**Explicação do código do projecto Tron**

**Constantes**

SP\_INICIAL EQU FDFFh ;endereço de memória onde começa a pilha

IO\_WRITE EQU FFFEh ;endereço associado à escrita na janela de texto

IO\_CONTROL EQU FFFCh ;endereço associado ao cursor da janela de texto

Inicial EQU 0110h ;linha 1 coluna 16 (primeira linha da moldura)

Final EQU 1610h ;linha 22 coluna 16 (última linha da moldura)

Fim EQU '@' ;utilizado na função EscreveStr (escreve string)

**Variáveis**

caracter2 STR '+------------------------------------------------+', Fim ;string da primeira e ultima linha

caracter3 STR '| |', Fim ;string das restantes 20

linhas com espaços no meio

**Rotinas Principais**

DesenhaMapa: PUSH R1 ;R1 vai receber a linha a escrever na janela de texto

PUSH R2 ;R2 vai receber a posição onde se começa a escrever cada linha

MOV R2, Inicial ;linha 1 coluna 16

MOV R1, caracter2 ;"+---------------------+"

CALL EscreveStr ;escreve a primeira linha (“+------------+”)

Colunas: ADD R2, 0100h ;muda para a linha seguinte

MOV R1, caracter3 ;”| |”

CALL EscreveStr ;escreve a linha “| |”

CMP R2, Final ;enquanto não chegarmos à última linha (linha 22 coluna 16)

BR.NZ Colunas ;fazemos este ciclo que acabará por chegar 20 linhas do tipo “| |”

MOV R1, caracter2 ;”+------------------+”

CALL EscreveStr ;escreve a última linha (“+---------------+”)

POP R2

POP R1

RET

PosicoesIni: PUSH R1 ;R1 recebe o simbolo associado ao jogador

PUSH R2 ;R2 recebe a posição inicial do jogador

MOV R1, jogador1 ;simbolo jogador 1 -> “X”

MOV R2, Posicao1 ;linha 12 coluna 24 (10+2 e 8+16)

CALL EscreveStr ;escreve o simbolo do jogador 1

MOV R1, jogador2 ;simbolo jogador 2 -> “#”

MOV R2, Posicao2 ;linha 12 coluna 56 (10+2 e 40+16)

CALL EscreveStr ;escreve o simbolo do jogador 2

POP R2

POP R1

RET

**Rotinas Auxiliares**

EscreveCar: PUSH R2 ;R2 recebe onde irá escrever o carater

PUSH R3 ;R3 recebe o carater

MOV M[IO\_CONTROL], R2 ;associa o cursor à posição

MOV M[IO\_WRITE],R3 ;escreve carater na janela de texto

POP R3

POP R2

RET

EscreveStr: PUSH R3

PUSH R1

PUSH R2

CicloEscString: MOV R3,M[R1] ;retira o primeiro carater da string

CMP R3,Fim ;se o carater for “@” (Fim) significa que já percorremos todos os carateres da string

BR.Z FimEscString

CALL EscreveCar ;escreve o carater

INC R1 ;selecionamos o carater seguinte

INC R2 ;o cursor passa a apontar para a posição seguinte

BR CicloEscString

FimEscString: POP R2

POP R1

POP R3

RET

**Código de chamada da rotina Desenha Moldura e PosicoesIni**

ORIG 0000h

MOV R7, SP\_INICIAL

MOV SP,R7

MOV R7,FFFFh ;estas duas linhas servem para

MOV M[IO\_CONTROL], R7 ;inicializar o cursor

JMP Inicio

Inicio: MOV M[FFFCh], R2

CALL DesenhaMapa

CALL PosicoesInic

Ciclo: JMP Ciclo ;ciclo infinito para o programa ficar “parado” à espera de que carreguemos numa interrupção, por exemplo

**Funcionamento do cursor da janela de texto**

Para converter a coluna X e linha Y para um valor que o cursor “perceba” basta converter os valores X e Y para hexadecimal e juntá-los segundo este modelo.

2bits hexadecimal -> Y | 2 bits hexadecimal -> X

Linhas | Colunas

Exemplo: linha 2 coluna 10 -> 020A (02 = linha 2 e 0A = coluna 10)