

## Introduction

The appeal and accessibility of Computer Games with home computers expanded rapidly with their extended graphics capabilities and creation of 2D object named Sprites. These Images are represented by the tiny dots, pixels, displayed to screen. As a Matrix written in columns and rows are stored in memory as Bitmaps.

Creating Pixel Characters and Background screens soon became a recognised Art form. Game Screens in the nineteen eighties were typically 256 by 192 pixels, limiting detail, where the best of the Sprite Designers could imply a face with an 8x8 pixel Bitmap.



## QBITS PIXEL Art

The original aspiration was to write a Program in SuperBASIC that could be used to create simple RETRO Games for the Sinclair QL Platform. Unfortunately, the performance of running a program with the Interpreter on a BBQL was unacceptably slow and Sprite Bitmaps require lots of memory. Today with the extended range of QL Platforms where speed and memory may not be the issue, the coding compatibility becomes the primary concern.

## QBITS Prog Concept

The approach taken was to produce a Program environment that build in stages the necessary elements of a Retro Game. **Stage One** being a Sprite Designer, which was a rewrite of code taken from the earlier **QBITS BITMap Designer** and **QLFont Editor**.

**Stage Two** was to assemble Screens backgrounds with Sprites designed as Tiles. These being copied across from the Spite Designer to form a Tile Library, then deployed to build sections of the background. Added to this a method to MAP links between multiple screens.

**Stage Three** to cover Action Sprites either Player controlled, independently moving or in fixed locations acting as Hazards or Rewards. A quick test to explore their actions linked with an Editable Game Title and setting Hazard or Reward of changes to the Score and Lives count.

**Stage Four** Firstly a Test Run of the constructed elements of the Game. Then the ability to save as an independent Retro Game with Title, Menu and Play Instructions.

## QBITS PIXELArt - Quick Overview

Developed using the QPC2 environment this offers several Colour Modes. The **COLOUR\_QL** Mode displays 0-7 as Black Red, Majenta, Green, Cyan, Yellow and White with 8-255 as a colour, contrast, and stipple combination. **COLOUR\_PAL** Mode uses 0 to 255 as a range of 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 PIXELArt - File Management

To change File **DIR**ectory Press (F) Select Drive and Enter. Then Press (E) to Edit Drive/SubDIRectory name. Enter to Complete.



Press (L) for **Load**. Select Drive with Up/Down Cursors and Enter. Program carries out a search for QBITS



**\_bmp** files. Select a file with Up/Down Cursors. Abort with Spacebar or Load with Enter.



Press (S) to **Save** Current File. Press (E) to Edit filename. A file saved in **SPRITE** Mode will contain basic header. A file saved in **SCREEN** Mode has an extended header with Screen BackGnd and **MAPing** information.



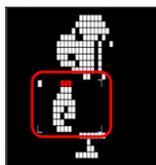
If file exists an Overwrite Y/N prompt is displayed.



## QBITS PIXELArt – (G)RID Mode

Loading a **SPRITE \_bmp** file will set the Grid Size. Press (G)RID to change, use Left/Right Cursors for Columns, Up/Down for Rows. Abort - **Spacebar** or Action - **Enter**. Then you can use **-/+** to select a Sprite Frame. If set to **0** the entries are displayed in Pixel Grid Size as a group across central screen. Select (N)**ew** (C)**opy** (D)**elete** for other Sprite Frame commands.

Use **xX yY**. to **Flip** horizontally and vertically. **Z** to **Rotate** anti-clockwise and **z** Clockwise 90°. To **PAN** or **SCROLL** use **Shift** with Left/Right Up/Down Cursors.



Use Max [#] Min to manipulate the whole Grid or a Selected area.



Select a Grid Cell with **Cursor Keys**.  
 Toggle Paint ON/OFF with **Spacebar**.  
 Clear Grid Cell with [R]ubber.

Use **TAB** or **< >** Chevrons to change colour.  
 Press (B) to change (B)ackGnd Colour.  
 Press (P) to change (P)alette Colours 8..15.



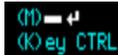
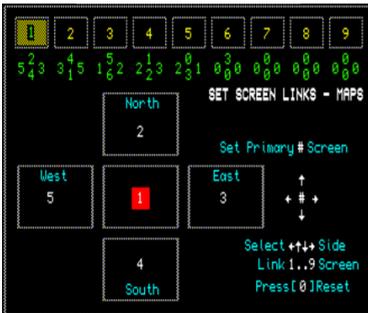
### QBITS PIXELArt – (T)ILE Mode

Use **-/+** to Select Tile to display in SPRITE Frame. Press **C** to copy Sprite Tile to Library. It is shown in Tile Frame top right with number below **<0001>** incremented. Select a Library Tile with **<>** Chevron keys.

Press **(D)** to Delete the Selected Tile.

Position Cursor and toggle Spacebar ON/OFF to **Paint** individual Tile, or as a row or column to SCREEN.

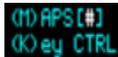
Use **[R]**ubber to Remove.



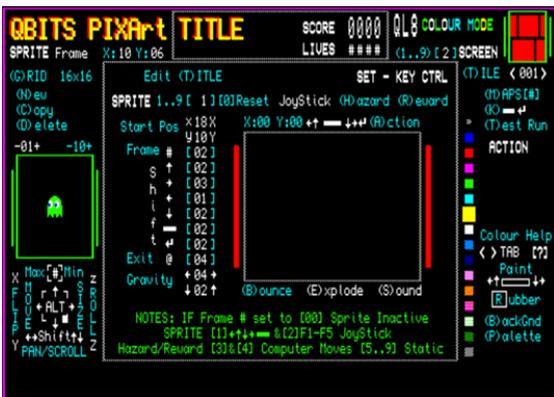
Press **(M)** to enter MAPS.

Press **#** to Select a Primary Screen **1..9**. Then use Cursors to Select a Compass direction and Set Screen Link **1..9**. The top of MAP screen displays the changed Link settings.

To Exit press Spacebar or Enter.



In Tile Mode Press **[#]** and move Cursor to an EXIT point to Test SCREEN Links.



### QBITS PIXELArt - ACTION

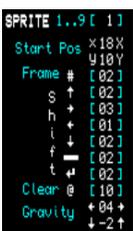
**Note:** Load a Sprite **\_bmp** File First.

Press **(K)ey CTRL**

Select SPRITE **1..9** Set as **(C)onTRL (H)azard** or **(R)eward** + Stationary or to Move using Gravity settings. Hazard and Reward need settings for Score and Lives



Edit (T)itle



Select **SPRITE 1..9**  
Start Position **xX yY**  
Select # Default frame  
  
Direction **Shift+Cursors**  
  
Jump **Shift+Spacebar** or  
**Enter Fire : Explode @**  
Gravity **X-X Y-Y axis**



Use **(A)**ction to Test SPRITE

Set Collision Response  
**(B)ounce** On/Off or **(E)xplore**  
with **(S)ound** On/Off

