

Introduction

The appeal of computer games prompted by eighties home computers expanded rapidly with extended graphics capabilities and creation of 2D images named as Sprites. These Raster Graphics Images are represented by the tiny dots, pixels, displayed to screen. As a Matrix of dots written as columns and rows and stored in memory as a Bitmap.

The creating and modifying of these Pixel Characters soon became a recognised Art form.



The Game screens in the nineteen eighties were typically 256x192 limiting size and detail, leaving the best Sprite Designers to imply a face with just a six, seven or eight-pixel matrix.

QBITS PIXEL Art

The original aspiration was to write a Program in SuperBASIC that could be used to create a simple RETRO Game 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. A more advanced QL Platform with speedier processing and extra memory was therefore seen as a necessity. 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 create a Game building environment in Stages. Starting with **Stage One** the Sprite Designer, this was in essence a rewrite utilising code from the QBITS BITMAP Designer and QLFont Editor.

Stage Two was to assemble Retro Screen Backgrounds by use of Sprites designed as Tiles produced in Stage One. These then being copied across to form a Tile Library and used in the build of background screens. Then a method to Map connections between different screens.

Stage Three was to cover Action Sprites to act as Hazards, Rewards or Player controlled. Some Hazards or Reward Sprites to move independently. Then the ability to explore these attributes with limited testing of their functions.

Stage Four would be to save as an independent working Retro Game with embedded Title, Menu and Play Instructions.



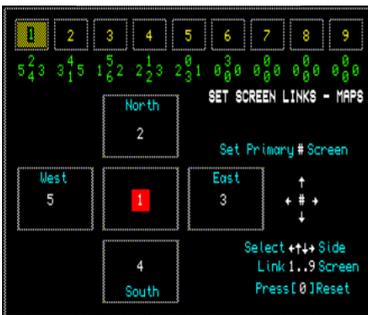
QBITS PIXELArt – (T)ILE Mode

Use **-/+** to Select Tile to display in SPRITE Frame. **(A)dd** Sprite Tile to Library, by pressing **(A)**. It is shown in Tile Frame top right with number below **<0001>** incremented. Select a Tile using **<>** Chevron keys.

Press **(D)** to Delete 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.

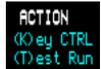


QBITS PIXELArt - ACTION

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

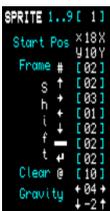
Press **(K)ey CTRL**

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



Added or Lost

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



Collision Responses can be set to **(B)ounce**, **(E)xplode** or **(J)ump** [as in jump to another screen] **(S)ound** ON/OFF for these actions

Use **(A)ction** to Test Sprite Functions
End by Pressing **(ESC)**.

