

General Information
Author: Sam Teague
Language: ACTION!

MODULE; BACKTRAK.ACT

; by Sam Teague

BYTE Stick0=\$278, Trig0=\$D010,
ChBas=\$2F4, Time=\$14,
MemTop=\$6A, Color0=\$2C4,
Color1=\$2C5, Color2=\$2C6,
Color3=\$2C7, Color4=\$2C8,
GraCtl=\$D01D, GPrior=\$26F,
SDmaCtl=\$22F, PMBase=\$D407,
Nmien=\$D40E, WSync=\$D40A,
Con=\$D01F, Atract=\$4D,
HPos0=\$D000, HPos1=\$D001,
HPos2=\$D002, HPos3=\$D003,
PMColor0=\$2C0, PMColor1=\$2C1,
PMColor2=\$2C2, PMColor3=\$2C3,
P0PF=\$D004, P2PF=\$D006,
P3PF=\$D007, P0PL=\$D00C,
HPosM0=\$D004, HPosM1=\$D005,
HPosM2=\$D006, HPosM3=\$D007,
HitClr=\$D01E, CrSinh=\$2F0,
StickFlag, SpaceFlag,
Dots, OldMemTop, Screen, Speed,
X, Y, X0, Y0, X1, Y1, X2, Y2,
X3, Y3, OldX0, OldY0, OldX2,
OldY2, OldX3, OldY3,
RobotCount1, RobotCount2,
BirdCount, Image0, Image1,
DotSound, BonusSound, DoorSound,
Lives, StartSpeed

INT RobotSpeed1, RobotSpeed2,
BirdDist, RobotDist1,
RobotDist2, Score, HighScore

CARD VDslst=\$200, SDLst=\$230,
ScRam=\$58

BYTE POINTER PMRam, Player0, Player1,
Player2, Player3,
MissileBase

BYTE ARRAY String(10)

; GOOD GUY PLAYER

BYTE ARRAY

Player0Data0= ;DOWN
[0 0 0 24 36 36 36 24 60 24 0 0 0 0],
Player0Data1= ;UP
[0 0 0 24 60 60 60 24 60 24 0 0 0 0],
Player0Data2= ;LEFT
[0 0 0 24 44 44 44 24 56 24 0 0 0 0],
Player0Data3= ;RIGHT
[0 0 0 24 52 52 52 24 28 24 0 0 0 0]

```

; BIRD PLAYER
BYTE ARRAY
Player1Data0=
[0 0 0 24 36 60 24 219
 126 60 36 0 0 0],
Player1Data1=
[0 0 0 24 36 60 24 24
 126 255 36 0 0 0],
Player1Data2=
[0 0 0 24 36 60 24 24
 255 60 36 0 0 0]

; ROBOT PLAYER
BYTE ARRAY
Player2Data0=
[0 0 0 255 152 60 102
 219 189 36 66 0 0 0],
Player2Data1=
[0 0 0 60 56 60 102 219
 189 36 24 0 0 0],
Player2Data2=
[0 0 0 255 29 60 102 219
 189 36 24 0 0 0],
Player2Data3=
[0 0 0 60 28 60 102 219
 189 36 36 0 0 0]

;FACE, HANDS & FEET PLAYER
BYTE ARRAY
Player3Data0=
[0 0 0 0 36 36 36 0 189 0 153 0 0 0],
Player3Data1=
[0 0 0 0 36 36 36 0 189 0 153 0 0 0],
Player3Data2=
[0 0 0 0 36 36 36 0 1 0 44 0 0 0],
Player3Data3=
[0 0 0 0 36 36 36 0 128 0 52 0 0 0]

CARD ARRAY PMData0(4), PMData1(4),
           PMData2(4), PMData3(4),
           MissileData(4)

PROC Move_Set()
BYTE i, chnum, bit
BYTE POINTER base, dest
BYTE ARRAY chars=
[ 255 255 255 255 255 255 255 255
 127 127 127 127 127 127 127 127 0
 0 254 254 254 254 254 254 254
 254 254 254 254 254 254 254 0
 0 127 127 127 127 127 127 127
 0 255 255 255 255 255 255 255
 127 127 127 127 127 127 127 127
 254 254 254 254 254 254 254 254
 0 90 60 66 66 60 90 0
 0 255 231 129 129 231 255 0
 0 254 254 254 254 254 254 0
 0 0 0 24 36 24 0 0
 0 126 126 126 126 126 126 126

```

```

126 126 126 126 126 126 126 0
0 127 127 127 127 127 127 0
0 255 255 255 255 255 255 0
126 126 126 126 126 126 126 126
255 255 255 255 255 255 255 0
0 24 24 126 24 60 102 0]

```

```

MemTop== -8
Graphics(17)
SDmaCtl=0
base=(MemTop)*$100
MoveBlock(base,$E000,$400)
bit=0
chnum=1
DO
    dest=base+8*chnum
    FOR i=0 TO 7
        DO
            dest ^= chars(bit)
            dest == +1 bit == +1
        OD
    IF chnum=1 THEN chnum=3
    ELSEIF chnum<12 THEN chnum == +1
    ELSEIF chnum=12 THEN chnum=14
    ELSEIF chnum=14 THEN chnum=15
    ELSEIF chnum=15 THEN chnum=26
    ELSEIF chnum=26 THEN chnum=27
    ELSEIF chnum=27 THEN chnum=29
    ELSEIF chnum=29 THEN chnum=30
    ELSEIF chnum=30 THEN chnum=32
    ELSEIF chnum=32 THEN chnum=56
    ELSEIF chnum=56 THEN chnum=57
    FI
UNTIL chnum>56
OD
RETURN

```

```

PROC Dli=*( )
    BYTE color1=$D017, color2=$D018
    [$48          ; PHA
        $8D WSync] ; STA WSync
    color1=14      ; Luminance
    color2=4       ; Gray background
    [$68          ; PLA
        $40]      ; RTI

```

```

PROC SetDLI( )
    VDsLst=Dli
    Nmien=$C0
RETURN

```

```

PROC Delay(BYTE jiffies)
    jiffies == +Time
    DO UNTIL jiffies=Time OD
RETURN

```

```

BYTE FUNC CheckLocate(BYTE x0,y0)
    BYTE errflag
    errflag=0
    IF x0>53 AND x0<200 AND y0>26

```

```

        AND y0<200 THEN
            errflag=1
    FI
RETURN(errflag)

PROC Song()
    BYTE ctr
    FOR ctr=1 TO 2
        DO
            Sound(0,108,10,8) Delay(18)
            Sound(0,96,10,8) Delay(9)
            Sound(0,91,10,8) Delay(9)
            Sound(0,81,10,8) Delay(18)
            Sound(0,91,10,8) Delay(9)
            Sound(0,96,10,8) Delay(9)
        OD
        Sound(0,121,10,8) Delay(18)
        Sound(0,108,10,8) Delay(9)
        Sound(0,96,10,8) Delay(9)
        Sound(0,91,10,8) Delay(18)
        Sound(0,96,10,8) Delay(9)
        Sound(0,108,10,8) Delay(9)
        Sound(0,121,10,8) Delay(54)
        Sound(0,0,0,0) Delay(10)
    RETURN

PROC Color(BYTE C0, BYTE C1, BYTE C2,
           BYTE C3, BYTE C4)

    Color0=C0
    Color1=C1
    Color2=C2
    Color3=C3
    Color4=C4
RETURN

PROC Sype(BYTE ARRAY temp_str
          BYTE POINTER address)

    BYTE i
    BYTE ARRAY ataint=[$40 $00 $20 $60]
    FOR i=1 TO temp_str(0)
        DO
            IF temp_str(i)#155 THEN
                address^=ataint((temp_str(i)
                RSH 5)&3)%(temp_str(i)&$9F)
                address==+1
            FI
        OD
    RETURN

PROC High_Score()
    StrI(HighScore,String)
    Sype(String,ScRam+493)
    Sype(" ",ScRam+468)
RETURN

PROC BGNoise()
    BYTE N=[40], S=[50], bgcount=[0]
    bgcount==+1
    IF bgcount=5 THEN

```

```

Sound(0,N,10,2)
N==+2
IF N>=S THEN
  N=-4
  S=N
  IF S=40 THEN
    S=50
  FI
FI
bgcount=0
FI
RETURN

```

```

PROC Inst()
Graphics(0)
CrSinh=1
Color(0,8,0,8,0)
SDmaCtl=$00
Position(0,1) Print(" ")
  Sype("BACKTRACK",ScRam+14)
  Sype("Your Mission is to get all of the",ScRam+81)
  Sype("pulsating globules in the maze with",ScRam+121)
  Sype("as little backtracking as possible.",ScRam+161)
  Sype("Each globule is worth one point. Each",ScRam+201)
  Sype("empty space will cost you one point.",ScRam+241)
  Sype("You have 3 lives. If you touch one of",ScRam+281)
  Sype("the robots you lose a life. An extra ",ScRam+321)
  Sype("life is awarded at 3,000 points. If",ScRam+361)
  Sype("you touch the bird it will cost you 50",ScRam+401)
  Sype("points. If you open one of the doors",ScRam+441)
  Sype("in the walls you will recieve 100 extra",ScRam+481)
  Sype("points. If you touch a multi-color ",ScRam+521)
  Sype("bonus globule your score is increased",ScRam+561)
  Sype("by the number of globules you've ",ScRam+601)
  Sype("already gotten on that maze. You must ",ScRam+641)
  Sype("return to home base after getting all ",ScRam+681)
  Sype("the globules. There are 3 screens. ",ScRam+721)
  Sype("When you finish screen 3 you are ",ScRam+761)
  Sype("returned to screen 1 and the action ",ScRam+801)
  Sype("speeds up. Press the fire button to ",ScRam+841)
  Sype("pause. Move the stick to resume play.",ScRam+881)
  Sype("                PRESS START FOR GAME.",ScRam+921)
SDmaCtl=62
DO
UNTIL Con=6
OD
DO
UNTIL Con#6
OD
RETURN

```

```

PROC Title()
Graphics(17)
Color(118,50,22,8,0)
Sype("BACK TRACK",ScRam+65)

```

```

Type("by",ScRam+149)
Type("sam teague",ScRam+165)
Type("press option",ScRam+304)
Type("FOR LEVEL",ScRam+325)
Type("press select",ScRam+364)
Type("FOR INSTRUCTIONS",ScRam+382)
Type("press start",ScRam+424)
Type("TO PLAY",ScRam+446)
Speed=3
  DO
  UNTIL Con#6
  OD
  DO
  Delay(10)
  IF Con=3 THEN
    Type("                ",ScRam+244)
    Speed== -1
    IF Speed=0 THEN
      Speed=3
    FI
  FI
  IF Speed=3 THEN
    Type("BEGINNER",ScRam+246)
  ELSEIF Speed=2 THEN
    Type("INTERMEDIATE",ScRam+244)
  ELSEIF Speed=1 THEN
    Type("ADVANCED",ScRam+246)
  FI
  IF Con=5 THEN Inst() EXIT FI
  UNTIL Con=6 OR Trig0=0
  OD
StartSpeed=Speed
  DO
  UNTIL Con#6 AND Trig0#0
  OD
RETURN

```

```

PROC Reset()
  Zero(PMRam+$300,$500)
  X0=64 Y0=197 X1=47 Y1=29 X2=88
  Y2=101 X3=160 Y3=101
  RobotSpeed1=4 RobotSpeed2=4
  RobotDist1=0 RobotDist2=0
  RobotCount1=7 RobotCount2=7
  OldX0=X0 OldY0=Y0 OldX2=X2 OldY2=Y2
  OldX3=X3 OldY3=Y3 BirdCount=0
  BirdDist=0 StickFlag=7
  Image0=0 Image1=0
  DotSound=0 BonusSound=0 DoorSound=0
RETURN

```

```

PROC Hit_Bad_Guy()
  SndRst()
  DoorSound=0 BonusSound=0
  Sound(0,81,10,8) Delay(5)
  Sound(0,91,10,8) Delay(5)
  Sound(0,96,10,8) Delay(5)
  Sound(0,108,10,8) Delay(5)
  Sound(0,121,10,8) Delay(10)

```

```

    SndRst()
RETURN

PROC Modify_Display_List()
    CARD dlistaddr=$230
    BYTE POINTER dlistpntr
    dlistpntr=dlistaddr+26
    dlistpntr^==+128 ; Set DLI
    dlistpntr=dlistaddr+28
    dlistpntr^=2 ; Graphics 0 line
RETURN

PROC Init()
    Screen=1 Lives=3 Score=0 Dots=0
    PMData0(0)=Player0Data0
    PMData0(1)=Player0Data1
    PMData0(2)=Player0Data2
    PMData0(3)=Player0Data3
    PMData1(0)=Player1Data0
    PMData1(1)=Player1Data2
    PMData1(2)=Player1Data1
    PMData1(3)=Player1Data2
    PMData2(0)=Player2Data0
    PMData2(1)=Player2Data1
    PMData2(2)=Player2Data2
    PMData2(3)=Player2Data3
    PMData3(0)=Player2Data0
    PMData3(1)=Player2Data1
    PMData3(2)=Player2Data2
    PMData3(3)=Player2Data3
    MissileData(0)=Player3Data0
    MissileData(1)=Player3Data1
    MissileData(2)=Player3Data2
    MissileData(3)=Player3Data3
    PMRam=(MemTop)*$100
    MissileBase=PMRam+$300
    Player0=PMRam+$400
    Player1=PMRam+$500
    Player2=PMRam+$600
    Player3=PMRam+$700
    Zero(PMRam+$300,$500)
    ChBas=MemTop PMBase=MemTop
    Modify_Display_List()
    GPrior=17 GraCtl=3
    PMColor0=66 PMColor1=34
    PMColor2=6 PMColor3=162
    Reset()
    High_Score()
RETURN

PROC Maze1()
    SDmaCtl=$00
    Sype(" !@@@@@@@@@@@@@@@@@! ", ScRam)
    Sype(" ", ScRam+20)
    Sype("&' '$&' '$ ( ", ScRam+40)
    Sype("#!!!%&' '$#!!!% ( ", ScRam+60)
    Sype(")(!)(!!)(!)( ", ScRam+80)
    Sype("!$#@%&!!!!$#@%&! ", ScRam+100)
    Sype("!)(!!!!)(! ", ScRam+120)

```



```

Type(")(",ScRam+200)
Type(")(",ScRam+220)
Type("!'!!!!',;!!!!!",ScRam+240)
Type(" #!@%&' '$#@!% ",ScRam+260)
Type(" )&!!!!$( ",ScRam+280)
Type(" )&'!!!!!!'$( ",ScRam+300)
Type(" )(!!!!!!!( ",ScRam+320)
Type(" )#@@@@@@@@%( ",ScRam+340)
Type(" &)( $ ",ScRam+360)
Type(" !@@@'==,;=='!!!!",ScRam+380)
Type(" ):(!!!!",ScRam+400)
Type(" )!!!!",ScRam+420)
Type(" !!!!!!!!!!!!!!!",ScRam+440)
SDmaCtl=62
RETURN

```

```

PROC Life_Display()
  BYTE ctr
  Type(" ",ScRam+477)
  FOR ctr=1 TO Lives
    DO
      Type("X",ScRam+476+ctr)
    OD
RETURN

```

```

PROC Over()
  Type("*****",ScRam+103)
  Type("* game over *",ScRam+123)
  Type("*press start*",ScRam+143)
  Type("*****",ScRam+163)
  DO
    UNTIL Con=6 OR Trig0=0
  OD
  IF Score>HighScore THEN
    HighScore=Score
  FI
  Score=0
  Dots=0
  Lives=3
  High_Score()
  Screen=1
  Speed=StartSpeed
  Color(194,10,240,246,0)
  Maze1()
  Life_Display()
  DO
    UNTIL Trig0#0 AND Con#6
  OD
RETURN

```

```

PROC NextMaze()
  Reset()
  SndRst()
  Song()
  Dots=0
  Screen==+1
  IF Screen=2 THEN
    Color(114,8,240,246,0)
    Maze2()
  ENDIF

```

```

ELSEIF Screen=3 THEN
  Color(20,10,240,246,0)
  Maze3()
ELSEIF Screen=4 THEN
  Speed== -1
  IF Speed<1 THEN
    Speed=1
  FI
  Color(194,10,240,246,0)
  Screen=1
  Maze1()
FI

```

```
RETURN
```

```
PROC PosPlayer()
```

```

  HitClr=0
  HPosM0=X0
  HPosM1=X0+2
  HPosM2=X0+4
  HPosM3=X0+6
  HPos0=X0
  MoveBlock(Player0+Y0,PMData0(Image1)
            ,14)
  HPos1=X1
  MoveBlock(Player1+Y1,PMData1(Image0)
            ,14)
  HPos2=X2
  MoveBlock(Player2+Y2,PMData2(Image0)
            ,14)
  HPos3=X3
  MoveBlock(Player3+Y3,PMData3(Image0)
            ,14)
  MoveBlock(MissileBase+Y0,
            MissileData(Image1),14)

```

```
RETURN
```

```
PROC Score_Display()
```

```

  StrI(Score,String)
  Sype("          ",ScRam+468)
  Sype(String,ScRam+468)

```

```
RETURN
```

```
PROC Flash()
```

```

  BYTE blink
  blink==+1
  IF blink>7 THEN
    IF Color1>10 THEN Color1=10
    ELSE Color1=54
  FI
  Color2=Rand(255)
  blink=0
FI

```

```
RETURN
```

```
PROC Sounds()
```

```

  IF DoorSound THEN
    DoorSound== -1
    IF DoorSound<15 THEN
      Sound(2,72,10,8)
    
```

```

        Sound(3,60,10,8)
    ELSE
        Sound(2,60,10,8)
        Sound(3,47,10,8)
    FI
ELSEIF BonusSound THEN
    BonusSound== -1
    IF BonusSound < 15 THEN
        Sound(2,29,10,8)
        Sound(3,35,10,8)
    ELSE
        Sound(2,35,10,8)
        Sound(3,45,10,8)
    FI
ELSE
    Sound(2,0,0,0)
    Sound(3,0,0,0)
FI
IF DotSound THEN
    DotSound== -1
    Sound(1,125,10,8)
ELSE
    Sound(1,0,0,0)
FI
RETURN

PROC Pause()
    SndRst()
    DO
        UNTIL Stick(0) <> 15
    OD
RETURN

PROC Animate()
    BYTE count
    count== +1
    IF count > 4 THEN
        count=0
        Image0== +1
        IF Image0=4 THEN
            Image0=0
        FI
    FI
RETURN

PROC Move_Stick()
    BYTE stickdir
    StickFlag== +1
    IF StickFlag > 7 THEN
        ; Go through tunnel
        IF X0 < 48 THEN X0=200
        ELSEIF X0 > 200 THEN X0=48
        FI
        OldX0=X0 OldY0=Y0
        stickdir=Stick0
        StickFlag=0
        IF SpaceFlag=1 THEN
            X=(X0-46)/8 Y=(Y0-24)/8
            IF X0 < 80 AND Y0 > 184 THEN

```

```

        Score==+1
    FI
    IF CheckLocate(X0,Y0)=1 THEN
        IF Locate(X,Y)=' THEN
            Score== -1
        FI
    FI

    FI
    IF stickdir=15 OR stickdir=10
        OR stickdir=6 OR stickdir=9
        OR stickdir=5 THEN
        SpaceFlag=0
    ELSE SpaceFlag=1 Atract=0
    FI
    FI
    IF stickdir=14 THEN
        Y0== -1 Imagel=1
    ELSEIF stickdir=7 THEN
        X0==+1 Imagel=3
    ELSEIF stickdir=13 THEN
        Y0==+1 Imagel=0
    ELSEIF stickdir=11 THEN
        X0== -1 Imagel=2
    FI
RETURN

```

```

PROC Move_Bird()
    BYTE dir, time, count=[0]
    count==+1
    IF BirdCount THEN
        BirdCount== -1
    FI
    IF count=Speed THEN
        IF BirdDist<1 THEN
            dir=Rand(9)
            BirdDist=Rand(40)
        FI
        IF BirdCount=0 THEN
            IF dir=1 THEN Y1== -1
            ELSEIF dir=2 THEN X1==+1 Y1== -1
            ELSEIF dir=3 THEN X1==+1
            ELSEIF dir=4 THEN X1==+1 Y1==+1
            ELSEIF dir=5 THEN Y1==+1
            ELSEIF dir=6 THEN X1== -1 Y1==+1
            ELSEIF dir=7 THEN X1== -1
            ELSEIF dir=8 THEN X1== -1 Y1== -1
            FI
        FI
        BirdDist== -1
        count=0
    FI
RETURN

```

```

INT FUNC ABS(INT i)
    IF i<0 THEN i=-i FI
RETURN(i)

```

```

PROC Move_Robot1()
    BYTE robotdir

```

```

INT disx, disy
RobotSpeed1==+1
  IF RobotSpeed1>Speed-1 THEN
    RobotSpeed1=0
    RobotCount1==+1
    IF RobotCount1>7 THEN
      RobotCount1=0
      IF X2<48 THEN X2=200
      ELSEIF X2>200 THEN X2=48
      FI
      OldX2=X2 OldY2=Y2
      RobotDist1==-1
      IF RobotDist1<1 THEN
        RobotDist1=Rand(8)
        IF Rand(3)#2 THEN
          disx=ABS(X0-X2)
          disy=ABS(Y0-Y2)
          IF disx>disy THEN
            IF X2>X0 THEN
              robotdir=4
            ELSE
              robotdir=2
            FI
          ELSE
            IF Y2>Y0 THEN
              robotdir=1
            ELSE
              robotdir=3
            FI
          FI
          ELSE robotdir=Rand(5)
        FI
      FI
    FI
  ;Adjust coordinates for robot
  IF robotdir=1 THEN Y2==-1
  ELSEIF robotdir=2 THEN X2==+1
  ELSEIF robotdir=3 THEN Y2==+1
  ELSEIF robotdir=4 THEN X2==-1
  FI
FI
RETURN

```

```

PROC Move_Robot2()
BYTE robotdir
INT disx, disy
RobotSpeed2==+1
  IF RobotSpeed2>Speed-1 THEN
    RobotSpeed2=0
    RobotCount2==+1
    IF RobotCount2>7 THEN
      RobotCount2=0
      IF X3<48 THEN X3=200
      ELSEIF X3>200 THEN X3=48
      FI
      OldX3=X3 OldY3=Y3
      RobotDist2==-1
      IF RobotDist2<1 THEN
        RobotDist2=Rand(8)

```

```

        IF Rand(3)#2 THEN
            disx=ABS(X0-X3)
            disy=ABS(Y0-Y3)
            IF disx>disy THEN
                IF X3>X0 THEN
                    robotdir=4
                ELSE
                    robotdir=2
                FI
            ELSE
                IF Y3>Y0 THEN
                    robotdir=1
                ELSE
                    robotdir=3
                FI
            FI
        ELSE robotdir=Rand(5)
        FI
    FI
    ;Adjust coordinates for robot
    IF robotdir=1 THEN Y3==--1
    ELSEIF robotdir=2 THEN X3==+1
    ELSEIF robotdir=3 THEN Y3==+1
    ELSEIF robotdir=4 THEN X3==--1
    FI
FI
RETURN

```

```

BYTE FUNC Hit_Man(BYTE playfield)
RETURN(P0PF&playfield)

```

```

BYTE FUNC Hit_Robot1(BYTE playfield)
RETURN(P2PF&playfield)

```

```

BYTE FUNC Hit_Robot2(BYTE playfield)
RETURN(P3PF&playfield)

```

```

PROC Collisions()
CARD offset
X=(X0-46)/8 Y=(Y0-24)/8
offset=(Y*20)+X
;Man hits super globule
    IF CheckLocate(X0,Y0)=1 THEN
        IF Hit_Man(4) THEN
            IF Locate(X,Y)='*' THEN
                Sype(" ",ScRam+offset)
                Score==+Dots
                BonusSound=30
            FI
        ;Man hits door
        ELSEIF Hit_Man(8) THEN
            IF Locate(X,Y)=' ' THEN
                Score==+101
                Sype(" ",ScRam+offset)
                DoorSound=30
            FI
        ;Man hits dot
        ELSEIF Hit_Man(2) THEN

```

```

        IF Locate(X,Y)=' THEN
            Sype(" ",ScRam+offset)
            Score==+2
            Dots==+1
            DotSound=2
        FI
    FI
    FI
;Man hits wall
    IF Hit_Man(1) THEN
        X0=OldX0
        Y0=OldY0
        StickFlag=7
        Score==+1
        IF X0<80 AND Y0>184 THEN
            Score== -1
        FI
    FI
;Robot1 hits wall
    IF Hit_Robot1(1) THEN
        X2=OldX2 Y2=OldY2
        RobotCount1=7 RobotDist1=1
    FI
;Robot1 hits door
    IF Hit_Robot1(8) THEN
        X=(X2-46)/8 Y=(Y2-24)/8
        IF Locate(X,Y)=' THEN
            offset=(Y*20)+X
            Sype(" ",ScRam+offset)
        FI
    FI
;Robot2 hits wall
    IF Hit_Robot2(1) THEN
        X3=OldX3 Y3=OldY3
        RobotCount2=7 RobotDist2=1
    FI
;Robot2 hits door
    IF Hit_Robot2(8) THEN
        X=(X3-46)/8 Y=(Y3-24)/8
        IF Locate(X,Y)=' THEN
            offset=(Y*20)+X
            Sype(" ",ScRam+offset)
        FI
    FI
;Man hits bird
    IF P0PL=2 THEN
        Zero(PMRam+$500,$100)
        Score== -50 X1=47 Y1=29
        BirdCount=100
        Hit_Bad_Guy()
    FI
;Man hits a robot
    IF P0PL=4 OR P0PL=8 THEN
        Lives== -1
        Life_Display()
        Hit_Bad_Guy()
        Reset()
    FI
;Bird hits boundry

```

```
IF X1>192 THEN X1== -2 BirdDist=0 FI
IF X1<47 THEN X1== +2 BirdDist=0 FI
IF Y1>200 THEN Y1== -2 BirdDist=0 FI
IF Y1<29 THEN Y1== +2 BirdDist=0 FI
```

```
RETURN
```

```
PROC Checks()
```

```
  BYTE extra_life
```

```
    IF Score=0 THEN
      extra_life=0
```

```
    FI
```

```
    IF Trig0=0 THEN Pause() FI
```

```
    IF      Dots>172 AND X0<74 AND
              Y0>192 AND Screen=1 THEN
      NextMaze()
```

```
    ELSEIF Dots>177 AND X0<74 AND
              Y0>192 AND Screen=2 THEN
      NextMaze()
```

```
    ELSEIF Dots>252 AND X0<74 AND
              Y0>192 AND Screen=3 THEN
      NextMaze()
```

```
    FI
```

```
    IF extra_life=0 AND Score>2999 THEN
      Lives==+1
      Life_Display()
      extra_life=1
```

```
    FI
```

```
    IF Lives<1 THEN
      Life_Display()
      Over()
```

```
    FI
```

```
RETURN
```

```
PROC Game()
```

```
  Reset()
```

```
  DO
```

```
    Flash()
```

```
    BGNoise()
```

```
    Collisions()
```

```
    Move_Stick()
```

```
    Move_Bird()
```

```
    Move_Robot1()
```

```
    Move_Robot2()
```

```
    Animate()
```

```
    Sounds()
```

```
    PosPlayer()
```

```
    Score_Display()
```

```
    Checks()
```

```
    Delay(1)
```

```
    IF Con=6 THEN EXIT FI
```

```
  OD
```

```
RETURN
```

```
PROC Main()
```

```
  HighScore=0
```

```
  DO
```

```
    OldMemTop=MemTop
```

```
    Title()
```

```
    SDmaCtl=0
```



```
Move_Set()
Init()
SetDLI()
Color(194,8,240,246,0)
Maze1()
Type("SCORE",ScRam+462)
Type("HIGH SCORE",ScRam+482)
Life_Display()
PosPlayer()
SndRst()
Game()
    IF Score>HighScore THEN
        HighScore=Score
    FI
HPos0=0 HPos1=0 HPos2=0 HPos3=0
HPosM0=0 HPosM1=0 HPosM2=0 HPosM3=0
SndRst()
MemTop=OldMemTop
OD
RETURN
```