



Sinclair QL Retro-Computing



Sinclair QL Retro-Computing



QBITS SuperBASIC Progs

QL 40th	Reflections
Introduction	In the beginning
QBITSBoot &Config	Source Device Settings
QBITSProgs	Progs Menu
File Tidy	File Management
QL Font Editor	Modify & Create New Fonts
QL Fonts	Font Viewer
PIXEL Art	Retro Game Design
3DGraphics	Revolving 3D Objects
QL Sounds	BEEP Tracker & SCORE Sheets
Organiser	1980's Style PDA
Enigma	WWII Cypher Machine
Lander	Lunar Lander
GAMES One	
Tic Tac Toe	Classic Nought and Crosses
Mine Detector	Clear the Mines
Tile Slider	Sliding Tiles Puzzle
Conundrum	Word Puzzle
Darts	301/501 or Clock Face
Golf	18 Holes Course + Scorecard
GAMES Two	
Warehouse	Manage Invoices & Stocks
Trader	Buy/Sell Stock Market Shares
Karnak	A Maze Game -5 Levels
Pandemic	Battle against a Global Outbreak
AD2375	A Galaxy Adventure

QBITS SuperBASIC Progs 2024

QL's 40th Anniversary

1984 Sinclair's Quantum Leap

Launch of the QL came with a ROM loaded operating system QDOS and a SuperBASIC Interpreter for command input. The price included the PISON Office suite Quill (Word processor), Abacus (Spreadsheet), Archive (Database), Easel (Graphics) which used a common interface with a Ribbon style command display and all of this four years before Microsoft Office.



1985 a QL plus Monitor could set you back £650.00 The Apple Mac was close to three times the price and without an Office suite. But it was GUI driven.

The two 100k microdrives a bit quirky, 3^{1/2} Floppy drives with extended RAM were soon available as were Printers and Software packages from 3rd Party suppliers, Forth, Supercharge, Text87, Spellbound, my list started to grow.

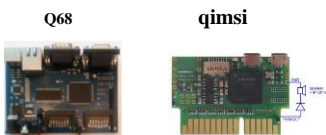
QL production terminated in 1986 after a sell off to Amstrad. At the time I considered an upgrade to a Miracal Systems Gold card or a CST Thor. Later a Miracal QXL card to go with my first PC, bought in the late 80's. Being busy at work and a growing family by the early nineties my QL involvement was in decline. Two BBQL's, RAM and floppy drive extension, software and documentation were all reluctantly stored away for another day.

A New Millenium

Into the 21st century and taking an early retirement offer with a half decent pension avoided Gordan Brown's 100 billion steal on final Salary schemes. As I had always intended, I started researching the family genealogy and writing a novel or two, by chance I found Jimmy Montesinos **QL2K Emulator** and interest in SuperBASIC was rekindled.

Later I found out the full extent of Tony Tebby **QDOS** upgrade to **SMSQ/E** and advances with the **QPC2 Emulator** under Marcel Kilgus. This prompted a revamping of old Progs and fulfilling some 1980 aspiration by extending of the **QBITS Suite of Progs**.

Today we have QL Emulators for MS Windows, Mac or Linux Operating Systems. For now, my preferred choice is Windows 10 and the **QPC2 Emulator**. However, I am again considering hardware variants, Peter Graf's development of the **qimsi** and **Q68** with Derick Stewart's production of the later have me once more intrigued.



Who knows what might come next?



My QL gear now gathered together in a new storage box is ready for a reawakening.



Coding verse Creativeness

The early QBITS SuperBASIC Program coding was more often than not a trial-by-error. Sometimes aspirations of the time were met, but in most cases if they attained a certain level of accomplishment, they were deemed acceptable. Such experimentation and lack of forward planning trends to levels of inconsistency and less than desirable coding, there often being more than one way to obtain a result. The obvious choice is for concise coding, but at times it comes down to just a question of preference.

For example: **(Esc)** **(E)**nd **(E)**xit or **(Q)**uit to leave a Program.
More ordered actions for File Management **(L)**oad & **(S)**ave.

Review of the QBITS Progs presents the opportunity to achieve more conformity, and for this the QPC2 Emulator has been a major influence. Most of the QBITSProgs can run standalone, however with QBITSBoot a common set of default and common Storage devices can be implemented via creation of QBITSConfig.

QBITSProgs presents a Menu of the Programs and has helped identify and tidy up bits of code that needed more structure. In some cases Program coding has been altered to smooth abnormalities and extended in others to add more functionality. Most were just tweaks to improving the appearance and flow of the Program.

QBITS Progs hopefully reflect a certain QL conformity of style and have evolved with a certain panache of their own.

A handwritten signature in black ink, reading "Cole Krill". The signature is stylized with a large, flowing 'C' and 'K'.



Introduction

Microcomputers released to the home market in the mid nineteen eighties came with a BASIC Interpreter. The **B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode constructed in a FORTRAN style of one-to-a-line statements. Variants evolved among the home computer manufactures with some to become quite sophisticated and yet small enough to fit within memory constraints of the day. Computer Magazines published BASIC code lists for Games and Utilities and for a while BASIC became the defacto standard for introducing beginners to Computer Programming.

In nineteen eighty-four during a College Summer Recess, I managed to get some work experience in the computing department of Aberystwyth University (Wales). I spent most of my time etching circuit boards, but it was also to be my first sighting of the Sinclair QL and much talked about multi-tasking operating system. The QL was under reviewed in particular the PSION business programs Quill (Word Processor), Abacus (Spreadsheet), Archive (Intelligent Database) and Easel (for Drawing Charts etc.). It had an external ROM which I believe included a release of the SuperBASIC Interpreter.

The Sinclair QL

I bought my first QL (Quantum Leap) computer in 1985 a few months before the price dropped from £399 to £199. My experience of programming at the time was fledgling, an introduction to machine code, a basic knowledge of some Forth commands and a few lessons of BASIC on an BBC micro. In starting a new job, we had just received our first IBM PC AT (£1200 Plus) DOS operating system and with twin 5¼ Floppy drives, the display green characters on a mono screen. The software IBM BASIC, MS Word, Lotus 123 Spreadsheet were bought as extras.

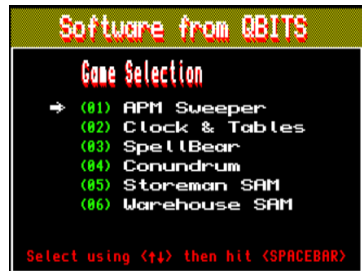
An early addition to my QL setup was a Trump card increasing RAM to 640kbytes and expanding my storage capacity with dual 3½" Floppy disk drives. It came with a release of Toolkit II which improved and extended the SuperBASIC list of Keywords. The drawback of those times, computer platforms weren't fast enough to satisfy the growing demands on running BASIC code through the Interpreter. Writing your program in Assembly or Machine code greatly increased the speed of execution. I did try my hand at writing some Assembly Code, then along came SUPERCHARGE and Compiling SuperBASIC was a much easier and less time-consuming prospect.

QBITS - It's all in the NAME

Located in a Village called Lolworth near Cambridge (UK), I attended **QUANTA** club meetings held in the old school hall. I gained a lot of programming advice and help from the members and first met Steve Bourne, who had just begun selling QL hardware. He encouraged me by suggesting he sold copies of my fledgling Progs. I vaguely recall a conversation discussing QL Bits and Bobs and from which the name QBITS for his Trader's name and my Software ensued. The QBITS Progs became an added contribution to Steve's wares as he trawled around different QL Club venues and shows back in the late eighties and early nineties.

QBITS Software

Not overly sophisticated, the beginnings of QBITS software were low-key. Shown is a selection of QBITS Games Steve carried as part of his Trading stock. There were other Progs such as FTidy128.exe which I recall as being Compiled with SUPERCHARGE.



QL SuperBASIC

The QL User's Guide introduces SuperBASIC and provides programming instructions. After setting up **WINDOW's** & **BORDER's** I began exploring the variety of ways in which to display Character fonts using **PRINT** with different colours and backgrounds utilising **PAPER**, **STRIP** & **INK**. A bigger impact was dropping the **AT** line/column Keyword for the more versatile **CURSOR**. Used with **CSIZE** and **OVER**, I could create different font sizes and even 3D affects. **CLS** options **FILL\$**, **LEN**, **SCROLL**, **PAN** added further to what could be achieved with character displays.

QL SuperBASIC Characters and Strings

CURSOR x,y:PRINT String\$& FILL\$(' ',LEN (String\$))

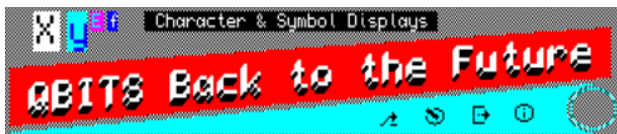
Maintains a set string length 'sl' by filling EOL with 'Spaces'

CURSOR x,y:PRINT FILL\$('0',4-LEN(number*)&number% [0000-9999]

Strings for units, ten, hundreds, thousands etc. where numbering expands from right to left.

CURSOR gx,gy,px,py:PRINT String\$ [graphic gx,gy coordinates px,py pixel offset]

The **CURSOR** option where **PRINT** is positioned relative to the Graphic coordinate system.



CSIZE 2.1:Str\$='QBITS back to the Future': FOR i=1 TO LEN(Str\$):**CURSOR** 20+i*12,60-i:PRINT str\$(i)

This uses a **FOR** Loop to create a **CURSOR** offset and **PRINT** a string - character by character.

QBITS Progs use the Cursor keys ← ↑ ↓ → to navigate and for Actions — Spacebar and ↵ Enter key.

BLOCK commands are used to provide Spacebar and Enter tail.

CURSOR 24,20:PRINT 'Select using ← ↑ ↓ → — ↵ :**BLOCK**#0,12,3,130,24,7:**BLOCK**#0,2,4,198,22,7

QBITS Boot & Config

At start up or reset the **SuperBASIC Interpreter** will Load and Run a program called **'Boot'** if found on the assigned default drive. This file is used to Load Extensions to the O/S and Executable Programs to memory. Whatever choice of default device, the Boot File will need to set a link to **LRUN QBITSBoot_bas**.

(See Page 9 for **QPCII Emulator** Download and Installation)

Bottom left of the **QPCII Configuration - Device** page are **Boot** options **FLP 1** or **2** or any of the **WIN1** to **WIN8** drives. The Default is **WIN1**.

QBITSBoot_bas file when **LRUN**, prompts for a default Source Drive **dev\$**. It then displays the generated **QBITSConfig** entries, these are common variables used by the **QBITS Programs**. Settings for **gx**, **gy** locate the QBITS backward compatible **512x256** screen size to sit within the higher screen resolutions of the **QPC11 Emulator**. When Exiting from a QBITS Program, **LRUN dn\$** [set to **'dev\$_QBITSProgs_bas'**] returns to the **QBITSProgs** Menu program.



For those Progs using **Load/Save** Options: Settings for **dn%**, **dm%** and **Drv\$(dn%)** are linked with Storage Device names such as: **mdv1_**, **flp1_**, **win1_**, etc.

To Edit Press **CTRL&Spacebar** and then type **'ED'** to change **QBITSConfig** settings.

Press a single KEY - First **DELETE**'s existing **QBITSConfig** file to creates a new one, it then **OPENs** and **Overwrites** the **QBITSProgs Menu** file default device with **dev\$**. Its last action is to **LRUN's dev\$&'QBITSProgs_bas'** and display the QBITS Progs Menu.

Note: **dev\$** must hold a copy of **QBITSProgs_bas**

Use **Cursor Keys** to Navigate through the Prog Entries, a brief description is given on each in the bottom window. Select one then Press **Enter** to **LRUN** the program.

Press **Esc** key to Quit QBITSProg



QBITS Progs

This Collection of **QBITS Progs** was written and edited with **QPC2 Emulator**. However, most of the Progs should **LRUN** on other **QL** Platforms with expanded memory and relative Toolkits of updated and expanded Keywords. You may need to tweak some of the code and bear in mind Progs run on a **BBQL** Platforms using the Interpreter will be slow. However, QBITS goal is to promote and explore the simple to the more complex use of the **SuperBASIC** Code Environment. Have fun...

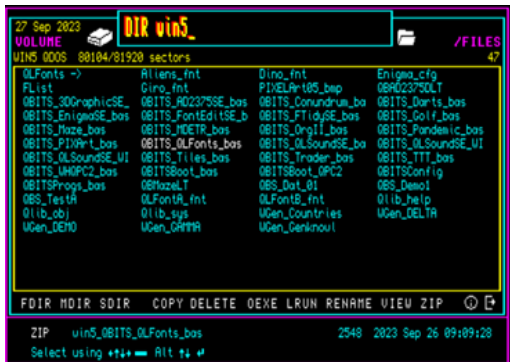
QBITS FTidySE

For the Management of an ever-growing number of Files this evolved to meet the needs of the day. The first QBITS coded Prog was called FileTidy then first released as FTidy128 and later shortened to just FTidy.



Opening Info Screen

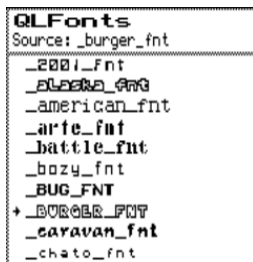
Compiler Screen.



Filenames shown in four columns

FTidySE includes **EXEC** for **_obj** Files and **ZIP** which links a Selected **_bas** file to be Compiled by **QLiberator**.

Note: ZIP only applies to the QPC2 environment.



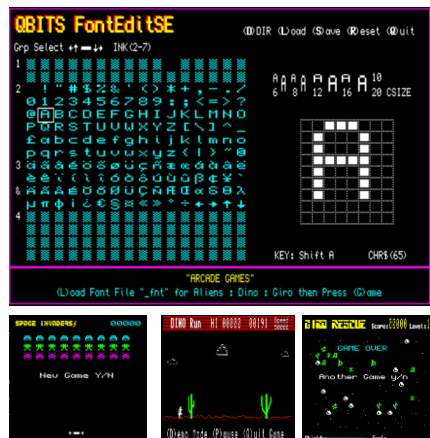
QBITS QLFontS

Source: Type Device[SubDIR] Press Enter: Scroll up/down the Character Font Themes displayed. Select and Press Enter: Returns to QBITSProgs Menu displayed with Selected Font.

Source: Leave Blank then Press Enter to Set QL Default Fonts

QBITS FontEditSE

This was written to Change and Create New Fonts. The default QL Character Fonts use an 8x9 (Eight bits by 9 Bytes) matrix. The Bitmap file holds a two-byte header which identifies a start Font and the number of Font Bitmaps that follow: 9-Bytes added for each Font in group.



DEMO Fonts to Load and Play Retro Games

Aliens_fnt

Dino_fnt

Giro_fnt

The aim, develop a Retro Game coding environment. Functions to scale and multicolour font sets and/or create new Spites or Tile backgrounds. Map background screens.

[illegible]

This investigates aspects of representing 3D objects in a two-dimensional environment. It all began with a Rotating Wireframe Cube and expanded too Solid and Coloured Frame Objects.

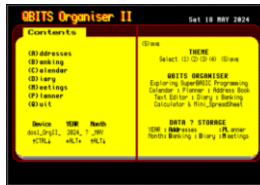
Page 5



QBITS Organiser II

The ambition was to create a nineteen eighties style Organiser with pages for Addresses, Banking, Diary, Meetings, plus double pages for a Calendar and Forward Planner.

Contents Page



Set Page Theme

Select Default Device
Set Date

QBITS Enigma

This WWII Enigma Simulation came about by way of a Family interest in Bletchley Park and a chance to Review a past QL Magazine article.

For DEMO Press (9)

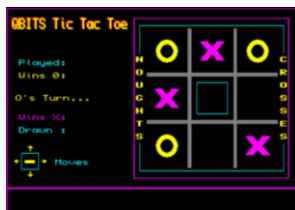


QBITS Lander

In celebration of the QL's 40th another icon of computer history is this tribute to the Lunar Lander Genre. SIM I a hark back to the original Text version. SIM II a 2D Graphics adaptation.

Both with DEMO options.

RESERVED Space

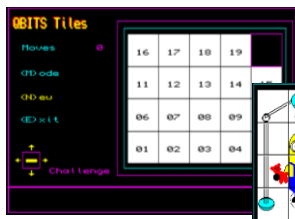


QBITS Tic Tac Toe

The classic coffee break challenge.

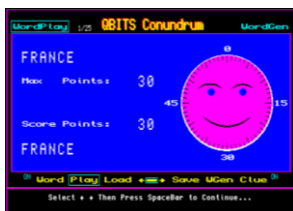
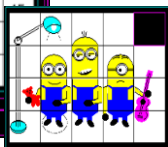
QBITS Mine Detector

Based on the 1980's Mine Sweeper.



QBITS Tile Slider

This is a take on Sliding Tiles Tablets with alternative Modes: Numbers or Minions Image.



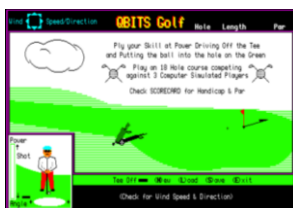
QBITS Conundrum

Victorian Hangman was the background to a QBITS Prog WordBear, bursting balloons, before upgrading to this version of the popular Word Game. It includes a Text Editor WordGen for creating new Word and Clue Files.



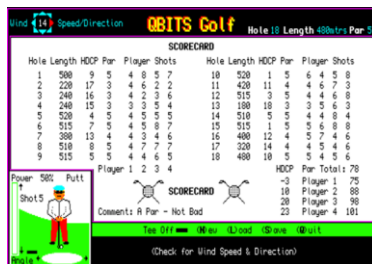
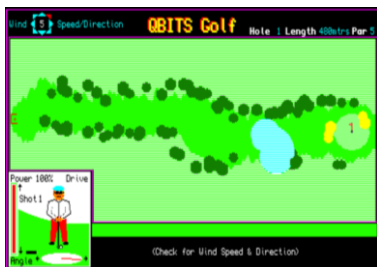
QBITS Darts

Classic 301 501 or Clock Face. Use Sliders to centre your Darts aim. At End of Play the Dartboard Scrolls up to show Winner and the number of Darts thrown.



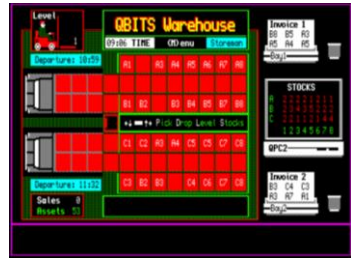
QBITS Golf

Play an 18 Holes course with Fairways variable in length and difficulty, the wind an added factor. The Scorecard results are displayed for each hole and at the end gives a Player's Handicap.



QBITS Warehouse

The Warehouse Prog started out as Storeman Sam driving the pickup (top left). It then expanded to a larger store with two lorry bays and four levels. The Printers show Sales Invoices and requested Stock Deliveries. The PC keeps Track of all held Stocks. Sales are generated by departing Lorries loaded with Goods. Game includes Hazards, Stock losses etc. and Progress Chart of Sales/Stock.



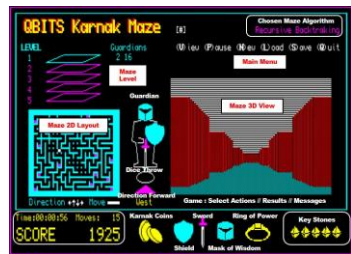
QBITS Trader

Here you take charge of a Trader's Portfolio. Buy and Sell Company Shares on the Stock Market over a three-year period. Hazards include Stock Market fluctuations in Share Values and changes to Company Dividends.



QBITS Karnak Maze

This shows a Plan and a moving 3D view around a Maze over five levels. Discover coins and other artifacts to collect while defeating the Phantom Guardians. The final level gives access to the Sphere of Destiny. Gamble your Score to correctly set the Stones of Destiny. This opens a Portal to the past and you save the day preventing a Human Genetic defect countdown to self-destruction.



QBITS Pandemic

The outbreak of a deadly virus and you are in charge of a group of Specialists deployed to find a Cure and Eradicate the Infection from all Cities. Any successful outcome will depend on actions taken, number of random Virus Outbreaks, Spread of City Infections and before your turns run out.



QBITS GALAXY AD2375

In the Role of Alliance or Republic, take over the Galaxy, Star system by Star system either by Acquisition or Military Force. Expect encounters with Enemy Fighters, Space Anomalies and unexpected Twists of Fortune along the way...

DEMO: Start New Game: Press F1 for Simulation Mode, then sit back and watch as things unfold.



QPC11 Emulator

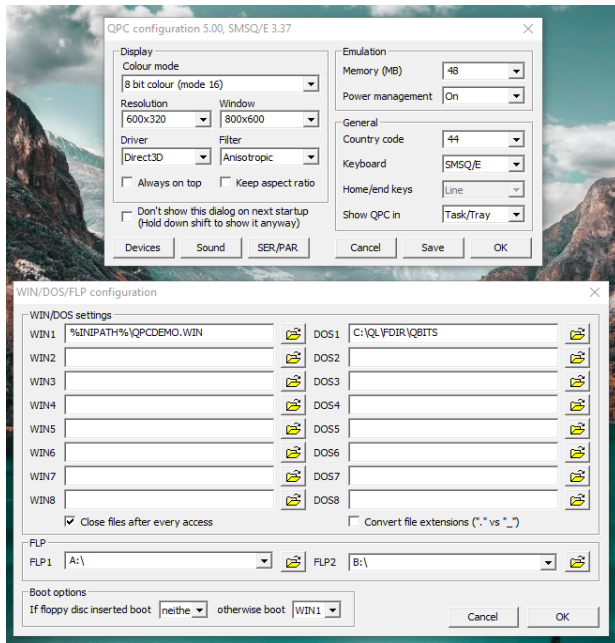
Installed and run on a Windows PC this Emulates a Sinclair QL Computer. However, it has a far more advanced O/S with Tony Tebby's SMSQ/E the successor to his QDOS and an updated expanded SBASIC to the QL SuperBASIC of Jan Jones day.

Downloads: <https://www.kilgus.net/qpc/downloads/>
Also Check out: <http://www.dilwyn.me.uk/emu/index.html>

QPC11 Manual

Check latest release of QPC11 it explains Installation, Concepts and SBASIC keywords. QPC Screen resolution and size is extended from the original 512x256 with additional Colour Palettes.

Download and follow the documentation's instructions to Install. Start **QPC11** and change the configuration to that shown below: -



Download **QBITS Progs** and UNZIP into a New Files Folder. In **QPC Configuration** Click on **Devices** and link **DOS1** to your **QBITS** Folder, press OK and then **Save**.

Press **Start** and with **QPC11** up and running exit from the demo page and in the SuperBasic Interpreter's Command Window type: - **LRUN Dos1_QBITSBoot_bas**

QBITSBoot_bas Code

All Journeys have a beginning, having Downloaded, UNZIPed and installed the **QBITS Suite of Progs**. Then check if present setting meets your local requirements. Feel free to make any changes.

100 REMark **QBITSBoot_bas** (QBITS Boot 2024 QL 40th - QPC2) vM30

120 **dev\$='win1_'**:MODE 4

140 WINDOW#2,512,222,0,0 :PAPER#2,0:BORDER#2,8,7:CLS#2:INK#2,7

150 WINDOW#1,492,208,12,6:PAPER#1,2:BORDER#1,1,0:CLS#1:INK#1,7

160 WINDOW#0,512,32,0,224:PAPER#0,0:BORDER#0,1,7:CLS#0:INK#0,7

180 CSIZE 2,1:**QBold 190,12,'QBITSBoot':QBold 180,74,'QBITSConfig':**CSIZE 1,1

190 BLOCK 320,90,86,100,0:**QBold 92,42,'Select Source Drive? [ie win1_]**

200 CURSOR 356,42:INPUT dr\$:IF dr\$<>":dev\$=dr\$:END IF :**QBold 356,42,dev\$**

210 str\$="Press a KEY to Continue OR 'CTRL+SpaceBar' to Edit Settings"

220 CSIZE 0,0:CURSOR 66,194:PRINT str\$:**RESTORE 260**

240 REMark *** **QBITSConfig Settings** ***

250 **gx=0:gy=0** :REMark Screen Coordinates offset

260 **dn\$=dev\$&'QBITSProgs_bas'** :REMark QBITS Menu Return Path

270 **dn%=0**:FOR d=0 TO 15:**READ d\$**:IF d\$=dev\$:dn%=d:END FOR d

280 **dm%=15** :REMark dn% source : **dm%** max

290 REMark *** **Device List** ***

300 DATA 'mdv1_', 'mdv2_', 'flp1_', 'flp2_', 'win1_', 'win2_', 'dos1_', 'dos2_'

310 DATA 'mdv3_', 'mdv4_', 'flp3_', 'flp4_', 'win3_', 'win4_', 'dos3_', 'dos4_'

Note: Change settings as required for default drive **dev\$** - Screen **gx gy** coordinates, return file address **dn\$** and for Drive allocations **dn% dm% drv\$(dn%)**

330 REMark *** **QBITSConfig Screen Display** ***

340 **RESTORE 260**:STRIP 0:INK 7

350 CURSOR 100,106:PRINT 'Screen Coordinates : gx=':**gx**':':gy=':**gy**

360 CURSOR 100,120:PRINT 'QBITS Menu Return : dn\$=':**dn\$**

370 CURSOR 100,140:PRINT 'drive default/max : dn%=':**dn%**':':dm%=':**dm%**

380 CURSOR 100,152:PRINT 'drives dn% 0 TO 15 : drv\$(dn%)' :INK 4

390 FOR d=0 TO 7:**READ d\$**:CURSOR 100+d*36,164:PRINT **d\$**

400 FOR d=8 TO 15:**READ d\$**:CURSOR -188+d*36,176:PRINT **d\$**

420 REMark *** **QBITSConfig Format** ***

Note: Code to Overwrite / Create NEW Config settings

430 PAUSE:**RESTORE 260**:DELETE **dev\$&'QBITSConfig'**

440 OPEN_NEW#9,**dev\$&'QBITSConfig'**:PRINT#9,**gx|gy|dn\$|dev\$|dn%|dm%**

450 FOR d=0 TO 15:**READ d\$**:PRINT#9,**d\$**:END FOR d:CLOSE#9

470 REMark *** **Set ALTKEY** ***

480 ALTKEY 'M','LRUN '&**dev\$&'QBITSProgs_bas'**&CHR\$(10)

490 ALTKEY 'P','LRUN '&**dev\$&'QBITS_FTidySE_bas'**&CHR\$(10)

510 REMark *** **QBITSProgs Set dev\$** ***

Note: Code to Set a Programs Default Drive

520 OPEN#9,**dn\$**:INPUT#9,a\$b\$c\$:CLOSE#9:c=c\$(1 TO 11)&**dev\$&c\$(17 TO**

530 OPEN#9,**dn\$**:PRINT#9,a\$b\$c\$:CLOSE#9:LRUN **dn\$**:STOP

550 **DEFine PROCedure QBold(x,y,str\$)**

560 OVER 1:FOR i=0 TO 1:CURSOR x+i,y:PRINT str\$:END FOR i:OVER 0

570 **END DEFine**

Set Source Drive and **LRUN QBITSProgs_bas** to display Menu.


```

1053 DEFine PROCEDURE QBold(ch,col,w,x,y,str$)
1054 OVER#ch,1:INK#ch,col
1055 FOR i=1 TO LEN(str$):CURSOR#ch,x+w*i,y :PRINT#ch,str$(i)
1056 FOR i=1 TO LEN(str$):CURSOR#ch,1+x+w*i,y:PRINT#ch,str$(i)
1057 OVER#ch,0
1058 END DEFine

1060 DEFine PROCEDURE QBMenu
1061 CSIZE 1,0:QBold 1,7,7,432,188,'Esc Exit':KEsc 1,7,157,5
1062 QBold 1,7,7,24,182,'Select Progs with ⬅️⬆️➡️⬇️ Cursors and Action with ⬅️⬇️ Enter'
1063 AT 0,0:CSIZE#1,3,0:BLOCK#1,2,4,370,190,7:col=0:x=230:y=1:max=9:c$='⬅️'
1064 REPEAT Menu_ip
1065 IF x=230:RESTORE 1077+y:ELSE RESTORE 1088+y
1066 CLS#0:READ str$:sl=LEN(str$):CURSOR#0,248-6*(sl/2),10:PRINT#0,str$
1067 CURSOR x,18+y*11:PRINT c$:k=CODE(INKEY$(-1)):BLOCK 18,10,x,18+y*11,0
1068 SELECT ON k
1069 =192:IF x=260:x=230:max= 9:col= 0:c$='⬅️':IF y>max:y=max
1070 =200:IF x=230:x=260:max=11:col=11:c$='➡️'
1071 =208:y=y-1:IF y<1 :y=max
1072 =216:y=y+1:IF y>max:y=1
1073 = 10:DD$=dev$&'QBITS.'&Prog$(y+col):QBDev:LRUN DD$
1074 = 27,69,101:CSIZE#2,0,0:INK#2,7:EXIT Menu_ip Note: (Esc) to exit QBITSProgs for Editing
1075 END SELECT
1076 END REPEAT Menu_ip
1077 END DEFine

```

```

1079 DATA 'A File Tidy Program - Review and Manage File Directories'
1080 DATA 'QLFont Editor to Modify Character Fonts - Plus Retro ARCADE Games'
1081 DATA 'QLFont Viewer - Scroll Up/Down to check out QL Character Fonts'
1082 DATA 'Explore PIXEL Art - Create Sprites & Retro Games etc.'
1083 DATA 'Exploring 3D Rotation Graphics - Plus Escape POD Rescue Game'
1084 DATA 'Exploring the Musical Attributes of the QL BEEP Commands'
1085 DATA 'Exploring the Functions of a 1980s Style Personal Organiser'
1086 DATA 'Enigma - A simulation of the WWII Encipher/Decipher Machine'
1087 DATA 'Lander – A hark back to the beginnings of the Lunar Lander Genre'
1088 DATA "
1089 DATA "
1090 DATA 'The Coffee Break Challenge - Classic Noughts & Crosses'
1091 DATA 'Clear a Mine Field - Based on Mine Sweeper of the 1980s'
1092 DATA 'A Sliding Tile Puzzle Game with Numbers or MINIONS Picture'
1093 DATA 'Type Correct Order of Letters that Spellout the Hidden Word'
1094 DATA 'Classic Darts - Play 301/501 or Around the Clock Face Game'
1095 DATA 'Compete over an 18 Hole Course - SCORECARD with Par & HandiCap'
1096 DATA 'Manage a WareHouse - Handle Invoice Requests & Stock Deliveries'
1097 DATA 'Solve the Maze - Your Mission Travel back in Time to Save Humanity'
1098 DATA 'As a Market Trader - Manage a Portfolio of Company Stocks & Shares'
1099 DATA 'As a Specialist - Lead a Team to Contain & Eradicate a Deadly Virus'
1100 DATA 'Galaxy Adventure AD2375 : The First Order - Alliance v Republic'

```

Note: QBDev Overwrites the Source Drive '**dev\$**' for QBITSConfig & Input of Common Settings.

```

1102 DEFine PROCEDURE QBDev
1103 OPEN#9,DD$:INPUT#9,a$|b$c$:CLOSE#9c$=c$(1 TO 11)&dev$&c$(17 TO)
1104 OPEN#9,DD$:PRINT#9,a$|b$c$:CLOSE#9
1105 END DEFine

```

1200 REMark QBITSProgs Graphics

```
1202 DEFine PROCEDURE QFLASH(ic,x,y)
1203 INK ic:FILL 1:LINE 40,90 TO 5-x,80+y TO 7-x,84+y TO 40,90:FILL 0
1204 FILL 1:LINE 142,90 TO 177+x,80+y TO 175+x,84+y TO 142,90:FILL 0
1205 END DEFine
```



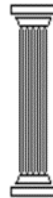
```
1207 DEFine PROCEDURE QL40
1208 FOR i=1 TO 4
1209   qx= 20+i*12:qy=15 -i:QPmnt qx,qy,c$(i):QPmnt qx,qy+1,c$(i)
1210   qx=402+i*12:qy= 9+i:QPmnt qx,qy,c$(i):QPmnt qx,qy+1,c$(i)
1211 END FOR i
1212 QTh 30,95:QTh 170.5,94
1213 END DEFine
```



```
1215 DEFine PROCEDURE QPmnt(qx,qy,q$)
1216 OVER 1:CUSOR qx,qy:PRINT q$:OVER 0
1217 END DEFine
```

```
1219 DEFine PROCEDURE QTh(qx,qy)
1220 LOCAL x1,y1,x2,y2::RESTORE 1222
1221 FOR i=1 TO 5:READ x1,y1,x2,y2:LINE qx+x1,qy+y1 TO qx+x2,qy+y2
1222 DATA -.5,0,+1,0, 0,0,0,-2, +1,0,+1,-2, +1,-.6,+2,-.6, +2,-.6,+2,-2
1223 END DEFine
```

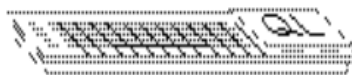
```
1225 DEFine PROCEDURE Pillar(ch,x,y)
1226 LINE#ch, x-5,y+21 TO x+6,y+21 TO x+6,y+23 TO x-5,y+23 TO x-5,y+21
1227 LINE#ch, x-3,y+18 TO x+4,y+18 TO x+5,y+20 TO x-4,y+20 TO x-3,y+18
1228 LINE#ch, x-3,y-18 TO x+4,y-18 TO x+5,y-20 TO x-4,y-20 TO x-3,y-18
1229 LINE#ch, x-5,y-21 TO x+6,y-21 TO x+6,y-23 TO x-5,y-23 TO x-5,y-21
1230 FOR c=1 TO 5
1231   x1=x-4+c*1.5-.3:x2=x-4+c*1.5+.3:y1=y+17:y2=y-17
1232   ARC#ch,x1,y1 TO x2,y1,-PI:LINE#ch TO x2,y2
1233   ARC#ch,x2,y2 TO x1,y2,-PI:LINE#ch TO x1,y1
1234 END FOR c
1235 END DEFine
```



```
1237 DEFine PROCEDURE KEsc(ch,col,x,y)
1238 INK#ch,col:CIRCLE#ch,x,y,2.2:LINE#ch,x,y TO x-2,y+2.8
1239 END DEFine
```

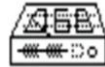


```
1241 DEFine PROCEDURE QLBB
1242 LINE 54,16 TO 70,16 TO 63,22.5 TO 47,22.5 TO 54,16
1243 LINE 5,21 TO 12,13 TO 69.5,13 TO 69.5,14 TO 12,14 TO 5,22 TO 47,22
1244 LINE 7,17 TO 13,11 TO 69,11 TO 69,13 TO 70,13 TO 70,16
1245 LINE 57,15 TO 61,15:LINE 64,15 TO 68,15:INK 255 :REMark microdrives
1246 FOR i=0 TO 4:LINE 13+i,20-i TO 47+i,20-i:LINE 9+i,20-i TO 11+i,20-i
1247 INK 7:CIRCLE 56,19.5,2.8,.5,PI/3
1248 LINE 56,19 TO 61,18:LINE 58.5,21.5 TO 62,18 TO 65,18
1249 INK 0:FOR i=0 TO 33 STEP 3:LINE 15+i,20 TO 21+i,14
1250 INK 7:LINE 25,16 TO 43,16
1251 END DEFine
```



1253 DEFine PROCEDURE Q68

```
1254 LOCAL x,y:x=95.5:y=20:INK 7:LINE 86,16 TO 104,16 TO 102,20 TO 89,20
1255 LINE TO 86,16 TO 86,11 TO 104,11 TO 104,16:CIRCLE 101.5,13,.8 :INK 248
1256 LINE 97.5,12.5 TO 99.5,12.5 TO 99.5,14.5 TO 97.5,14.5 TO 97.5,12.5:INK 7
1257 BLOCK 8,5,241,172,248:BLOCK 8,5,253,172,248:LINE 87.5,13.5 TO 97,13.5
1258 LINE x-3,y-3 TO x-3,y-1 TO x-5,y-1 TO x-7,y-3 TO x-3,y-3 TO x-4,y-2 TO x-2,y-4
1259 LINE x+1,y-1 TO x-1,y-1 TO x-1,y-3 TO x+1.5,y-3 TO x+1.5,y-2 TO x-1,y-2
1260 LINE x+3,y-2 TO x+5,y-2 TO x+6,y-3 TO x+3,y-3 TO x+3,y-1 TO x+5,y-1 TO x+6,y-3
1261 END DEFine
```



Note: 'Q'
Note: '6'
Note: '8'

1263 DEFine PROCEDURE QIMSi

```
1264 INK 7:LINE 120,16 TO 134,16 TO 133,20.5 TO 121.5,20.5 TO 120,16
1265 LINE TO 120,13 TO 121,13:LINE 133,13 TO 134,13 TO 134,16
1266 LINE 122,14 TO 121.5,11.5 TO 132.5,11.5 TO 132.5,15
1267 INK 7:OVER 1:CURSOR 332,160:PRINT 'qimsi':OVER 0
1268 BLOCK 26,4,334,174,248:INK 6:LINE 129,11.5 TO 129,14
1269 END DEFine
```



Note: Examples of Character Printouts using QL SuperBASIC Keywords.

ch channel bi background ink ci character ink cw character width cd character depth cx cy coordinates.

1001 DEFine PROCEDURE TBold(ch,ci,cw,cd,cx,cy,c\$)

Note: Print Title

```
1002 OVER#ch,1:CSIZE#ch,cw,cd
1003 INK#ch,bi:FOR i=0 TO 1:CURSOR#ch,cx+i,cy :PRINT#ch,c$
1004 INK#ch,ci:FOR i=2 TO 3:CURSOR#ch,cx+i,cy+1:PRINT#ch,c$
1005 OVER#ch,0:CURSOR#ch,0,0:CSIZE#ch,0,0
1006 END DEFine
```



Note: CSIZE of 20 pixels high can cause problems if following PRINT Statements fall outside Window size.

Note: OVER printing with CSIZE 0,0 blurs the characters but with a Character offset cw.



1100 DEFine PROCEDURE CBold(ch,ci,cw,cx,cy,c\$)

Note: Print Character

```
1101 INK#ch,col:OVER#ch,1
1102 FOR a=1 TO LEN(str$)
1103   FOR b=0 TO cs:CURSOR#ch,cx+b+a*(6+cs),cy:PRINT#ch,str$(a)
1104 END FOR a:OVER#ch,0
1105 END DEFine
```



Note: Horizontal Offset

1200 DEFine PROCEDURE SBold(ch,ci,cx,cy,c\$)

Note: Print String

```
1201 INK#ch,ci :CURSOR#ch,cx,cy :PRINT#ch,c$
1202 OVER#ch,1:CURSOR#ch,cx,cy+1:PRINT#ch,c$:OVER#ch,0
1203 END DEFine
```



Note: Vertical Offset

1301 DEFine PROCEDURE DPrnt

```
1302 LOCAL ch,i,x,y,p$:READ ch,i,x,y,p$:INK#ch,CP(i):CURSOR#ch,x,y:PRINT#ch,p$
1303 END DEFine
```

Note: Print DATA String



Note: The use of PAPER#ch,colour with STRIP#ch,colour are further Keywords to extend Print displays.

