reinicio imediato do jogo		
;Composto por: Apaga_tudo, Reset, Wall, Desenha_Nave e Recontagem		
Reinit:	CALL Apaga_tudo ;limpa ecran CALL Reset ;reset de valores da memoria CALL Intro ;ecran de inicio de jogo CALL Reset ;reset de valores da memoria CALL Desenha_nave ;desenha a nave CALL Wall ;desenha espaco de jogo CALL Recontagem ;reinicia o contador RET	
;-----		
;_____Rotina Temporizador_____		
;Tratamento de eventos dependentes do temporizador		
;vide fluxograma n. 3		
;-----		
Temporizador:	PUSH R5 PUSH R6 MOV R5,2 MOV R6,6 CALL Saiu_limites ;se o tiro ou obj celestes saem CALL Atualiza_tiro ;incrementa coords do tiro INC M[F_COUNT_2] CMP R5, M[F_COUNT_2] BR.P desenhar CALL Atualiza_Cel ;atualiza coords dos obj celestes CALL Colisoes_nave ;testa colisoes nave<-->obj celestes CMP M[F_GAME_OVER],R0 BR.NZ sair_temp ;se game over, sai da rotina CALL Desenha_cel ;desenha os objetos celestes MOV M[F_COUNT_2],R0 INC M[F_COUNT_6] CMP R6,M[F_COUNT_6] BR.P desenhar CALL Cria_aster_bh ;cria obj celestes MOV M[F_COUNT_6],R0 	
desenhar:	DSI CALL Desenha_tiro ;desenha os tiros CALL Colisoes_tiro ;testa colisoes tiro<-->obj celestes	
sair_temp:	ENI POP R6 POP R5 COM M[F_TICKTACK] ;repoe flag temporizador CALL Recontagem ;reinicia relógio RET	
;_____Rotina Saiu_limites_____		
;Verifica se o tiro vai sair dos limites, apaga se sim		
;Tambem faz o mesmo para objetos celestes		
Saiu_limites:	PUSH R1 PUSH R2 PUSH R3 PUSH R5 MOV R5,P_TIROS 	
s_tiro_loop:	CMP R5,P_aster ;percorre a lista	

Dec 02, 16 21:34	Rtype_g10.as	Page 11/18
	BR.NN s_tiro_saida CMP M[R5],R0 ;ve se existe posicao de tiros BR.Z s_prox_tiro MVBL R3,M[R5] CMP R3, 004Dh ;ver se vai sair dos limites BR.NP s_prox_tiro MOV R1,M[R5] ;limpa o tiro do ecran MOV R2, VAZIO CALL Escrita MOV M[R5],R0 ;apaga o tiro da memoria s_prox_tiro: INC R5 ;avanca para a proxima posicao BR s_tiro_loop s_tiro_saida: MOV R5,P_ASTER ;PARTE DOS OBJ CELESTES MOV R2, VAZIO s_cel_loop: CMP R5, F_INTERRUPT ;se ainda estamos dentro dos tabs BR.NN s_cel_saida MVBL R3,M[R5] CMP R3,R0 ;verifica a coluna BR.P s_prox_cel CMP R0,M[R5] BR.Z s_prox_cel ;verifica se existe na memoria MOV R1,M[R5] CALL Escrita ;limpa o ecran MOV M[R5],R0 ;apaga a memoria s_prox_cel: INC R5 BR s_cel_loop s_cel_saida: POP R5 POP R3 POP R2 POP R1 RET	
;_____Rotina Atualiza_tiro_____		
;Registos usados:R6 ;em todas as posicoes ocupadas da lista de tiros, incrementa coordenadas		
Atualiza_tiro:	PUSH R6 MOV R6,P_TIROS a_tiro_loop: CMP R6,P_ASTER BR.NN a_tiro_saida ;fim da lista de tiros CMP M[R6],R0 BR.Z next_pos_t ;tiro nulo INC M[R6] next_pos_t: INC R6 ;inc coord. tiro BR a_tiro_loop a_tiro_saida: POP R6 RET	
;_____Rotina Atualiza_Celeste_____		
;Registos usados:R6 ;em todas as posicoes ocupadas da lista de obj celestes, decrementa coordenadas		
Atualiza_Cel:	PUSH R6 MOV R6,P_ASTER ;inicio da lista P_ASTER a_cel_loop: CMP R6,F_INTERRUPT ;fim da lista de B_HOLE BR.NN a_cel_saida CMP M[R6],R0 ;verifica se posicao nao nula BR.Z next_pos_cel DEC M[R6] ;Dec coord. obj celeste next_pos_cel: INC R6	

Dec 02, 16 21:34	Rtype_g10.as	Page 1
a_cel_saida:	BR a_cel_loop POP R6 RET	
;_____Rotina Colisoos_nave_____		
;verifica se nenhum obj celeste colidiu com a nave		
Colisoos_nave:	PUSH R5 PUSH R6 MOV R6, P_ASTER ;inicio lista obj cel MOV R5, M[P_NAVE] ;coord. nave	
ciclo_c_nave:	CMP R6, F_INTERRUPT BR.NN saida_c_nave CMP M[R6],R0 BR.Z next_c_nave_oc CMP M[R6],R5 ;pos canhao BR.Z col_n_inc_fgo DEC R5 ;pos cabine CMP M[R6],R5 BR.Z col_n_inc_fgo ADD R5,100h ;pos asa baixo CMP M[R6],R5 BR.Z col_n_inc_fgo SUB R5,200h ;pos asa cima CMP M[R6],R5 BR.Z col_n_inc_fgo	
next_c_nave_oc:	INC R6 MOV R5, M[P_NAVE] BR ciclo_c_nave	
col_n_inc_fgo:	COM M[F_GAME_OVER] ;se houve colisao, game_over	
saida_c_nave:	POP R6 POP R5 RET	
;_____Rotina Desenha_cel_____		
;Registos usados: R1,R2,R4		
;Argumentos:vetor de posicoes de asteroides (P_ASTER)		
; e buracos negros (P_BHOLE)		
Desenha_cel:	PUSH R1 PUSH R2 PUSH R4 MOV R4,P_ASTER ;inicio da lista P_ASTER	
d_aster_loop:	CMP R4,P_B_HOLE BR.NN d_b_h_loop CMP M[R4],R0 ;verifica se asteroide nao nulo BR.Z next_p_d_ast MOV R1,M[R4] MOV R2,ASTER CALL Escrita ;desenha asteroide atualizado INC R1 MOV R2,VAZIO ;apaga o asteroide anterior CALL Escrita	
next_p_d_ast:	INC R4 ;incrementa posicao da lista BR d_aster_loop	
d_b_h_loop:	CMP R4,F_INTERRUPT BR.NN saida_d_cel CMP M[R4],R0 ;verifica se b. negro nao nulo BR.Z next_p_d_b_h	

Dec 02, 16 21:34	Rtype_g10.as	Page 13/18
next_p_d_b_h:	<pre> MOV R1,M[R4] MOV R2,B_HOLE CALL Escrita ;desenha b. negro atualizado INC R1 MOV R2,VAZIO ;apaga o b. negro anterior CALL Escrita BR d_b_h_loop ;incrementa posicao da lista saida_d_cel: POP R4 POP R2 POP R1 RET </pre>	
; _____ Rotina Random _____		
;Random: gera um novo numero pseudo-aleatorio		
; argumentos: M[D_RANDOM_N]-ultimo pseudo aleatorio gerado		
; gera P_INIT_CEL, posicao inicial do proximo obj celeste		
Random:	<pre> PUSH R1 PUSH R7 MOV R1,M[D_RANDOM_N] RORC R1,1 BR.NC random_2 BR ror_random random_1: random_2: MOV R1,RAND_MASK XOR M[D_RANDOM_N],R1 ROR M[D_RANDOM_N],1 ror_random: pos_random: MOV R7,20 MOV R1,M[D_RANDOM_N] DIV R1,R7 ADD R7,2 SHL R7,8 ADD R7,004Eh MOV M[P_INIT_CEL],R7 POP R7 POP R1 RET </pre>	
; _____ Rotina Cria_ASTER_BH _____		
Cria_ASTER_BH:	<pre> PUSH R1 PUSH R2 PUSH R3 PUSH R4 INC M[F_COUNT_3] MOV R3,3 CALL Random MOV R1,M[P_INIT_CEL] ;posicao aleatoria CMP M[F_COUNT_3],R3 ;3 asteroides ja foram? BR.P cria_b_h MOV R4,P_ASTER ciclo_aster: CMP M[R4],R0 ;encontra uma posicao livre em P_ASTER BR.Z esc_p_aster INC R4 BR ciclo_aster esc_p_aster: MOV M[R4],R1 ;escreve posicao do asteroide na posicao MOV R2,ASTER CALL Escrita DEC R3 </pre>	

Dec 02, 16 21:34	Rtype_g10.as	Page 1
cria_b_h:	<pre> BR saida_ast_b_h MOV R4,P_B_HOLE ciclo_b_h: CMP M[R4],R0 ;encontra uma posicao livre em P_B_HOLE BR.Z esc_p_b_h INC R4 BR ciclo_b_h esc_p_b_h: MOV M[R4],R1 ;escreve posicao do b_h no espaco livre MOV R2,B_HOLE CALL Escrita MOV M[F_COUNT_3],R0 saida_ast_b_h: POP R4 POP R3 POP R2 POP R1 RET </pre>	
; _____ Rotina Desenha_tiro _____		
;Registos usados: R1,R2,R4		
;Argumentos:vetor de posicoes dos tiros (P_TIROS)		
Desenha_tiro:	<pre> PUSH R1 PUSH R2 PUSH R4 MOV R4,P_TIROS ;inicio da lista P_TIROS d_tiro_loop: CMP R4,P_ASTER BR.NN saida_d_tiro CMP M[R4],R0 ;verifica se tiro nao nulo BR.Z next_p_d_t MOV R1,M[R4] MOV R2,TIROS CALL Escrita ;desenha tiro atualizado DEC R1 MOV R2,VAZIO ;apaga o tiro anterior CALL Escrita next_p_d_t: INC R4 ;incrementa posicao da lista BR d_tiro_loop saida_d_tiro: POP R4 POP R2 POP R1 RET </pre>	
; _____ Rotina Colisoes_tiro _____		
;verifica colisoes entre tiros e objetos celestes		
Colisoes_tiro:	<pre> PUSH R1 PUSH R2 PUSH R3 PUSH R4 PUSH R6 PUSH R7 MOV R2,VAZIO MOV R6,P_TIROS MOV R7,P_ASTER ciclo_col_t: CMP R6,P_ASTER JMP.NN saida_col_t CMP M[R6],R0 JMP.Z next_shot_col MVBH R3,M[R6] ciclo_col_cel: CMP R7,F_INTERRUPT JMP.NN next_shot_col </pre>	

Dec 02, 16 21:34	Rtype_g10.as	Page 15/18
testel:	CMP M[R7],R0 JMP.Z next_cel MVBH R4,M[R7] CMP R3,R4 ;tiro/cel na mesma linha? JMP.NZ next_cel MOV R1,M[R7] CMP R1,M[R6] BR.NZ teste2 CALL Escrita MOV M[R6],R0 BR col_aster_bh INC R1 CMP R1,M[R6] BR.NZ next_cel CALL Escrita MOV M[R6],R0 CMP R7, P_B_HOLE BR.NN col_des_bh ;se for b_hole, redesenha MOV R1,M[R7] CALL Escrita ;se for asteroide, limpa do ecran MOV M[R7],R0 ;e apaga as coordenadas INC M[D_POINTS] ;incrementa pontos CALL Formata_pontos CALL Escreve_pontos ;atualiza display de 7 segmentos MOV R4, FFFFh MOV M[LEDS], R4 ;acende leds BR next_shot_col MOV R2, B_HOLE MOV R1, M[R7] CALL Escrita ;repoe o buraco negro no ecran MOV R2, VAZIO BR next_shot_col INC R7 JMP ciclo_col_cel INC R6 MOV R7,P_ASTER JMP ciclo_col_t POP R7 POP R6 POP R4 POP R3 POP R2 POP R1 RET	
;----- ;Rotinas da Placa ;-----		
;----- ;Rotina Formata_pontos ;Divide a pontuacao por unidades, dezenas, centenas e milhares ; e coloca na posicap de memoria respetiva		
Formata_pontos:	PUSH R4 PUSH R5 MOV R4,D_POINTS MOV R5,000Ah formata_ciclo: CMP M[R4],R5 BR.N formata_saida MOV M[R4],R0	

Dec 02, 16 21:34	Rtype_g10.as	Page 1
formata_saida:	INC R4 CMP R4,TXT_PREP BR.NN formata_saida INC M[R4] BR formata_ciclo POP R5 POP R4 RET	
;----- ;Rotina Escreve_pontos ;Escreve o numero de pontos no display de 7 segmentos		
Escreve_pontos:	PUSH R4 PUSH R5 PUSH R6 MOV R4, DISPLAY MOV R5, D_POINTS ciclo_e_pontos: MOV R6, M[R5] MOV M[R4],R6 INC R4 INC R5 CMP R5, TXT_PREP BR.N ciclo_e_pontos saida_e_pontos: POP R6 POP R5 POP R4 RET	
;----- ;Rotina Inicia_LCD ;Inicia o LCD com as coordenadas linha e coluna da nave		
Inicia_LCD:	PUSH R3 PUSH R4 PUSH R5 PUSH R6 MOV R4, M[P_NAVES] MOV R5, Ah MVBH R6, R4 ;linha da nave SHR R6, 8 MVBH R4, R0 ;coluna da nave DIV R4, R5 ADD R4,'0' ;dezenas e unidades da linha da nave ADD R5,'0' MOV R3, 8000h MOV M[LCD_CONTROLO],R3 MOV R1,'C' MOV M[LCD_ESCRITA],R1 INC R3 MOV M[LCD_CONTROLO],R3 MOV R1,':' MOV M[LCD_ESCRITA],R1 INC R3 MOV M[LCD_CONTROLO],R3 MOV M[LCD_ESCRITA],R4 INC R3 MOV M[LCD_CONTROLO],R3	

Dec 02, 16 21:34

Rtype_g10.as

Page 17/18

```

MOV     M[LCD_ESCRITA], R5

MOV     R5, Ah
DIV     R6, R5    ;dezenas e unidades da coluna da nave
ADD     R6,'0'
ADD     R5,'0'
MOV     R3, 800Ch
MOV     M[LCD_CONTROLO],R3
MOV     R1,'L'
MOV     M[LCD_ESCRITA],R1
INC     R3
MOV     M[LCD_CONTROLO],R3
MOV     R1,':'
MOV     M[LCD_ESCRITA],R1
INC     R3
MOV     M[LCD_CONTROLO],R3
MOV     M[LCD_ESCRITA],R6
INC     R3
MOV     M[LCD_CONTROLO],R3
MOV     M[LCD_ESCRITA],R5

POP     R6
POP     R5
POP     R4
POP     R3
RET

```

```

;-----Rotina Escreve_L_LCD-----
;Escreve as coordenadas atualizadas da linha no LCD
;so escreve o numero novo

```

```

Escreve_L_LCD:  PUSH    R1
                PUSH    R2
                PUSH    R3
                MVBH    R1,M[P_NAVE]
                SHR     R1,8
                MOV     R2,Ah
                DIV     R1,R2
                ADD     R1,'0'
                ADD     R2,'0'
                MOV     R3,8002h
                MOV     M[LCD_CONTROLO],R3
                MOV     M[LCD_ESCRITA],R1
                INC     R3
                MOV     M[LCD_CONTROLO],R3
                MOV     M[LCD_ESCRITA],R2
                POP     R3
                POP     R2
                POP     R1
                RET

```

```

;-----Rotina Escreve_C_LCD-----
;Escreve as coordenadas atualizadas da coluna no LCD
; so escreve o numero novo

```

```

Escreve_C_LCD:  PUSH    R1

```

Dec 02, 16 21:34

Rtype_g10.as

Page 1

```

PUSH     R2
PUSH     R3
MVBH     R1,M[P_NAVE]
MOV     R2,Ah
DIV     R1,R2
ADD     R1,'0'
ADD     R2,'0'
MOV     R3,800Eh
MOV     M[LCD_CONTROLO],R3
MOV     M[LCD_ESCRITA],R1
INC     R3
MOV     M[LCD_CONTROLO],R3
MOV     M[LCD_ESCRITA],R2
POP     R3
POP     R2
POP     R1
RET

```

```

;-----
;Main
;-----
;R7 e usado MAS NÃO e CONSERVADO

```

```

inicial:  ORIG     0000h
          MOV     R7, FFFFh
          MOV     M[CURSOR], R7    ; inicialização do cursor
          MOV     R7, MASCARA
          MOV     M[RECETOR_MAS], R7 ; inicialização de interrupção
          ENI
          MOV     R7, SP_INIT
          MOV     SP, R7           ; inicialização da stack

```

```

          CALL     Apaga_tudo
          CALL     Intro

          CALL     Recontagem      ;inicio do relógio

```

```

rotinas_init:  CALL     Wall
               CALL     Desenha_nave

```

```

main:         CMP     M[F_INTERRUPT], R0
               BR.Z    tempo

```

```

tempo:        CALL     Interrupt
               CMP     M[F_TICKTACK],R0
               BR.Z    main
               MOV     M[LEDS], R0
               CALL     Temporizador
               CMP     M[F_GAME_OVER],R0

```

```

fim:          CALL.NZ  Game_over
               BR      main

```