

Hearts and Bones v1.0

[hearts.zip](#)

This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.

Hearts.bas:

```

'*****
'*** Hearts and Bones v1.0  adapted from a game
'*** appearing on the HP 200LX palmtop computer.
'*** MMBasic version created by Hugh Buckle April 2012
'*** Requires MMBasic v3.2 or later and Hearts.fnt
'*****

'load hearts and bones font
Font Load "Hearts.fnt" As #4

'*** Grid definitions
x0=100:  y0=40      'grid offset (top left)
x1=x0-1:  y1=y0-4    'Cursor top left
x2=x1+16: y2=y1+18   'Cursor bottom right
xmin=0:   ymin=0     'Cell range, x=columns and y=rows
xmax=14:  ymax=8

'*** Cell definitions
UnmarkedCell =0      'These 3 variables are used to test a valid range of
values
HeartCell     =1      'in a cell using Int(cell(x,y)/10) thus testing
MarkedCell    =2      'with a single test
Bone          =40
MarkedBone    =50

'*** Default direction key assignments - Numeric Keypad
GoSub DefaultKeys

'*** Other variables
true=1
false=0
NumHearts = 8
StartBones = 20      ' default number of bones
Score=0
Dim cell(xmax+1,ymax+1)      ' stores cell content
Dim Stack((xmax+1)*(ymax+1)) ' used in revealing zero cells
Dim ZeroStack((xmax+1)*(ymax+1))
'*** date statements give x,y values for position of hearts
Data 0,4,0,8,7,0,7,4,7,8,14,0,14,4,14,8

Cls

```

```
GoSub PrintText
GoSub PrintGrid
GoSub PlaceHearts
'Load current player's HiScore, direction keys and starting bones
GoSub LoadGame
NumBones=StartBones
GoSub PrintInitialFnKeys
GoSub PrintHighScore
GoSub PrintScore
GoSub printBones
GoSub PrintMarks
Do
    GoSub GetKey
    GoSub ClearHelpText
    GoSub ProcessFnKeys
    If quit=true Then End
Loop Until Asc(k$)=145 ' F1 pressed
Line (0,0)-(90,19*12),0,BF
```

Start:

```
Do '*** Setup a new game
    GameEnd=false
    ZeroStackTop=0
    CorrectMarks=0
    GoSub ClearCells
    GoSub PrintText
    GoSub PrintGameFnKeys
    GoSub PrintHighScore
    GoSub PrintScore
    GoSub printBones
    GoSub PrintMarks
    GoSub PrintGrid
    GoSub PlaceHearts
    GoSub PlaceBones
    GoSub CountBones
    x=0: y=0 'Cursor to top left
    PrintNum(x,y,Cell(x,y))
    GoSub AddCursor
    If Cell(x,y)=0 Then GoSub FindAdjacentZeros

    '*** main game loop starts here
Do
    GoSub GetKey
    GoSub DeleteCursor
    GoSub ProcessMove
    GoSub AddCursor
    Loop While GameEnd=false And HeartsFound<NumHearts And Quit=false

    If quit=false Then
```

```
If GameEnd Then
    GoSub Finish
    Score=0
    NumBones=StartBones
    GoSub getKey
    If Asc(k$)=27 Then
        Quit=true
    EndIf
    GoSub ClearMessage
Else
    GoSub LevelCompleted
EndIf
EndIf
Loop Until quit=true

GoSub ClearMessage
Font #2,,1
Print @(13,0*10) " Bye "
Font #1,,0
Print @(5,3*10) " Thanks for"
Print @(5,4*10) " playing."
Print @(5,5*10) " Hope you had"
Print @(5,6*10) " fun."

Print @(0,300);
End

GetKey:
Do
    k$=Inkey$
Loop While k$=""
Return

LevelCompleted:
Font #1,,1
Print @(5,0*10) " LEVEL "
Print @(5,1*10) " COMPLETED "
Font #1,,0
If CorrectMarks=NumBones And CorrectMarks=Marks Then
    Score=Score+NumHearts
    If Score>HighScore Then
        HighScore=Score
    EndIf
    Print @(5,3*10) " All Hearts &"
    Print @(5,4*10) "Bones located."
    Print @(5,5*10) "BONUS 8 Hearts"
Else
    Print @(5,3*10) "Not all bones"
    Print @(5,4*10) " marked so"
    Print @(5,5*10) " NO BONUS"
EndIf
```

```
    NumBones=NumBones+2
Return

ProcessMove:
    '*** up left arrow
    If Asc(k$) = UL Then
        If x>xmin And y>ymin Then
            Move(-1,-1)
        Else
            GoSub MarkOff
        EndIf
    EndIf
    '*** up arrow
    If Asc(k$) = UN Or Asc(k$)= UA Then
        If y>ymin Then
            Move(0,-1)
        Else
            GoSub MarkOff
        EndIf
    EndIf
    '*** up right arrow
    If Asc(k$) = UR Then
        If x<xmax And y>ymin Then
            Move(1,-1)
        Else
            GoSub MarkOff
        EndIf
    EndIf
    '*** left arrow
    If Asc(k$) = LN Or Asc(k$)= LA Then
        If x>xmin Then
            Move(-1,0)
        Else
            GoSub MarkOff
        EndIf
    EndIf
    '*** right arrow
    If Asc(k$) = RN Or Asc(k$)= RA Then
        If x<xmax Then
            Move(1,0)
        Else
            GoSub MarkOff
        EndIf
    EndIf
    '*** down left arrow
    If Asc(k$) = DL Then
        If x>xmin And y<ymin Then
            Move(-1,1)
        Else
```

```
        GoSub MarkOff
    EndIf
EndIf
'*** down arrow
If Asc(k$) = DN Or Asc(k$)= DA Then
    If y<ymax Then
        Move(0,1)
    Else
        GoSub MarkOff
    EndIf
EndIf
'*** down right arrow
If Asc(k$) = DR Then
    If x<xmax And y<ymax Then
        Move(1,1)
    Else
        GoSub MarkOff
    EndIf
EndIf
'*** Space toggle mark
If Asc(k$) = MK Then
    If mark=1 Then
        GoSub MarkOff
    Else
        mark=1
        Print @(5,3*10) "    Press a"
        Print @(5,4*10) "direction key"
        Print @(5,5*10) "to mark a BONE"
    EndIf
EndIf
'*** F1 Re-Start
If Asc(k$) = 145 Then
    GoTo Start
EndIf
'*** Esc Quit
If Asc(k$) = 27 Then
    Quit=true
EndIf
' Clear Message area
If Not Mark Then
    GoSub ClearMessage
EndIf
Return

ProcessFnKeys:
'*** F1 Play
If Asc(k$) = 145 Then
    ' Do nothing - starts play

'*** F2 Increase Bones
ElseIf Asc(k$) = 146 Then
```

```
GoSub IncBones

'*** Shift/F2 Decrease bones
ElseIf Asc(k$) = 178 Then
    GoSub DecBones

'*** F3 Set Keys
ElseIf Asc(k$) = 147 Then
    GoSub SetKeys

'*** F4 Help
ElseIf Asc(k$) = 148 Then
    GoSub Help

'*** Esc Quit
ElseIf Asc(k$) = 27 Then
    Quit=true
EndIf
Return

PrintGrid:
    Line (x0-2,y0-5)-(x0-2+xmax*18+18,y0+15+ymax*20),0,BF 'Clear old grid
    For i = 0 To xmax
        For j = 0 To ymax
            Line (x0-2+i*18,y0-5+j*20)-(x0-2+i*18+18,y0+15+j*20),1,B
        Next j,i
    Return

DeleteCursor:
    ' remove old Cursor
    Line (x1+x*18,y1+20*y)-(x2+18*x,y2+20*y),0,b
    Line (x1+x*18+1,y1+20*y+1)-(x2+18*x-1,y2+20*y-1),0,b
Return

AddCursor:
    ' draw new Cursor
    Line (x1+x*18,y1+20*y)-(x2+18*x,y2+20*y),1,b
    Line (x1+x*18+1,y1+20*y+1)-(x2+18*x-1,y2+20*y-1),1,b
Return

ClearCells:
    For i=xmin To xmax
        For j= ymin To ymax
            cell(i,j)=0
        Next j,i
    Marks=0
    HeartsFound=0
Return
```

```
ClearMessage:
    Line (0,0)-(90,80),0,BF
Return

ClearHelpText:
    Line (0,19*12)-(MM.HRes,MM.VRes),0,BF
Return

PrintText:
    Font #2,1,1
    Print @(MM.HRes/2-149,0) "    Hearts and Bones    "
    Font #4
    Print @(MM.HRes-62,0) "$"
    Print @(MM.HRes-85,20) "&"
    Print @(MM.HRes-40,20) "%"
    Print @(MM.HRes-62,40) "$"
    Font #1
Return

PrintHighScore:
    Font #1,1,1
    Print @(MM.HRes-85,7*12) "High score"
    Font #1,1,0
    Print @(MM.HRes-75,8*12) Format$(HighScore,"%5g")
Return

PrintScore:
    Font #1,1,1
    Print @(MM.HRes-85,10*12) "    Score    "
    Font #1,1,0
    Print @(MM.HRes-75,11*12) Format$(Score,"%5g")
Return

PrintBones:
    Font #1,1,1
    Print @(MM.HRes-85,13*12) "    Bones    "
    Font #1,1,0
    Print @(MM.HRes-75,14*12) Format$(NumBones,"%5g")
Return

PrintMarks:
    Font #1,1,1
    Print @(MM.HRes-85,16*12) "    Marks    "
    Font #1,1,0
    Print @(MM.HRes-75,17*12) Format$(Marks,"%5g")
Return

PrintInitialFnKeys:
    Print @(10,7*12) "(F1)  Play"
    Print @(10,9*12) "(F2)  Bones"
    Print @(10,11*12) "(F3)  Keys"
```

```
Print @(10,13*12) "(F4)  Help"
Print @(10,17*12) "(Esc) Quit"
Return

PrintGameFnKeys:
  Print @(10,13*12) "(F1)  Re-Start"
  Print @(10,15*12) "(Esc) Quit"
Return

Sub PrintNum(x,y,Count)
  Print @(x0+x*18,y0+y*20) Count
End Sub

Sub PrintSprite(x,y,Txt$)
  Font #4
  Print @(x0+x*18,y0+y*20-3) Txt$
  Font #1
End Sub

Sub ClearSprite(x,y)
  Line (x1+x*18,y1+20*y) - (x2+18*x,y2+20*y),0,bf
End Sub

PlaceHearts:
  Restore
  Font #4
  For i = 1 To 8
    Read j,k
    cell(j,k)=HeartCell*10      '*** will later have number of adjacent
bones added
    PrintSprite(j,k,"$")
  Next
  Font #1
Return

PlaceBones:
  Cell(0,0)=1      '*** top left cell must not be a bone
  For i=1 To numbones
    Do
      j=Int(Rnd()*xmax)
      k=Int(Rnd()*ymax)
      Loop Until cell(j,k)=0      'ignore if bone or heart already there
      cell(j,k)=Bone              'place bone in cell
    Next
  Cell(0,0)=0      '*** clear top left cell
Return

CountBones:
  For i=xmin To xmax      'look at each cell
```



```

    For j=ymin To ymax
        If cell(i,j)=Bone Then      'add one to each adjacent non-bone cell
            For k=i-1 To i+1
                For l=j-1 To j+1
                    If k>=xmin And k<=xmax And l>=ymin And l<=ymax Then
                        If cell(k,l)<>Bone Then 'if adjacent cell not a bone
                            cell(k,l)=cell(k,l)+1
                            If Int(cell(k,l)/10)=HeartCell Then
                                PrintSprite(k,l,"$")
                            EndIf
                        EndIf
                    EndIf
                EndIf
            Next l,k
        EndIf
    Next j,i
Return

Sub Move(i,j)
    If Mark=1 Then
        MarkBone(i,j)
        'Move if target cell is bone, unmarked cell or heart
    ElseIf cell(x+i,y+j)=bone Or Int(cell(x+i,y+j)/10)=UnmarkedCell Or
Int(cell(x+i,y+j)/10)=HeartCell Then
        x=x+i
        y=y+j
        If Int(cell(x,y)/10)=HeartCell Then
            GoSub AddToScore
            '*** Once a heart cell has been visited,
            '*** it becomes an ordinary visited cell
            ClearSprite(x,y)
            cell(x,y)=Cell(x,y)+10*(UnmarkedCell-HeartCell)
        EndIf
        If Cell(x,y)=0 And NotOnZeroStack(x,y,ZeroStackTop) Then
            GoSub FindAdjacentZeros
        EndIf
        If cell(x,y)=Bone Then
            GameEnd=true
        Else
            PrintNum(x,y,Cell(x,y))
        EndIf
    EndIf
End Sub

Sub MarkBone (i,j)
    'You can mark only Bone and unmarked empty cells and
    'you can un-mark only marked cells.
    If cell(x+i,y+j)=Bone Then      'mark an existing bone
        CorrectMarks=CorrectMarks+1
        cell(x+i,y+j)=MarkedBone
        PrintSprite(x+i,y+j,"&")
        marks=marks+1
    End If
End Sub

```

```
ElseIf cell(x+i,y+j)=MarkedBone Then          'unmark an existing bone
  CorrectMarks=CorrectMarks-1
  cell(x+i,y+j)=Bone
  ClearSprite(x+i,y+j)
  Marks=Marks-1
ElseIf Int(cell(x+i,y+j)/10)=UnmarkedCell Then
  cell(x+i,y+j)=cell(x+i,y+j)+10*MarkedCell  'mark an empty cell
  PrintSprite(x+i,y+j,"&")
  Marks=marks+1
ElseIf Int(cell(x+i,y+j)/10)=MarkedCell Then
  cell(x+i,y+j)=cell(x+i,y+j)-10*MarkedCell  'unmark an empty cell
  ClearSprite(x+i,y+j)
  ' Don't show value 'cause it may not have been visited
  Marks=Marks-1
EndIf
GoSub PrintMarks
GoSub MarkOff
End Sub

FindAdjacentZeros:
  Savex=x          ' Save current cursor position
  Savey=y
  StackPtr=0
  StackTop=1
  Stack(StackTop)=x*100+y  ' put current location on stack

  Do ' If you step on a zero then all adjacent zeros are displayed
    ' and the boardering non-zero cells.
    StackAdded=false
    For i=x-1 To x+1
      For j=y-1 To y+1
        If i>=xmin And i<=xmax And j>=ymin And j<=ymax Then
          If Int(cell(i,j)/10)<>Heartcell And
Int(cell(i,j)/20)<>MarkedCell Then
            PrintNum(i,j,Cell(i,j))
            'EndIf
          EndIf
          If Cell(i,j)=0 Or cell(i,j)-10*HeartCell=0 Then
            GoSub IsCellOnStack
            If NotOnStack Then
              StackTop=StackTop+1          'inc stack pointer
              Stack(StackTop)=i*100+j      'code cell address as xxyy
              StackAdded=true
            EndIf
          EndIf
        EndIf
      Next j,i

      If Not StackAdded Then 'If no new zero cells, point at previous one
```

```
StackPtr=StackPtr-1 ' Otherwise go to the top of the stack
Else
StackPtr=StackTop
EndIf

'get cell x,y from top of stack
If stackptr >=0 Then
x=Int(stack(stackPtr)/100)
y=stack(Stackptr)-x*100
Else
stackPtr=0
EndIf
Loop Until StackPtr=0

GoSub AccumulateStack
If StackTop > MaxStackTop Then
MaxStackTop=StackTop
EndIf

x=Savex 'Restore the entry cell
y=Savey

Return

AccumulateStack:
For i=0 To StackTop
ZeroStack(ZeroStackTop)=stack(i)
ZeroStackTop=zeroStackTop+1
Next
Return

Function NotOnZeroStack(i,j,z)
NotOnZeroStack=true
For k=0 To z
If ZeroStack(k)=i*100+j Then
NotOnZeroStack=false
Exit For
EndIf
Next k
End Function

IsCellOnStack: 'Checks to see if the cell is already on the stack
NotOnStack=true
For k=0 To StackTop
If Stack(k)=i*100+j Then
NotOnStack=false
Exit For
EndIf
Next k
Return

AddToScore: 'Score one point for each Heart visited
```

```
HeartsFound=HeartsFound+1
Score=Score+1
GoSub PrintScore
If Score>HighScore Then
    HighScore=Score
    GoSub PrintHighScore
EndIf
Return

ShowAllBones:
    For i=0 To xmax
        For j=0 To ymax
            If Cell(i,j)=bone Then
                PrintSprite(i,j,"&")
            ElseIf Int(Cell(i,j)/10)=MarkedCell Then
                PrintSprite(i,j,"%")
            EndIf
        Next j
    Next i
Return

Markoff:
    mark=0
    Print @(12*10,MM.VRes-12*2) Space$(40)
Return

DefaultKeys:
    UL=55    ' Up Left
    UN=56    ' Up numeric
    UA=128   ' Up arrow
    UR=57    ' Up Right
    LN=52    ' Left numeric
    LA=130   ' Left Arrow
    RN=54    ' Right numeric
    RA=131   ' Right Arrow
    DL=49    ' Down Left
    DN=50    ' Down Numeric
    DA=129   ' Down Arrow
    DR=51    ' Down Right
    MK=32    ' Mark (space)
Return

IncBones:
    GoSub PrintF2Help
    NumBones=NumBones+1
    GoSub PrintBones
Return

DecBones:
```

```

GoSub PrintF2Help
If NumBones<19 Then
    Print
    Print Space$(10) "C'mon, let's not make it too easy!! 18 is small
enough."
Else
    NumBones=NumBones-1
    GoSub PrintBones
EndIf
Return

PrintF2Help:
    i=14                                ' Left indent
    j=MM.VRes-16*12                    ' Lines from bottom of screen
    Print @(0,j) Space$(i-1);
    Font #1,,1
    Print " Increase or reduce the starting number of Bones. "
    Font #1,,0
    Print
    Print Space$(i) "<F2> increses the number of starting Bones."
    Print Space$(i) "<shift+F2> reduces the number."
Return

Setkeys:
    i=14                                ' Left indent
    j=MM.VRes-17*12                    ' Lines from bottom of screen
    Print @(0,j) Space$(i-1);
    Font #1,,1
    Print " Set the direction keys and the key to mark a bone. "
    Font #1,,0
    Print
    Print Space$(i); "The defaults are the numeric keypad and spacebar."
    Print Space$(i); "Press <Esc> to exit without saving, <Enter> for
defaults,"
    Print Space$(i); "or follow the prompts to set your own direction keys."
    Print
    Print Space$(i); "Press a key for... Up Left   ? "; GoSub getkey
    i=i+19

    If Asc(K$)=13 Then
        GoSub defaultKeys
        Print "<F1>"
    ElseIf Asc(k$)<>27 Then
        UL=Asc(k$): Print K$
        Print Space$(i); "Up           ? "; GoSub getkey: UN=Asc(K$): Print k$
        Print Space$(i); "Up Right  ? "; GoSub getkey: UR=Asc(K$): Print k$
        Print Space$(i); "Left      ? "; GoSub getkey: LN=Asc(K$): Print k$
        Print Space$(i); "Right     ? "; GoSub getkey: RN=Asc(K$): Print k$
        Print Space$(i); "Down Left ? "; GoSub getkey: DL=Asc(K$): Print k$
        Print Space$(i); "Down      ? "; GoSub getkey: DN=Asc(K$): Print k$
        Print Space$(i); "Down Right? "; GoSub getkey: DR=Asc(K$): Print k$

```

```
Print Space$(i); "Mark Bone ? "; GoSub getkey: MK=Asc(K$): Print k$
EndIf
' Clear the text
GoSub ClearHelpText
Return

Help:
Print @(0,MM.VRes-16*12);
?" Your task is to capture all of the Hearts without stepping on any
Bones."
?" On capturing the last Heart, you move to a new level with 2 more
Bones."
?" The game ends when you step on a Bone. All Bones (skulls) and any"
?" incorrectly marked Bones (crossed bones) then are revealed."
?" You score one point for each Heart captured and if you correctly
mark all"
?" of the Bones, you score a bonus of 8 points for that level. You
forfeit"
?" the bonus if you leave a mark on a square that doesn't contain a
Bone."
?" You move the cursor using the numeric keypad and mark a Bone by
pressing"
?" the space bar followed by a direction key. Un-mark a Bone in the same
way."
?" Each square you step on reveals the number of Bones in adjacent
squares."
?" Before you start a game you can set the starting number of Bones."
?" Press <F2> to increase; <Shift+F2> to reduce. As you may not have a
numeric"
?" keypad, <F3> allows you to define direction keys and these are saved
at the"
?" end of the game along with your highest score and starting number of
Bones."
? Space$(17) "Good Luck! Press any key to exit Help."
Return

LoadGame:
Print @(0,21*12) Space$(10) "Please enter your name so I can load your
highest score,"
Print Space$(10) "starting number of bones and direction keys";
Input FName$
If Len(FName$)>8 Then FName$=Left$(FName$,8)
FName$=FName$+".xls"
Option Error Continue
Open FName$ For input As #1
If MM.Errno=0 Then
Input #1,HighScore
Input #1,UL,UN,UA,UR,LN,LA ' direction key assignments
```

```
    Input #1,RN,RA,DL,DN,DA,DR,MK
    Input #1,StartBones
    Close #1
EndIf
Option Error Abort
GoSub ClearHelpText
Return
```

SaveGame:

```
Open FName$ For output As #1
Print #1,HighScore
Print #1,UL","UN","UA","UR","LN","LA ' direction key assignments
Print #1,RN","RA","DL","DN","DA","DR","MK
Print #1,StartBones
Close #1
Return
```

Finish:

```
GoSub ShowAllBones
GoSub SaveGame
i=0:j=10
Font #2,,1
Print @(j,i*12) " Game "
Print @(j,(i+2)*12) " Over "
Font #1,,0
Print @(j-2,(i+4)*12) "No More Hearts"
Print @(j-2,(i+5)*12) "   For You   "
Font #2,,1
For k=1 To 3
    Pause 300
    Print @(j,i*12) Space$(6)
    Print @(j,(i+2)*12) Space$(6)
    Pause 300
    Print @(j,i*12) " Game "
    Print @(j,(i+2)*12) " Over "
Next
Font #1,,0
Print @(0,MM.VRes-6*12)
Return
```

From:

<https://fruitoftheshed.com/wiki/> - FotS

Permanent link:

https://fruitoftheshed.com/wiki/doku.php?id=mmbasic_original:hearts_and_bones_v1_0

Last update: **2024/01/19 09:39**

