



Introduction

The 1980's appeal and accessibility of Computer Games with home computers expanded rapidly as with their growing graphics capabilities. Character Fonts evolved into 2D creations called Sprites. These were represented by a matrix of tiny dots, pixels displayed to screen and were stored in memory as Bitmaps.

Creating Pixel Characters and Background Screens soon became a recognised Art form. Screens of the 1980's were typically 256 by 192 pixels, limiting detail and where the best Sprite Designers made recognisable characters within a 24x24 or less Pixel Grid.



QBITS PIXEL Art

The 1980's ambition was to write a Program in SuperBASIC to create simple RETRO Games for the Sinclair QL Platform. Unfortunately, the performance of Programs run with the QL Interpreter could be painfully slow and Sprite Bitmaps required lots of memory. Today with the extended range of QL Platforms and available Memory, the issue is more likely to be compatibility.

QBITS PIXEL Art Concept

Viewed as Four Stages the **First** was a SPRITE Designer with code taken from earlier **QBITS** Progs **BITMap Designer** and **QLFont Editor**. The code then extended to allow SPRITE _bmp files to Merge and the Import and ReSize QL 9x8 Font sets, maybe generated from other Retro Games. ReSizing of the Edit areas within a Frame Grid and ReColour then Copy Edit area or whole Grid to new Frame.

The **Second** Stage was Screen Backgrounds built with selected Frame designs copied across from the SPRITE Designer to form a Tile Library. A Tile Set as Solid, Decorative or act as Hazard or Reward. Plus a Map Link to manage Multiple Levels/Screens.

Stage **Three** Action settings for SPRITE movement by a Player or under Program control. Setting of Score & Lives counters with Hazard or Reward Gains or Losses. Plus an Editable Game Title.

Stage **Four** Play Checks on RETRO Game and a SAVE option as a standalone Program.

QBITS PIXELArt - Palette Choice

Developed with the QPC2 environment with two Colour Modes. The **COLOUR_QL** displays 0-7 as **Black Red Magenta Green Cyan Yellow White** with 8...255 as a Colour, Contrast and Stipple combination. **COLOUR_PAL** uses 0...255 as colours and shades from a Palette based on 24Bit RGB values (sixteen million variations).



Select with Left [QL8 or PAL] Right Cursor and Enter.
Note: Default is QL8 [Reverting to Mode 4 for BBQL]

QBITS PIXArt - File Management

Press (F) to change File **DIR**ectory Select Drive and Enter.
Press (E) to Edit **Drive/SubDIR**ectory name. Enter to Action.

Press (L)oad - Select Drive with Up/Down Cursors then Enter.
Select QBITS **_bmp _fnt** file with Up/Down Cursors. Press
Enter to Load or (M)erge a Sprite **_bmp** Abort with Spacebar.

(S)ave current **_bmp** File. (E)dit new filename. **SPRITE**
Mode has a basic header followed by BITMaps Frames.
SCREEN Mode has an extended header with Screen
MAPing information followed by **TILE** Library BITMaps.

RETRO Game will Save as a Standalone [WIPd]

If file exists an Overwrite Y/N prompt is displayed.

```
(F)DIR (L)oad (S)ave (W)uit
FDIR dos1_
Select + ↑ ↓ (E)dit + [ ] + =
```

```
(F)DIR (L)oad (S)ave (W)uit
Load vni_
Select + ↑ ↓
```

```
(F)DIR (L)oad (S)ave (W)uit
Load QB0Star.bmp
Select + ↑ ↓ = (M)erge
```

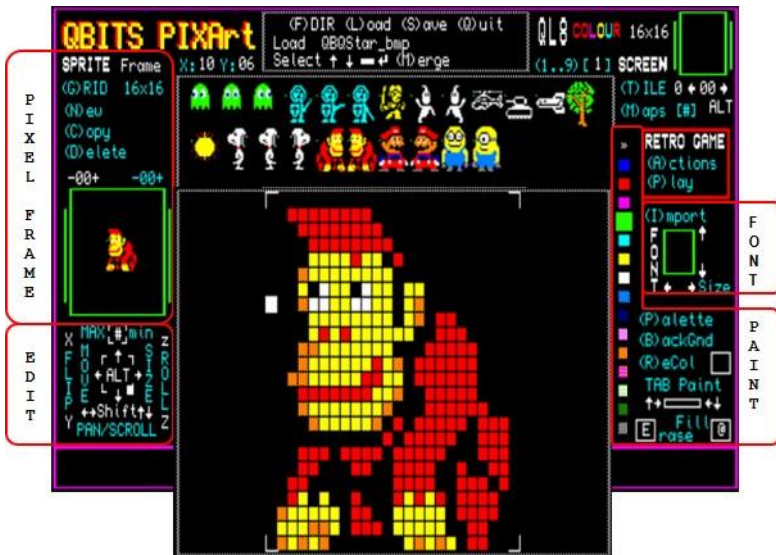
```
(F)DIR (L)oad (S)ave (W)uit
Save QBPIXArtGen3_01
Action = ↑ ↓ (E)dit + [ ] + =
```

```
(F)DIR (L)oad (S)ave (W)uit
Save QBPIXArtGen4_02.bmp
Action = ↑ ↓ (E)dit
```

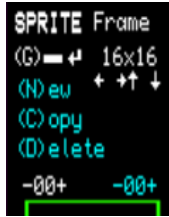
```
(F)DIR (L)oad (S)ave (W)uit
Save QBPIXArtGen4_02.bmp
Overwrite = ↑ ↓
```

QBITS PIXELArt - SPRITE Mode

Loading a **SPRITE_bmp** file will set the Grid Size. If Frame number is set to **-00+** the **SPRITES** are displayed in Pixel Grid Size as a group across central screen. Load a **QL FONT_fnt** file, set Size and Import to a **PIXEL** Art Frame.



QBITS PIXEL Art - (G)RID



Press 'G' for **SPRITE Mode** or to change **Frame** Gride use Cursors to size Columns and Rows. Abort **Spacebar** or Action **Enter**. Then you can use **-00+** to select a **SPRITE Frame**.

QBITS PIXEL Art - Frame

Press 'N' to add a **New Frame**.

Press 'C' to **Copy** present Frame to Another

Press 'D' to **Delete** current Frame.



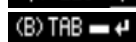
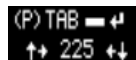
QBITS PIXEL Art - Paint

Select **Paint** Colour with **TAB/ShiftTAB** keys. Colour is shown as **Enlarged Square** that moves Up/Down the **Palette Bar**.

(P)alette 0..7 colours **Black Red Magenta Green Cyan Yellow White**

8..15 hold default QL8 or PAL colours. Press 'P' and use cursor keys to change nnn value and display.

[Abort with **Spacebar** or Action with **Enter**]



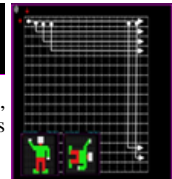
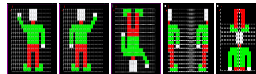
(B)ackGnd sets the background colour for all frames. Press 'B' Select a Colour from **Palette Bar**. Abort with **Spacebar** or **Enter** to Action.

Select **Grid** position [**Up** **Right** **Left** **Down**] Press **Spacebar** to toggle **Paint ON/OFF** moving the cursor extends this into other columns and/or rows. Press 'E' and move over Painted Cells to restore them to Background colour. Press '@' to Fill Frame Edit area with chosen colour or use with [E]rase to Reset cells to Background colour. To use (R)eCol - Set a New Colour **TAB** and Press 'R' then select the existing Frame colour you wish to change. Press **Enter** or **Spacebar** to Abort.

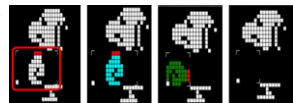


QBITS PIXEL Art - Edit

Press 'xX' to **FLIP** Horizontal or 'yY' on Vertical axis, 'zZ' to **ROLL** 90° clockwise or anticlockwise. To move cells within Frame Grid use **Shift**+Cursors to **PAN** or **SCROLL**.



Press '#' to MAX[#]min the Edit area of Frame Grid. The Edit Area can be Re-Positioned or Re-Sized (set Screen Cursor on a corner and drag) with **ALT**+Cursors **FLIP**, **ROLL**, **PAN**, **SCROLL**, **FILL** or (R)eColour within the Edit Area or all of Grid Frame.



Note: First Load a **FONT_fnt** File.

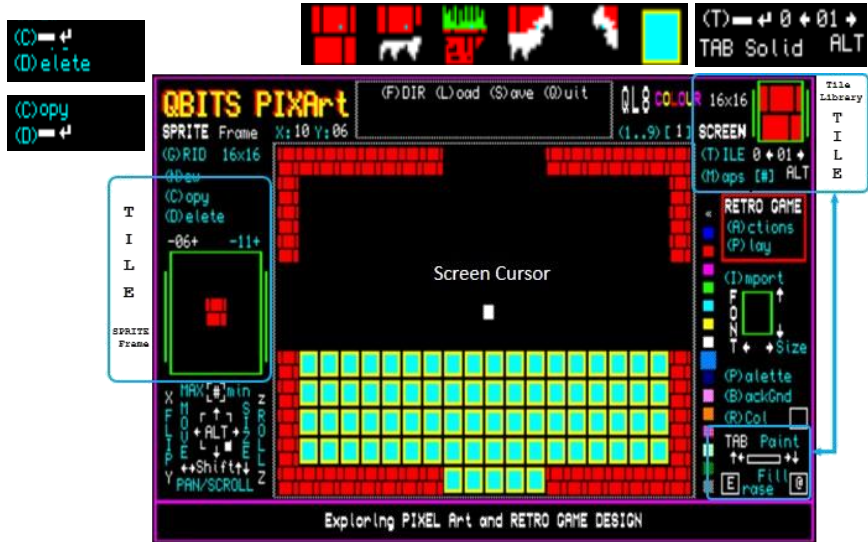


Press (I) Select **-/+** Font and use cursors **Up** **Right** **Left** **Down** to resize. Press **Enter** to Import and add to Frame Sequence or **Spacebar** to Abort.

Use **Edit Tools** with All or ReSized Grid Area and **Paint Options**, **ReColour** etc together with **Frame Copy** to build colourfully designed **SPRITES**.

QBITS PIXEL Art - SCREEN Mode

Load a SPRITE _bmp file Press 'T' for SCREEN Mode and then Select SPRITE Frame with -/+ Press (C) to Copy to TILE Library alternatively Load a SCREEN/TILE _bmp file. The Tile Image is shown in Frame top right with Library number below. Press (D) to Delete the Library Tile displayed. Press 'T' to change TILE Attribute, this determines Action when in contact with a moving Sprites. [Use **TAB** Key and Enter to Action or Spacebar to Abort]. The Tile Library can now be used to build Background Screens...



Select a Library Tile with **ALT** ← nn → keys and Position on screen with ↑ ↓ ← → then Toggle Spacebar **ON/OFF** to **Paint** individual Tile to SCREEN or extend as a row or column. Use **[E]**rase to Remove.

QBITS PIXEL Art - Maps

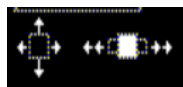


Press (M) for **MAP LINKS** to exit Press Spacebar/Enter.

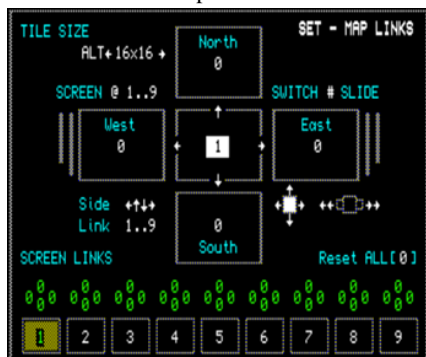
Change BackGnd Tile Size **ALT** ← →

Press @ to Select a Primary Screen **1.9**. Then use Cursors to Select a Compass direction and Set a Screen Link **1.9**.

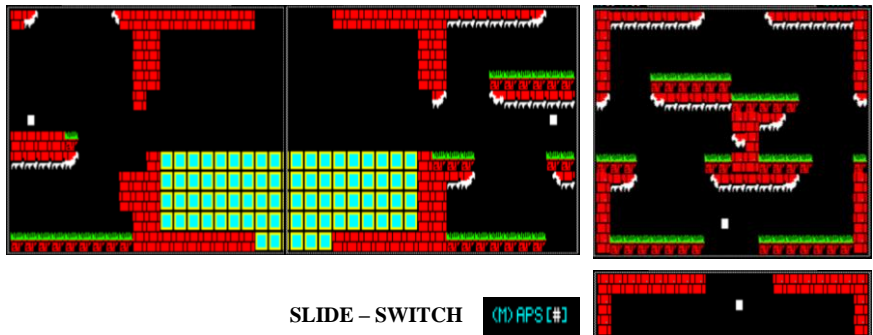
The bottom screen MAP displays changed Link Settings.



Select **SWITCH** or **SLIDE** with **[#]** Back in SCREEN Mode Press **[#]** and move Cursor to an EXIT point and Test SCREEN Links.



QBITS PIXEL Art - Screen Links



Note: Designing Tiles in one size and Displaying/Saving in another might not work as expected.

QBITS PIXEL Art - Actions

.

UNDER
CONSTRUCTION

QBITS PIXEL Art - BITMaps

The 1980's Games utilised character fonts with one colour. **QBITS Font Edit** Special Edition has three Demo Games that explored this approach. **QBITS BITMap Design** explored multi colour Objects with various Grid sizes. The BITMaps saved contained Headers to describe file Type and info such as Grid Size of Bitmap Data that followed.

QL FONT _fnt Files

For the QL default Font Character set the First Byte holds the Start Character code and second Byte the number of Character Bitmaps to follow ie 32/96

PIXEL Art _bmp Files [type sz]

IF **sz**=0:**bm**=**frt**:mlth= 8+**bm*****cm*****rm** :addr=ALCHP(mlth):ptr=addr+8
IF **sz**=1:**bm**=**frt**:mlth=16+**bm*****cm*****rm** :addr=ALCHP(mlth):ptr=addr+16
IF **sz**=2:**bm**=**tm**:mlth=52+2160+**bm*****256** :addr=ALCHP(mlth):cm=16:rm=16
IF **sz**=3:**bm**=**tm**:mlth=52+2160+200+**bm*****cm*****rm**+**bm**:addr=ALCHP(mlth):**tsz**=**cm**

bm BITmaps Max : **cm** Columns Max : **rm** Rows Max: **bg** BackGnd Colour :**sz** Type

SPRITE _bmp	frt Frame total	Bytes
Bytes 0... 7	QL8 or PAL bm cm rm bg sz	8
Bytes 8...15	Changeable Palette Colours	16
	bm*cm*rm SPRITE Bitmaps ???	
TILE _bmp	tm Tile Max	
Bytes 0.....15	QL8 PAL bm cm rm bg sz + Palette	16
Bytes 16.....51	Screen Links 9x4 = 36	52
Bytes 52.....2212	Screen Tiles 9x20x12 = 2160	2112
Bytes 2212....2412	Key CTRL 9x20 = 180 +	
	bm*cm*rm TILES Bitmaps ??? + bm TILE Action	

Note: **tsz** BackGnd Tile size

QBITS PIXEL Art uses sz=1 for SPRITE and sz=3 for SCREEN/Tile _bmp Files

WORK IN
PROGRESS

QBITS PIXEL Art - RETRO Play

The constructed Retro Game is accessed with **(P)**lay to check performing as expected.
[Esc] Returns to PIXEL Art for further changes.

UNDER
CONSTRUCTION

If all is Satisfactory Press '**S**' to **(S)**ave as standalone code, which can be Loaded and Run independently of QBITS PIXEL Art...

QBITS PIXEL Art WIPvA07

1000 REMark **QBITS_PIXELArt_WIPvA07** [QBITS PIXEL Art Game Design 2024 - QPC2] vA07

1002 dev\$='win1_':MODE 4:gx=0:gy=0 :REMark Basic Settings

1004 **WHEN ERor**:eck=1:CONTINUE:**END WHEN** :COLOUR_QL

1006 REMark **Import QBITSConfig Settings - QPC2**

1007 OPEN _IN#9,dev\$&'QBITSConfig':INPUT#9,gx\gy\dn\$\dev\$\dn%\dm%

1008 DIM drv\$(dm%,16):FOR d=0 TO dm%:INPUT#9,drv\$(d):END FOR d:CLOSE#9

1010 REMark **PIXEL Array Settings**

1011 DIM FG(32,64,64),TG(64,64),File\$(50,20),CP(15),BEEP\$(4,34)

1012 DIM Tile(60,24,24),TAss(120),TScn(9,40,24),WScn(20,12),TMap(9,4),SK(9,20)

1013 DIM TAss\$(9,6),RGNS\$(13),fnt\$(9,8),CP(15):FOR i=0 TO 7:CP(i)=i

1014 Fntaddr=ALCHP(1164):PFile\$="":TFile\$="":SFile\$=""

1015 QPC\$='COLOUR':QBT\$='QBITS PIXArt':QGT\$='TITLE':RGN\$='RETRO-MASTER'

1016 sg%=0:cg%=0:ch%=0:k=0:lks=0:csx=1:csy=1:tz=4:tsz=tsz&'x'&tsz

1018 COLOUR_QL:Init_Screens:Init_Palette:Init_Layout:PIXArt_Menu

1020 REMark **PIXArt SetUP**

1022 **DEFine PROCEDURE Init_Screens**

1023 LOCal w,d,x,y:**RESTORE 1028**

1024 WINDOW#0,512, 22,gx,gy+234:PAPER#0,0:BORDER#0,1,3:CLS#0

1025 WINDOW#1,508,234,gx+2,gy+1:PAPER#1,0

1026 WINDOW#2,512,234,gx,gy :PAPER#2,0:BORDER#2,1,3:CLS#2

1027 FOR i=3 TO 8,10,11:OPEN#,scr_:**READ w,d,x,y**: WINDOW#,w,d,gx+x,gy+y

1028 DATA 328,196,92,36, 72,68,12,96, 200,32,156,2, 90,196,2,36

1029 DATA 32,30,470,3, 90,196,420,36, 80,72,8,168, 24,24,456,118

1030 CSIZE#11,3,1:BORDER#5,1,248:BORDER#3,1,248

1031 **QTitle 5,2,24,2,QBT\$** : **QTitle 3,3,64,10,RGN\$**:CSIZE#3,1,1

1032 FOR a=1 TO 6:**READ bx,by,str\$** : **QBold 3,7,bx,by,str\$**:PAUSE 5

1033 DATA 100,35,', FONT IMPORT "',104,60,', SPRITE Gen "',108,85,', TILE Sets"

1034 DATA 108,110,', TILE MAPS "',100,135,', ACTION CTRL "

1035 DATA 54,160,'EXPORT / SAVE - RETRO GAME': PAUSE 100

1036 **END Define**

1038 **DEFine PROCEDURE Init_Palette**

1039 RESTORE 1051:CLS#5:CSIZE#5,0,1:CSIZE 1,1:pm\$='QL8':**QBold 1,7,360,2,pm\$**

1040 IF VER\$(1)>=2.98

1041 COLOUR_QL:INK#5,5:CUSOR#5,15,6:PRINT#5,'Select Palette '

1042 INK#5,7:PRINT#5,'QL8 ◀ ▶ PAL ◀ ▶:BLOCK#5,2,6,184,12,7

1043 **REPEAT P_lp**

1044 **QBold 1,7,360,2,pm\$**:k=CODE(INKEY\$(-1))

1045 IF k=192:pm\$='QL8' :**RESTORE 1051**:tcol%=255

1046 IF k=200:pm\$='PAL' :**RESTORE 1052**:tcol%=10

1047 IF k= 10:IF pm\$='PAL':COLOUR_PAL:END IF :**EXIT P_lp**

1048 **END REPEAT P_lp**

1049 END IF

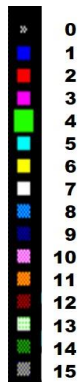
8...15 Default QL8 or PAL

1050 CSIZE#5,0,0:CSIZE 0,0:FOR i=0 TO 15:**READ CP(i)**

1051 DATA 0,1,2,3,4,5,6,7,225,200,227,230,210,31,224,254

1052 DATA 0,4,2,5,3,7,6,1,25,50,31,22,46,36,61,10

1053 **END Define**




```

1055 DEFINE PROCEDURE Init_Layout
1056 CSIZE 1,1 : QBold 1,7,360,2,pm$: QTitle 1,2,2,4,QBT$
1057 CSIZE 0,0:FOR i=1 TO 6:QBold 1,RND(2 TO 7),380+i*6,6,QPC$(i)
1058 CLS#5:DIRMenu:Info:PAUSE 200:CLS#3:CLS#4:CLS#7
1059 CLS#6:BLOCK 2,22,464,6,CP(4) :BLOCK 2,22,502, 6,CP(4):BORDER#7,1,CP(4)
1060 CLS#8:BLOCK 2,50,6,106,CP(4) :BLOCK 2,50,84,106,CP(4):BORDER#4,1,CP(4)
1061 BLOCK#8,14,12,20,182,CP(7) :BLOCK#8,14,12,73,182,CP(7)
1062 BLOCK#8,66,38,20,28,CP(2) :BLOCK#8,62,34,22,30,0
1063 QBold 1,7, 4,24,'SPRITE' :QBold 8,7,22,30,'RETRO GAME'
1064 QBold 1,7,420,24,'SCREEN' :OVER 1:CURSOR 451,24:PRINT 'N'
1065 str$=FONT':FOR i=1 TO 4:QBold 8,7,26,71+i*9,str$(i)
1066 QBold 8,7,62,78,'↑':QBold 8,7,62,98,'↓':QBold 8,7,36,107,'←' →'
1067 FOR i=0 TO 15:BLOCK#8,8,6,37+10*i,CP(i):END FOR i
1068 RESTORE 1069:FOR i=1 TO 22:QPrnt:END FOR i
1069 DATA 1,7,48,25,'Frame',1,5,92,25,'X': Y':1,5,356,25,'(1..9)[ ]'
1070 DATA 6,5,2,2,'(G)RID',6,5,4,14,'(N)ew',6,5,4,25,'(C)opy',6,5,4,36,'(D)elete'
1071 DATA 1,7,428,7,tz$&'x'&tz$,8,5,2,2,'(T)ILE',8,5,2,14,'(M)aps [#]'
1072 DATA 8,5,22,42,'(A)ctions',8,5,22,52,'(P)lay',8,5,20,70,'(I)mport'
1073 DATA 8,5,62,108,'Size',8,5,20,124,'(P)alette',8,5,20,135,'(B)ackGnd'
1074 DATA 8,5,20,147,'(R)Col',8,7,22,160,TAB',8,7,26,168,'↑' ←' →'↓'
1075 DATA 8,5,50,160,'Paint',8,5,36,186,'rase',8,5,48,179,'Fill'
1076 BORDER#11,1,CP(4):INK#8,CP(7):Init_Guide
1077 BLOCK#8,14,12,73,146,CP(7):BLOCK#8,12,10,74,147,0:BLOCK#8,26,5,40,172,CP(7)
1078 RESTORE 1079:FOR i=0 TO 9:READ TAS$(i)
1079 DATA 'Solid','Floor','Hazard','Reward','???','???','???','???'
1080 END DEFINE

```



```

1082 DEFINE PROCEDURE Init_Guide
1083 ch=10:INK#ch,CP(7):CURSOR#ch,0,4:PRINT#ch,'X' z:M1$='FLIPROLL'
1084 M2$='MOVESIZE' :CURSOR#ch,0,48:PRINT#ch,'Y' z:INK#ch,CP(5)
1085 FOR i=0 TO 3:CURSOR#ch,0,14+i*8:PRINT#ch,M1$(i+1),' 'M1$(i+5)
1086 FOR i=0 TO 3:CURSOR#ch,12,10+i*8:PRINT#ch,M2$(i+1),' 'M2$(i+5)
1087 BLOCK#ch,24,20,28,17,CP(7):BLOCK#ch,22,18,29,18,0:BLOCK#ch,16,20,32,17,0
1088 CURSOR#ch,12,0:PRINT#ch,'MAX min':BLOCK#ch,14,11,33,0,CP(7)
1089 BLOCK#ch,12,9,34,1,0:BLOCK#ch,14,3,33,4,0:BLOCK#ch,6,11,37,0,0
1090 BLOCK#ch,5,5,47,33,CP(7):RESTORE 1092
1091 FOR i=1 TO 7:READ pc,px,py,p$:INK#ch,CP(pc):CURSOR#ch,px,py:PRINT#ch,p$
1092 DATA 5,9,52,'PAN/SCROLL',7,12,42,'←' →' Shift ↑ ↓',7,37,1,'#'
1093 DATA 7,37,12,'↑' ↓',7,22,22,'←' →',7,31,23,'ALT',7,37,33,'↓'
1094 END DEFINE

```



```

1096 DEFine PROCEDURE QTitle(ch,chz,tx,ty,str$)
1097 OVER#ch,1:CSIZE#ch,chz,1
1098 INK#ch,CP(2):FOR i=0 TO 1:CURSOR#ch,tx+i,ty :PRINT#ch,str$
1099 INK#ch,CP(6):FOR i=2 TO 3:CURSOR#ch,tx+i,ty+1:PRINT#ch,str$
1100 OVER#ch,0:CURSOR#ch,0,0:CSIZE#ch,0,0
1101 END DEFINE

```



```

1103 DEFine PROCEDURE QBold(ch,bc,bx,by,B$)
1104 INK#ch,CP(bc):CURSOR#ch,bx,by:PRINT#ch,B$
1105 OVER#ch,1:CURSOR#ch,bx,by+1:PRINT#ch,B$:OVER#ch,0
1106 END DEFINE

```



```

1108 DEFine PROCEDURE QPrnt
1109 LOCAL ch,i,x,y,p$:READ ch,i,x,y,p$:INK#ch,CP(i):CURSOR#ch,x,y:PRINT#ch,p$
1110 END DEFINE

```



1112 DEFINE PROCEDURE DIRMenu

1113 CLS#5:INK#5,CP(7):CURSOR#5,16,1:PRINT#5,'(F)DIR (L)oad (S)ave (Q)uit'

1114 OVER#5,1:CURSOR#5,22,1:PRINT#5,'F L S Q':OVER#5,0

1115 END DEFINE

(F)DIR (L)oad (S)ave (Q)uit

(F)DIR (L)oad (S)ave (Q)uit
Drive dos!_
Select ↑ ↓ ← → (E)dit [] + =

Note: Press 'F' then 'E' to Edit Drive/SubDIR

(F)DIR (L)oad (S)ave (Q)uit
Load @B0Star.bmp
Select ↑ ↓ ← → (M)erge

To Load Press 'L' Select a Drive then Select File from List presented or Press 'M' to Merge with File loaded

(F)DIR (L)oad (S)ave (Q)uit
Save @BPIXaGen4_02.bmp
Action ← → (E)dit

To Save Press 'S' Select a Drive then 'E' to Edit Filename if required

1117 REMark PIXEL Art Menu

1119 DEFINE PROCEDURE PIXArt_Menu

1120 INK#8,CP(7):k=0:ic=4:oc=4:sic=4:bg=0:pn=0:tn=0:tm=0:sn=1:obj=0:lks=0

1121 frm=31:fr=0:frt=0:cm=16:rm=16:cur=0:rub=0:FrSet:GLoc:GFrame

1123 REPEAT Main_ip

1124 IF fed=1 AND fr>0

1125 IF rub=7:c=x:r=y:FG(fr,c,r)=255:GPBit

1126 IF cur=7:c=x:r=y:FG(fr,c,r)=ic:GPBit

1127 END IF

1128 IF fed=2 AND obj=0

1129 IF rub=7:tx=2+x*16:ty=1+y*16:BLOCK#3,16,16,tx,ty,CP(bg):TScn(sn,x,y)=0

1130 IF cur=7:tx=2+x*16:ty=1+y*16:TDraw 0:TScn(sn,x,y)=tn

1131 END IF

1132 BLOCK#8,24,3,41,173,CP(cur):INK CP(7):INK#8,CP(7):GCErase:OVER 0

1133 CURSOR 106,24:PRINT FILL\$(0',2-LEN(x+1));x+1

1134 CURSOR 136,24:PRINT FILL\$(0',2-LEN(y+1));y+1

1135 GPos:k=CODE(INKEY\$(-1)):ox=x:oy=y:GPos:GLoc

1136 SELECT ON k

1137 =32: IF cur=0:cur=7:rub=0:ELSE cur=0

:REMark SB ON/OFF

1138 =69,101:IF rub=0:rub=7:cur=0:ELSE rub=0

:REMark (E>raser

1139 =70,102:SelDrv 1,'FDIR 'DIRMenu

:REMark (F)DIR

1140 =76,108:BMLoad:DIRMenu

:REMark (L)oad

1141 =83,115:BMSave:DIRMenu

:REMark (S)ave

1142 =81,113:QExit:BLOCK#5,18,10,160,12,0

:REMark (Q)uit

1143 =47, 63:col%=CP(6):Info:PAUSE:IF fed=1:GFrame :ELSE GTile :REMark (?)Help

1144 END SELECT

1145 IF fed=1:SGen

1146 IF fed=2:TGen

1147 END REPEAT Main_ip

1148 END DEFINE

1150 DEFINE PROCEDURE GCErase

1151 BLOCK#8,12,10,21,183,CP(rub):OVER#8,-1:CURSOR#8,24,183:PRINT#8,'E':OVER#8,0

1152 BLOCK#8,12,10,74,183,CP(cur):OVER#8,-1:CURSOR#8,77,183:PRINT#8,'@':OVER#8,0

1153 END DEFINE

1155 DEFINE PROCEDURE GPos

1156 OVER#3,-1:BLOCK#3,gpw,gph,gpx,gpy,CP(7):OVER#3,0

1157 END DEFINE

1159 **DEfIne PROCEDURE GLoc**

1160 **SElect ON k**

1161 =192:IF x> 0:x=x-1 :REMark ← x \
1162 =200:IF x<xm:x=x+1 :REMark → x Grid Tile
1163 =208:IF y> 0:y=y-1 :REMark ↑ y Position
1164 =216:IF y<ym:y=y+1 :REMark ↓ y /

1165 **END SElect**

1166 IF fed=1:gpw=cw+1:gph=rh+1:gpx=-1+zx+x*cw:gpy=-1+zy+y*rh

1167 IF obj=1 AND lks=0:**Scn_Switch**

1168 IF obj=1 AND lks=1:**Scn_Slide**

1169 IF fed=2:gpw=tsz/2:gph=tsz/2:gpx=6+x*tsz:gpy=5+y*tsz

1170 **END DEfIne**

1172 **DEfIne PROCEDURE GFrame**

1173 BLOCK#8,30,10,54,1,0:BLOCK#8,18,10,70,12,0:**TNum 5:obj=0:HLObj 5:CLS#7**

1174 **INK#6,CP(7):FrGrid cm,rm:fed=1:sz=1:k=0:FrSet:FrNum frt,5,56,50**

1175 **END DEfIne**

Note: Switches to GRID Mode – Set TILE Mode Inactive - Change Settings and Show SPRITE Frame

1177 **DEfIne PROCEDURE GTile**

1178 **INK#6,CP(5):FrGrid cm,rm:QBold 8,7,54,1,' ← → 'tz=tsz/4:Tile_Sise**

1179 **CURSOR#8,70,12:PRINT#8,'ALT':TNum 7:tmax=120:fed=2:sz=3:k=0**

1180 **xm=320 DIV tsz:x=xm div 2:ym=192 DIV tsz:y=ym DIV 2**

1181 **xm=xm-1:ym=ym-1:cur=0:rub=0:obj=0:GLoc:HLObj 5:Tile_MAP**

1182 **END DEfIne**

Note: Switches to TILE Mode – Set GRID Mode Inactive - Change Settings for SCREEN/TILE Background

1184 **DEfIne PROCEDURE Retro_Act**

1185 **PAPER#3,0:CLS#3:BLOCK#3,170,48,80,46,CP(7):BLOCK#3,168,46,81,47,0**

1186 **QTitle 3,2,132,50,'UNDER':QTitle 3,2,90,70,'CONSTRUCTION'**

1187 **QBold 3,7,240,4,'SET - ACTIONS':PAUSE:k=32:GTile**

1188 **END DEfIne**

Note: Actions Under Construction

1190 **DEfIne PROCEDURE Retro_Play**

1191 **BLOCK 152,35,0,0,0:CLS#6:BLOCK 152,35,356,0,0:CLS#8**

1192 **CLS#3:CLS#5:Tile_MAP:INK CP(7):CSIZE 1,1**

1193 **CURSOR 50,2:PRINT 'SCORE: 99999':CURSOR 390,2:PRINT '00 :LIVES'**

1194 **CSIZE 0,0:CURSOR 98,24:PRINT '01:15:25':CURSOR 360,24:PRINT 'LEVEL:'**

1195 **CURSOR 4,24:PRINT '(L)oad (S)ave' :CURSOR 418,24:PRINT '(Q)uit (R)eset'**

1196 **FOR i=1 TO 6:BLOCK 6,120+i*10,i*12,72-i*5,i:BLOCK 6,120+i*10,502-i*12,72-i*5,i**

1197 **QTitle 5,2,36,4,'GAMEMASTER'**

1198 **BLOCK#3,170,48,80,46,CP(7):BLOCK#3,168,46,81,47,0**

1199 **QTitle 3,2,132,50,'UNDER':QTitle 3,2,90,70,'CONSTRUCTION':PAUSE**

1200 **BLOCK 152,34,0,0,0:CLS#6:BLOCK 152,34,356,0,0:CLS#8**

1201 **Init_Layout:k=32:IF fed=1:GFrame:ELSE GTile**

1202 **END DEfIne**

Note: Retro Play Under Construction

1204 **DEfIne PROCEDURE QExit**

1205 **CURSOR#5,160,12:PRINT#5,'Y/N':IF GAns=0:k=0:RETurn**

1206 **CLOSE#10:CLOSE#8:CLOSE#7:CLOSE#6:CLOSE#5:CLOSE#4:CLOSE#3:CLS#2:COLOUR_QL**

1207 **CURSOR#0,0,0:PRINT#0,'Bye...':LRUN dn\$**

1208 **END DEfIne**

1210 REMark SPRITE Generator



Note: SPRITE Frames 1...32 max - Saved as _bmp File

Bytes: [0 to 2] **QL8** or **PAL** ID Header followed by

[3] **bm** number of SPRITE Bitmaps GRID Size [4] **cm** columns [5] **rm** rows

[6] **bg** BackGnd colour [7] **sz** File Loading Reference

[8 to 15] BackGnd Colours range 8...255 of selected Palette **QL** or **PAL**

[16.....nn.] SPRITE BITMaps [**bm** * **cm** * **rm**]

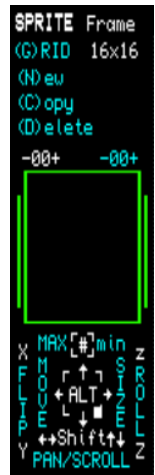
1212 DEFine PROCEDURE SGen

1213 SElect ON k

```

1214 =84,116:GTile :REMark (T)iles
1215 =45,95:IF fr>0 :fr=fr-1:FrChk :REMark - Frame Chg
1216 =43,61:IF fr<frt:fr=fr+1:FrChk :REMark + Frame Chg
1217 =10,32,9,253::oic=ic :PCol :REMark <> Paint Colour Chg
1218 =73,105:IF frt<frm AND cg%>0:FrntShow:FrntAdd :REMark (I)mort FONT
1219 =78,110:IF frt<frm:FrNEW :CURSOR#6,21,14:PRINT#6,'ew ' :REMark (N)ew
1220 =67,99:IF frt>1 :FrCOPY:CURSOR#6,21,25:PRINT#6,'opy ' :REMark (C)opy
1221 =68,100:IF frt>0 :FrDEL :CURSOR#6,21,36:PRINT#6,'elete' :REMark (D)elete
1222 =39,64:IF frt>0 :FrChg 1:GDraw :REMark (@) Grid Fill
1223 =82,114:IF frt>0 :FrReCol :REMark (R)eColour
1224 =70,103 :GSize :REMark (G)rid
1225 =66,98:IF frt>0 :GBGnd :REMark (B)kGnd
1226 =80,112 :GPalette :REMark (P)alette
1227 =35:IF fr>0 :EMaxMin :REMark # Edit Area
1228 =193:EMove:LEft :EMove :REMark ALT ¼
1229 =201:EMove:ERight :EMove :REMark ALT ½ Move
1230 =209:EMove:EUp :EMove :REMark ALT ¾ SIZE
1231 =217:EMove:EDown:EMove :REMark ALT ¿
1232 =196:dm=0:pa=cs+cn-1:pb=cs:pc=cs-1:pd=cs:GSlid :REMark Shift ¼ PAN
1233 =204:dm=0:pa=cs:pb=cs+cn-1:pc=cs:pd=cs-1:GSlid :REMark Shift ½ PAN
1234 =212:dm=1:sa=rs+rn-1:sb=rs:sc=rs-1:sd=rs :GSlid :REMark Shift ¾ SCROLL
1235 =220:dm=1:sa=rs:sb=rs+rn-1:sc=rs:sd=rs-1 :GSlid :REMark Shift ¿ SCROLL
1236 =88,120:xf=cs+cn-1:yf=rs:xz=-1:yz=1 :GFlip :REMark X FLIP
1237 =89,121:yf=rs+rn-1:xf=cs:yz=-1:xz=1 :GFlip :REMark Y FLIP
1238 =122:rxm=m-1:rym=0:xt=-1:yt=1 :GRoll :REMark z ROLL Anti-CW
1239 =90:rym=m-1:rxm=0:yt=-1:xt=1 :GRoll :REMark Z ROLL ClockWise
1240 END SElect
1241 END DEFine

```



1242 DEFine PROCEDURE PCol

```

1243 BLOCK#8,12,10,4,35+oic*10,0:BLOCK#8,8,6,6,37+oic*10,CP(oic)
1244 SElect ON k=44,60,253:ic=ic-1:IF ic<pn:ic=15:END IF :END SElect
1245 SElect ON k=46,62, 9:ic=ic+1:IF ic>15:ic=pn:END IF :END SElect
1246 BLOCK#8,12,10,4,35+ic*10,CP(ic) :BLOCK#8, 4,3,8,38,255
1247 IF ic=0:BLOCK#8,12,10,4,35,CP(7):BLOCK#8,10,8,5,36,0
1248 END DEFine

```



Highlighted Palette Colour

1250 DEFine FuNction GAns(ax,ay,a\$)

```

1251 REPEAT lp
1252 k=CODE(INKEY$(-1))
1253 SElect ON k=10,89,121:RETURN 1
1254 SElect ON k=32,78,110:RETURN 0
1255 END REPEAT lp
1256 END DEFine

```

1258 REMark **SPRITE GRID** Edit

1260 **DEFine PROCedure EMaxMin**

1261 IF cn<cm OR m<rm

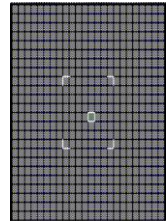
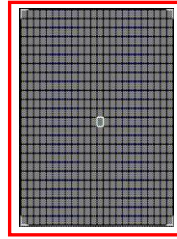
1262 **EMove**:cs=0:cn=cm:rs=0:rn=rm:**EMove**

1263 ELSE

1264 **EMove**:cs=cm/2-2:cn=4:rs=rm/2-2:rn=4:**EMove**

1265 END IF

1266 **END DEFINE**



1268 **DEFine FuNction EPos**

1269 IF cs=x AND rs=y :**RETurn** 1 :REMark Top Left

1270 IF cs=x AND rs=m-1=y :**RETurn** 2 :REMark Bottom Left

1271 IF cs+cn-1=x AND rs=y :**RETurn** 3 :REMark Top Right

1272 IF cs+cn-1=x AND rs+m-1=y :**RETurn** 4 :REMark Bottom Right

1273 **RETurn** 0

1274 **END DEFINE**

MAX  min Edit Area

Position Edit Area

ReSIZE Edit Area

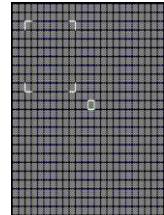
1276 **DEFine PROCedure EUUp**

1277 IF rs>0 AND EPos=0:rs=rs-1

1278 IF rs>0 AND EPos=1 OR rs>0 AND EPos=3:m=m+1:y=y-1:rs=rs-1

1279 IF m>4 AND EPos=2 OR m>4 AND EPos=4:m=m-1:y=y-1

1280 **END DEFINE**



1282 **DEFine PROCedure EDown**

1283 IF rs+m<rm AND EPos=0:rs=rs+1

1284 IF m>4 AND EPos=1 OR m>4 AND EPos=3:m=m-1:y=y+1:rs=rs+1

1285 IF rs+m<rm AND EPos=2 OR rs+m<rm AND EPos=4:m=m+1:y=y+1

1286 **END DEFINE**

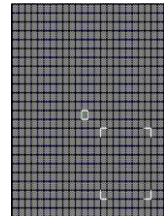
1288 **DEFine PROCedure ELeft**

1289 IF cs>0 AND EPos=0:cs=cs-1

1290 IF cs>0 AND EPos=1 OR cs>0 AND EPos=2:cn=cn+1:x=x-1:cs=cs-1

1291 IF cn>4 AND EPos=3 OR cn>4 AND EPos=4:cn=cn-1:x=x-1

1292 **END DEFINE**



1294 **DEFine PROCedure ERight**

1295 IF cs+cn<cm AND EPos=0:cs=cs+1

1296 IF cn>4 AND EPos=1 OR cn>4 AND EPos=2:cn=cn-1:x=x+1:cs=cs+1

1297 IF cs+cn<cm AND EPos=3 OR cs+cn<cm AND EPos=4:cn=cn+1:x=x+1

1298 **END DEFINE**

1300 **DEFine PROCedure EMove**

1301 x1=-1+zx+cs*cw:x2=-1+zx+(cs+cn)*cw:y1=-1+zy+rs*rh:y2=-1+zy+(rs+m)*rh

1302 OVER#3,-1:col%=CP(7)

1303 BLOCK#3,cw+1,1,x1+1,y1,col%:BLOCK#3,1,rh+1,x1,y1,col%

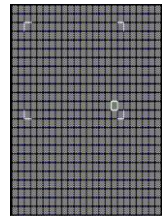
1304 BLOCK#3,cw,1,x2-cw,y1,col%:BLOCK#3,1,rh+1,x2,y1,col%

1305 BLOCK#3,cw+1,1,x1+1,y2,col%:BLOCK#3,1,rh+1,x1,y2-rh,col%

1306 BLOCK#3,cw,1,x2-cw,y2,col%:BLOCK#3,1,rh+1,x2,y2-rh,col%

1307 OVER#3,0

1308 **END DEFINE**



Repositioning Grid Cells

GSide	PAN Left	dm=0:pa=cs+cn-1:pb=cs:pc=cs-1:pd=cs	:
GSlide	PAN Right	dm=0:pa=cs:pb=cs+cn-1:pc=cs:pd=cs-1	
GSlide	SCROLL Up	dm=1:sa=rs+m-1:sb=rs:sc=rs-1:sd=rs	
GSlide	SCROLL Down	dm=1:sa=rs:sb=rs+m-1:sc=rs:sd=rs-1	
GFlip	FLIP X	xf=cs+cn-1:yf=rs:xz=-1:yz=1	
GFlip	FLIP Y	yf=rs+m-1:xf=cs:yz=-1:xz=1	
GRoll	ROLL z	rxm=m-1:rym=0:xt=-1:yt=1	Anti-CW
GRoll	ROLL Z	rym=m-1:rxm=0:yt=-1:xt=1	Clockwise

Note: Edit Area reduced to Square of shortest side

1310 DEFINE PROCEDURE GSlid

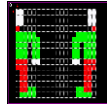
Grid Slide

```

1311 IF dm=0
1312   FOR r=rs TO rs+m-1
1313     FOR c=1 TO cn-1:TG(pa,r)=FG(fr,pb,r):TG(c+pc,r)=FG(fr,c+pd,r)
1314   END FOR r
1315 ELSE
1316   FOR r=1 TO m-1
1317     FOR c=cs TO cs+cn-1:TG(c,sa)=FG(fr,c,sb):TG(c,r+sc)=FG(fr,c,r+sd)
1318   END FOR r
1319 END IF
1320 cur=0:GTSet:GDraw
1321 END DEFINE

```

Left/Right



Up/Down



1323 DEFINE PROCEDURE GFlip

Grid Flip XX YY

```

1324 FOR r=0 TO rn-1
1325   FOR c=0 TO cn-1:TG(xf+c*xz,yf+r*yz)=FG(fr,cs+c,rs+r)
1326 END FOR r
1327 cur=0:GTSet:GDraw
1328 END DEFINE

```



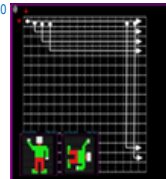
1330 DEFINE PROCEDURE GRoll

Grid Rotate 90°

```

1331 IF cs+m>cm:RETURN:ELSE EMove:cn=rn:EMove
1332 FOR r=0 TO rn-1
1333   rx=rxm+(r*xt):ry=rym
1334   FOR c=0 TO m-1:TG(cs+c,rs+r)=FG(fr,cs+rx,rs+ry):ry=ry+yt
1335 END FOR r
1336 cur=0:GTSet:GDraw
1337 END DEFINE

```



1339 DEFINE PROCEDURE GTSet

Grid Temp

```

1340 FOR r=rs TO rs+m-1:FOR c=cs TO cs+cn-1:FG(fr,c,r)=TG(c,r):END FOR c:END FOR r
1341 END DEFINE

```

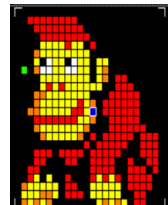
1343 DEFINE PROCEDURE GDraw

Grid Draw

```

1344 FOR r=rs TO rs+m-1:FOR c=cs TO cs+cn-1:GPBit:END FOR c:END FOR r
1345 END DEFINE

```



1347 DEFINE PROCEDURE GPBit

Grid Pixel Bit

```

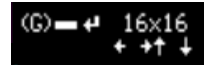
1348 IF FG(fr,c,r)=255:pcol%=CP(bg):ELSE pcol%=CP(FG(fr,c,r))
1349 IF fed=1:BLOCK#3,cw-1,rh-1,zx+c*cw,zy+r*rh,pcol%
1350 BLOCK#4,1,1,px+c,py+r,pcol%
1351 END DEFINE

```

1353 REMark **SPRITE** Frame Actions

1355 **DEFine PROCEDURE GSize**

```
1356 INK#6,CP(7):CURSOR#6,2,2:PRINT#6,'(G)':FrMes 6,20,2
1357 CURSOR#6,46,10:PRINT#6,' ← → ↑ ↓ ':ocm=cm:orm=rm:zx=66:zy=1
```



1358 **REPEAT Size_Ip**

```
1359 FrGrid cm,rm:k=CODE(INKEY$(-1))
```

1360 **SElect ON k**

```
1361 =192:IF cm>=12:cm=cm-4
```

```
1362 =200:IF cm<=56:cm=cm+4
```

```
1363 =208:IF rm<=56:rm=rm+4
```

```
1364 =216:IF rm>=12:rm=rm-4
```

```
1365 = 32:cm=ocm:rm=orm:EXIT Size_Ip
```

```
1366 = 10:FrSet:GLoc:EXIT Size_Ip
```

1367 **END SElect**

1368 **END REPEAT Size_Ip**

```
1369 BLOCK#6,36,10,46,10,0:INK#6,CP(5):CURSOR#6,2,2:PRINT#6,'(G)RID '
```

```
1370 INK#6,CP(7):FrGrid cm,rm
```

```
1371 END DEFine
```

Note: Grid Size multiples of 8
Range 08 to 60

1373 **DEFine PROCEDURE FrMes(ch,mx,my)**

```
1374 INK#ch,CP(7):CURSOR#ch,mx,my:PRINT#ch,' ← '
```

```
1375 BLOCK#ch,10,3,mx,my+4,CP(7):BLOCK#ch,2,4,mx+18,my+2,CP(7)
```

```
1376 END DEFine
```

Note: Frame Message Prompt
Abort Spacebar or Action Enter

1378 **DEFine PROCEDURE FrGrid(cm,rm)**

```
1379 CURSOR#6,50,2:PRINT#6,FILL$('0',2-LEN(cm));cm;'x';FILL$('0',2-LEN(rm));rm
```

```
1380 END DEFine
```

1382 **DEFine PROCEDURE FrSet**

Note: Frame Set - Parameters

```
1383 IF cm>rm:cm=192 DIV cm:rm=cw:ELSE cw=192 DIV rm:rh=cw
```

```
1384 xm=cm-1:ym=rm-1:zx=162-(cm*cw)/2:zy=98-(rm*rh)/2:px=34-cm/2:py=33-rm/2
```

```
1385 x=cm/2:y=rm/2:cs=0:cn=cm:rs=0:m=rm:CLS#3:CLS#4:FrChk
```

```
1386 END DEFine
```

1388 **DEFine PROCEDURE FrChk**

Note: Frame Change Check

```
1389 IF fed=1 AND fr=0:FrShow:FrNum fr,7,8,50:RETurn
```

```
1390 IF fed=1:FrNum fr,7,8,50:cs=0:cn=cm:rs=0:m=rm:CLS#3:CLS#4:GDraw:EMove
```

```
1391 IF fed=2:FrNum fr,7,8,50:CLS#4:GDraw
```

```
1392 END DEFine
```

1394 **DEFine PROCEDURE FrNum(fr%,fi%,fx%,fy%)**

```
1395 INK#6,CP(fi%):CURSOR#6,fx%,fy%:PRINT#6,'-':FILL$('0',2-LEN(fr%));fr%:'+'
```

```
1396 END DEFine Note: Frame Numbering displays Current and Total SPRITES
```

1398 **DEFine PROCEDURE FrShow**

```
1399 PAPER 0:CLS#3:nx=320 DIV cm:ny=192 DIV rm:max=nx*ny
```

```
1400 fx=2:fy=1:IF max>frt:max=frt
```

```
1401 FOR f=1 TO max
```

```
1402 FOR r=0 TO rm-1
```

```
1403 FOR c=0 TO cm-1
```

```
1404 IF FG(f,c,r)=255:pcol%=bg:ELSE pcol%=CP(FG(f,c,r)):END IF
```

```
1405 BLOCK#3,1,1,fx+c,fy+r,pcol%
```

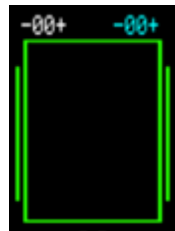
```
1406 END FOR c
```

```
1407 END FOR r
```

```
1408 fx=2+(f MOD nx)*cm:fy=1+(f DIV nx)*rm
```

```
1409 END FOR f
```

```
1410 END DEFine
```



Note: -00+ Frame Option for Display of **SPRITE** Grp



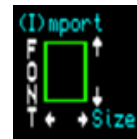
Note: (!)import of QL FONT_fnt File BITMaps and ReSize into SPRITE Frames. First Load a FONT_fnt File. Press 'I' and select with -/+ set to '0' to display Set.



```

1412 DEFINE PROCEDURE FntAdd
1413 CHAR_USE#11,Fntaddr,0:QBold 8,7,20,70,'(!)-/+ ':FrMes 8,62,70:ocsn=csn:ofr=fr
1414 REPEAT char_lp
1415 CURSOR#11,3,0:PRINT#11,CHR$(ch%+sg%):IF ch%=0:CLS#3:FntShow
1416 CURSOR#8,45,108:PRINT#8,csx:CURSOR#8,62,89:PRINT#8,csy
1417 k=CODE(INKEY$(-1))
1418 SElect ON k
1419 =45,95:ch%=ch%-1:IF ch%<=0:ch%=0 :REMark - Chr$(n)
1420 =43,61:ch%=ch%+1:IF ch%>=cg%:ch%=cg% :REMark + Chr$(n)
1421 =192:csx=csx-1 :IF csx<=1:csx=1
1422 =200:csx=csx+1 :IF csx>=6:csx=6
1423 =208:csy=csy+1 :IF csy>=6:csy=6
1424 =216:csy=csy-1 :IF csy<=1:csy=1
1425 =32:csn=ocsn:fr=ofr:EXIT char_lp
1426 =10:csn=ch%:FntSize csn,csx,csy:EXIT char_lp
1427 END SElect
1428 END REPEAT char_lp
1429 INK#8,CP(5):CURSOR#8,20,70:PRINT#8,'(!)mport ':FrChk
1430 END DEFINE

```



```

1432 DEFine PROCEDURE FntShow
1433 WINDOW#3,260,180,gx+150,gy+38:CSIZE#3,3,1:CHAR_USE#3,Fntaddr,0
1434 CLS#3:PRINT#3:FOR i=1+sg% TO sg%+cg%:PRINT#3,CHR$(i);
1435 WINDOW#3,324,194,gx+94,gy+37:CSIZE#3,0,0:CHAR_USE#3,0,0
1436 END DEFINE

```

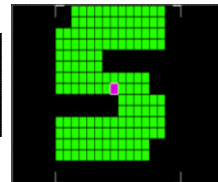
Note: CHAR_USE



```

1438 DEFine PROCEDURE FntSize(csn,csx,csy)
1439 frt=frt+1:fr=frt:FrCLS fr
1440 FOR r=0 TO 8:fnt$(r)=BIN$(PEEK(Fntaddr+2+r+csn*9),8)
1441 FOR r=0 TO 8
1442 yz=csy*r:yr=0
1443 FOR c=0 TO 7
1444 xz=csx*c:xc=0
1445 REPEAT clp
1446 IF fnt$(r,c+1)='1':FG(fr,xz+xc,yz)=ic:ELSE FG(fr,xz+xc,yz)=bg
1447 xc=xc+1:IF xc>=csx:EXIT clp
1448 END REPEAT clp
1449 END FOR c
1450 REPEAT rlp
1451 yr=yr+1:IF yr=csy:EXIT rlp
1452 FOR c=0 TO 8*csx:FG(fr,c,yz+yr)=FG(fr,c,yz)
1453 END REPEAT rlp
1454 END FOR r
1455 FrNum fr,7,8,50:FrNum frt,5,56,50:rs=0:rn=rm:cs=0:cn=cm
1456 END DEFINE

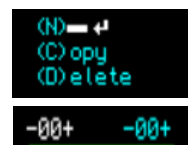
```



```

1458 DEFine PROCEDURE FrNew
1459 INK#6,CP(7):FrMes 6,21,14:IF GAns=0:INK#6,CP(5):RETurn
1460 IF fr=frt:fnt=frt+1:fr=fr+1:FrCLS fr
1461 IF fr<frt:FOR f=frt TO fr STEP -1:FrChg 4:END FOR f:frt=frt+1:FrCLS fr
1462 FrNum fr,7,8,50:FrNum frt,5,56,50:INK#6,CP(5)
1463 CLS#3:cs=0:cn=cm:rs=0:rn=rm:xm=cm-1:ym=rm-1:fed=1:ch=3:GDraw:EMove
1464 END DEFINE

```




```

1466 DEFine PROCEDURE FrCOPY
1467 INK#6,CP(7):FrMes 6,21,25:FrNum fr,5,8,50:f2=fr
1468 REPEAT Copy_LP
1469 FrNum f2,7,56,50:GDraw
1470 k=CODE(INKEY$(-1))
1471 SElect ON k
1472 =45, 95:IF f2>1 :f2=f2-1
1473 =43, 61:IF f2<frt:f2=f2+1
1474 =78,110,32:EXIT Copy_LP
1475 =89,121,10:FrChg 6:fr=f2:GDraw:fed=1:EXIT Copy_LP
1476 END SElect
1477 END REPEAT Copy_LP
1478 FrNum fr,7,8,50:FrNum frt,5,56,50
1479 INK#6,CP(5):EMove:cs=0:cn=cm:rs=0:m=rm:EMove
1480 END DEFine

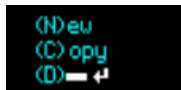
```



```

1482 DEFine PROCEDURE FrDEL
1483 BLOCK#6,42,10,21,36,0:INK#6,CP(7):FrMes 6,21,36:IF GAns=0:INK#6,CP(5):RETURN
1484 IF frt=1:fr=1:FrCLS:frt=0:fr=0
1485 IF frt>1 AND fr=frt:FrCLS frt:frt=frt-1:fr=frt
1486 IF frt>1 AND fr<frt:FOR f=fr TO frt:FrChg 3:END FOR f:FrCLS frt:frt=frt-1
1487 FrNum fr,7,8,50:FrNum frt,5,56,50:INK#6,CP(5)
1488 CLS#3:cs=0:cn=cm:rs=0:m=rm:ch=3:GDraw:EMove
1489 END DEFine

```



```

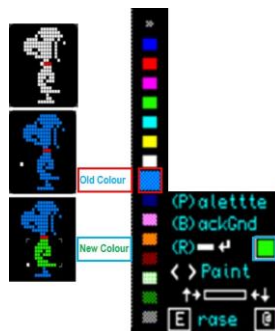
1491 DEFine PROCEDURE FrCLS(f%)
1492 FOR r=0 TO 63:FOR c=0 TO 63:FG(f%,r,c)=255:END FOR c:END FOR r
1493 END DEFine

```

```

1495 DEFine PROCEDURE FrReCol
1496 orub=rub:sic=ic:FrMes 8,38,146:BLOCK#8,10,8,75,148,CP(sic)
1497 REPEAT P_lp
1498 k=CODE(INKEY$(-1))
1499 SElect ON k
1500 =44,46,60,62,9,253:oic=ic:PCol
1501 =10:IF oic=bg:EXIT P_lp:ELSE FrChg 2:GDraw:EXIT P_lp
1502 =32:oic=sic:EXIT P_lp
1503 END SElect
1504 END REPEAT P_lp
1505 INK#8,CP(5):CURSOR#8,38,146:PRINT#8,'eCol':k=0
1506 rub=0:cur=0:sic=ic:PCol:BLOCK#8,10,8,75,148,CP(sic)
1507 END DEFine

```



Note: First Select **New** Colour from **< >** Palette. Press **[R]** then Select **Old** [existing] Grid Colour Press **Enter** and Colours are exchanged within Whole Grid or Resized Edit Area.

```

1509 DEFINE PROCEDURE FrChg(ck)
1510 FOR r=rs TO rs+m-1
1511   FOR c=cs TO cs+cn-1
1512     IF ck=1 AND rub=0:FG(fr,c,r)=ic:cur=7      :REMark Set Edit Area New Colour
1513     IF ck=1 AND rub=7:FG(fr,c,r)=255          :REMark Set Edit Area BackGnd
1514     IF ck=2 : IF FG(fr,c,r)=ic:FG(fr,c,r)=sic  :REMark Chg Edit Area Colours
1515     IF ck=3 : FG(fr,c,r)=FG(f+1,c,r)           :REMark Delete Frame
1516     IF ck=4 : FG(f+1,c,r)=FG(fr,c,r)           :REMark Insert New Frame
1517     IF ck=5 AND FG(fr,c,r)=bg:FG(fr,c,r)=255   :REMark Restore BackGnd
1518     IF ck=6 : FG(f2,c,r)=FG(fr,c,r)           :REMark Copy Frame f2 - fr
1519   END FOR c
1520 END FOR r
1521 END DEFINE

```

Note: Frame Grp Array Change of Settings

Note: Press 'B' and Select Palette Colour with < > chevron keys shown as ➡

```

1523 DEFINE PROCEDURE GBGnd
1524 tic=ic:obg=bg:oic=ic:ic=bg:INK#8,CP(7)
1525 CURSOR#8,20,135:PRINT#8,'(B)TAB ' :FrMes 8,60,135
1526 REPEAT BGnd_ip
1527   PCol:k=CODE(INKEY$(-1))
1528   SELECT ON k
1529     =44,46,60,62,9,253:oic=ic
1530     =10:IF ic=bg:EXIT BGnd_ip:ELSE bg=ic:FrChg 5:GDraw:EXIT BGnd_ip
1531     =32:EXIT BGnd_ip
1532   END SELECT
1533 END REPEAT BGnd_ip
1534 INK#8,CP(5):oic=ic:ic=tic:PCol:CURSOR#8,20,135:PRINT#8,'(B)ackGnd ' :FrChg
1535 END DEFINE

```



```

1537 DEFINE PROCEDURE GPalette
1538 INK#8,CP(7):pn=8:IF ic<8:oic=ic:ic=8:PCol:END IF
1539 CURSOR#8,20,124:PRINT#8,'(P)TAB ' :FrMes 8,60,124
1540 CURSOR#8,22,135:PRINT#8,' ↑ →   ← ↓ ' :tic=CP(ic):oic=ic
1541 REPEAT Col_ip
1542   PCol:col%=CP(ic):CURSOR#8,46,135:PRINT#8,FILL$('0',3-LEN(col%)):col%
1543   k=CODE(INKEY$(-1)):oic=ic
1544   SELECT ON k
1545     =192:IF col%> 8:col%=col% -1:CP(ic)=col%
1546     =200:IF col%<255:col%=col% +1:CP(ic)=col%
1547     =208:IF col%<192:col%=col%+64:CP(ic)=col%
1548     =216:IF col%> 64:col%=col%-64:CP(ic)=col%
1549     = 32:CP(ic)=tic :PCol:EXIT Col_ip
1550     = 10:CP(ic)=col%:PCol:FrChk:EXIT Col_ip
1551   END SELECT
1552 END REPEAT Col_ip
1553 BLOCK#8,60,20,20,124,0:INK#8,CP(5):CURSOR#8,20,124:PRINT#8,'(P)alette '
1554 CURSOR#8,20,135:PRINT#8,'(B)ackGnd ' :pn=0
1555 END DEFINE

```



Note: Only Palette Colours 8.5 are changeable. Select colour and change Number displayed with Cursor Keys. The New Colour is shown in Enlarged Square on the Palette Bar.

1557 REMark SCREEN/TILE Generator

1559 DEFine PROCEDURE TGen

1560 SElect ON k

```

1561 =70,103 :GFrame :REMark (G)rid
1562 =45,95:IF fr>0 :fr=fr-1 :FrChk :REMark - Frame
1563 =43,61:IF fr<frt:fr=fr+1 :FrChk :REMark + Frame
1564 =193 :IF tn>1 :tn=tn-1 :TNum 7 :REMark ◀ TILE
1565 =201 :IF tn<tm :tn=tn+1 :TNum 7 :REMark ▶ TILE
1566 =65,97:IF frt>0 :Retro_Act :REMark (A)ctions
1567 =80,112 :Retro_Play :REMark (P)lay Game
1568 =49 TO 57:sn=k-48:IF tm>0 :Tile_MAP :REMark (1...9)
1569 =67,99:IF fr>0 AND tm<tmax :Tile_ADD:CURSOR#6,21,25:PRINT#6,'opy '
1570 =68,100:IF tm>0 :Tile_DEL:CURSOR#6,21,36:PRINT#6,'elete'
1571 =66,98:GBGnd:IF k=10 :Tile_MAP :REMark (B)kGnd
1572 =84,116:IF tn>0 :Tile_Ass :REMark (T)ile Asset
1573 =77,109:CLS#3:INK#3,CP(6) :Scn_MAP :REMark (M)AP Scn Links
1574 =35:IF obj=0:obj=1:HObj 7 :ELSE obj=0:HObj 5 :REMark # Screen Links
1575 END SElect
1576 END DEFine

```

1578 DEFine PROCEDURE Scn_Num(sn)

Note: SCREEN Num

(1..9) [1] SCREEN

```

1579 INK CP(7):CURSOR 402,24:PRINT sn:cur=7
1580 END DEFine

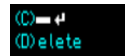
```

1582 DEFine PROCEDURE Tile_ADD

```

1583 FrMes 6,21,25:IF frt=0 OR GAns=0:INK#6,CP(5):RETURN
1584 IF tn=0 OR tn=tm:tn=tn+1
1585 IF tn<tm:FOR td=tm TO tn STEP -1:Tile_Chg 2:END FOR td
1586 td=tn:Tile_Chg 1:tm=tm+1:TNum 7:INK#6,CP(5)
1587 END DEFine

```



1589 DEFine PROCEDURE Tile_DEL

```

1590 BLOCK#6,30,10,32,36,0:FrMes 6,21,36:IF GAns=0:INK#6,CP(5):RETURN
1591 IF tn<tm:FOR td=tn TO tm:Tile_Chg 3:END FOR td:tm=Tile_Chg 4
1592 tm=tm-1:IF tn>tm:tn=tm:END IF :TNum 7:INK#6,CP(5)
1593 END DEFine

```



1595 DEFine PROCEDURE TNum(ni)

Note: Tile Status & Library number

```

1596 INK#8,CP(ni):CURSOR#8,44,2:PRINT#8,TAss(ni)
1597 TDraw 2 :CURSOR#8,63,2:PRINT#8,FILL$( '0',2-LEN(ni))&tn
1598 END DEFine

```

1600 DEFine PROCEDURE Tile_Chg(tck)

Note tic TILE Check

```

1601 FOR r=0 TO tsz-1
1602 FOR c=0 TO tsz-1
1603 IF tck=1:Tile(td,c,r)=FG(fr,cs+c,rs+r) :REMark Add New Tile
1604 IF tck=2:Tile(td+1,c,r)=Tile(td,c,r) :REMark Insert Tile Space
1605 IF tck=3:Tile(td,c,r)=Tile(td+1,c,r) :REMark Cut Tile from List
1606 IF tck=4:Tile(td,c,r)=255 :REMark CLS Tile
1607 END FOR c
1608 END FOR r
1609 END DEFine

```

1611 DEFine PROCEDURE HLObj(oi)

(1) ops [#]

```

1612 INK#8,CP(oi):CURSOR#8,50,14:PRINT#8,'#'
1613 END DEFine

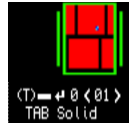
```

Note: ON/OFF # hash highlight for SWITCH - SLIDE

```

1615 DEFINE PROCEDURE Tile_Ass
1616 CURSOR#8,2,2:PRINT#8,'(T)':FrMes 8,20,2:atn=TAss(tn):atn=TAss(tn)
1617 REPEAT Ass_lp
1618 CURSOR#8,44,2:PRINT#8,atn:CURSOR#8,2,14:PRINT#8,TAB';TAS$(atn)
1619 k=CODE(INKEY$(-1))
1620 IF k=9:atn=atn+1:IF atn>7:atn=0      Note: 'TAB' through Assets
1621 IF k=32:atn=TAss(tn):EXIT Ass_lp      Abort with Spacebar
1622 IF k=10:TAss(tn)=atn:EXIT Ass_lp      Action with Enter
1623 END REPEAT Ass_lp
1624 CURSOR#8,44,2:PRINT#8,atn:BLOCK#8,60,12,4,14,0:INK#8,CP(5)
1625 CURSOR#8,2,2:PRINT#8,'(T)ILE':CURSOR#8,2,14:PRINT#8,'(M)aps [#]'
1626 END DEFINE

```

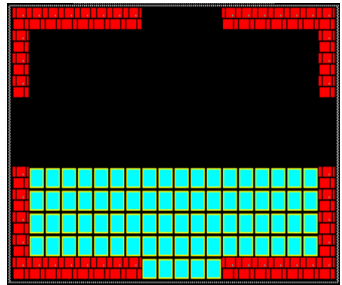


```

1628 DEFINE PROCEDURE Tile_Chk
1629 tn=TScn(0,x,y)
1630 IF TAss(tn)=0:x=ox:y=oy                :REMark Solid Blocks Move
1631 IF TAss(tn)=1:x=x:y=y                  :REMark Floor
1632 IF TAss(tn)=2:x=x:y=y                  :REMark Hazard
1633 IF TAss(tn)=3:x=x:y=y                  :REMark Reward
1634 IF TAss(tn)=4:x=x:y=y                  :REMark ???
1635 IF TAss(tn)=5:x=x:y=y                  :REMark ???
1636 IF TAss(tn)=6:x=x:y=y                  :REMark ???
1637 IF TAss(tn)=7:x=x:y=y                  :REMark ???
1638 END DEFINE

```

Note: The Screen Background is a Matrix of Tiles the **Tile_MAP** searches TScn array to identify occupied Cells by its Library TILE number and uses **TDraw** to display.



```

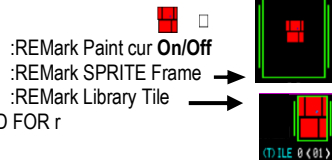
1640 DEFINE PROCEDURE Tile_MAP
1641 PAPER#3,CP(bg):CLS#3:Scn_Num sn :tcz=320 DIV tsz:trz=192 DIV tsz
1642 FOR tr=0 TO trz-1
1643   FOR tc=0 TO tcz-1
1644     TScn(0,tc,tr)=TScn(sn,tc,tr) MOD 64
1645     tn=TScn(0,tc,tr):IF tn>0:tx=2+tc*tsz:ty=1+tr*tsz:TDraw 0
1646   END FOR tc
1647 END FOR tr
1648 cur=0:rub=0:snp=sn:snw=TMap(snp,4):sne=TMap(snp,2):ec=0:wc=19
1649 END DEFINE

```

```

1651 DEFINE PROCEDURE TDraw(ver)
1652 LOCAL x,y
1653 IF ver=0:ch=3:w=1:h=1:x=bx:y=ty
1654 IF ver=1:ch=4:w=1:h=1:x=26:y=25:CLS#4
1655 IF ver=2:ch=7:w=1:h=1:x=(28-tsz)/2:y=(26-tsz)/2:CLS#7
1656 FOR r=0 TO tsz-1:FOR c=0 TO tsz-1:PDraw:END FOR c:END FOR r
1657 END DEFINE

```



```

1659 DEFINE PROCEDURE PDraw
1660 pcol%=Tile(tn,c,r):IF pcol%=255:pcol%=bg
1661 BLOCK#ch,w,h,x+c*w,y+r*h,CP(pcol%)
1662 END DEFINE

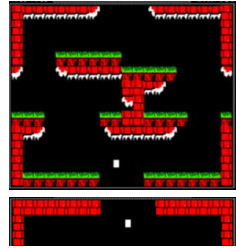
```

Note: Checks for Background. Fill Block at TILE x y coordinates with TILE Colour

```

1664 DEFine PROCEDURE Scn_Switch
1665 ns=0 :IF TScn(0,x,y)>0:Tile_Chk:END IF :REMark Check IF Move Blocked
1666 IF y=0 :ns=TMap(sn,1):IF ns>0:SEnt ns,1,11,18,11 :x=c+1:y=10 :END IF
1667 IF x=xm:ns=TMap(sn,2):IF ns>0:SEnt ns,0,1,0,10 :x=1 :y=r+1 :END IF
1668 IF y=ym:ns=TMap(sn,3):IF ns>0:SEnt ns,1,0,18,0 :x=c+1:y=1 :END IF
1669 IF x=0 :ns=TMap(sn,4):IF ns>0:SEnt ns,19,1,19,10 :x=18 :y=r+1 :END IF
1670 IF ns>0 :sn=ns:Tile_MAP
1671 END DEFine

```



Note: See Maps for Links Set between Screens. Test by Pressing [#]

```

1673 DEFine PROCEDURE SEnt(ns,ax,by,cx,dy)
1674 FOR r=by TO dy:FOR c=ax TO cx:IF TScn(ns,c,r)=0:x=c:y=r:RETurn
1675 END DEFine

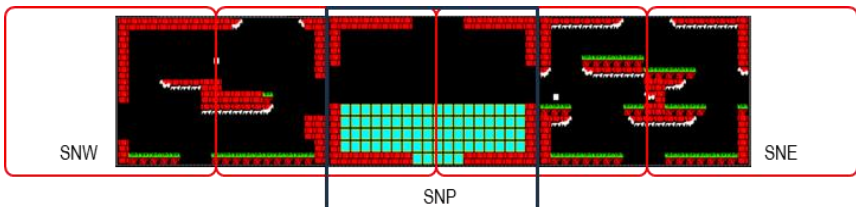
```

Note: This Option for Platform Games that use Sliding Screens and where Player's change in Levels are accomplished with a **Jump** action from pressing the **Spacebar**.

```

1677 DEFine PROCEDURE Scn_Slide
1678 IF TScn(0,x,y)>0:Tile_Chk:END IF :REMark Check IF Move Blocked
1679 IF x+1>xm:PAN#3,-16:x=ox:PANEast:ec=ec+1:wc=wc+1:IF wc>19 :wc=19
1680 IF x-1<0 :PAN#3, 16:x=ox:PANWest:wc=wc-1:ec=ec-1:IF ec<0:ec=0
1681 END DEFine

```



```

1683 DEFine PROCEDURE PANEast
1684 IF ec>19:snp=sne:ec=0:wc=19:sne=TMap(snp,2):Scn_Num snp
1685 IF wc<19 AND wc>0:sne=snp:ec=wc+1
1686 FOR tr=0 TO 11
1687 FOR tc=1 TO 19:TScn(0,tc-1,tr)=TScn(0,tc,tr):END FOR tc
1688 TScn(0,19,tr)=TScn(sne,ec,tr):tn=TScn(0,19,tr)
1689 IF tn>0:tx=2+19*16:ty=1+tr*16:TDraw 0
1690 END FOR tr
1691 END DEFine

```

```

1693 DEFine PROCEDURE PANWest
1694 IF wc<0:snp=snw:wc=19:ec=0:snw=TMap(snp,4):Scn_Num snp
1695 IF ec>0 AND ec<19:snw=snp:wc=ec-1
1696 FOR tr=0 TO 11
1697 FOR tc=18 TO 0 STEP -1:TScn(0,tc+1,tr)=TScn(0,tc,tr):END FOR tc
1698 TScn(0,0,tr)=TScn(snw,wc,tr):tn=TScn(0,0,tr)
1699 IF tn>0:tx=2:ty=1+tr*16:TDraw 0
1700 END FOR tr
1701 END DEFine

```

1703 REMark MAPS SCREEN Links

1705 DEFINE PROCEDURE Scn_MAP

```

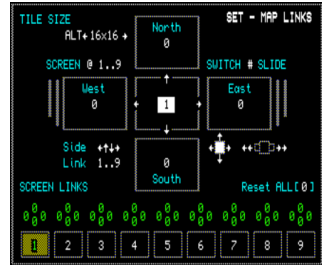
1706 obg=bg:PAPER#3,0:CLS#3:QBOLD 3,7,224,4,'SET - MAP LINKS'
1707 FOR i=0 TO 1:BLOCK#3,68,40,52+i*152,54,CP(15):BLOCK#3,64,38,54+i*152,55,0
1708 FOR i=4 TO 5:BLOCK#3,3,i*7,5+i*8,56,CP(15):BLOCK#3,3,i*7,317-i*8,56,CP(15)
1709 FOR i=0 TO 2:BLOCK#3,68,40,128,8+i*46,CP(15):BLOCK#3,64,38,130,9+i*46,0
1710 FOR i=1 TO 9
1711   mx=-27+i*35:BLOCK#3,30,20,mx,170,CP(15):BLOCK#3,28,18,mx+1,171,0
1712   CURSOR#3,-16+i*35,176:PRINT#3,i
1713 END FOR i
1714 RESTORE 1718:FOR i=1 TO 30:QPrtnt
1715   BLOCK#3,11,10,212,102,CP(15):BLOCK#3,9,8,213,103,0
1716   BLOCK#3,21,6,254,104,CP(15):BLOCK#3,19,4,255,105,0
1717   BLOCK#3,11,10,259,102,CP(15):BLOCK#3,9,8,260,103,0
1718   DATA 3,5,6,6,'TILE SIZE',3,7,54,18,'ALT'
1719   DATA 3,7,214,94,'',3,7,214,110,'',3,7,205,102,'',
1720   DATA 3,7,158,24,TMap(sn,1),3,7,158,114,TMap(sn,3)
1721   DATA 3,7,82,70,TMap(sn,4),3,7,236,70,TMap(sn,2)
1722   DATA 3,0,155,52,'',3,7,126,69,'%',1/2,3,0,155,90,' '
1723   DATA 3,7,158,50,'%',3,5,146,12,'North',3,5,226,58,'East'
1724   DATA 3,7,158,88,'%',3,5,146,126,'South',3,5,72,58,'West'
1725   DATA 3,5,34,40,'SCREEN',3,7,77,40,'@ 1..9',3,5,203,40,'SWITCH SLIDE'
1726   DATA 3,7,246,40,'#',3,5,52,102,'Side',3,7,88,102,'',
1727   DATA 3,5,52,114,'Link',3,7,88,114,'1..9',3,7,240,102,'',
1728   DATA 3,5,6,132,'SCREEN LINKS',3,5,240,132,'Reset ALL',
1729   FrMes 8,22,14:ScnTie:mx=158:my=70:sl=0:Tile_Size:Scn_Links
1730 END DEFINE

```

```

(T)ILE 0 (01)
(0) - (0)
(R)dd FONT

```



Note: Use @ and 1..9 to Select SCREEN. Use Cursor Keys to Select Side and 1..9 to set Link. A LINK will connect on the opposite side of SCREEN Chosen: ie WEST Side Links to the EAST Side of selected SCREEN.

1732 DEFINE PROCEDURE Scn_Links

1733 REPEAT Scn_lp

```

1734 PrntScn 7,7,0:PrntMap:k=CODE(INKEY$(1)):PrntScn 0,0,7
1735 SELECT ON k
1736   =193 :tz=tz-1:IF tz<2:tz=6:END IF :Tile_Size
1737   =201 :tz=tz+1:IF tz>6:tz=2:END IF :Tile_Size
1738   =35 :IF lks=0:lks=1:ScnTie:ELSE lks=0:ScnTie
1739   =39,64:mx=158:my=70:sl=0 :REMark @ Select Screen
1740   =192 :mx=82:my=70:sl=4:so=2 :REMark West Exit East Entry
1741   =200 :mx=236:my=70:sl=2:so=4 :REMark East Exit West Entry
1742   =208 :mx=158:my=24:sl=1:so=3 :REMark North Exit South Entry
1743   =216 :mx=158:my=114:sl=3:so=1 :REMark South Exit North Entry
1744   =48 :DIM TMap(9,4):RESTORE 1720:FOR i=1 TO 4:QPrtnt
1745   =49 TO 57:IF sl=0
1746     PrntCLS:sn=k-48:RESTORE 1720:FOR i=1 TO 4:QPrtnt
1747     ELSE
1748       TMap(sn,sl)=k-48:TMap(k-48,so)=sn
1749     END IF
1750   =10,32 :tsz=tz:Tile_Size=obg:PAPER#3,CP(bg):CLS#3:EXIT Scn_lp
1751 END SELECT
1752 END REPEAT Scn_lp
1753 INK#8,CP(5):CURSOR#8,20,14:PRINT#8,'aps [#]':obj=0:tsz=tz*4:GTile
1754 END DEFINE

```

Note sl screen link
so screen opposite

1756 **DEfIne PROCEDURE Tile_Size**

Note: tz=2 ...6 *4 : Grid 8x8 24x24

1757 tz\$=tz*4:tz\$=FILL\$('0',2-LEN(tz\$))&tz\$

1758 CURSOR#3,80,18:PRINT#3,tz\$;'x';tz\$

1759 CURSOR#1,428,7:PRINT#1,tz\$;'x';tz\$

1761 **END DEfIne**

1763 **DEfIne PROCEDURE ScnTie**

1764 IF lks=0:BLOCK#3,9,8,213,103,CP(7):BLOCK#3,9,8,260,103,0

1765 IF lks=1:BLOCK#3,9,8,213,103,0:BLOCK#3,9,8,260,103,CP(7)

1766 **END DEfIne**

1768 **DEfIne PROCEDURE PrntScn(bink,sink,pink)**

1769 BLOCK#3,18,12,mx-6,my-1,CP(bink):STRIP#3,CP(sink):INK#3,CP(pink)

1770 CURSOR#3,mx,my:IF sl=0:PRINT#3,sn:ELSE PRINT#3,TMap(sn,sl)

1771 **END DEfIne**

1773 **DEfIne PROCEDURE PrntMap**

1774 STRIP#3,0:INK#3,CP(4)

1775 BLOCK#3,26,16,-25+sn*35,172,240:CURSOR#3,-16+sn*35,176:PRINT#3,sn

1776 FOR i=1 TO 9:CURSOR#3,-25+i*35,154:PRINT#3,TMap(i,4);' ';TMap(i,2)

1777 FOR i=1 TO 9:CURSOR#3,-16+i*35,148:PRINT#3,TMap(i,1)

1778 FOR i=1 TO 9:CURSOR#3,-16+i*35,158:PRINT#3,TMap(i,3)

1779 STRIP#3,2:INK#3,0

1780 **END DEfIne**

1782 **DEfIne PROCEDURE PrntCLS**

1783 BLOCK#3,26,16,-25+sn*35,172,0

1784 STRIP#3,0:INK#3,CP(5):CURSOR#3,-16+sn*35,176:PRINT#3,sn

1785 **END DEfIne**

2038 REMark File Management

2040 **DEFine PROCEDURE SelDrv(act%,Act\$)**

2041 INK#5,CP(7):CURSOR#5,6,20:PRINT#5,'Select ↑ ↓ ← → ':BLOCK#5,2,4,78,22,CP(7)

2042 IF act%=1:PRINT#5,'(E)dit':END IF :INK CP(5)

2043 **REPEAT drv_lp**

2044 CURSOR#5,6,11:PRINT#5,Act\$&drv\$(dn%):CLS#5,4

2045 k=CODE(INKEY\$(-1))

2046 **SElect ON k**

2047 =208:dn%=dn%-1:IF dn%<0:dn%=dm%

2048 =216:dn%=dn%+1:IF dn%>dm%:dn%=1

2049 =101,69:IF act%=1:**EditName 5,act%,42,11,16,drv\$(dn%)** :REMark Drive/SubDIR

2050 = 10:**EXIT drv_lp**

2051 **END SElect**

2052 **END REPEAT drv_lp**

2053 **END DEFine**

```
(F)DIR (L)oad (S)ave (Q)uit
Load winl_
Select ↑ ↓ ← →
```

```
(F)DIR (L)oad (S)ave (Q)uit
Drive dosl_
Select ↑ ↓ ← → (E)dit +[ ] → ←
```

2055 **DEFine PROCEDURE SelFile**

2056 CURSOR#5,6,20:PRINT#5,'Select ↑ ↓ (M)erge':INK#5,CP(5):FrMes 5,72,20

2057 **REPEAT FSel_lp**

2058 CURSOR#5,6,11:PRINT#5,'Load 'File\$(sf%):CLS#5,4:k=CODE(INKEY\$(-1))

2059 **SElect ON k**

2060 =208:sf%=sf%-1:IF sf%<1:sf%=ft% :REMark Up

2061 =216:sf%=sf%+1:IF sf%>ft%:sf%=1 :REMark Dn

2062 =110,78,32:chk=0:**EXIT FSel_lp** :REMark Y/N Spacebar

2063 =121,89,10:chk=1:**EXIT FSel_lp** :REMark Y/N Enter

2064 =109,77 :chk=2:**EXIT FSel_lp** :REMark (M)erge

2065 **END SElect**

2066 **END REPEAT FSel_lp**

2067 **END DEFine**

```
(F)DIR (L)oad (S)ave (Q)uit
Load QBStar_bmp
Select ↑ ↓ ← → (M)erge
```

2069 **DEFine PROCEDURE DIR_File**

2070 CURSOR#5,6,20:PRINT#5,'Checking...':CLS#5,4:**DIR_List:PAUSE 20**

2071 OPEN _IN#9,drv\$(dn%)&'FList':dl%=LEN(drv\$(dn%)):chk=1

2072 **REPEAT dir_lp**

2073 IF EOF(#9) OR sf%>fm%:sf%=sf%-1:ft%=sf%:CLOSE#9:**EXIT dir_lp**

2074 INPUT#9,F\$:F\$=F\$(dl%-4 TO):fl%=LEN(F\$)

2075 IF fl%<=20 AND '_bmp' INSTR F\$>0:File\$(sf%)=F\$:sf%=sf%+1

2076 IF fl%<=20 AND '_fnt' INSTR F\$>0:File\$(sf%)=F\$:sf%=sf%+1

2077 **END REPEAT dir_lp**

2078 IF fl%<1:chk=0:CURSOR#5,6,11:PRINT#5,'No Files Found...':PAUSE 30

2079 **END DEFine**

```
(F)DIR (L)oad (S)ave (Q)uit
No Files Found...
Checking...
```

2081 **DEFine PROCEDURE DIR_List**

2082 DELETE drv\$(dn%)&'FList':OPEN_NEW#9,drv\$(dn%)&'FList':DIR#9,drv\$(dn%):CLOSE#9

2083 **END DEFine**

2085 **DEFine PROCEDURE DIR_Chk**

2086 OPEN _IN#9,drv\$(dn%)&'FList':dl%=LEN(drv\$(dn%))

2087 **REPEAT dir_lp**

2088 IF EOF(#9):CLOSE#9:chk=0 :**EXIT dir_lp**

2089 INPUT#9,F\$:F\$=F\$(dl%-4 TO)

2090 IF F\$==SFile\$:CLOSE#9:chk=1:**EXIT dir_lp**

2091 **END REPEAT dir_lp**

2092 **END DEFine**

2094 DEFINE PROCEDURE BMLoad

```

2095 chk=0:SelDrv 2,'Load '.INK CP(7):sf%=1:ft%=0:fm%=50:frz=frt
2096 DIR_File :IF chk=0:RETURN :END IF :SelFile:IF chk=0:RETURN :END IF
2097 BLOCK#5,190,20,2,10,0:CUSOR#5,6,11:PRINT#5,'Loading.':CLS#5,4:ID$=""
2098 OPEN _In#9,drv$(dn%)&File$(sf%):ID$=INKEY$(#9)&INKEY$(#9)&INKEY$(#9)
2099 IF '_fnt' INSTR File$(sf%)
2100   sg%=CODE(ID$(1)):cg%=CODE(ID$(2)):CLOSE#9
2101   LBYTES drv$(dn%)&File$(sf%),Fntaddr:GFrame
2102 END IF
2103 IF ID$='QL8' OR ID$='PAL'
2104   bm=CODE(INKEY$(#9)):cm=CODE(INKEY$(#9)):rm=CODE(INKEY$(#9))
2105   bg=CODE(INKEY$(#9)):sz=CODE(INKEY$(#9)):CLOSE#9
2106   IF sz>1 AND chk=2:fr=1:GFrame:RETURN
2107   IF sz=0:frt=bm:base=ALCHP( 8+frt*cm*rm):ptr=base+8 :PFile$=File$(sf%)
2108   IF sz=1:frt=bm:base=ALCHP(8+8+frt*cm*rm):ptr=base+16 :PFile$=File$(sf%)
2109   IF sz=2:tm=bm:base=ALCHP(8+8+36+2160+tm*cm*rm) :TFile$=File$(sf%) :tsz=cm
2110   IF sz=3:tm=bm:base=ALCHP(8+8+36+2160+200+tm*cm*rm+bm) :TFile$=File$(sf%) :tsz=cm
2111   LBYTES drv$(dn%)&File$(sf%),base
2112   IF sz=2:SnLoad base+16,base+52:ptr=base+2212
2113   IF sz=3:SnLoad base+16,base+52:AnLoad base+2212:ptr=base+2412
2114   FOR i=8 TO 15:BLOCK#8,8,6,6,36+10*i,CP(i)
2115   FrLoad bm,cm,rm,ptr:RECHP base:IF sz<2:fr=1:GFrame:ELSE tn=1:sn=1:GTile
2116 END IF
2117 END DEFINE

```

(F)DIR (L)oad (S)ave (M)ult
Load win1_
Select ↑ ↓ ← →

(F)DIR (L)oad (S)ave (M)ult
Loading

2119 DEFINE PROCEDURE SnLoad(ptr1,ptr2)

```

2120 FOR sn=1 TO 9
2121   FOR sd=1 TO 4:TMap(sn,sd)=PEEK(ptr1):ptr1=ptr1+1:END FOR sd
2122   FOR tr=0 TO 11
2123     FOR tc=0 TO 19:TScn(sn,tc,tr)=PEEK(ptr2):ptr2=ptr2+1:END FOR tc
2124   END FOR tr :END FOR sn
2126 END DEFINE

```

Note: SCREEN Maps

Screen Maps/Links 1..9
Grid Size **rn** Tile Size
Screen Grid Tile ID's

2128 DEFINE PROCEDURE AnLoad(ptr4)

```

2129 FOR i=1 TO 13:RGNS$(i)=CHR$(PEEK(ptr4)):ptr4=ptr4+1:END FOR i:ptr4=ptr4+1
2130 FOR ka=1 TO 9
2131   FOR kb=1 TO 20:SK(ka,kb)=PEEK(ptr4):ptr4=ptr4+1:END FOR kb
2132 END FOR ka
2133 END DEFINE

```

Note: Action Keys
Game Name

Actions

2135 DEFINE PROCEDURE FrLoad(bmax,cmax,rmax,ptr)

```

2136 IF sz<2 AND chk<2:DIM FG(32,64,64)
2137 IF sz>0:FOR i=8 TO 15:CP(i)=PEEK(base+i)
2138 OVER#5,1:frs=1:frmax=bmax:stp=0
2139 IF chk=2:frs=frz:IF bmax>frm-frz:frmax=frm:ELSE frmax=frz+bmax-1:END IF :frt=frmax
2140 FOR f=frs TO frmax
2141   CURSOR#5,48+stp,11:PRINT#5,'.':PAUSE 1:stp=stp+3
2142   FOR r=0 TO max-1
2143     FOR c=0 TO cmax-1
2144       IF sz<2 :FG(f,c,r)=PEEK(ptr) :END IF
2145       IF sz>1 :Tile(f,c,r)=PEEK(ptr):END IF
2146       ptr=ptr+1:REMark IF frt=0:FG(f,c,r)=PEEK(ptr) :END IF :ptr=ptr+1
2147     END FOR c :END FOR r :END FOR f
2148   OVER#5,0:IF sz>1:FOR f=1 TO frmax:TAss(f)=PEEK(ptr):ptr=ptr+1:END FOR f
2151 END DEFINE

```

Note: SPRITE/TILE Bitmaps

Palette Colours 8 to 15
chk=2 - (M)erge _bmp File

Progress Aid

Sprite Frame
Screen Tile

```

2155 DEFine PROCEDURE BMSave
2156 IF frt>0 OR tm>0:chk=0:eck=0:SelDrv 2,'Save ':ELSE RETURN
2157 IF sz<2:SFile$=PFile$:ELSE SFile$=TFile$:END IF
2158 IF SFile$=' ':SFile$='QBDefault_bmp':END IF
2159 IF LEN(SFile$)>20:SFile$=SFile$(1 TO 16) & '_bmp'
2160 INK CP(7):CURSOR#5,6,20:PRINT#5,'Action (E)dit':FrMes 5,48,20
2161 REPEAT chk_lp
2162 CURSOR#5,42,11:PRINT#5,SFile$:FILL$(' ',20-LEN(SFile$)):k=CODE(INKEY$(-1))
2163 SElect ON k=101,69:EditName 5,2,42,11,20,SFile$
2164 SElect ON k=10:DIR_List:DIR_Chk:EXIT chk_lp
2165 SElect ON k=32:BLOCK#5,190,20,2,10,0:RETURN
2166 END REPEAT chk_lp
2167 CURSOR#5,6,20:
2168 IF eck=1:PRINT#5,'DEVICE ERROR...':CLS#5,4:PAUSE 30:eck=0:RETURN
2169 IF chk=1:PRINT#5,'Overwrite':CLS#5,4:FrMes 5,72,20:IF GAns=0:RETURN
2170 DELETE drv$(dn%)&SFile$:BLOCK#5,190,20,2,10,0
2171 CURSOR#5,6,11:PRINT#5,'Saving':CLS#5,4
2172 IF sz=1:bm=frt:mlth=16+bm*cm*rm :addr=ALCHP(mlth):ptr=addr+16
2173 IF sz=3:bm=tm :mlth=52+2160+200+bm*tsz*tsz+bm :addr=ALCHP(mlth):cm=tsz:rm=tsz
2174 FOR i=0 TO 2:POKE addr+i,CODE(pm$(i+1)):END FOR i
2175 POKE addr+3,bm:POKE addr+4,cm:POKE addr+5,rm:POKE addr+6,bg:POKE addr+7,sz
2176 IF sz=3:SnSave addr+16,addr+52:AnSave addr+2212:ptr=addr+2412
2177 OVER#5,1:FrSave bm,cm,rm,ptr:OVER#5,0
2178 SBYTES drv$(dn%)&SFile$,addr,mlth:RECHP addr
2179 END DEFine

```

```

2181 DEFine PROCEDURE SnSave(ptr2,ptr3)
2182 FOR sn=1 TO 9
2183 FOR sd=1 TO 4:POKE ptr2,TMap(sn,sd):ptr2=ptr2+1:END FOR sd
2184 FOR tr=0 TO 11
2185 FOR tc=0 TO 19:POKE ptr3,TScn(sn,tc,tr):ptr3=ptr3+1:END FOR tc
2186 END FOR tr
2187 END FOR sn
2188 END DEFine

```

Note: SCREEN Maps
Screen Maps/Links 1..9

Grid Size nn Tile Size
Screen Grid Tile ID's

```

2190 DEFine PROCEDURE AnSave(ptr4)
2191 FOR i=1 TO 13:POKE ptr4,CODE(RGN$(i)):ptr4=ptr4+1:END FOR i:ptr4=ptr4+1
2192 FOR ka=1 TO 9
2193 FOR kb=1 TO 20:POKE ptr4,SK(ka,kb):ptr4=ptr4+1:END FOR kb
2194 END FOR ka
2195 END DEFine

```

Note: Action keys
Game Name
SPRITE 1..9
Actions

```

2197 DEFine PROCEDURE FrSave(bmax,cmax,rmax,ptr)
2198 IF sz>0:FOR i=8 TO 15:POKE addr+i,CP(i)
2199 FOR f=1 TO bmax
2200 IF f<48:CURSOR#5,42+f*3,11:PRINT#5,'.':PAUSE 1
2201 FOR r=0 TO rmax-1
2202 FOR c=0 TO cmax-1
2203 IF sz<2:POKE ptr,FG(f,c,r):END IF
2204 IF sz>1:POKE ptr,Tile(f,c,r):END IF :ptr=ptr+1
2205 END FOR c : END FOR r : END FOR f
2206 IF sz>1:FOR f=1 TO bmax:POKE ptr,TAss(f):ptr=ptr+1:END FOR f
2207 END DEFine

```

Note: SPRITE/TILE Grids
Palette Colours 8 to 15

Progress Aid

Frame Bitmap
Tile Bitmap

2211 REMark Text Editor

2213 **DEFine PROCEDURE** EditName(ch%,act%,sx%,sy%,sm%,str\$)

2214 IF act%=2:sl%=(' _bmp' INSTR str\$)-1:str\$=str\$(1 TO sl%)

2215 INK#ch%,CP(7):CURSOR#ch%,124,20:PRINT#ch%, '%[_]½ ¼'

2216 BLOCK#ch%,10,3,156,24,CP(7):BLOCK#ch%,2,4,172,22,CP(7)

2217 temp\$=str\$:sl%=LEN(str\$):cp%=sl%+1

2218 **REPeat Ed_ip**

2219 **Ln_Prn:Ln_Cur:**k\$=INKEY\$(#0,-1):k=CODE(k\$)

2220 **SElect ON k**

2221 = 10:IF sl%=0:str\$=temp\$:END IF **:EXIT Ed_ip**

2222 = 32:str\$=temp\$:**EXIT Ed_ip**

2223 = 48 TO 57,65 TO 90,95, 97 TO 122:**Add_chr**

2224 =194:IF cp%>1:cp%=cp%-1:**Del_chr**

2225 =202:**Del_chr**

2226 =192:IF cp%>1:cp%=cp%-1

2227 =200:IF cp%<sl%+1:cp%=cp%+1

2228 **END SElect**

2229 **END REPeat Ed_ip**

2230 IF act%=1

2231 IF sl%<5:str\$=temp\$:**RETurn**

2232 IF str\$(sl%)<>'_':IF sl%<sm%:str\$=str\$&'_' :ELSE str\$(sl%)='_'

2233 IF str\$(5)<>'_':str\$(5)='_'

2234 END IF

2235 k=0:BLOCK#ch%,60,10,122,20,0:IF act%=2:str\$=str\$&' _bmp'

2236 **END DEFine**



Edit Cursor Position

2238 **DEFine PROCEDURE** Ln_Prn

2239 IF LEN(str\$)>sm%:str\$=str\$(1 TO sm%):cp%=sm%

2240 INK#ch%,5:CURSOR#ch%,sx%,sy%:PRINT#ch%,str\$:CLS#ch%,4

2241 **END DEFine**

2243 **DEFine PROCEDURE** Ln_Cur

2244 BLOCK#ch%,6,1,sx%+cp%*6-6,sy%+8,CP(7)

2245 **END DEFine**

2247 **DEFine PROCEDURE** Add_chr

2248 IF cp% =1 AND sl%=0 :str\$=str\$&k\$

2249 IF cp%>=1 AND cp%<sl%:str\$=str\$(1 TO cp%-1)&k\$&str\$(cp% TO sl%)

2250 IF cp%>=1 AND cp%=sl%:str\$=str\$(1 TO cp%-1)&k\$&str\$(cp%)

2251 IF cp%> 1 AND cp%>sl%:str\$=str\$&k\$

2252 IF cp%=sm%:str\$(cp%)=k\$

2253 IF sl%<sm%:sl%=sl%+1 :ELSE sl%=sm%

2254 IF cp%<sm%:cp%=cp%+1 :ELSE cp%=sm%

2255 **END DEFine**

2257 **DEFine PROCEDURE** Del_chr

2258 IF cp%=sl%:str\$=str\$(1 TO sl%-1):sl%=sl%-1

2259 IF cp%>=1 AND cp%<sl%:str\$=str\$(1 TO cp%-1)&str\$(cp%+1 TO sl%):sl%=sl%-1

2260 IF cp%=sm%:str\$=str\$(1 TO sm%-1):cp%=cp%-1:sl%=sm%-1

2261 IF cp%=1 AND sl%=1:str\$="" :sl%=0

2262 **END DEFine**

2264 REMark PIXArt Help Screen

2266 DEFINE PROCEDURE Info

```

2267 LOCAL ik%,ix%,iy%,ip$ :CLS#3:CSIZE#3,0,0
2268 QBold 3,6,6, 4,'PIXELArt' :QBold 3,6,270, 4,'HELP [?]'
2269 QBold 3,6,8, 28,'SPRITE' :QBold 3,6,254, 28,'GRID'
2270 QBold 3,6,8,129,'SCREEN':QBold 3,6,245,129,'TILE' ◀ ▶
2271 QBold 3,6,8,154,'RETRO GAME'
2272 OVER#3,1:INK#3,CP(7):RESTORE 2274
2273 FOR i=1 TO 28:READ ix%,iy%,ip$:CURSOR#3,ix%,iy%:PRINT#3,ip$
2274 DATA 66, 4,"Use ◀ ▶ ↕ ↘ Cursor Keys to Navigate"
2275 DATA 54, 14,"Abort/Rtn Spacebar Action ◀ Enter"
2276 DATA 50,29,"Frame '-/+ ' 'G' Set Frame Size 16x16"
2277 DATA 12,43,"'X'FLIP'Y' 'z'ROLL'Z'",12,54,"'#' MAX min Edit"
2278 DATA 65,69,◀,65,87,◀,89,87,%/,80,93,◀ ▶,89,99,◀
2279 DATA 18,79,"MOVE ◀ ALT ▶ ReSize",38,102,"Set/Drag CURSOR"
2280 DATA 18,113,"PAN'◀▶ Shift ◀ ▶ SCROLL"
2281 DATA 160,40,"'N'New 'C'Copy 'D>Delete",160,50,"'I' Import FONT"
2282 DATA 160,60,"'P' Palette Colour Chg",160,70,"'B' BackGnd Colour Chg"
2283 DATA 160,81,"'R' ReCol",157,91,"'SB'",184,91,"Paint ON OFF"
2284 DATA 160,101,"'E' Erase ON OFF",160,112,"'@' Fill"
2285 DATA 50,130,"(1..9) 'T' Set Asset 0..",278,130,0 00'
2286 DATA 12,141,"'M' Maps SCREEN Links - TILE Size - Switch[#]Slide"
2287 DATA 140,154,"'A' Actions 'P' Play"
2288 DATA 12,168,"'F' DIR Settings 'L' Load _fnt _bmp 'M' Merge _bmp"
2289 DATA 12,180,"'S' Save SPRITE _bmp SCREEN _bmp RETROGAME _bas"
2290 OVER#3,1:ik%=CP(7) :BLOCK#3,12,3,112,18,ik% :BLOCK#3,2,4,234,16,ik%
2291 BLOCK#3,12,9,61, 54,ik% :BLOCK#3,10,7,62,55,0 :REMark MAX[ ]min
2292 BLOCK#3,6,5,100, 56,ik% :BLOCK#3,4,3,101,57,0
2293 BLOCK#3,18,4,244,93,ik% :BLOCK#3,16,4,293, 93,ik% :BLOCK#3,14,2,294, 94,0
2294 BLOCK#3,12,9,247,101,ik%:BLOCK#3,12,9,296,101,ik% :BLOCK#3,10,7,297,102,0
2295 BLOCK#3,6,5,40,67,ik% :BLOCK#3,5,4,41,68,0 :REMark [ALT] Edit Area
2296 BLOCK#3,6,5,90,67,ik% :BLOCK#3,5,4,90,68,0
2297 BLOCK#3,6,5,40,96,ik% :BLOCK#3,5,4,41,96,0 :BLOCK#3,3,7,5,89,96,ik%
2298 BLOCK#3,12,9,247,81,ik% :BLOCK#3,10,7,248,82,0
2299 BLOCK#3,12,9,247,112,ik%:BLOCK#3,10,7,248,113,0
2300 QBold 0,7,130,4,'Exploring PIXEL Art and RETRO GAME DESIGN'
2301 END DEFINE

```

