Sinclair QL retro gaming

Sinclair QL retro gaming

QL Screen in High Resolution with 512x256 pixels grid.
Note: Dual byte horizontal, Single byte vertical.

QBITS Warehouse

QBITS Game Ideas

Reviewing my early nineteen nineties QBITS floppies I came across two programs STOREMAN_SAM and WAREHOUSE_SAM. I'm not sure what was I thinking at the time, I do remember a neighbour who we still keep in touch with had started a new job as Storeman for 3M. As for myself my companies IT team had been informed that the firm was to invest in a new innovation a Computer controlled Automated Storage and Retrieval System (ASRS). Being the mid nineteen eighties that was to be quite a project.

Amongst some documentation of QL programs, sketches and notes I came across a two sided sheet of graph paper headed ASRS. That was my starting point, the program STOREMAN_SAM revealed a simple layout of storage bays and a loading bay. Goods were moved around the store with a Pick-Up truck, then loaded onto a Lorry. This had to be done within a set time frame as the Lorry would drive off once the Departure time had expired. The Orders and Deliveries were identified by a printer output. A second level of stacked pallets added more gaming difficulties and introduced the graphics of Sam seated on his Pick-Up Truck to indicate which level was being accessed.

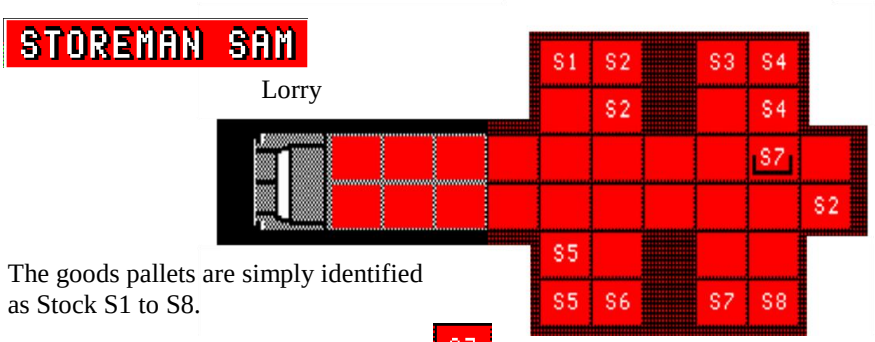
My original sketch showed dual loading bays, WAREHOUSE_SAM incorporated these, extending the code to up the ante so to speak. A Store Computer displayed the Warehouse Stock. To replenish diminishing stock a Stock Request was added and used the Store Computer to highlight the items for selection. Whereas STOREMAN_SAM had a simple Profit scoring on Orders being fulfilled, for WAREHOUSE_SAM I had ventured to add a different scoring method. This was in the form of an Audit Report, which when called displayed a Profit and Loss chart on the Store Computer.

As I remember it, running the SuperBASIC on an original QL hardware was not altogether fast. I have since found what appeared to be compiled versions. However having never quite achieved all that was expressed with the original concepts, I now set myself the daunting task of getting to grip with my earlier coding and hopefully to come up with a fully functioning programme.

QBITS Warehouse

STOREMAN SAM

A review of Storeman Sam showed four, four by four storage bays together with a temporary stacking area. A passageway for the Pick-Up Truck to move back and forth along, then a loading bay providing six storage positions when a lorry is backed into it.



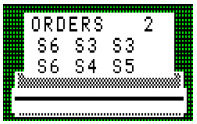
The goods pallets are simply identified as Stock S1 to S8.

The Pick-Up depicted by a tray shape 

is moved around using the cursor keys and spacebar to allow directional change within a single square.

Stacking the Goods to a second level is depicted by SAM and his Pick-Up Truck.

For Orders and Deliveries a printer readout lent itself as an ideal graphical representation.



The basic layout in place how were things to proceed. Having used the PAN SuperBASIC command for the Lorry in and out it seems I had employed the novel idea to PAN the introductory text. Then a Skill level - Forman, Storeman and Trainee (this altered the time allotted for loading and unloading).

Picking up and Depositing Stock on/off the lorry and in/out of the storage bays I adopted (P)ick and (D)rop and (L)evel for stacking the different heights.

So as SAM in charge of the store you load (Orders) or unload (Deliveries) from the lorries as they arrive and the game proceeds with a score generated by correctly loaded lorries being dispatched to their customers.

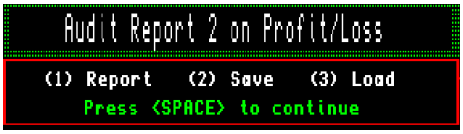
QBITS Warehouse

WAREHOUSE SAM

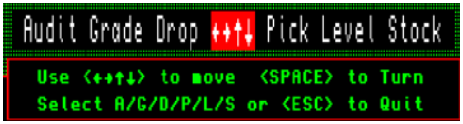
As a second generation, the layout of the display changed to include the dual Loading bays with their own individual printer displays. The Store Computer showing the current status of the Stocks held. Four levels not Two and not eight but twenty-four different types of good to handle (Labelled a1 to 8 : b1 to 8 and c1 to 8).

To replace diminishing Stocks a Stock Request to call for Deliveries, not a random generated Delivery as in STOREMAN SAM.

The Scoring now dealt with by calling for an Audit Report displayed on the Store Computer.



Skill and Speed alters the time elements of the Game Time.



Grading and normal move functions are all called from within the main Game loop. As before the various skill levels set the timing of Lorry Arrival and Departures and vary the game length. However, you can use the Esc key to quit at any time. The Stock at the start of the Game were defined for level one and randomly chosen for level three with level two and four left empty. In addition, I added the option to SAVE a current game and LOAD it at a later date so as to continue play.

QBITS Warehouse

QBITS Warehouse
Found amongst my QL retrieved floppies, were further bits of program code that indicated a third generation to STOREMAN_SAM and WAREHOUSE_SAM Intrigued I have spent more than a few hours getting to grips with the coding and have hopefully achieved a fully functioning program which I have now dubbed QBITS Warehouse...

There has been some further changes to the layout, for one I brought back the PickUp Truck and SAM to show the Level and a more logical continuously updated Score display. The Printers have moved more in line with their respective Loading Bays with the Store Computer now sitting between them. An added touch was to screw up each printout as its Lorry departed and deposit it in a waste paper basket. Recycling and help save the planet, that was probably my thoughts when writing the code back in the early nineteen nineties. Anyway I though it added nice a touch of thoughtful sophistication.

An Introductory page was added with a revamp of the Menu with added touches to the options for New Load Save Exit. An (M) key to access the Menu while playing a Game to SAVE and re-LOAD for later continuation, then an updated version of the Asset/Sales (Profit & Loss) Chart display.

Menu Options
Game status (**Gs**) allows or inhibits actions within the **Menu**. Loading and running QBITS Warehouse starts with the introductory page, any key takes you to the **Menu (1)NEW (2)LOAD (3)SAVE (4)EXIT**. The **SAVE** option is disabled at this point (activated after first Invoice is fulfilled in a Game). You can **EXIT** or Start a **NEW** game or **LOAD** a previously saved one. Once a Game is active, you can call the **Menu** by pressing the (**M**) key. If an Invoice has been completed all Menu options are made available. You can return to continue the present Game by pressing the Spacebar.

Gs=0 Menu called from Introductory page.
Gs=1 Return to continue Game
Gs=2 LOAD Game
Gs=3 SAVE Game go to GAME END
Gs=4 GAME END

SAVE or **LOAD** activates a **PATH** selection to identify a Drive name (**dn**) and an Audit Filename (**af**). To return without selecting press Spacebar. However if a **SAVE** is selected, after completion you automatically go to **GAME END**. A **LOAD** takes you straight into the Game with the next Invoice from the last one when the Game was saved.

QBITS Warehouse

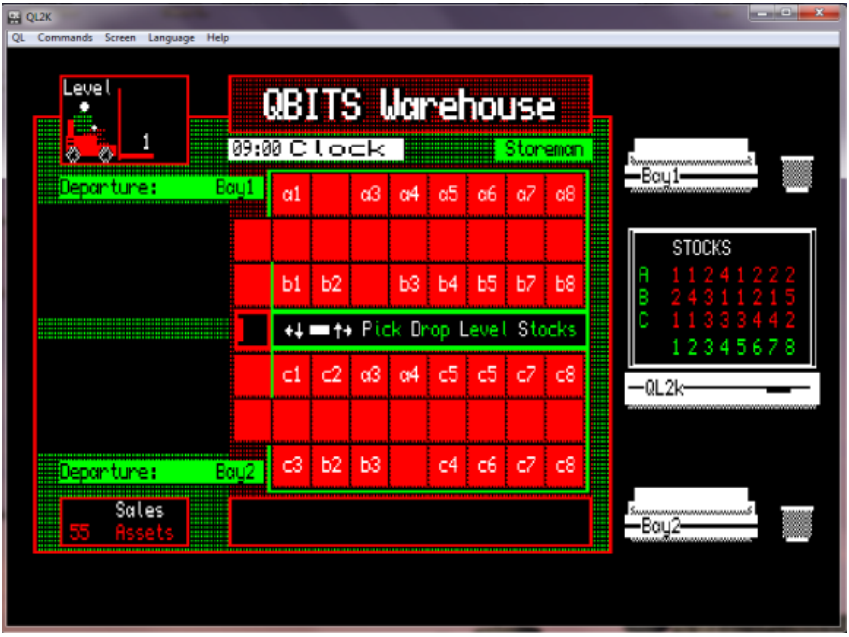
QBITS Warehouse Channels/Windows

For the movement of Lorries and Printer outputs the graphic displays presented a number of choices. Open a large number of new channels (windows) or simply resize a window according to the action to take place. I chose the latter.

Channel / Window defined by - w-*width*, d-*depth*, x, y-*coordinates*

Intro and Menu WINDOW#1 Clear for Opening Introduction and resize for options Menu.

Store Layout	ch=3:OPEN #ch,scr_356x194a16x30:PAPER #ch,32:CLS #ch:BORDER #ch1,2
Game Title	ch=3:WINDOW#ch,216,26,13,x10:PAPER #ch,24:CLS #ch:BORDER #ch1,2
Pick-Up Truck	ch=4:OPEN #ch,scr_80x40a32,12:PAPER #ch,0:CLS #ch :BORDER #ch1,2
	ch=4:WINDOW #ch,20,30,70,13
	(Reduce Window area for graphics to show Storage Bin Level)
Print/Lorries	ch=5:OPEN #ch,scr_10x10a10,10
	(Open Window #5 reconfigure for Printer & Lorry Bays)
Bay1 Print	ch=5:WINDOW #ch,116,24,380,40
Bay2 Print	ch=5:WINDOW #ch,116,24,380,195
Bay1 Lorry	ch=5:WINDOW #ch,120,40,18,75
Bay2 Lorry	ch=5:WINDOW #ch,120,40,18,135
Store Score	ch=6:OPEN #ch,scr_80x22a32,198:PAPER #ch,0:CLS #ch :BORDER #ch1,2
QL2K Display	ch=7:OPEN #ch,scr_120x806a380,80:PAPER #ch,2:CLS #ch
Stock Request	ch=8:OPEN #ch,scr_216x22a136,198:PAPER #ch,2:CLS #ch:BORDER #ch1,2
Keyboard Input	ch=9:OPEN #ch,con_20x10a10x10 [Consul for k$=INKEY$:k=COD$(k)]
Note:	BLOCK, BORDER, CURSOR, INK, PRINT, OVER, STRIP etc.
	- x, y coordinates specific to Channel / Window.



New third Generation QBITS Warehouse layout

QBITS Warehouse

QBITS Warehouse Stock Movement
The basic graphics for the Warehouse display began with the Storage bays and Truck gangways. Here I continued with the Array of 7 rows with 15 columns. Truck movement was restricted to cells holding a 0. Cells containing Stock would be Set 1 to 24 (a1-8 b1-8 c1-8) or for the storage bays if empty set to 32. All other cells would be set to 255. As for the six cells for a Lorry if empty they would be set as 0 for Truck movement or if the bay was empty set back to 255.

QBIT Warehouse PickUp
The return of the Storeman Sam Truck and Level indicator, this has less detail mainly due to its reduced size to fit in with the new layout. However, it presented the opportunity to be displayed as part of the Introductory page.

QBITS Warehouse QL2K
This was mostly a repositioning. Each of the Store Array Cells rows 1,3,5,7 and columns 6 to 14 and levels 1 to 4 are checked (Stock_aud) and the sum of each type of Stock (a1 to c8) is displayed stk(1 to 24) on the Store Computer. The one problem I had to calculate for was only a single number could be displayed. Therefore an upper limit of 8 is set for each Stock item any additional ones found are automatically deleted (Cell = 32 ie. Empty).

QBITS Warehouse Printers and Lorries
The printers are now presented in line with their respective loading bay. Printouts have to cater for Invoices and Deliveries (Stock Requests) and as such, the top of each show whether they are an Invoice or a Delivery. The six spaces of an empty Lorry are to be filled as shown by the Invoice. For a Delivery, the six spaces displayed with the Stock Request items. As a lorry departs, the displayed printout is now screwed up into a ball then further reduced to finally drop into the wastepaper basket

QBITS Warehouse Sales & Assets
As Invoices (**Sales**) become fulfilled the Store Stock is reduced, Deliveries provide the means to re-Stock (**Assets**). Lorry Departures of Invoices generate **Sales**, the Store Stock, which may have been added with a Delivery is checked and both corresponding **Sales** and **Assets** values are displayed.

Each unit of Stock (**Assets**) is valued as 1credit. If an Invoice (**Sales**) is fulfilled as shown on the Printout each unit is worth 2 credits, if not in correct order then it counts as only one unit. You cannot make a Stock Request if one is already in progress or you have less than 12 Sales credits.

To gain an overall Profit more Sales credits than Assets the game will need to run to a higher number of Invoices. The Game run is between 10 and 40.

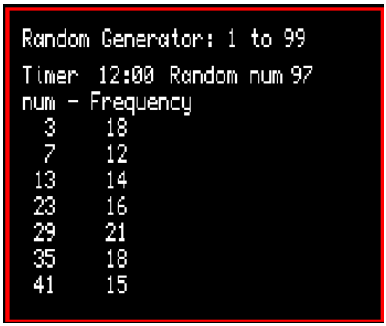
QBITS Warehouse

QBITS Warehouse Random Numbers

A program of this nature uses Random numbers from selecting which Lorry bay to be used and their Departure Times, to generating the stock items on Invoices and when to deploy any Store hazards, Computer clichés, Revenue Tax, Energy Bill payments to lost or Stolen Stock etc.

I therefore found it useful to write a short program to check the frequency of numbers used in such a program. To begin with it has a range within which the Random numbers are to be Generated (RGen). Then Specific numbers to be check for their frequency within a time period (Timer) and an imposed Frame Delay (FDelay) waiting for any keyboard input.

```
100 REMark RanGen
101 WINDOW 492,220,10,0:PAPER 0:CLS
102 FDelay=20:RGen=99:RanGen
103 :
104 DEFine PROCedure RanGen
105 ch=1:WINDOW#ch,200,120,16,12:BORDER#ch,1,2:INK#ch,7
106 CURSOR#ch,0, 6:PRINT#ch,' Random Generator: 1 to ';RGen;
107 CURSOR#ch,0,20:PRINT#ch,' Timer       Random  num';
108 clkold=DATE:a=0:b=0:c=0:d=0:e=0:f=0:g=0
109 PRINT#ch,\' num - Frequency'\'   3'\'   7'\'  13'\'  23'\'  29'\'  35'\'  41'
110 REPeat lp
111    clk=(DATE)-clkold:clock$=DATE$(clk*60)
112    CURSOR#ch, 46,20:PRINT#ch,clock$(13 TO 17)
113    k=CODE(INKEY$(FDelay))
114    IF k=32:EXIT lp
115    num=RND(1 TO RGen):CURSOR#ch,148,20:PRINT#ch,num;'  '
116    SELect ON num
117      = 3:a=a+1:CURSOR#ch,50,40:PRINT#ch,a;'  '
118      = 7:b=b+1:CURSOR#ch,50,50:PRINT#ch,b;'  '
119      =13:c=c+1:CURSOR#ch,50,60:PRINT#ch,c;'  '
120      =21:d=d+1:CURSOR#ch,50,70:PRINT#ch,d;'  '
121      =29:e=e+1:CURSOR#ch,50,80:PRINT#ch,e;'  '
122      =35:f=f+1:CURSOR#ch,50,90:PRINT#ch,f;'  '
123      =41:g=g+1:CURSOR#ch,50,100:PRINT#ch,g;'  '
124    END SELect
125 END REPeat lp
126 WINDOW 492,200,10,0
127 END DEFine
```



Note: The Random Generator in this example ran for 12 minutes. With eight numbers the frequency of occurrence is an average of 14 time for each within the 720 seconds ie. once every 50 seconds. The QBITA Warehouse Game applies these random choices with other restrains such as minimum Sales credits or Stock Assets before deduction are made.

QBITS Warehouse

QBITS Warehouse Program
The use of variables keeps track of Truck movement, Store Levels, Pick and Drop of Stock locations. And last but not least, to keep tab on the changes to Stock (**Assets)** after Deliveries and Invoices (**Sales**) have been processed for the Store Computer and to update the Score.

QBITS Tracking Variables
The Warehouse Game's Global variables can be split into those dealing with the **Menu** options those dealing with **Movement** and those that invoke **Hazards**.

At the start of a **New** Game selected options for Skill (**s**) and number of Invoices (**Inv**) are chosen. Here a multiplier (**m)** is used to create multiples of 5 from 10 to 40 (**Invmax**). As an Invoice is called, the **Inv** is incremented for the next. However, because of multiple unloading bays it was necessary to not only check Invoices in (**Inv**) but also out separately (**Ino**).

When the Menu (**M** key) is called it opens a window to show the current status of **Asset**/**Sales** in chart form and the number of Invoices (**Inv**) completed against the maximum (**Invmax**) number chosen

Asset/Sales chart
After each completed Invoice as the lorry vacates its loading bay the Asset and Sales values are recorded **ASRep[Ino,1(Asset), 2(Sales)]**.

Calling the Menu once a game is in play the Asset & Sales Report is displayed in the Menu's window in chart form.

Lorry Movement
Key to all is the Warehouse Clock against which the **Arrival** and **Departure** of Lorries are checked. The clock start time is created using the QL Date (**ClkOld**=DATE). The clock (**Clk**) updates are by reading the QL clock and deducting the **ClkOld** to give a time span. Divided by a clock multiple (**cm**) in this case 60 to change seconds into minutes on the Warehouse Clock.

The two loading bay's Departure Times are held by (**Dtime1** & **Dtime**2)
Bay1& **Bay2** =0 when empty =1 for Invoice and =2 for Delivery

For deliveries, **Del** identifies if a delivery is pending and which loading bay it will be delivered to. **Din** is activated when a Delivery comes in.

Del=0 1or 2 **Din**=0 or 1

QBITS Warehouse

Truck Movement

The Pickup Truck is moved by the cursor keys and Spacebar along track ways to access Stocks. Warehouse location, levels and loading bays are identified as part of an array by row (**r)**, column (**c**) and level (**l**), **Stock(r, c, l).** The Truck level is identified by (**lev**). The Stock identified by **stk** and **stk$**, and written to the Warehouse Computer Display. Goods carried by the Truck are **box** & **box$**.

QBIT Warehouse Hazards

A Game requires actions that randomly upset the flow and this one presents a number of options. The hazards chosen here are a Store Computer glitch which is just annoying, losses due to missing Stock which reduces the Asset and various others such as Tax Revenues, Energy Bills paid, Stolen Goods each deducting credits from accrued Sales gains.

QBITS Code Testing

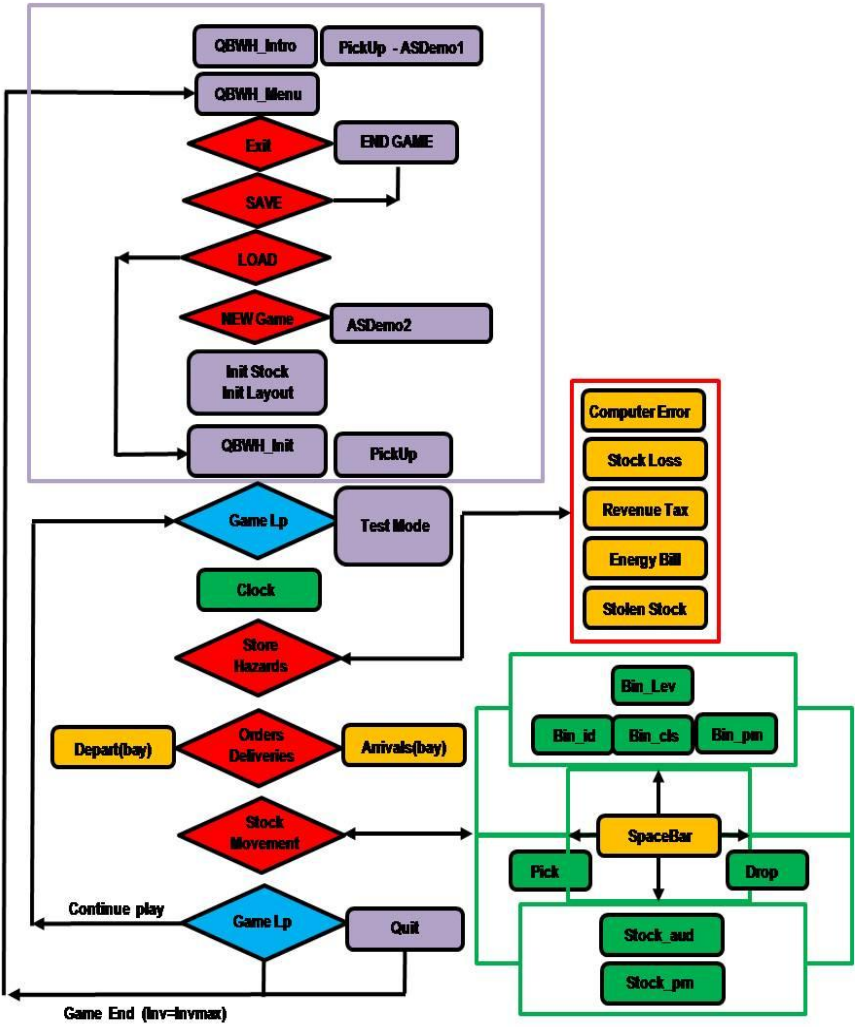
In reviewing my earlier QBITS Warehouse code and trying to understand it, I decided to add a Test Mode. This was to allow checking the various Procedures, Invoices and Delivery printouts and Lorry movements, Store Clock, Stock Request, Game End. Also Store Hazards, Stock (Asset) Loss, Computer error, loss of Sales credits for Tax Revenues, Energy Bill payments and Stolen Lorry Stock. F1 key invokes a sub menu and halts any other action by the main program. You can return by pressing the F1 key again after the selected action(s) are completed.

Special Keys F1 - Invokes the keys below for selection.

=49	:bay1=1:**Arrival(1)**	(1) Bay1_in Print & Lorry
=50	:**Depart(1):**bay1=0	(2) Bay1_out Print & Lorry
=51	:bay2=1:**Arrival(2)**	(3) Bay2_in Print & Lorry
=52	:**Depart(2):**bay2=0	(4) Bay2_out Print & Lorry
=53	:**Store_err**	(5) Compute Crash
=54	:**Stock_loss** :PAUSE	(6) Assets Lost
=55	:**Asset_Tax** :PAUSE	(7) Loss of Sales credits
=56	:**Energy_Bill** :PAUSE	(8) Loss of Sales credits
=57	:**Stolen_Stock** :PAUSE	(9) Loss of Sales credits
=97	:Sales=Sales+6 :**Stock_aud**	(a) Increase Sales
=65	:Sales=Sales-6 :**Stock_aud**	(A) Decrease Sales
=67	:Tim=Tim+3600:**Store_Clk**	(C) Clock 1hr increments
=71	:Inv=Invmax+1	(G) Game End
=83,115:**Stock_Request**		(S s)
=232	:F1=0:EXIT Test	(F1) End Test Mode

QBITS Warehouse

QBWH Flowchart



QBITS Warehouse

Define Procedures

QBWH_Intro	Main instruction on Game
QBWH_Menu	(1)New (2)Load (3)Save (4)Exit options Menu
New_Game	Set up Select New Skill Foreman Storeman Trainee and number of Invoices
Save_Audit	SAVE Game in progress for re- LOAD at later date
Load_Audit	LOAD previously saved Game
Sel_Path	Selects drive & Audit file
QBWH_Game	The main repeat loop for Game play until Inv=Invmax (Completed Total Orders).
Init_clk	Displays a digital counter hour : minutes
QBWH_Init	Graphics – Stock bins, Lorry bays, Clocks, Title, Truck, Score, Stock Request, Stock Display, Printers & Baskets.
Init_stock	Loads Data Array of Warehouse stock
Init_Layout	Displays Stock in Bin locations
PickUP	Graphics for Truck
Bin_lev	Use L level-key to change Bin Level in store
Bin_id	Convert Bin Stock number (1 to 24) to string (a1- c 8).
Bin_cls	Clear Bin
Bin_prn	Print Bin Stock number
Stock_aud	Checks Stock held in Store
Stock_prn	Print Stock numbers in QL2K Stock Display Screen.
Truck_pos(n)	Use Arrow-keys and Spacebar to move and show direction of Pick-Up truck.
Truck_Pick	Pick selected Bin Stock
Truck_Drop	Drop selected Bin Stock
Arrival(bay)	Bay 1 or 2: Generates Random Order or loads Delivery from Stock Request.
Prn_in	Prints Order or Delivery details to bay Printer
Lorry_in	PANS Lorry graphic into appropriate bay - Delivery stock shown loaded .
Depart(bay)	Bay 1 or 2: Graphics for Lorry Departure.+ Updates - Score / Assets.
Prn_out	Throws away paper to Basket.
Lorry_out	PANS Graphics for Lorry Departure
Stock_Request	Stock requested for next Delivery (maintain Stocks in Warehouse).
Stock_loss	Randomly deletes Warehouse Stock
Stock_er	Computer crash – Warehouse QL2K screen shows all stocks at zero.
Asset_Tax	**?** paid. Random amount taken from sales credits
Energy_Bill	**?** paid. Random amount taken from sales credits
Stolen_Stock	**?** lost. Random amount taken from sales credits
ASReport	Generates Asset /Sales Graph of Players progress
AS_Demo1	Asset/Sales graph for QBWH_Intro page
As_Demo2	Asset/Sales graph for QBWH_Menu page

QBITS Warehouse

QBITS Program

```
100 REMark QBWH_Game [QBITS 2017 v1.6]

102 MODE 4:REMark JM/JS ROMS + TK2
103 WINDOW#1,512,256,0,0:PAPER#1,0:CLS#1
104 WINDOW#2,492,220,10,0:PAPER#2,0:CLS#2

106 REMark Store Variables
107 ClkOld=DATE:Tim=3600    :REMark ClkOld=Start Time Tim=1hr Increment
108 Clk=0:cm=60              :REMark Clk=Clock cm=? Alters Game Speed
109 Asset=0:Sales=0          :REMark Asset=Stock Sales=Orders fulfilled
110 Invmax=10:Inv=1:Ino=1    :REMark Oders Invmax=Max Inv=Current Ino=ASRep
111 Skill=6                  :REMark 3/6/9 Sets Departure Interval
112 r=4:c=5:l=1:lev=1        :REMark Warehouse row,column,level & Bin Level
113 stk=32:stk$='  '         :REMark Stocks (1-24;32=Empty)Stk$='a1 to c8'
114 p=192                    :REMark ¼ Truck direction (¼¾ SpaceBar ¿½)
115 box=0:box$='  '          :REMark Truck - Pick/Drop status
116 Dtime1=0:Dtime2=0        :REMark Scheduled Lorry Departure Times Bay 1&2
117 bay1=0:bay2=0            :REMark Bay 0=empty 1=Order 2=Delivery
118 Del=0:Din=0:Gs=0         :REMark Del Din Delivery 0/1/2  Gs 0/1/2/3 Game Satus

120 REMark Save & Load - Devicename (dn) & Audit file (af)
121 DIM SDR$(8,5),Audit$(8,10):RESTORE 124:dn=3:af=1
122 FOR f=1 TO 8:Audit$(f)='Audit'&f&'_dat'
123 FOR d=1 TO 8:READ SDR$(d)
124 DATA 'mdv1_','mdv2_','flp1_','flp2_','win1_','win2_','nfa1_','nfa2_'
125 :
126 REMark Arrays
127 DIM Aud(24),InB(2,6),Stock(7,14,4),SReq(6),ASRep(40,2)

129 REMark Warehouse data
130 DATA 225,225,225,225,255,1,32,3,4,5,6,7,8,255
131 DATA 225,225,225,225,0,0,0,0,0,0,0,0,0,255
132 DATA 225,225,225,225,0,9,10,32,11,12,13,15,16,255
133 DATA 225,225,225,225,0,255,255,255,255,255,255,255,255,255
134 DATA 225,225,225,225,0,17,18,3,4,21,21,23,24,255
135 DATA 225,225,225,225,0,0,0,0,0,0,0,0,0,255
136 DATA 225,225,225,225,255,19,10,11,32,20,22,23,24,255

138 QBWH_Intro
139 QBWH_Menu
```

QBITS Warehouse

```
141 DEFine PROCedure QBWH_Intro
142 ch=1:WINDOW#ch,440,200,36,12:PAPER#ch,0:CLS#ch:INK#ch,7:CSIZE#ch,2,1
143 OVER#ch,1:CURSOR#ch,64,4:PRINT#ch,'Welcome to QBITS Warehouse'
144 CURSOR#ch,65,4:PRINT#ch,'Welcome to QBITS Warehouse':OVER#ch,0
145 :
146 CSIZE#ch,0,0:CURSOR#ch,0,30
147 PRINT#ch,'        As Lorries arrive you have to Load (Invoice) or Unload (Delivery).'
148 PRINT#ch,' Invoices are shown on the relevant Bay Printout. Use the ¼¾¿½ arrows and'
149 PRINT#ch,'  spacebar to move the Pick-UP. Then Pick and Drop using the P & D keys.'
150 PRINT#ch,'        The Warehouse has four levels use the L key to access other stocks.'
151 PRINT#ch,'         Use the S key for Stock Requests (new stock). Lorries leave at the'
152 PRINT#ch,'                Departure time irrespective of their load or unload status.'
153 PRINT#ch,\'       As Invoices are fulfil your Sales increase, but keep an eye on the'
154 PRINT#ch,'           Assets this shows the Stocks currently held in the Warehouse.'
155 PRINT#ch,\'          To access the Menu (New Load Save Exit) press the M key.'
156 PRINT#ch, \\\'                                press any key to continue...'
157 ch=9:OPEN#ch,scr_80x40a72x162:Init_PickUp
158 WINDOW#ch,100,60,360,150:AS_Demo1:CLOSE#ch:PAUSE
159 END DEFine

161 DEFine PROCedure QBWH_Menu
162 ch=1:PAPER#ch,0:WINDOW#ch,220,144,138,53
163 FOR i=1 TO 20
164  INK#ch,0:FILL#ch,1:CIRCLE#ch,60,50,i*3:FILL#ch,0:PAUSE 1
165 END FOR i
166 BORDER#ch,1,2:CLS#ch
167 IF demo=2:AS_Demo2:demo=0:ELSE ASReport
168 IF Gs=1:ch=8:CLS#ch:INK#ch,4:CURSOR#ch,50,4:PRINT#ch,'Spacebar to return'
169 IF Gs=4:ch=8:CLS#ch:INK#ch,4:CURSOR#ch,84,4:PRINT#ch,'GAME END'
170 REPeat lp
171 ch=1:INK#ch,7:CURSOR#ch,18,120:PRINT#ch,'(1)New (2)Load (3)Save (4)Exit'
172 k=CODE(INKEY$(-1))
173 SELect ON k
174   =32:IF Gs=1:CLS#8:Init_Layout:QBWH_Game          :REMark continue
175   =49:New_Game  :QBWH_Game
176   =50:Load_Audit:IF Gs=2:EXIT lp
177   =51:Save_Audit:IF Gs=3:EXIT lp
178   =52:EXIT lp                                      :REMark Game End
179 END SELect
180 END REPeat lp
181 IF Gs=2:QBWH_Game
182 ch=1:CLS#ch:OVER#ch,1:CSIZE#ch,2,1:INK#ch,7
183 FOR i=0 TO 1:CURSOR#ch,56+i,60:PRINT#ch,'Game End'
184 PAUSE 100:FOR ch=3 TO 9:CLOSE#ch
185 WINDOW#1,512,230,0,0:PAPER#1,0:CLS#1:STOP
186 END DEFine
```

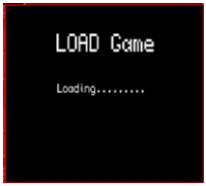
QBITS Warehouse

```
188 DEFine PROCedure New_Game
189 ch=1:Sk$=' ':s=2:m=3 :REMark SK$/s/m Skill & Invoice Multiplyer
190 CLS#ch:CSIZE#ch,2,1:INK#ch,7:CURSOR#ch,60,20:PRINT#ch,'New Game'
191 CSIZE#ch,3,0:CURSOR#ch,14,90:PRINT #ch,'¼         ½'\'    ¾  ¿'
192 CSIZE #ch,0,0:INK#ch,2
193 CURSOR#ch,20,80 :PRINT#ch,'Vary Game Time - Select Skill'
194 PRINT#ch,\\'          Invoice Number'\'       then press Spacebar...'
195 REPeat Choice
196   INK#ch,4:CURSOR#ch,30,90:PRINT#ch,' Foreman Storeman Trainee '
197   IF s=1:Sk$=' Foreman'
198   IF s=2:Sk$='Storeman'
199   IF s=3:Sk$='Trainee '
200   INK#ch,7:CURSOR#ch,s*54-24,90:PRINT#ch,Sk$
201   INK#ch,7:CURSOR#ch,104,100   :PRINT#ch,5+m*5
202   k=CODE(INKEY$(#ch,20))
203   SELect ON k
204     =192:s=s-1:IF s<1:s=3
205     =200:s=s+1:IF s>3:s=1
206     =208:m=m+1:IF m>7:m=7
207     =216:m=m-1:IF m<1:m=1
208     = 32:Set=0:Skill=(s*3):Invmax=5+(m*5):EXIT Choice
209     = 27:STOP
210   END SELect
211 END REPeat Choice
212 Inv=1:Ino=1:Sales=0:FOR i=1 TO 40:ASRep(i,1)=0:ASRep(i,2)=0
213 Init_Stock:QBWH_Init:Init_Layout:Gs=1
214 END DEFine
```


```
269 DEFine PROCedure Sel_Path
270 ch=1:CSIZE#ch,0,0:INK#ch,2
271 CURSOR#ch,6,100:PRINT#ch,'Exit<Spacebar> ¼ ½      ¾¿ <Enter>'
272 INK#ch,7
273 REPeat Path_lp
274   CURSOR#ch,48,90:PRINT#ch,'Select:';SDR$(dn)&Audit$(af)
275   k=CODE(INKEY$(-1))
276   SELect ON k
277     =192:dn=dn-1:IF dn<1:dn=1
278     =200:dn=dn+1:IF dn>8:dn=8
279     =208:af=af+1:IF af>8:af=8
280     =216:af=af-1:IF af<1:af=1
281     = 10:file=1:EXIT Path_lp
282     = 32:file=0:EXIT Path_lp
283   END SELect
284 END REPeat Path_lp
285 END DEFine
```

QBITS Warehouse

```
216 DEFine PROCedure Load_Audit
217 ch=1:CLS#ch:CSIZE#ch,2,1,:INK#ch,7
218 CURSOR#ch,54,20:PRINT#ch,'LOAD Game':Sel_Path
219 IF file=0:CLS#ch:RETurn
220 CURSOR#ch,56,60:PRINT#ch,'Searching...':CLS#ch,2
221 DELETE SDR$(dn)&'FList'
222 ch=9:OPEN_NEW#ch,SDR$(dn)&'FList':DIR#ch,SDR$(dn):CLOSE#ch
223 OPEN_IN#ch,SDR$(dn)&'FList'
224 REPeat Dir_lp
225   IF EOF(#ch)
226     CLOSE#ch:CURSOR#1,54,60:PRINT#1,'File Not Found...'
227     PAUSE 50:CLS#1:file=0:RETurn
228   END IF
229   INPUT#ch,Fchk$:IF Fchk$==Audit$(af):CLOSE#ch:EXIT Dir_lp
230 END REPeat Dir_lp
231 ch=1:CURSOR#ch,56,60:PRINT#ch,'Loading...'
232 ch=9:OPEN_IN#ch,SDR$(dn)&Audit$(af)
233 FOR l=1 TO 4
234  FOR r=1 TO 7
235   FOR c=1 TO 14
236    INPUT#ch,Stock(r,c,l)
237   END FOR c
238   CURSOR#1,104+r*6,60:PRINT#1,'.';
239  END FOR r
240 END FOR l
241 FOR n=1 TO 40:INPUT#ch,ASRep(n,1)\ASRep(n,2)
242 INPUT#ch,Inv\Ino\Invmax\Sales\Sk$:PAUSE 50:CLOSE#ch
243 QBWH_Init:Init_Layout:Gs=2
244 END DEFine
```




```
246 DEFine PROCedure Save_Audit
247 IF Ino<2:RETurn
248 ch=1:CLS#ch:CSIZE#ch,2,1,:INK#ch,7
249 CURSOR#ch,54,20:PRINT#ch,'SAVE Game':Sel_Path
250 IF file=0:CLS#ch:RETurn
251 FOR r=2 TO 3:FOR c=2 TO 4:Stock(r,c,1)=255:END FOR c:END FOR r
252 FOR r=5 TO 6:FOR c=2 TO 4:Stock(r,c,1)=255:END FOR c:END FOR r
253 CLS#8:ch=1:CURSOR#ch,56,60:PRINT#ch,'Saving...':CLS#ch,2
254 ch=9:DELETE SDR$(dn)&Audit$(af)
255 OPEN_NEW#ch,SDR$(dn)&Audit$(af)
256 FOR l=1 TO 4
257  FOR r=1 TO 7
258   FOR c=1 TO 14
259    PRINT#ch,Stock(r,c,l)
260   END FOR c
261   CURSOR#1,104+r*6,60:PRINT#1,'.';
262  END FOR r
263 END FOR l
264 FOR n=1 TO 40:PRINT#ch,ASRep(n,1)\SRep(n,2)
265 Inv=Ino:PRINT#ch,Inv\Ino\Invmax\Sales\Sk$
266 CLOSE#ch:Gs=3
267 END DEFine
```

QBITS Warehouse

```
287 DEFine PROCedure QBWH_Game
288 Bin_lev:Stock_aud:Score:Gs=1
289 OPEN #9,con_20x10a10x10:PAPER#9,0:CLS#9
290 REPeat Game_lp
291  Store_Clk:Truck_Pos(p)
292  IF Inv>7 AND Sales>24 AND Del=0
293    num=RND(1 TO 99)
294    IF num= 3:Store_err
295    IF num= 7:Stock_loss
296    IF num=13:Asset_Tax
297    IF num=21:Energy_Bill
298    IF num=29:Stolen_Stock
299  END IF
300  IF Inv>Invmax:Gs=4:QBWH_Menu
301  IF bay1=0 AND Inv<=Invmax AND RND(1 TO 24)=7:bay1=1:Arrival(1)
302  IF bay1>0 AND Clk>=Dtime1:Depart(1):bay1=0
303  IF bay2=0 AND Inv<=Invmax AND RND(1 TO 24)=7:bay2=1:Arrival(2)
304  IF bay2>0 AND Clk>=Dtime2:Depart(2):bay2=0
305  k=CODE(INKEY$(#9,20))
306  SELect ON k
307   = 32:p=p+8:Bin_cls:IF p>216:p=192
308   =192:p=192:Bin_cls:IF Stock(r,c-1,1)=0:c=c-1
309   =200:p=200:Bin_cls:IF Stock(r,c+1,1)=0:c=c+1
310   =208:p=208:Bin_cls:IF Stock(r-1,c,1)=0:r=r-1
311   =216:p=216:Bin_cls:IF Stock(r+1,c,1)=0:r=r+1
312   =68,100:IF box<>0:Truck_Drop        :REMark (D d)
313   =80,112:IF box=0 :Truck_Pick        :REMark (P p)
314   =76,108:lev=lev+1:Bin_lev           :REMark (L l)
315   =83,115:Stock_Request               :REMark (S s)
316   =77,109:Gs=1:QBWH_Menu              :REMark (M m)
317   =232   :Test_Mode                   :REMark  F1
318   =27    :STOP
319  END SELect
320 END REPeat Game_lp
321 END DEFine
323 DEFine PROCedure Init_Stock
324 RESTORE 130
325 FOR r=1 TO 7
326   FOR c=1 TO 14
327     READ dat:Stock(r,c,1)=dat
328     IF c>=6 AND c<=13
329       n=RND(1 TO 32):IF n>24:n=32
330       Stock(r,c,3)=n
331       Stock(r,c,2)=32
332       Stock(r,c,4)=32
333     END IF
334   END FOR c
335 END FOR r
336 END DEFine
```

QBITS Warehouse

```
338 DEFine PROCedure QBWH_Init
339 ch=1:WINDOW#ch,492,220,10,0:CLS#ch
340 F1=0:stk=0:r=4:c=5:l=1:p=192:bay1=0:bay2=0
341 :
342 REMark Store Layout
343 ch=3:OPEN#ch,scr_356x194a16x30:PAPER#ch,32:CLS#ch:BORDER#ch,1,2
344 BLOCK#ch,120,50,0,38,0:BLOCK#ch,120,50,0,98,0:REMark Drive Bay 1&2
345 :
346 REMark QBITS Title
347 ch=3:WINDOW#ch,224,26,136,12:PAPER#ch,24:CLS#ch:BORDER#ch,1,2
348 OVER#ch,1:CSIZE#ch,2,1
349 INK#ch,0:FOR i=0 TO 3:CURSOR#ch,20+i,2  :PRINT#ch,'QBITS Warehouse'
350         FOR i=0 TO 2:CURSOR#ch,20+i,2+i:PRINT#ch,'QBITS Warehouse'
351 INK#ch,7:FOR i=0 TO 1:CURSOR#ch,18+i,3  :PRINT#ch,'QBITS Warehouse'
352 OVER#ch,0
353 :
354 REMark Headings
355 ch=3:WINDOW#ch,352,192,18,31
356 CSIZE#ch,2,0:INK#ch,0:STRIP#ch,7
357 CURSOR#ch,118,9 :PRINT#ch,'   Clock '
358 CSIZE#ch,0,0:INK#ch,0:STRIP#ch,4
359 CURSOR#ch,282,9 :PRINT#ch,' ';Sk$;' '
360 CURSOR#ch,8,26 :PRINT#ch,'Departure:      '
361 CURSOR#ch,8,152:PRINT#ch,'Departure:      '
362 :
363 REMark PickUp
364 ch=4:OPEN#ch,scr_80x40a32x12:PAPER#ch,0:CLS#ch:BORDER#ch,1,2
365 Init_PickUp:INK#ch,7:CURSOR#ch,0,0:PRINT#ch,'Level'
366 WINDOW#ch,20,38,70,13        :REMark bin_lev window ch=4
367 :
368 REMark Printers Bay 1 & 2
369 ch=5:OPEN#ch,scr_10x10a0x0
370 FOR b=1 TO 2
371  IF b=1:y=40:ELSE y=195
372  WINDOW#ch,82,24,380,y       :REMark printer
373  BLOCK#ch,78,16,2,6,7
374  BLOCK#ch,74,16,4,8,240
375  BLOCK#ch,82,10,0,12,7
376  BLOCK#ch,82,2,0,16,0
377  BLOCK#ch,70,10,6,0,7
378  STRIP#ch,70:INK #ch,0:CURSOR #ch,10,12:PRINT #ch,'Bay'&b
379  WINDOW#ch,20,16,476,y+8     :REMark waste basket
380  BLOCK#ch,20,2,0,0,7
381  BLOCK#ch,14,2,3,14,7
382  BLOCK#ch,18,12,1,2,240
383 END FOR b
384 :
385 REMark Score
386 ch=6:OPEN#ch,scr_80x22a32x199:PAPER#ch,0:CLS#ch:BORDER#ch,1,2,
387 INK#ch,7:CURSOR#ch,32, 0:PRINT#ch,'Sales '
388 INK#ch,2:CURSOR#ch,32,10:PRINT#ch,'Assets'
389 :
```

QBITS Warehouse

```
390 REMark QL2K PC
391 ch=7:OPEN#ch,scr_120x80a380x80:PAPER#ch,0:CLS#ch
392 BLOCK#ch,116,62,2,0,7:BLOCK#ch,114,60,3,1,0
393 BLOCK#ch,110,60,5,1,7:BLOCK#ch,108,58,6,2,0
394 BLOCK#ch,120,14,0,64,7:BLOCK#ch,112,1,2,70,0:BLOCK#ch,116,2,2,78,240
395 FOR i=1 TO 2:BLOCK#ch,14,2,86+i,71,0
396 CSIZE#ch,0,0:INK#ch,0:STRIP#ch,7:CURSOR#ch,12,66:PRINT#ch,'QL2k'
397 WINDOW#ch,100,56,388,84:PAPER#ch,0:CLS#ch
398 INK#ch,7:FOR i=0 TO 1  :CURSOR#ch,20+i,0:PRINT#ch,'STOCKS'
399 INK#ch,4:CURSOR#ch,0,12:PRINT#ch,'A'\'B'\'C'
400 INK#ch,4:FOR i=1 TO 8  :CURSOR#ch,10+(i*10),44:PRINT#ch,i
401 :
402 REMark Requests & Messages
403 ch=8:OPEN#ch,scr_224x22a136x199:PAPER#ch,0:CLS#ch:BORDER#ch,1,2
404 END DEFine

406 DEFine PROCedure Init_Layout
407 ch=3:WINDOW#ch,352,192,18,31
408 BLOCK#ch,224,148,118,20,16                    :REMark Warehouse
409 BLOCK#ch,196,1,142,23,4                      :REMark Warehouse Walls
410 BLOCK#ch,2,143,338,23,4
411 BLOCK#ch,196,1,142,165,4
412 BLOCK#ch,2,20,142,24,4
413 BLOCK#ch,2,60,144,64,4
414 BLOCK#ch,2,20,142,145,4
```

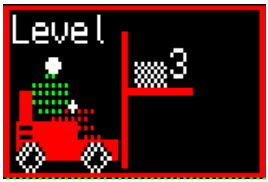

```
415 BLOCK#ch,196,18,144,85,4:BLOCK#ch,192,14,146,87,0
416 OVER#ch,1:STRIP#ch,0:BLOCK#ch,12,4,168,92,7
417 INK#ch,7:CURSOR#ch,152,89:PRINT#ch,'¼¿  ¾½'
418         CURSOR#ch,200,89:PRINT#ch,'P    D    L    S'
419 INK#ch,4:CURSOR#ch,200,89:PRINT#ch,' ick  rop  evel  tocks'
420 OVER#ch,0
421 FOR d=1 TO 7
422   FOR a=1 TO 14
423     IF Stock(d,a,1)<200:BLOCK #ch,22,18,2+a*24,5+d*20,2
424   END FOR a
425 END FOR d
426 END DEFine
```


```
446 DEFine PROCedure Store_Clk
447 Clk=Tim+25200+((DATE-ClkOld)*cm):clock$=DATE$(Clk)
448 ch=3:BLOCK#ch,30,10,120,9,7:STRIP#ch,7:INK#ch,0
449 CURSOR#ch,120,9:PRINT#ch,clock$(13 TO 17)
450 END DEFine
```

QBITS Warehouse

```
428 DEFine PROCedure Init_PickUp
429 INK#ch,16 :FILL#ch,1:CIRCLE#ch,42,32,5,3,-.8:FILL#ch,0
430 INK#ch,16 :FILL#ch,1:CIRCLE#ch,35,36,5,3,0:FILL#ch,0
431 INK#ch,32 :FILL#ch,1:CIRCLE#ch,24,48,12:FILL#ch,0
432 INK#ch,7  :FILL#ch,1:CIRCLE#ch,26,65,4:FILL#ch,0
433 FILL#ch,1:INK#ch,2
434 LINE#ch,4,8 TO 64,8 TO 64,20 TO 40,20 TO 40,30 TO 12,30 TO 12,50 TO 5,50 TO 5,30 TO
4,30 TO 4,8
435 FILL#ch,0:INK#ch,0
436 INK#ch,0:LINE#ch,5,21 TO 14,21:LINE#ch,5,26 TO 14,26
437 INK#ch,2:LINE#ch,68,4 TO 68,90:LINE#ch,70,4 TO 70,90
438 INK#ch,2:LINE#ch,72,7 TO 112,7:LINE#ch,72,9 TO 112,9
439 INK#ch,240:FILL#ch,1:CIRCLE#ch,12,7,8:FILL#ch,0
440 INK#ch,0  :FILL#ch,1:CIRCLE#ch,12,7,3:FILL#ch,0
441 INK#ch,240:FILL#ch,1:CIRCLE#ch,51,7,8:FILL#ch,0
442 INK#ch,0  :FILL#ch,1:CIRCLE#ch,51,7,3:FILL#ch,0
443 INK#ch,7  :FILL#ch,1:CIRCLE#ch,38,42,2.5:FILL#ch,0
444 END DEFine
```


```
452 DEFine PROCedure Bin_lev
453 LOCal r,c:IF lev>4:lev=1
454 ch=4:PAPER#ch,0:CLS#ch:INK#ch,7
455 BLOCK#ch,20,2,0,42-(8*lev),2
456 IF box>=1 AND box<=24:BLOCK#ch,12,6,2,36-(8*lev),240
457 CURSOR#ch,14,32-(8*lev):PRINT#ch,lev
458 FOR r=1 TO 7 STEP 2
459 FOR c=6 TO 13
460   stk=Stock(r,c,lev):Bin_id:Bin_cls:Bin_prn
461  END FOR c
462 END FOR r
463 END DEFine

465 DEFine PROCedure Bin_id
466 stk$='  '
467 SELect ON stk
468  = 1 TO  8:stk$='a'&stk
469  = 9 TO 16:stk$='b'&(stk-8)
470  =17 TO 24:stk$='c'&(stk-16)
471 END SELect
472 END DEFine

474 DEFine PROCedure Bin_cls
475 ch=3:BLOCK#ch,22,18,2+c*24,5+r*20,2
476 IF c<=5 AND lev>1:lev=1:Bin_lev
477 END DEFine

479 DEFine PROCedure Bin_prn
480 ch=3:INK#ch,7:PAPER#ch,2
481 CURSOR#ch,7+c*24,9+r*20:PRINT#ch,stk$:stk$='  '
482 END DEFine
```

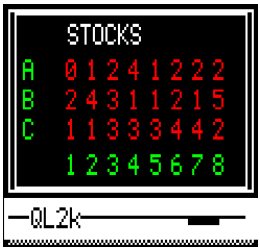


QBITS Warehouse

```
484 DEFine PROCedure Stock_aud
485 LOCal r,c,l:Asset=0
486 FOR i=1 TO 24:Aud(i)=0
487 FOR l=1 TO 4
488  FOR r=1 TO 7 STEP 2
489   FOR c=6 TO 13
490    IF Stock(r,c,l)>=1 AND Stock(r,c,l)<=24
491      Aud(Stock(r,c,l))=Aud(Stock(r,c,l))+1
492      IF Aud(Stock(r,c,l))>8:Stock(r,c,l)=32:ELSE Asset=Asset+1
493    END IF
494   END FOR c
495  END FOR r
496 END FOR l
497 FOR stk=1 TO 24:Stock_prn(stk)
498 END DEFine
```

Note: Update checks of Store Stock (Asset)

```
506 DEFine PROCedure Stock_prn(stk)
507 ch=7:INK#ch,2
508 SELect ON stk
509  = 1 TO  8:CURSOR #ch,(stk*10)+ 10,12:PRINT #ch,Aud(stk)
510  = 9 TO 16:CURSOR #ch,(stk*10)- 70,22:PRINT #ch,Aud(stk)
511  =17 TO 24:CURSOR #ch,(stk*10)-150,32:PRINT #ch,Aud(stk)
512 END SELect
513 END DEFine
```

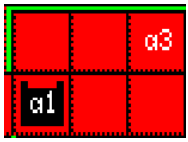


Note: Prints the results of Stock_aud to store Computer

```
500 DEFine PROCedure Score
501 ch=6:BLOCK#ch,24,19,4,0,0
502 INK#ch,7:CURSOR#ch,4,0:PRINT#ch,Sales
503 INK#ch,2:CURSOR#ch,4,10:PRINT#ch,Asset
504 END DEFine
```

Note: Store Sales and Asset recorded for each Invoice and provide Info for Asset\Sales Report.

QBITS Warehouse

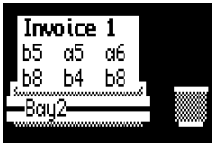



```
515 DEFine PROCedure Truck_Pos(p)
516 ch=3:BLOCK#ch,18,14,4+c*24,7+r*20,0
517 SELect ON p
518  =192:BLOCK#ch,4,10,4+c*24,9+r*20,2
519  =200:BLOCK#ch,4,10,18+c*24,9+r*20,2
520  =208:BLOCK#ch,12,4,7+c*24,6+r*20,2
521  =216:BLOCK#ch,12,4,7+c*24,18+r*20,2
522 END SELect
523 INK #ch,7:STRIP#ch,0:CURSOR #ch,7+c*24,9+r*20:PRINT #ch,box$
524 END DEFine

526 DEFine PROCedure Truck_Pick
527 rt=r:ct=c:l=lev                              :REMark rt,ct,lt temp hold of r,c,lev
528 IF p=208:r=r-1
529 IF p=216:r=r+1
530 IF c>5 AND c<14
531   IF Stock(r,c,l)>=1 AND Stock(r,c,l)<=24
532     Aud(Stock(r,c,l))=Aud(Stock(r,c,l))-1
533     stk=Stock(r,c,l):Stock(r,c,l)=32:Stock_prn(stk):pick=1
534     Bin_cls:Bin_prn                          :REMark Bin Clear & Print
535     Bin_id:box=stk:box$=stk$                 :REMark Pick up box Bin_id
536   END IF
537 END IF
538 IF p=192:c=c-1
539 IF p=200:c=c+1
540 IF c<5 AND Stock(r,c,1)<25
541   stk=Stock(r,c,1):Stock(r,c,1)=0
542   Bin_cls:Bin_prn                            :REMark Bin Clear & Print
543   Bin_id:box=stk:box$=stk$                   :REMark Pick up box Bin_id
544 END IF
545 r=rt:c=ct:Bin_prn
546 IF box<>0:ch=4:BLOCK#ch,12,6,2,36-(8*lev),240
547 END DEFine

549 DEFine PROCedure Truck_Drop
550 rt=r:ct=c:l=lev
551 IF p=208:r=r-1                                :REMark facing bins Up
552 IF p=216:r=r+1                               :REMark facing bins Down
553 IF c>5 AND c<14
554   IF Stock(r,c,l)=32
555     Stock(r,c,l)=box:REMark Stock_prn(box)
556     Aud(Stock(r,c,l))=Aud(Stock(r,c,l))+1:Stock_prn(box)
557     stk$=box$:Bin_prn:box=0:box$='  ':pick=0
558   END IF
559 END IF
560 IF p=192:c=c-1                               :REMark loading lorry Left
561 IF p=200:c=c+1                              :REMark loading lorry Right
562 IF c<5 AND Stock(r,c,1)=0
563   Stock(r,c,1)=box:box=0
564   stk$=box$:Bin_prn:box$='  '
565 END IF
566 r=rt:c=ct
567 IF box=0:ch=4:BLOCK#ch,12,6,2,36-(8*lev),0
568 END DEFine
```

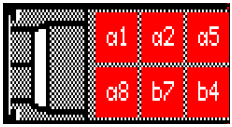
QBITS Warehouse

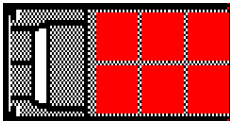
```
570 DEFine PROCedure Arrival(bay)
571 IF bay=1:py=20:ly=75:rl=2:ELSE py=175:ly=135:rl=5
572 ar=r:ac=c:Prn_in:Lorry_in
573 n=0:ch=5:WINDOW#ch,352,192,18,31
574 FOR r=rl TO rl+1
575  FOR c=2 TO 4
576   Stock(r,c,1)=0:Bin_cls
577   IF Del=bay
578     n=n+1:stk=SReq(n):Stock(r,c,l)=stk:Bin_id:Bin_prn
579   END IF
580  END FOR c
581 END FOR r
582 r=ar:c=ac:IF Del=bay:Din=1
583 IF bay=1 AND bay2=1:dlay=300*Skill:ELSE dlay=0
584 IF bay=1:ty=26 :Dtime1=dlay+Clk+(1200*Skill):Dept$=DATE$(Dtime1)
585 IF bay=2 AND bay1=1:dlay=300*Skill:ELSE dlay=0
586 IF bay=2:ty=152:Dtime2=dlay+Clk+(1200*Skill):Dept$=DATE$(Dtime2)
587 STRIP#ch,4:INK#ch,0:CURSOR#ch,76,ty:PRINT#ch,Dept$(13 TO 17)
588 END DEFine
```


```
590 DEFine PROCedure Prn_in
591 ch=5:WINDOW#ch,70,30,386,py:PAPER#ch,0:CLS#ch
592 n=0:PAPER#ch,7:INK#ch,0:OVER#ch,1:SCROLL#ch,-10
593 IF Del=bay
594  String$='Delivery':CLS#8                    :REMark clear Stock Request Window
595 ELSE
596  String$='Invoice  '&Inv:Inv=Inv+1
597 END IF
598 FOR i=0 TO 1:CURSOR#ch,2+i,20:PRINT#ch,String$
599 FOR d=1 TO 2
600  PAUSE 2:SCROLL #ch,-10
601  FOR a=1 TO 3
602   n=n+1:IF Del=bay:stk=SReq(n):ELSE stk=RND(1 TO 24):InB(bay,n)=stk
603   Bin_id:CURSOR#ch,-22+a*24,20:PRINT#ch,stk$:stk$='  '
604  END FOR a
605 END FOR d
606 END DEFine
```

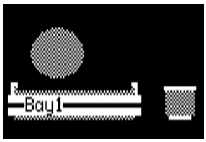


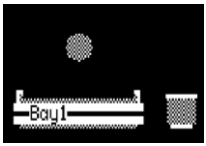

```
608 DEFine PROCedure Lorry_in
609 ch=5:WINDOW#ch,120,40,18,ly:PAPER#ch,240
610 FOR i=1 TO 19:PAN#ch,4:PAUSE 2
611 BLOCK#ch,2,40,0,0,0
612 FOR i=1 TO 5
613  PAN#ch,4:BLOCK#ch,4,40,0,0,240
614  BLOCK#ch,4,2,0,4,0:BLOCK#ch,4,2,0,36,0
615 END FOR i
616 BLOCK#ch,2,32,0,4,0
617 FOR i=1 TO 2
618  PAN#ch,4:BLOCK#ch,4,2,0,4+i,0
619  BLOCK#ch,4,29-i,0,6+i,7:BLOCK#ch,4,2,0,36-i,0
620 END FOR i
621 BLOCK#ch,2,28,0,6,0
622 FOR i=1 TO 3
623  PAN#ch,4:BLOCK #ch,4,2,0,6,0
```





```
624  BLOCK#ch,4,2,0,19,0:BLOCK#ch,4,2,0,32,0
625 END FOR i
626 BLOCK#ch,6,4,0,0,7 :BLOCK#ch,4,4,0,0,0
627 BLOCK#ch,6,4,0,36,7:BLOCK#ch,4,4,0,36,0
628 BLOCK#ch,1,36,0,2,7
629 PAPER#ch,0:PAN#ch,4
630 END DEFine


632 DEFine PROCedure Depart(bay)
633 IF bay=1:lr=2:py=20:ly=75:ty=26:ELSE lr=5:py=175:ly=135:ty=152
634 dr=r:dc=c:n=0
635 FOR r=lr TO lr+1
636  FOR c=2 TO 4
637   IF Del=bay
638     IF Stock(r,c,1)=0:Sales=Sales-1:END IF
639   ELSE
640     n=n+1:IF Stock(r,c,1)=InB(bay,n):Sales=Sales+2:END IF
641     IF Stock(r,c,1)>=1 AND Stock(r,c,1)<=24:Sales=Sales+1:END IF
642   END IF
643   Stock(r,c,1)=255
644  END FOR c
645 END FOR r
646 IF Del=bay AND Din=1
647    Del=0:Din=0
648 ELSE
649    Stock_aud:Score:ASRep(Ino,1)=Asset:ASRep(Ino,2)=Sales:Ino=Ino+1
650 END IF
651 ch=5:WINDOW#ch,352,192,18,31:BLOCK#ch,30,10,76,ty,4:REMark Clr DTime
652 r=dr:c=dc:Prn_out:Lorry_out:Stock_aud:Score
653 END DEFine
```






```
655 DEFine PROCedure Prn_out
656 ch=5:WINDOW#ch,70,30,386,py:PAPER#ch,0:CLS#ch:INK#ch,240
657 FILL#ch,1:CIRCLE#ch,60,60,40:PAUSE 10:CLS#ch
658 FILL#ch,1:CIRCLE#ch,90,80,20:PAUSE 10:CLS#ch
659 FILL#ch,1:BLOCK#ch,70,10,0,20,7
660 ch=5:WINDOW#ch,20,30,476,py-2:PAPER#ch,0:CLS#ch:INK#ch,240
661 FILL#ch,1:CIRCLE#ch,20,80,10
662 FOR i=1 TO 15:SCROLL#ch,2:PAUSE 1
663 END DEFine


665 DEFine PROCedure Lorry_out
666 ch=5:WINDOW#ch,120,40,18,ly:BLOCK#ch,72,40,48,0,240
667 PAPER#ch,0:FOR i=1 TO 30:PAN#ch,-4:PAUSE 1
668 IF c<5
669   IF bay=1 AND r<4:c=5:r=4:Bin_cls:Truck_Pos(192)
670   IF bay=2 AND r>4:c=5:r=4:Bin_cls:Truck_Pos(192)
671 END IF
672 END DEFine
```

QBITS Warehouse

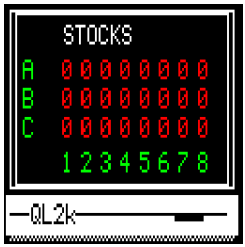
```
674 DEFine PROCedure Stock_Request
675 IF Del>0 OR Sales<12:RETurn
676 ch=8:CLS #ch:x=1:y=0:n=1:s=0
677 INK #ch,7:CURSOR #ch,4,0  :PRINT #ch,'Stock Request'
678 INK #ch,4:CURSOR #ch,36,10:PRINT #ch,'Select 6 Items ¼½  ¿¾'
679 BLOCK #ch,20,4,140,14,4                :REMark SpaceBar
680 Stock_aud:FOR stk=1 TO 24:Stock_prn(stk)
681 REPeat Stock_lp
682  Store_Clk
683  IF n>6:Del=RND(1 TO 2):RETurn
684  ch=7:STRIP #ch,4:INK #ch,0:Stock_prn(x+8*y)
685  k$=INKEY$(#9,20):k=CODE(k$)
686  ch=7:STRIP #ch,0:INK #ch,2:Stock_prn(x+8*y)
687  SELect ON k
688    = 27:STOP
689    =192:IF x>1:x=x-1:ELSE x=1
690    =200:IF x<8:x=x+1:ELSE x=8
691    =208:IF y>0:y=y-1:ELSE y=0
692    =216:IF y<2:y=y+1:ELSE y=2
693    = 32:IF n<=6
694          stk=x+y*8:SReq(n)=stk:Bin_id
695          ch=8:INK #ch,7:CURSOR #ch,80+(n*18),1:PRINT #ch,stk$
696          n=n+1:str$=' '
697        END IF
698  END SELect
699 END REPeat Stock_lp
700 END DEFine
```

Note:, As Invoices are fulfilled the Store Stock reduces, to ensure there is always adequate goods make Stock Requests to replenish dwindling Stock. Once sales have reached 12 credits, a Stock Request can be made. A stock item is highlighted among the Stock displayed on the Store Computer. Use the cursor keys to move around highlighting different items and select with Spacebar. Only six items at a time can be Requested.

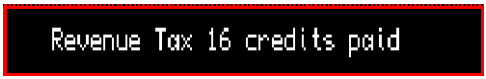
Any Deliveries must be completed and Sales credits reach 12 or above before the next Stock Request can be made.

QBITS Warehouse


```
702 DEFine PROCedure Stock_loss
703 IF Sales<36:RETurn
704  ch=8:CLS#ch:INK#ch,7:OVER#ch,1
705 CURSOR#ch,48,5:PRINT#ch,'Attention Stock LOSS!'
706 att=RND(1 TO 7)
707 IF att=1:FOR i=8 TO RND(9 TO 11):Stock(1,i,1)=32:Stock(7,i,1)=32
708 IF att=3:FOR i=1 TO RND(2 TO  3):Stock(3,i,3)=32:Stock(5,i,3)=32
709 IF att=5:FOR i=7 TO RND(8 TO 10):Stock(1,i,1)=32:Stock(7,i,1)=32
710 IF att=7:FOR i=4 TO RND(5 TO  7):Stock(3,i,3)=32:Stock(5,i,3)=32
711 Stock_aud:Score:FOR stk=1 TO 24:Stock_prn(stk)
712 END DEFine
```




```
714 DEFine PROCedure Store_err
715 ch=8:CLS#ch:INK#ch,7
716 CURSOR#ch,36,5:PRINT#ch,'Store Computer Error!';
717 FOR stk=1 TO 24:Aud(stk)=0:Stock_prn(stk)
718 FOR t=1 TO 5:PAUSE 30:Store_Clk:ch=8:PRINT#ch,'!';
719 CLS#ch:Stock_aud:FOR stk=1 TO 24:Stock_prn(stk)
720 END DEFine
```




```
722 DEFine PROCedure Asset_Tax
723 IF Sales<42:RETurn
724 ch=8:CLS#ch:INK#ch,7:Tr=INT(Sales/10):Sales=Sales-Tr
725 CURSOR#ch,20,5:PRINT#ch,'Revenue Tax ';Tr;' credits paid':Score
726 END DEFine
```

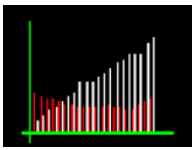

```
728 DEFine PROCedure Energy_Bill
729 IF Sales<36:RETurn
730 ch=8:CLS#ch:INK#ch,7:Ec=INT(Asset/20):Sales=Sales-Ec
731 CURSOR#ch,20,5:PRINT#ch,'Energy Bill ';Ec;' credits paid':Score
732 END DEFine
```


```
734 DEFine PROCedure Stolen_Stock
735 IF Sales<24:RETurn
736 ch=8:CLS#ch:INK#ch,7:Ls=RND(2 TO 6):Sales=Sales-Ls
737 CURSOR#ch,20,5:PRINT#ch,'Lorry Theft ';Ls;' credits lost':Score
738 END DEFine
```

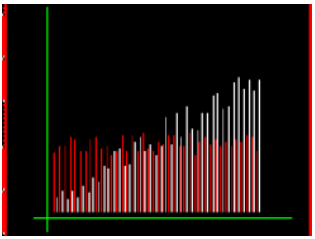
```
740 DEFine PROCedure ASReport
741 INK#ch,4:LINE#ch,10,20 TO 100,20:LINE#ch,15,15 TO 15,85
742 INK#ch,2:CURSOR#ch,12,4:PRINT#ch,'Assets';
743 INK#ch,4:PRINT#ch,'\';:INK#ch,7:PRINT#ch,'Sales','Invoice num: ';Ino-1;'/';Invmax
744 x=16:y=22:z=2:IF Asset>120 OR Sales>120:z=4
745 FOR n=2 TO 80 STEP 2
746   INK#ch,2:LINE#ch,x+n,y TO x+n,y+ASRep(n/2,1)/z
747   INK#ch,7:LINE#ch,x+n+1,y TO x+n+1,y+ASRep(n/2,2)/z
748 END FOR n
749 END DEFine
```

Note: Sales Report for each Invoice (scale z)

```
751 DEFine PROCedure AS_Demo1
752 INK#ch,4:LINE#ch,10,20 TO 110,20:LINE#ch,15,15 TO 15,90
753 a=50:s=0:x=16:y=22:demo=2
754 FOR n=2 TO 80 STEP 4
755   o=INT(RND(1 TO 24)/RND(4 TO 8)):d=RND(5 TO 6)
756   IF a>RND(8 TO 24):s=s+o*3:a=a-o
757   IF s>28 AND RND(1 TO 24)=7:a=a+d:s=s-d
758   IF n>40 AND RND(1 TO 3)=3:a=a+6
759   INK#ch,2:LINE#ch,x+n,y TO x+n,y+a/2
760   INK#ch,7:LINE#ch,x+n+2,y TO x+n+2,y+s/2
761 END FOR n
762 END DEFine
```



Note: Demo1 for Intro Page



```
764 DEFine PROCedure AS_Demo2
765 FOR n=1 TO 40
766  ASRep(n,1)=(50+RND(-12 TO 2))
767  ASRep(n,2)=(n*2+RND(-6 TO 18))
768 END FOR n
769 Ino=41:Invmax=40:ASReport
770 ino=1:invmax=20:FOR n=1 TO 40:ASRep(n,1)=0:ASRep(n,2)=0
771 END DEFine
```

Note: Demo2 - Representation of a completed 40 Invoices

QBITS Warehouse

```
773 DEFine PROCedure Test_Mode
774 F1=1
775 REPeat Test
776 ch=8:CLS#ch:PRINT#ch,'Test Mode':Score
777 tk=CODE(INKEY$(-1))
778 SELect ON tk
779   =49   :bay1=1:Arrival(1)          :REMark (1)Bay1_in Print & Lorry
780   =50   :Depart(1):bay1=0           :REMark (2)Bay1_out Print & Lorry
781   =51   :bay2=1:Arrival(2)          :REMark (3)Bay2_in Print & Lorry
782   =52   :Depart(2):bay2=0           :REMark (4)Bay2_out Print & Lorry
783   =53   :Store_err                  :REMark (5)Compute Crash
784   =54   :Stock_loss     :PAUSE      :REMark (6)Assets Lost
785   =55   :Asset_Tax      :PAUSE      :REMark (7)Loss of Sales credits
786   =56   :Energy_Bill    :PAUSE      :REMark (8)Loss of Sales credits
787   =57   :Stolen_Stock   :PAUSE      :REMark (9)loss of Sales credits
788   =97   :Sales=Sales+6:Stock_aud     :REMark (a)Increase Sales
789   =65   :Sales=Sales-6:Stock_aud     :REMark (A)Decrease Sales
790   =67   :Tim=Tim+3600 :Store_Clk     :REMark (C)Clock 1hr increments
791   =71   :Inv=Invmax+1               :REMark (G)Game End
792   =83,115:Stock_Request             :REMark (S s)
793   =232  :F1=0:EXIT Test             :REMark end Test Mode
794  END SELect
795 END REPeat Test
796 ch=8:CLS#ch
797 END DEFine
```

Checks

Store Clock
Increments – minutes and hours rotate through 24 hrs

Arrival(bay) & Depart(bay)

Lorry bay1 & bay2	Invoices incremented.
Lorry Arrival	Invoices Empty / Delivery with Goods
Lorry Departs	Stock Asset and Sales increased/decrease

Stock Request
Not active until Sales credits >12:No further Request until current one fulfilled

Lorry bay1 or bay2	Delivery
As lorry Departs	Stock Asset /Sales credits updated

Sales
Sales Increase and Decrease for testing other Hazards etc

Hazards

Store_err	Computer crash
Stock_loss	checks Asset Loss
Asset_Tax	checks Sale credits reduction
Energy_Bill	checks Sale credits reduction
Stolen_Stock	checks Sale credits reduction

QBITS Warehouse

QBITS WAREHOUSE

Having obtained a copy of **QBWH_Game** SuperBASIC code and loaded it into a recognised QL device. Use the QDOS command LRUN, as shown:-

LRUN flp1_QBWH_Game

Follow the instructions on the intro screen and all being well you will soon be striving to complete orders and see your sales rise...

Notes on QL2K emulator

Both the **QLAY & QL2K emulators** use an application tool to create a QDOS directory file and append or delete files in it. Creating a new qlay.dir file first open a Windows **Command Prompt** (Win 7 Press Start button in *search programs and files* box type **command prompt:** Win 10 in *ask me anything* box type **command prompt**.)

Activate the command prompt window then navigate with DOS commands to the drive and Windows File Directory folder that holds your QL Files.

i.e C:\>**chdir** H:\QL\FDIR\WIN1_ H:\QLFDIR\WIN1_>dir

This will list the files as a DOS directory. This needs to also contain a copy of **QLAYT-86.EXE or QLAY-X64.EXE** downloaded with **QLAY** or **QL2K**

At the DOS prompt now enter this command: -

i.e. H:\QL\FDIR\WIN1_>**qlayt-x64.exe –c qlay.dir**

This should create a directory file qlay.dir to which you can now append files. For example:-

i.e. H:\QL\FDIR\WIN1_>**qlayt-x64.exe –i Boot**

This will append the File named '**Boot**' to the qlay.dir .

Once you have appended your files you can use the following command to list them:-

i.e. H:\QL\FDIR\WIN1_> **qlayt-x64.exe –l**

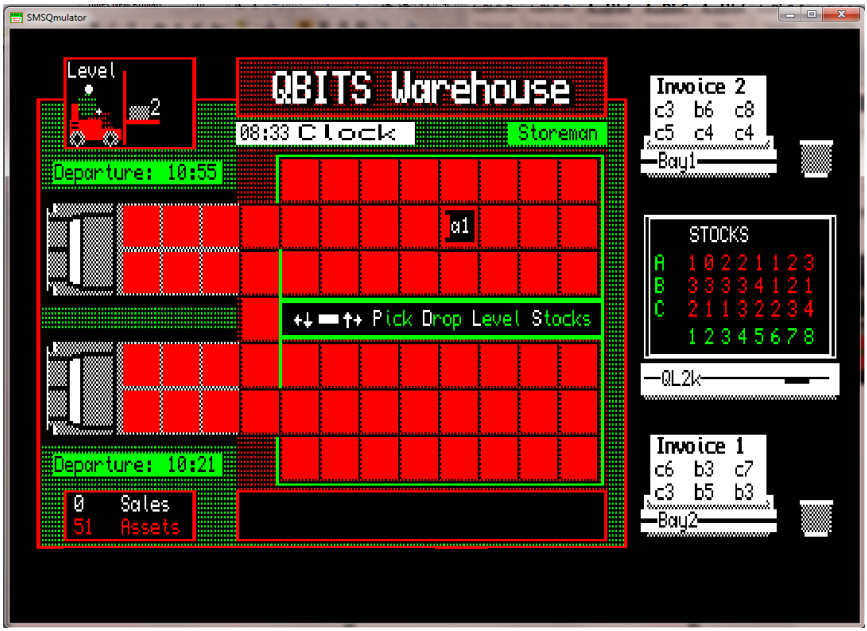
A list of files should now be shown contained within the qlay.dir

QL Emulators

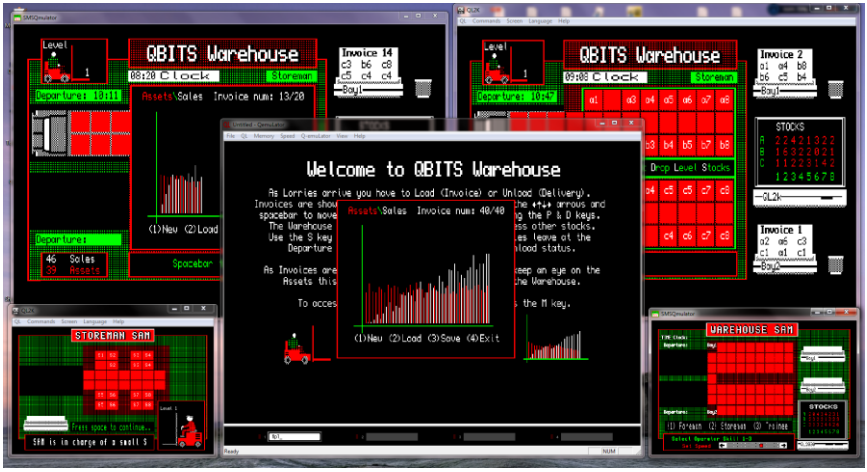
There are several available for the original QL as well as its later spin offs. You can download these and run them on PC's, Desktops Laptops and Tablets under the Windows, Mac or Linux operating systems. Then there are the additional ROM's and toolkit extensions and an extensive number of useful programs all with plenty of helpful documentation available.

Check out Dilwyn's web site below for downloads, helpful information and links to other suppliers of QL software and documentation.

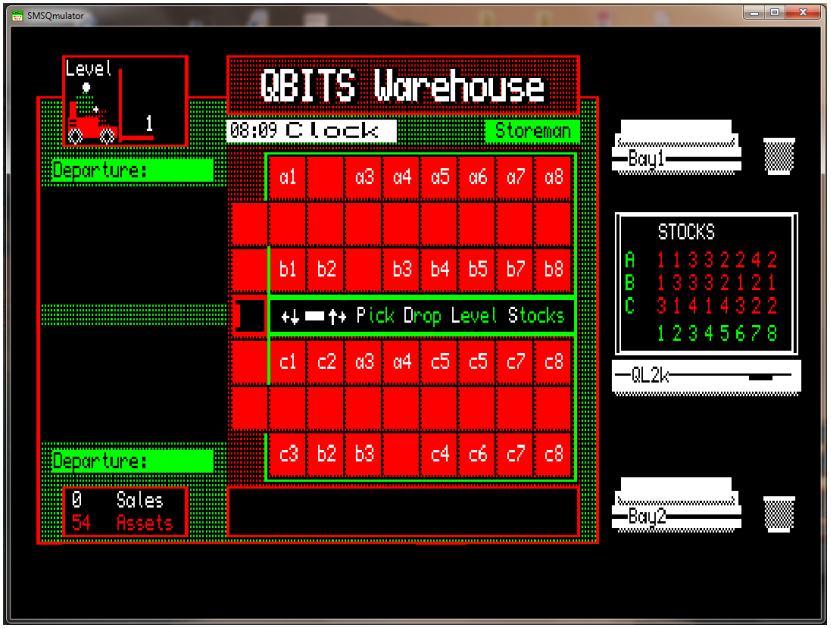
http://www.dilwyn.me.uk/ Updated: 07.09.2015



STOREMAN_SAM WAREHOUSE_SAM QBWH_Game
has been tried with **Qemulator** in original QL mode
and with **QL2K** and **SMSQmulator**




SMSQmulator
Welcome to QBITS Warehouse
As Lorries arrive you have to Load (Invoice) or Unload (Delivery).
Invoices are shown on the relevant Bay Printout. Use the ←↑↓→ arrows and
spacebar to move the Pick-UP. Then Pick and Drop using the P & D keys.
The Warehouse has four levels use the L key to access other stocks.
Use the S key for Stock Requests (new stock). Lorries leave at the
Departure time irrespective of their load or unload status.
As Invoices are fulfilled your Sales increase, but keep an eye on the
Assets this shows the Stocks currently held in the Warehouse.
To access the Menu (New Load Save Exit) press the M key.
press any key to continue...


SMSQmulator
Level
1
QBITS Warehouse
08:09 Clock
Storeman
Departure:
a1 a3 a4 a5 a6 a7 a8
b1 b2 b3 b4 b5 b7 b8
←↓ ↑→ Pick Drop Level Stocks
c1 c2 a3 a4 c5 c5 c7 c8
c3 b2 b3 c4 c6 c7 c8
Departure:
0 Sales
54 Assets
Bay1
STOCKS
A 1 1 3 3 2 2 4 2
B 1 3 3 3 2 1 2 1
C 3 1 4 1 4 3 2 2
1 2 3 4 5 6 7 8
QL2k
Bay2