

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 3/19

```

|_|_|_|_|_|', FIM_TEXTO
strfim09      STR      ' ', FIM_TEXTO
strfim0A      STR      ' ', FIM_TEXTO
strfim0B      STR      ' ', FIM_TEXTO
strfim0C      STR      ' ', FIM_TEXTO
strfim0D      STR      ' ', FIM_TEXTO
strfim0D2     STR      ' ', FIM_TEXTO
strfim0E      STR      ' ', FIM_TEXTO
strfim10      STR      ' ', FIM_TEXTO
strfim11      STR      ' ', FIM_TEXTO
strtrc        STR      ' ', FIM_TEXTO
-----
strtrcBlank   STR      ' ', FIM_TEXTO
stringdistancia STR      'distancia:00000', FIM_TEXTO
stringcolunas STR      'colunas', FIM_TEXTO
APAGANUMESC   STR      '0000', FIM_TEXTO
NUMEROAESC    STR      '0000', FIM_TEXTO
NUMEROAESC2   STR      '0', FIM_TEXTO
; =====TABELA DE INTERRUPCOES =====
INT0          WORD     BotaoI0
INT1          WORD     BotaoI1
INT2          WORD     BotaoI2
INT3          WORD     BotaoI3
INT4          WORD     Endgame
INT5          WORD     Endgame
INT6          WORD     Endgame
INT7          WORD     Endgame
INT8          WORD     Endgame
INT9          WORD     Endgame
INTA          WORD     Endgame
INTB          WORD     Endgame
INTC          WORD     Endgame
INTD          WORD     Endgame
INTE          WORD     Endgame
ORIG          WORD     FE0Fh
INT15         WORD     Atualiza

; ===== ROTINAS DE INTERRUPCAO =====

;Funcao:
;      Entradas:
;      Saida:
;      Efeitos:

;BotaoI0: rotina que indica que foi pressionado o botao I0, e utilizado para
;      ativar a rotina Saltar e para terminar o jogo depois do gameover
;      Entradas:---
;      Saida:---
;      Efeitos: ativacao da flag M[Salto]
BotaoI0:      INC      M[Salto]
RTI

;BotaoI1: rotina que indica que foi pressionado o botao I1, e utilizado para
;      iniciar o jogo, diminuir a dificuldade depois do jogo iniciado e para
;      terminar o jogo depois do gameover
;      Entradas:---
;      Saida:---
;      Efeitos: ativacao da flag M[DimDif]
BotaoI1:      INC      M[DimDif]

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 4/19

```

RTI

;BotaoI2: rotina que indica que foi pressionado o botao I2, e utilizado para
;      aumentar a dificuldade depois do jogo ter sido iniciado
;      Entradas:---
;      Saida:---
;      Efeitos: ativacao da flag M[AumDif]
BotaoI2:      INC      M[AumDif]
RTI

;BotaoI3: rotina que indica que foi pressionado o botao I3, e utilizado para
;      reiniciar o jogo depois do gameover
;      Entradas:
;      Saida:
;      Efeitos: ativacao da flag M[ResetJogo]
BotaoI3:      INC      M[ResetJogo]
RTI

;Endgame: rotina que indica que foi pressionado qualquer botao desde o I4 ate
;      ao IE, e utilizado para terminar o jogo depois do gameover
;      Entradas:
;      Saida:
;      Efeitos: ativacao da flag M[KillGame]
Endgame:      INC      M[KillGame]
RTI

;Atualiza: rotina que incrementa os valores dos contadores
;      Entradas:---
;      Saida:---
;      Efeitos: incrementa os valores dos contadores
;      CONTGRAV, CONTCAIR e CONTROLCOL
Atualiza:     PUSH     R1
INC           M[CONTGRAV]
INC           M[CONTCAIR]
INC           M[CONTROLLOCOL]
MOV           R1, INTERVALO
MOV           M[TEMPORIZADOR], R1
MOV           R1, BITATIV
MOV           M[ATIVADOR], R1
POP           R1
RTI

; ===== INICIA O PROGRAMA =====
ORIG          0000h
JMP           Inicio

; ===== ROTINAS =====

;EscreveEcraIni: rotina que escreve o ecra inicial
;      Entradas:---
;      Saida:---
;      Efeitos: escreve as strings que formam o ecra inicial
EscreveEcraIni: PUSH     R1
PUSH         R2
DSI
MOV          R1, strini02
MOV          R2, POSINI01
PUSH         R1
PUSH         R2
CALL         EscreveStr
MOV          R1, strini03

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 5/19

```

ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini04
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini05
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini06
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini09
MOV     R2, POSINI02
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini0A
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini0B
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini0C
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini0D
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini0E
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini11
MOV     R2, POSINI03
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini12
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini13
ADD     R2, 0100h
PUSH    R1

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 6/19

```

PUSH    R2
CALL    EscreveStr
MOV     R1, strini14
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
MOV     R1, strini15
ADD     R2, 0100h
PUSH    R1
PUSH    R2
CALL    EscreveStr
ENI
POP     R2
POP     R1
RET

;Colunas: rotina que move as colunas
;      Entradas:---
;      Saidas:---
;      Efeitos: apaga todas as colunas (uma de cada vez)
;               e escreve-os uma coluna a esquerda e
;               decrementa o valor da posicao das colunas
Colunas:    PUSH    R1
            PUSH    R2
            PUSH    R3
            PUSH    R4
            PUSH    R5
            DSI
            CALL    ATUALIZADIST
            MOV     R1, M[NUM_OBST]
            MOV     R2, colunal
ColunasEsc: MOV     R3, BLANK
            MOV     R4, M[R2]
            PUSH    R3
            PUSH    R4
            CALL    ESCREVECOL      ;apaga coluna
            MOV     R4, M[R2]
            AND     R4, 00FFh
            CMP     R4, 0
            JMP     .Z FimCicloEsc
            MOV     R4, M[R2]
            MOV     R3, OBSTACULO
            DEC     R4
            MOV     M[R2], R4
            PUSH    R3
            PUSH    R4
            CALL    ESCREVECOL      ;escreve coluna
            MOV     R3, 0015h
            AND     R4, 00FFh
            CMP     R4, R3
            BR.NZ   ColunasJmp1; verifica se o bico pode bater na coluna
            MOV     R5, 0001h
            MOV     M[ATIVACOLISAO], R5
            MOV     M[COLUNAATUAL], R2
            PUSH    R2
            CALL    COLISAO
            POP     R2
ColunasJmp1: DEC     R3
            CMP     R4, R3
            BR.NZ   ColunasJmp2; verifica se o corpo pode bater na coluna
            PUSH    R2

```

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 7/19

ColunasJmp2: CALL.Z COLISAO
              POP      R2
              MOV      R3, 0013h
              CMP      R4, R3
              BR.NZ    ColunasJmp3; verifica se o passaro passou a coluna
; para incrementar a pontuacao
              PUSH     R2
              CALL     IncPontuacao
              POP      R2
ColunasJmp3:  INC      R2
              DEC      R1
              JMP.NN   ColunasEsc; verifica se ja moveu todas as colunas
FimCicloEsc:  INC      M[ATIVGERACOL]
              MOV      M[CONTROLOCOL], R0
              ENI
              POP      R5
              POP      R4
              POP      R3
              POP      R2
              POP      R1
              RET

;COLISAO:rotina que verifica se o passaro colidiu
;
; Entradas: pilha:
; M[SP+6] - endereco da memoria que contem a coluna a comparar
; Saidas:---
; Efeitos: faz gameover se o passaro colidiu
COLISAO:     PUSH     R1
              PUSH     R2
              PUSH     R3
              PUSH     R4
              DSI
              MOV      R4, 5
              MOV      R3, M[SP+6]
              MOV      R1, M[R3]
              MOV      R2, R1
              ADD      R2, 0500h
              AND      R1, FF00h
CICLOCOLISAO: ADD      R1, 0100h
              MOV      R3, R7
              AND      R3, FF00h
              CMP      R3, R1
              BR.Z     FIMCOLISAO
              DEC      R4
              CMP      R4, R0
              BR.NZ    CICLOCOLISAO
              CALL     FIM
FIMCOLISAO:  ENI
              POP      R4
              POP      R3
              POP      R2
              POP      R1
              RET

;INCREMENTARNUM: rotina que incrementa um dado numero de forma decimal
;
; Entradas: pilha :
; M[SP+5] - endereco da variavel que contem o numero a incrementar
; Saidas:---
; Efeitos: coloca no endereco da memoria recebida o valor do

```

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 8/19

;
; numero atualizado
INCREMENTARNUM: PUSH     R1
                PUSH     R2
                PUSH     R3
                DSI
                MOV      R1, M[SP+5]
                MOV      R2, M[R1]
                CMP      R2, 9999h
                JMP.Z    Somar1
                AND      R2, 0FFFh
                CMP      R2, 0999h
                JMP.Z    Somar2
                AND      R2, 00FFh
                CMP      R2, 0099h
                JMP.Z    Somar3
                MOV      R2, M[R1]
                AND      R2, 000Fh
                CMP      R2, 0009h
                JMP.Z    Somar4
                INC      M[R1]
                JMP      RETORNA
Somar1:        MOV      M[R1], R0
                JMP      RETORNA
Somar2:        MOV      R2, M[R1]
                AND      R2, F000h
                ADD      R2, 1000h
                MOV      M[R1], R2
                JMP      RETORNA
Somar3:        MOV      R2, M[R1]
                AND      R2, FF00h
                ADD      R2, 0100h
                MOV      M[R1], R2
                JMP      RETORNA
Somar4:        MOV      R2, M[R1]
                AND      R2, FFF0h
                ADD      R2, 0010h
                MOV      M[R1], R2
RETORNA:       ENI
                POP      R3
                POP      R2
                POP      R1
                RETN     1

;IncPontuacao: rotina que incrementa a pontuacao
;
; Entradas:---
; Saidas:---
; Efeitos: incrementa Pontuacao e desativa ATIVACOLISAO
IncPontuacao:  DSI
                DEC      M[ATIVACOLISAO] ;Desativa ATIVACOLISAO
                PUSH     Pontuacao
                CALL     INCREMENTARNUM
                CALL     EscPontuacao
                ENI
                RET

;EscPontuacao: Rotina que efetua a escrita da pontuacao no display de
;
; sete segmentos
; Entradas: ---
; Saidas: ---
; Efeitos: ---
EscPontuacao:  PUSH     R1
                PUSH     R2

```

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 9/19

EscPontCiclo:
    PUSH    R3
    DSI
    MOV     R2, NUM_NIBBLES
    MOV     R3, DISPLAY_SEG
    MOV     R1, M[Pontuacao]
    AND     R1, NIBBLE_MASK
    MOV     M[R3], R1
    ROR     M[Pontuacao], BITS_PER_NIBBLE
    INC     R3
    DEC     R2
    BR.NZ   EscPontCiclo
    ENI
    POP     R3
    POP     R2
    POP     R1
    RET

;ESCREVECOL: rotina que escreve um obstaculo com um dado caracter
;
;      Entradas: pilha:
;      M[SP+7] - caracter a escrever
;      M[SP+6] - endereco que contem a posicao da coluna M
;
;      Saidas:---
;      Efeitos:---
ESCREVECOL:
    PUSH    R1
    PUSH    R2
    PUSH    R3
    PUSH    R4
    DSI
    MOV     R1, M[SP+6]
    MOV     R2, M[SP+7]
    MOV     M[JANELA_CONTROLO], R1
    PUSH    R2
    CALL    EscreveCar
    SUB     R1, 0100h
    MOV     R3, R1
    AND     R3, FF00h
    CMP     R3, 0000h
    BR.NZ   esccolcima ;ciclo que escreve a parte de cima
;da coluna
    MOV     R1, M[SP+6]
    ADD     R1, 0600h
    MOV     R2, M[SP+7]
    MOV     M[JANELA_CONTROLO], R1
    PUSH    R2
    CALL    EscreveCar
    ADD     R1, 0100h
    MOV     R3, R1
    AND     R3, FF00h
    CMP     R3, 1700h
    BR.NZ   esccolbaixo; ciclo que escreve a parte de baixo
;da coluna
    ENI
    POP     R4
    POP     R3
    POP     R2
    POP     R1
    RETN    2

;GERACOL: rotina que gera uma nova coluna
;
;      Entradas:---
;      Saidas:---
;
;      Efeitos: atualiza os conteudos das memorias que contem as

```

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 10/19

;
;      posicoes das colunas.
GERACOL:
    PUSH    R1
    PUSH    R2
    PUSH    R3
    DSI
    MOV     R1, M[NUM_OBST]
    MOV     R2, colunai
    ADD     R2, R1
    MOV     R1, M[R2]
    MOV     M[R2+1], R1
    DEC     R2
    CMP     R2, colunai
    BR.NN   CICLOGERA
    CALL    RANDOM
    MOV     R3, M[PROXIMONUM]
    SHL     R3, 8
    ADD     R3, 004Eh
    MOV     M[colunai], R3
    MOV     R1, M[NUM_OBST]
    CMP     R1, 000Fh
    BR.Z    TERMINA
    INC     M[NUM_OBST]
    MOV     M[ATIVGERACOL], R0
    ENI
    POP     R3
    POP     R2
    POP     R1
    RET

;RANDOM: rotina que gera um novo numero pseudo aleatorio em um dado intervalo
;
;      Entradas:---
;      Saidas:---
;
;      Efeitos: altera M[NIMOD] e M[PROXIMONUM]
RANDOM:
    PUSH    R3
    PUSH    R4
    DSI
    MOV     R3, M[NIMOD]
    AND     R3, 0001h
    CMP     R3, R0
    BR.NZ   CONDICA0
    MOV     R3, M[NIMOD]
    BR      ACABA
CONDICA0:
    MOV     R3, M[NIMOD]
    MOV     R4, MASCARARAND
    XOR     R3, R4
    ROR     R3, 1
    MOV     M[NIMOD], R3
    MOV     R4, LIMSUP
    DIV     R3, R4
    CMP     R4, R0
    BR.NZ   NAOZERO
    ADD     R4, LIMINF
    MOV     M[PROXIMONUM], R4
    ENI
    POP     R4
    POP     R3
    RET

;DelPassaro: rotina que apaga o passaro do ecran
;
;      Entradas: R7
;
;      Saidas:---
;
;      Efeitos:---

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 11/19

```

DelPassaro:  DSI
              MOV     M[JANELA_CONTROLO], R7
              PUSH    BLANK
              CALL    EscreveCar
              INC     R7
              MOV     M[JANELA_CONTROLO], R7
              PUSH    BLANK
              CALL    EscreveCar
              DEC     R7
              ENI
              RET

;EscPassaro: rotina que escreve o passaro no ecra
;
;      Entradas: R7
;      Saidas:---
;      Efeitos:---
EscPassaro:  PUSH     R1
              DSI
              MOV     M[JANELA_CONTROLO], R7
              PUSH    CORPO
              CALL    EscreveCar
              INC     R7
              MOV     M[JANELA_CONTROLO], R7
              PUSH    BICO
              CALL    EscreveCar
              DEC     R7
              ENI
              POP     R1
              RET

;Cair: rotina que escreve o passaro uma posicao abaixo no ecra
;
;      Entradas: R7
;      Saidas:---
;      Efeitos: aumenta o valor de R7 em 0100h
Cair:       PUSH     R1
              PUSH     R2
              PUSH     R3
              DSI
              MOV     R3, M[Cairn]
CicloCair:  MOV     R2, R7
              CMP     R2, YINFLIM
              JMP.NN  FIM; caso chegue ao limite inferior salta
;para o gameover
              CALL    DelPassaro
              MOV     R1, 0100h
              ADD     R7, R1
              CALL    EscPassaro
              MOV     R1, 0001h
              CMP     M[ATIVACOLISAO], R1
              PUSH    M[COLUMAATUAL]
              CALL.Z   COLISAO; caso a flag ATIVACOLISAO esteja ativa
; verifica se houve colisao na nova posicao do passaro
              POP     R1
              DEC     R3
              JMP.NN  CicloCair; apos a velocidade de queda ser 1 linha
; por ciclo passa a efetuar varias quedas seguidas do passaro consoante
; a aceleracao
              MOV     R1, CONTCAIR
              MOV     M[R1], R0
              ENI
              POP     R3
              POP     R2

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 12/19

```

              POP     R1
              RET

;Gravidade: rotina que altera a taxa de atualizacao da rotina Cair
;
;      Entradas:---
;      Saidas:---
;      Efeitos: altera M[TEMPOCAIR2]
Gravidade:  PUSH     R1
              PUSH     R2
              PUSH     R3
              DSI
              MOV     R1, TEMPOCAIR2
              MOV     R2, M[R1]
              MOV     R3, R2
              SUB     R3, GRAVIDADEINI
              CMP     R3, R0; verifica se ja chegou a velocidade de
;uma linha a cada 0.1s e aumenta o numero de vezes que cai por ciclo
              BR.NP   GravidadeFimB
              SUB     R2, GRAVIDADEINI
              MOV     M[R1], R2
              BR      GravidadeFimA
GravidadeFimB: INC     M[Cairn];
GravidadeFimA: MOV     R1, CONTGRAV
              MOV     M[R1], R0
              ENI
              POP     R3
              POP     R2
              POP     R1
              RET

;EscreveCar: rotina que escreve um caracter numa dada posicao do ecra
;
;      Entradas: pilha - caracter M[SP+3]
;      Saidas:---
;      Efeitos: escreve o dado caracter na posicao dada por
;              M[JANELA_CONTROLO]
EscreveCar: PUSH     R1
              MOV     R1, M[SP+3]
              MOV     M[JANELA_WRITE], R1
              POP     R1
              RETN    1

;EscreveStr: rotina que escreve uma cadeia de carcteres no ecra
;
;      Entradas: pilha:
;              M[SP+6] - endereco da string a escrever
;              M[SP+5] - posicao de escrita inicial no ecra
;      Saidas:---
;      Efeitos:---
EscreveStr: PUSH     R1
              PUSH     R2
              PUSH     R3
              DSI
              MOV     R1, M[SP+6] ; endereco da string a escrever
              MOV     R2, M[SP+5] ; posicao de escrita no ecra
              MOV     R3, M[R1]
              MOV     M[JANELA_CONTROLO], R2
              CMP     R3, FIM_TEXTO
              BR.Z     FimEsc
              PUSH     R3
              CALL    EscreveCar
              INC     R2
              INC     R1
              BR      Ciclo
Ciclo:

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 13/19

```

FimEsc:      ENI
             POP      R3
             POP      R2
             POP      R1
             RETN     2

;EscTraco: rotina que escreve os limites do campo de jogo
;           Entradas:---
;           Saidas:---
;           Efeitos:---
EscTraco:    DSI
             PUSH     strtrc
             PUSH     R0
             CALL     EscreveStr
             PUSH     strtrc
             PUSH     1700h
             CALL     EscreveStr
             ENI
             RET

;ATUALIZADIST: rotina que incrementa a distancia percorrida em colunas
;           Entradas:---
;           Saidas:---
;           Efeitos: altera M[contadordist] e M[contadordist2]
ATUALIZADIST: PUSH     R1
              PUSH     R2
              PUSH     R3
              DSI
              MOV      R1, 9999h
              CMP      M[Contadordist], R1
              BR.NZ    SALTODIST; verifica se e necessario incrementar o numero
mais significativo da distancia
              PUSH     contadordist2
              CALL     INCREMENTARNUM
              MOV      R1, 000Fh
              AND      M[contadordist2], R1
              MOV      R1, NUMEROAESC2
              MOV      R2, M[contadordist2]
              ADD      R2, '0'
              MOV      M[R1], R2
              MOV      R2, CONTROLLCD2
              DEC      R2
              PUSH     R1
              PUSH     R2
              CALL     ESCREVEDISPLAY
SALTODIST:   PUSH     Contadordist
              CALL     INCREMENTARNUM
              MOV      R3, NUMEROAESC
              ADD      R3, 0003h
              MOV      R2, M[Contadordist]
CICLOATUALI: MOV      R1, R2
              AND      R1, 000Fh
              ADD      R1, '0'
              MOV      M[R3], R1
              SHR      R2, 4
              DEC      R3
              CMP      R3, NUMEROAESC
              BR.NN    CICLOATUALI
              PUSH     NUMEROAESC
              PUSH     CONTROLLCD2
              CALL     ESCREVEDISPLAY

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 14/19

```

             ENI
             POP      R3
             POP      R2
             POP      R1
             RET

;ESCREVEDISPLAY: rotina que escreve no lcd a distancia percorrida
;           Entradas: pilha:
;           M[SP+5] - posicao de escrita do lcd
;           M[SP+6] - string a escrever
;           Saidas:---
;           Efeitos:---
ESCREVEDISPLAY: PUSH     R1
                PUSH     R2
                PUSH     R3
                DSI
                MOV      R1, M[SP+5] ; CONTROLO
                MOV      R2, M[SP+6] ; STRING
CICLODISPLAY: MOV      R3, FIM_TEXTO
                CMP      R3, M[R2]
                BR.Z     FIMDISPLAY
                MOV      R3, R1
                MOV      M[PONTEIRO_LCD], R3
                MOV      R3, M[R2]
                MOV      M[DISPLAY_LCD], R3
                INC      R2
                INC      R1
                BR       CICLODISPLAY
FIMDISPLAY:   ENI
                POP      R3
                POP      R2
                POP      R1
                RETN     2

;ApagaEcra: rotina que apaga todo o ecra
;           Entradas:---
;           Saidas:---
;           Efeitos:---
ApagaEcra:    PUSH     R1
              PUSH     R2
              DSI
              MOV      R1, strtrcBlank
              MOV      R2, R0
CicloApaga:   PUSH     R1
              PUSH     R2
              CALL     EscreveStr
              ADD      R2, 0100h
              CMP      R2, 1800h
              BR.NZ    CicloApaga
              ENI
              POP      R2
              POP      R1
              RET

;EscreveFim: rotina que escreve a janela de fim de jogo com pontuacao
;           Entradas: M[Pontuacao]
;           Saidas:---
;           Efeitos:---
EscreveFim:   PUSH     R1
              PUSH     R2
              PUSH     R3

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 15/19

```

        PUSH    R4
        DSI
        MOV     R3, M[Pontuacao]
        MOV     R1, strfim09
        MOV     R2, POSFIM01
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        MOV     R1, strfim0A
        ADD     R2, 0100h
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        MOV     R1, strfim0B
        ADD     R2, 0100h
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        MOV     R1, strfim0C
        ADD     R2, 0100h
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        MOV     R1, strfim0D
        ADD     R2, 0100h
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        MOV     R2, POSFIM02
        MOV     R4, 0004h
        ROL     R3, 4
        MOV     R1, R3
        AND     R1, 000Fh
        ADD     R1, '0'
        MOV     M[JANELA_CONTROLO], R2
        PUSH    R1
        CALL    EscreveCar
        INC     R2
        DEC     R4
        CMP     R4, R0
        BR.NZ   EscFimPont
        MOV     R1, strfim0D2
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        MOV     R1, strfim0E
        MOV     R2, POSFIM03
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        MOV     R1, strfim10
        ADD     R2, 0100h
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        MOV     R1, strfim11
        ADD     R2, 0100h
        PUSH    R1
        PUSH    R2
        CALL    EscreveStr
        ENI
        POP     R4

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 16/19

```

        POP     R3
        POP     R2
        POP     R1
        RET

;DiminuiDif: rotina que inicia o jogo ou diminui a dificuldade do jogo
;      Entradas:---
;      Saidas:---
;      Efeitos: inicio do jogo ou aumento da taxa de atualizacao
;              das colunas
DiminuiDif:  PUSH    R1
              PUSH    R2
              DSI
              MOV     R1, M[DIFICULDADE]
              MOV     R2, 12
              CMP     R1, R2
              BR.NN   FimDiminuiDif
              ADD     R1, FATOR
              MOV     M[DIFICULDADE], R1
              SHL     M[NIVEL], 1
              MOV     R1, M[NIVEL]
              MOV     M[DISPLAY_LEDS], R1
FimDiminuiDif: MOV     M[DimDif], R0
              ENI
              POP     R2
              POP     R1
              RET

;AumentaDif: rotina que aumenta a dificuldade do jogo
;      Entradas:---
;      Saidas:---
;      Efeitos: diminuicao da taxa de atualizacao das colunas
AumentaDif:  PUSH    R1
              PUSH    R2
              DSI
              MOV     R1, M[DIFICULDADE]
              MOV     R2, R1
              SUB     R2, FATOR
              CMP     R2, R0
              BR.NP   FimAumentaD
              SUB     R1, FATOR
              MOV     M[DIFICULDADE], R1
              SHRA    M[NIVEL], 1
              MOV     R1, M[NIVEL]
              MOV     M[DISPLAY_LEDS], R1
FimAumentaD:  MOV     M[CONTROLOCOL], R0
              MOV     M[AumDif], R0
              ENI
              POP     R2
              POP     R1
              RET

;Saltar: rotina que efetua o salto do passaro
;      Entradas:---
;      Saidas:---
;      Efeitos: alteracao do registo R7 e atualiza a posicao do passaro
;              no ecran
Saltar:      PUSH    R1
              PUSH    R2
              DSI
              CMP     R7, YSUPLIM

```


dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 17/19

```

BR.NP    Saltarfim
CALL     DelPassaro
SUB      R7, JMPISE
CALL     EscPassaro
MOV      R1, 0001h
CMP      M[ATIVACOLISAO], R1
PUSH     M[COLUMAATUAL]
CALL.Z   COLISAO
POP      R1
Saltarfim: MOV    R1, CONTGRAV
MOV      M[R1], R0
MOV      R1, TEMPOCAIR2
MOV      R2, GRAVIDADE0
MOV      M[R1], R2
MOV      R1, CONTCAIR
MOV      M[R1], R0
MOV      M[Salto], R0
MOV      M[Cairn], R0
ENI
POP      R2
POP      R1
RET

;ResetarJogo: rotina que reinicia todos os contadores e flags e recomeca o jogo
;
; Entradas:---
;
; Saidas:---
;
; Efeitos: reinicia as variaveis que influenciam o estado de jogo
ResetarJogo: MOV    M[ATIVADOR], R0
CALL     ApagaEcra
MOV      M[Pontuacao], R0
CALL     EscPontuacao
MOV      M[Contadordist], R0
MOV      M[contadordist2], R0
MOV      R1, 8000h
MOV      M[CONTROLOLCD], R1
MOV      M[NIMOD], R0
MOV      M[PROXIMONUM], R0
MOV      M[CONTGRAV], R0
MOV      M[CONTCAIR], R0
MOV      M[ATIVGERACOL], R0
MOV      M[CONTROLOCOL], R0
MOV      M[NUM_OBST], R0
MOV      R1, 13
MOV      M[DIFICULDADE], R1
MOV      R1, 8000h
MOV      M[NIVEL], R1
MOV      M[DISPLAY_LEDS], R1
MOV      R1, GRAVIDADE0
MOV      M[TEMPOCAIR2], R1
MOV      M[Cairn], R0
MOV      M[Salto], R0
MOV      M[DimDif], R0
MOV      M[AumDif], R0
MOV      M[ResetJogo], R0
MOV      M[ATIVACOLISAO], R0
MOV      M[COLUMAATUAL], R0
MOV      R1, colunal
MOV      R2, 17
ResetColunas: MOV    M[R1], R0
INC      R1
DEC      R2
CMP      R2, R0

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 18/19

```

BR.NZ    ResetColunas

Inicio:   MOV      R1, SP_INICIAL; inicializa o SP
MOV      SP, R1
MOV      R1, FFFFh
MOV      M[JANELA_CONTROLO], R1; inicializa o porto de controlo
MOV      R1, INT_MASK
MOV      M[INT_MASK_ADDR], R1; inicializa a mascara de interrupco

es        PUSH     stringdistancia; escreve os caracteres iniciais no lcd
PUSH     CONTROLOLCD
CALL     ESCREVEDISPLAY

PUSH     stringcolunas
PUSH     CONTROLOLCD3
CALL     ESCREVEDISPLAY

CALL     EscreveEcraIni

ENI
PreJogo:  INC      M[NIMOD]; gera um numero inicial a ser usado pela
;rotina RANDOM pseudo aleatorio
CMP      M[DimDif], R0
BR.Z     PreJogo
MOV      M[DimDif], R0
DSI

CALL     ApagaEcra

CALL     RANDOM
MOV      R1, 004Eh
MOV      R2, M[PROXIMONUM]
SHL      R2, 8
ADD      R1, R2
MOV      M[coluna1], R1

InicioJogo: CALL    EscTraco

MOV      R7, YBIRDi ; posicao inicial do passaro
CALL     EscPassaro

MOV      R1, INTERVALO ;ativa o temporizador
MOV      M[TEMPORIZADOR], R1
MOV      R1, BITATIV
MOV      M[ATIVADOR], R1
ENI
MOV      R1, M[NIVEL]; liga os leds do nivel de dificuldade inici

al        MOV      M[DISPLAY_LEDS], R1

CicloJogo: CMP      M[Salto], R0; verifica se o botao I0 foi clicado
CALL.P   Saltar

CMP      M[DimDif], R0; verifica se o botao I1 foi clicado
CALL.P   DiminuiDif

CMP      M[AumDif], R0; verifica se o botao I2 foi clicado
CALL.P   AumentaDif

```

dez 02, 15 15:38 C:\Users\rodri\Desktop\ProjetoPassaroBamb.as Page 19/19

```

MOV     R1, CONTCAIR
MOV     R2, M[TEMPOCAIR2]
CMP     M[R1], R2
CALL.NN Cair; move o passaro consoante a velocidade atual

MOV     R2, INTCAIR
MOV     R1, CONTGRAV
CMP     M[R1], R2
CALL.Z  Gravidade; a cada 0.7s aumenta a velocidade do passaro

MOV     R1, CONTROLOCOL
MOV     R2, M[DIFICULDADE]
CMP     M[R1], R2
CALL.Z  Colunas; move as colunas consoante a dificuldade
MOV     R2, M[ATIVGERACOL]
CMP     R2, 0005h
CALL.Z  GERACOL; gera uma nova coluna a cada 6 ciclos
JMP     CicloJogo

FIM:     CALL     EscreveFim
          CALL     EscPassaro
          MOV      M[ResetJogo], R0
          MOV      M[KillGame], R0
          MOV      M[AumDif], R0
          MOV      M[DimDif], R0
          MOV      M[Salto], R0
Final:   CMP      M[ResetJogo], R0; verifica se os botoes estao foram pres
sionados
;se for primido o botao I3 faz reset se for outro botao termina o programa
          JMP.NZ   ResetarJogo
          CMP      M[KillGame], R0
          BR.NZ    Final2
          CMP      M[AumDif], R0
          BR.NZ    Final2
          CMP      M[DimDif], R0
          BR.NZ    Final2
          BR       Final

Final2:   DSI
          CALL     ApagaEcra
          MOV      M[PONTEIRO_LCD], R0
          MOV      M[DISPLAY_LEDS], R0
          MOV      M[Pontuacao], R0
          CALL     EscPontuacao
Final3:   BR       Final3

```