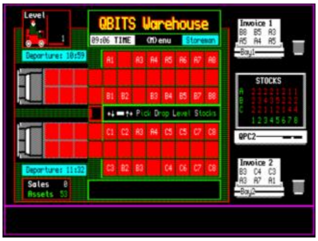
QBITS SuperBASIC Progs

GAMES Two

QBITS Warehouse

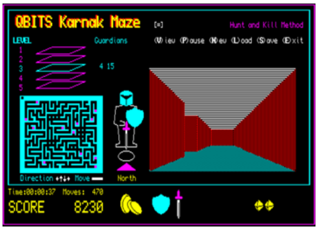
The Warehouse Prog started out as Storeman Sam the Guy driving the pickup. It then expanded to a larger store with two lorry bays and multi levels. The Printers show Sales Invoices and requested Stock Deliveries. The PC keeps Track of all held Stocks. The Sales are generated by departing Lorries loaded with Goods. Includes, hazards Stock loss etc. and Progress Chart of Sales/Stock.



QBITS Karnak Maze

This utilises a 3-Dimentional view around a Maze over five levels. Discover coins and other artifacts to collect while avoiding the Phantom Guardians.

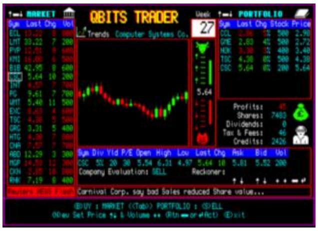
Your gained Wealth is used in solving a puzzle to unlock a Time Portal, but before this you have to defeat all the Phantom Guardians.



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.

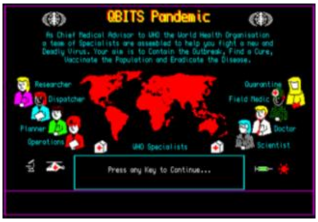
Your achievements are assessed at the end.



QBITS Pandemic

An 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.

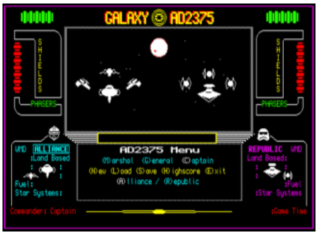
The number of turns, random Virus outbreaks and choices made, affects any successful outcome.

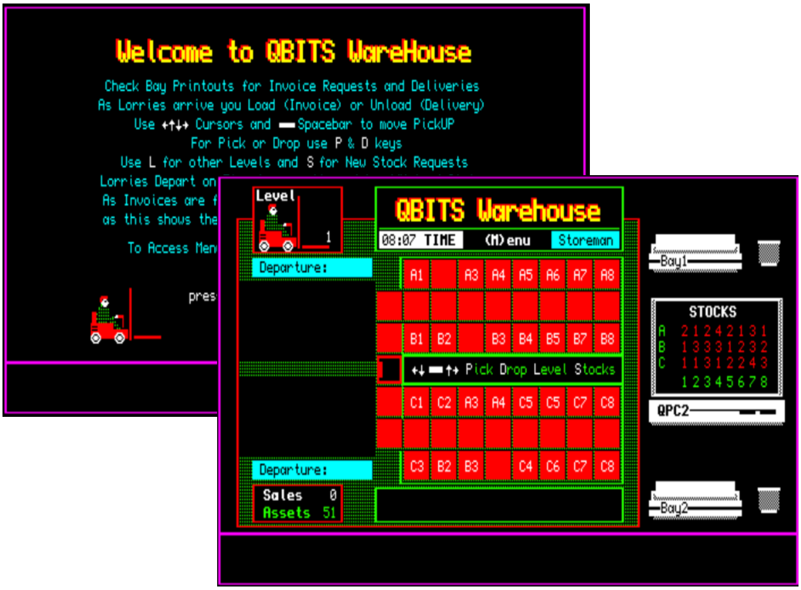


QBITS GALAXY AD2375

Play as Alliance or Republic. Take over the Galaxy, Star system by Star system by Acquisition or Military Force. Expect some Twists of Fortune along the way and encounters with enemy Fighters.

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





Welcome to QBITS WareHouse
Check Bay Printouts for Invoice Requests and Deliveries
As Lorries arrive you Load (Invoice) or Unload (Delivery)
Use ←↑↓→ Cursors and ▬ Spacebar to move PickUP
For Pick or Drop use P & D keys
Use L for other Levels and S for New Stock Requests
Lorries Depart on
As Invoices are f
as this shows the
To Access Men
pres

QBITS Warehouse
08:07 TIME (M)enu Storeman
Level 1
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:
Sales 0
Assets 51
Bay1
STOCKS
A 2 1 2 4 2 1 3 1
B 1 3 3 3 1 2 3 2
C 1 1 3 1 2 2 4 3
1 2 3 4 5 6 7 8
QPC2
Bay2

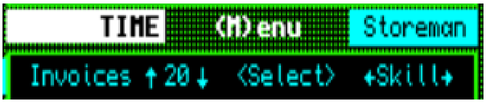
Introduction

Amongst old documentation of QL Programs, sketches and notes, I came across a two-sided sheet of graph paper headed ASRS. In the mid-eighties the company I work for at the time invested in a new innovation a computer controlled Automated Storage and Retrieval System (ASRS) This was quite a Project and I guess a starting point for writing my QBITS Game.

QBITS WareHouse Game

The aim is to fulfil Invoices of requested goods from the Warehouse Stocks that leads to an overall **Profit.** To achieve this **Sales** credits will need to overtake **Assets cost**. To accomplish this the Game can be is set to run between 10 and 40 Invoices. The Store holds 128 bins and 24 different types of goods. The maximum held for each item is 8, this allows the Store Computer to display available stock in three rows A/B/C and eight columns 1-8, the Stock items number shown as between 0 to 8. Lorries are loaded / unloaded within a set time frame, each Lorry Departs on time. The Invoices and Deliveries are identified by printer outputs.

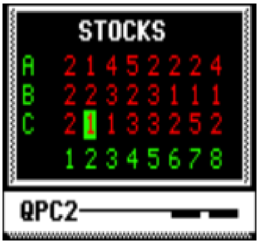
A Game requires actions that randomly upset the flow, the hazards chosen here are for a Store **Computer glitch** which is just annoying, losses due to **Missing Stock** which reduces the **Asset** and various others such as **Tax Revenues**, **Energy Bills,** and **Stolen Goods**, each of which deducts credits from the accrued **Sales**.



TIME (M)enu Storeman
Invoices ↑20↓ <Select> ←Skill→

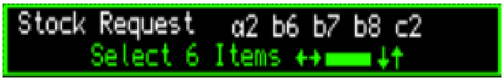
At the start of a **New** Game select **Skill** (Forman Storeman Trainee) and number of **Invoices**. These are chosen in multiples of 5 from 10 to 40. As an Invoice is called, the count is incremented for the next. Upon each Lorry Departure or Hazzard, **Assets** and **Sales** are checked and the corresponding results are displayed.

QBITS WareHouse Sales & Assets



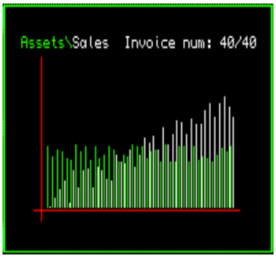
```
    STOCKS
A  2 1 4 5 2 2 2 4
B  2 2 3 2 3 1 1 1
C  2 1 1 3 3 2 5 2
   1 2 3 4 5 6 7 8
QPC2
```

As **Sales** are fulfilled the Store Stocks (**Assets**) are reduced. To replenish diminishing stock, the (**S**)tock Request brings in new Goods Deliveries to provide a means of restocking (**Assets**). It uses the Store Computer display to highlight the Stock item for selection.



```
Stock Request    a2 b6 b7 b8 c2
   Select 6 Items ↔▬▬▬↓↑
```

You cannot make a Stock Request if you have less than 12 Sales credits, or a delivery is already being processed.

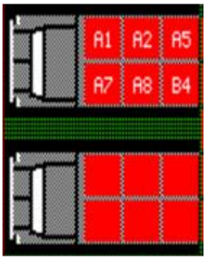


Lorry Departures with loaded Invoices generate **Sales.** Each unit of Stock (**Assets**) is valued as 1credit. If an Invoice is fulfilled and Lorry loaded as shown on the Printout each unit is worth 2 credits. If incorrectly loaded the items only counts as one credit.

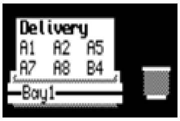
The Score is shown as **Sales/Assets**, press (**M**) for Menu and the window is opened to present an Audit Report Chart displaying the status of Profitability of completed transactions.

Variables keep track of Truck movement, storage Level, the Pick and Drop of Stock and to keep tab of changes to Stock (**Assets)** following a Delivery and/or move to fulfil an Invoice (**Sales**). These are processed to update the Store Computer Stocks (**Assets**) and the Score **(Sales)**.

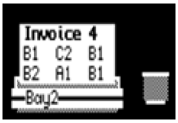
QBITS WareHouse Printers and Lorries



The printers are 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 display the Stock items Requested. As a lorry departs, the displayed printout is screwed up into a ball then further reduced to finally drop into the wastepaper basket.

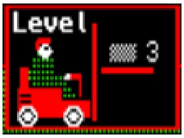


```
Delivery
A1 A2 A5
A7 A8 B4
Bay1
```



```
Invoice 4
B1 C2 B1
B2 A1 B1
Bay2
```

Bay1& **Bay2** =0 when empty =1 for an Invoice or =2 for a Delivery. For deliveries, **Del=0/1/2** identifies if a Delivery is pending and which loading bay it will be delivered to. **Din=0 or 1** active, when a Delivery comes in.



The Pickup Truck is moved by the cursor keys along track ways to access Stocks. Direction for Pick or Drop is cycled around with the Spacebar. 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).** Having four levels of storage added more gaming difficulties and introduced the graphics of Sam seated on his Pick-Up Truck to indicate which level was being accessed and is identified by (**lev**).

QBITS Warehouse Stock Movement

The basic graphics for the Warehouse display began with the Storage bays and Truck gangways. This is an 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 and if empty set to 32. All other cells are 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 to 255.

QBITS Warehouse Timing

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)

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.
the - x, y coordinates are specific to Channel / Window.

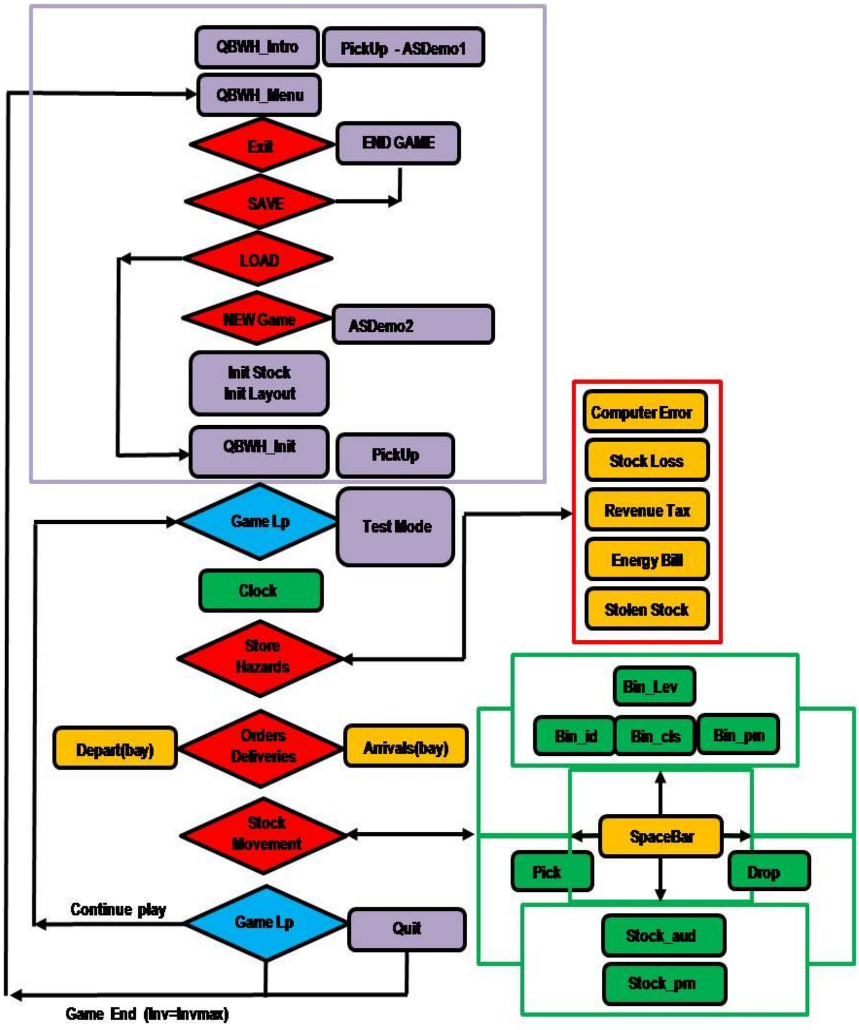
QBITS WareHouse Procedures

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
Sel_Path	Selects drive & Audit file
FCheck	Checks Drive & Audit File is available
Save_Audit	SAVE Game in progress for re- LOAD at later date
Load_Audit	LOAD previously saved Game
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

QBWH Flowchart

In working out various actions and how they relate to each in a Program it helps to create a Flowchart as shown below.

Pickup and bin movement see Green Section. Hazards are highlighted by a Red box.
The Menu - New Load Save Exit in Mauve box below.



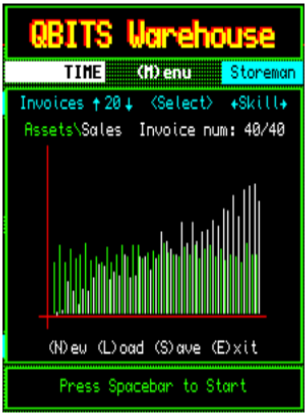
QBITS Warehouse Program Code

```
1000 REMark QBITS_WHQPC2_bas [QBITS WareHouse SE 2023 Review - QPC2]
:
1002 dev$='dos7_':MODE 4:gx=0:gy=0   :REMark Basic Settings

1004 WHEN ERRor :eck=1:CONTINUE:END WHEN

1006 REMark Import QBITSConfig Settings - QPC2
1007 OPEN_IN#9,dev$&'QBITSConfig':INPUT#9,gx\gy\dn$\dev$\dn%\dm%
1008 DIM drv$(dm%,5):FOR d=0 TO dm%:INPUT#9,drv$(d):END FOR d:CLOSE#9

1010 REMark Store Variables
1011 ClkOld=DATE:Tim=3600         :REMark ClkOld=Start Time Tim=1hr Increment
1012 Clk=0:cm=60:Skill=6           :REMark Clk=Clock cm=60 Skill 3/6/9 Timings
1013 Asset=0:Sales=0               :REMark Asset=Stock Sales=Orders fulfilled
1014 Invmax=10:Inv=1:Ino=1         :REMark Invmax=Max Orders Inv=Current Ino Chk Sales
1015 stk=32:stk$='  '              :REMark Stocks (1-24;32=Empty)Stk$='a1 to c8'
1016 r=4:c=5:l=1:lev=1             :REMark Warehouse row,column,level & Bin Level
1017 p=192:box=0:box$='  '         :REMark Truck Moves/Pick/Drop Status
1018 Dtime1=0:Dtime2=0             :REMark Scheduled Lorry Departure Times Bay 1&2
1019 bay1=0:bay2=0                 :REMark Bay 0=empty 1=Order 2=Delivery
1020 Del=0:Din=0:Gs=0              :REMark Delivery Del/Din 0/1/2 Game Status Gs 0/1/2/3
1021 SD$='QBWHAudit_':fnum=0       :REMark Audit Data Files 0 - 9

1023 REMark Arrays
1024 DIM Aud(24),InB(2,6),Stock(7,14,4),SReq(6),ASRep(40,2),Mes$(10,60)
1025 DIM Sk$(3,8)  :RESTORE 1026:FOR sk=1 TO 3:READ Sk$(sk)
1026 DATA 'Foreman ','Storeman','Trainee '

1028 QBWH_Intro:QBWH_Init:QBWH_Menu

1030 DEFine PROCedure QBWH_Menu
1031 ch=1:WINDOW#ch,220,144,gx+138,gy+49:CSIZE#1,0,0:INK#ch,0
1032 FOR i=1 TO 20:FILL#ch,1:CIRCLE#ch,60,50,i*3:FILL#ch,0:PAUSE 1
1033 BORDER#ch,1,4:PAPER#ch,0:CLS#ch:dn=5:af=1
1034 IF demo=2:AS_Demo2:demo=0:ELSE ASReport
1035 IF Gs=1:ch=8:CLS#ch:INK#ch,4:CURSOR#ch,50,4:PRINT#ch,'Spacebar to Return'
1036 IF Gs=4:ch=8:CLS#ch:INK#ch,4:CURSOR#ch,84,4:PRINT#ch,'GAME END'
1037 ch=1:INK#ch,7:CURSOR#ch,28,130:PRINT#ch,'(N)ew (L)oad (S)ave (E)xit'
1038 REPeat lp
1039   k=CODE(INKEY$(-1))
1040   SELect ON k
1041     =32:IF Gs=1:CLS#8  :Init_Layout:QBWH_Game     :REMark continue
1042     =78,110:New_Game :QBWH_Game                   :REMark (N)ew Game
1043     =76,108:Load_Audit :IF Gs=2:QBWH_Game         :REMark (L)oad Game
1044     =83,115:Save_Audit                            :REMark (S)ave Game
1045     =69,101:CLS#8 :GExit:CLS#8                    :REMark (E)xit Game
1046   END SELect
1047 END REPeat lp
1048 END DEFine
```

(N)ew (L)oad (S)ave (E)xit

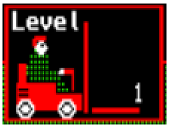
```
1050 DEFine PROCedure GExit
1051 CURSOR#8,84,4:PRINT#8,'Exit Y/N':PAUSE:IF KEYROW(5)=64:LRUN dn$
1052 END DEFine
```

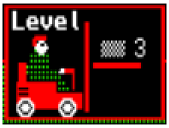
```
1054 DEFine PROCedure New_Game
1055 IF bay1>0:Depart 1:bay1=0
1056 IF bay2>0:Depart 2:bay2=0
1057 ch=1:sk=2:iv=3:CLS#8:INK#8,4:CURSOR#8,38,4:PRINT#8,'Press Spacebar to Start'
1058 INK#ch,5:CURSOR#ch,8,2:PRINT#ch,'Invoices ↑   ↓ <Select> ←Skill→ '
1059 REPeat Choice
1060   ch=3:BLOCK#ch,60,11,276,7,5:QBold 0,0,276,8,Sk$(sk)
1061   ch=1:CURSOR#ch,71,2:PRINT#ch,FILL$(' ',2-LEN(5+iv*5))&5+iv*5
1062   k=CODE(INKEY$(#ch,20))
1063   SELect ON k
1064    =192:sk=sk-1:IF sk<1:sk=3
1065    =200:sk=sk+1:IF sk>3:sk=1
1066    =208:iv=iv+1:BLOCK#1,12,10,71,2,0:IF iv>7:iv=7
1067    =216:iv=iv-1:BLOCK#1,12,10,71,2,0:IF iv<1:iv=1
1068    =69,101:LRUN dn$
1069    = 32:Skill=(sk*3):Invmax=5+(iv*5):CLS#8:EXIT Choice
1070   END SELect
1071 END REPeat Choice
1072 Inv=1:Ino=1:Sales=0:FOR i=1 TO 40:ASRep(i,1)=0:ASRep(i,2)=0
1073 Init_Stock:Init_Layout:Gs=1
1074 END DEFine
```


QBITS Warehouse
TIME
(M)enu
Storeman
Invoices ↑20↓ <Select> ←Skill→
Assets\Sales Invoice num: 40/40
(N)ew (L)oad (S)ave (E)xit
Press Spacebar to Start

```
1076 DEFine PROCedure QBWH_Game
1077 Bin_lev:Stock_aud:Score:Gs=1:r=4:c=5
1078 REPeat Game_lp
1079  Store_Clk:Truck_Pos p
1080  IF Inv>7 AND Sales>24 AND Del=0
1081    num=RND(1 TO 99)
1082    IF num=  3:Store_err
1083    IF num=  7:Stock_loss
1084    IF num=13:Asset_Tax
1085    IF num=21:Energy_Bill
1086    IF num=29:Stolen_Stock
1087  END IF
1088  IF Inv>Invmax:Gs=4:QBWH_Menu
1089  IF bay1=0 AND Inv<=Invmax AND RND(1 TO 24)=7:bay1=1:Arrival(1)
1090  IF bay1>0 AND Clk>=Dtime1:Depart 1:bay1=0
1091  IF bay2=0 AND Inv<=Invmax AND RND(1 TO 24)=7:bay2=1:Arrival(2)
1092  IF bay2>0 AND Clk>=Dtime2:Depart 2:bay2=0
1093  k=CODE(INKEY$(#1,20))
1094  SELect ON k
1095   =  32:p=p+8:Bin_cls:IF p>216:p=192
1096   =192:p=192:Bin_cls:IF Stock(r,c-1, 1)=0:c=c-1:IF c<6:lev=1:Bin_lev
1097   =200:p=200:Bin_cls:IF Stock(r,c+1,1)=0:c=c+1
1098   =208:p=208:Bin_cls:IF Stock(r-1,c, 1)=0:r=r-1
1099   =216:p=216:Bin_cls:IF Stock(r+1,c,1)=0:r=r+1
1100   =68,100:IF box<>0 :Truck_Drop          :REMark (D)rop
1101  =80,112:IF box= 0   :Truck_Pick          :REMark (P)ick
1102   =76,108:lev=lev+1  :Bin_lev             :REMark (L)evel
1103   =83,115            :Stock_Request       :REMark (S)tock
1104   =77,109:Gs=1       :QBWH_Menu           :REMark (M)enu
1105   =232               :Test_Mode           :REMark  F1
1106  END SELect
1107 END REPeat Game_lp
1108 END DEFine
```

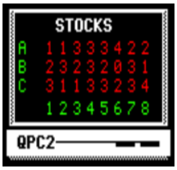
```
1110 DEFine PROCedure Store_Clk
1111 Clk=Tim+25200+((DATE-ClkOld)*cm):clock$=DATE$(Clk)
1112 ch=3:BLOCK#ch,30,10,124,8,7:STRIP#ch,7:INK#ch,0
1113 CURSOR#ch,126,8:PRINT#ch,clock$(13 TO 17)
1114 END DEFine

1116 DEFine PROCedure Bin_lev
1117 LOCal r,c:IF lev>4:lev=1
1118 ch=4:CLS#ch:INK#ch,7:BLOCK#ch,24,2,0,44-(8*lev),2
1119 IF box>=1 AND box<=24 :BLOCK#ch,12,6,4,36-(8*lev),248
1120 CURSOR#ch,20,34-(8*lev):PRINT#ch,lev
1121 FOR r=1 TO 7 STEP 2
1122   FOR c=6 TO 13:stk=Stock(r,c,lev):Bin_id:Bin_cls:Bin_prn
1123 END FOR r
1124 END DEFine
```





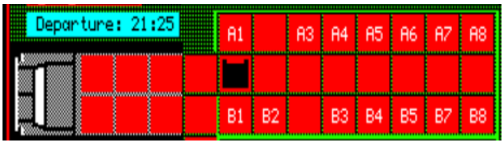
```
1126 DEFine PROCedure Bin_id
1127 stk$='  '
1128 SELect ON stk
1129  =  1 TO   8:stk$='A'&stk
1130  =  9 TO 16:stk$='B'&(stk-8)
1131  =17 TO 24:stk$='C'&(stk-16)
1132 END SELect
1133 END DEFine

1135 DEFine PROCedure Bin_cls
1136 ch=3:BLOCK#ch,22,18,2+c*24,5+r*20,2:1137 IF c<=5 AND lev>1:lev=1:Bin_lev
1137 END DEFine

1139 DEFine PROCedure Bin_prn
1140 ch=3:INK#ch,7:PAPER#ch,2:CURSOR#ch,7+c*24,10+r*20:PRINT#ch,stk$:stk$='  '
1141 END DEFine

1143 DEFine PROCedure Stock_aud
1144 LOCal r,c,l:Asset=0
1145 FOR i=1 TO 24:Aud(i)=0
1146 FOR l=1 TO 4
1147  FOR r=1 TO 7 STEP 2
1148   FOR c=6 TO 13
1149    IF Stock(r,c,l)>=1 AND Stock(r,c,l)<=24
1150      Aud(Stock(r,c,l))=Aud(Stock(r,c,l))+1
1151      IF Aud(Stock(r,c,l))>8:Stock(r,c,l)=32:ELSE Asset=Asset+1
1152    END IF
1153   END FOR c
1154  END FOR r
1155 END FOR l
1156 FOR stk=1 TO 24:Stock_prn stk
1157 END DEFine

1159 DEFine PROCedure Score
1160 ch=6:INK#ch,7
1161 CURSOR#ch,54, 1:PRINT#ch,FILL$(' ',3-LEN(Sales))&Sales:INK#ch,4
1162 CURSOR#ch,54,12:PRINT#ch,FILL$(' ',3-LEN(Asset))&Asset
1163 END DEFine
```

```
1165 DEFine PROCedure Stock_prn(stk)
1166 ch=7:INK#ch,2
1167 SELect ON stk
1168  = 1 TO  8:CURSOR #ch,(stk*10)+ 10,12:PRINT #ch,Aud(stk)
1169  = 9 TO 16:CURSOR #ch,(stk*10)- 70,22:PRINT #ch,Aud(stk)
1170  =17 TO 24:CURSOR #ch,(stk*10)-150,32:PRINT #ch,Aud(stk)
1171 END SELect
1172 END DEFine
```

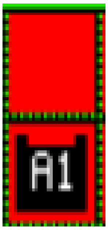


```
STOCKS
A  1 1 3 3 3 4 2 2
B  2 3 2 3 2 0 3 1
C  3 1 1 3 3 2 3 4
   1 2 3 4 5 6 7 8
QPC2
```

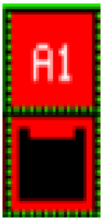
```
1174 DEFine PROCedure Truck_Pos(p)
1175 ch=3:BLOCK#ch,18,14,4+c*24,7+r*20,0
1176 SELect ON p
1177  =192:BLOCK#ch,4,10,4+c*24,9+r*20,2
1178  =200:BLOCK#ch,4,10,18+c*24,9+r*20,2
1179  =208:BLOCK#ch,12,4,7+c*24,6+r*20,2
1180  =216:BLOCK#ch,12,4,7+c*24,18+r*20,2
1181 END SELect
1182 INK#ch,7:STRIP#ch,0:CURSOR #ch,7+c*24,9+r*20:PRINT#ch,box$
1183 END DEFine
```



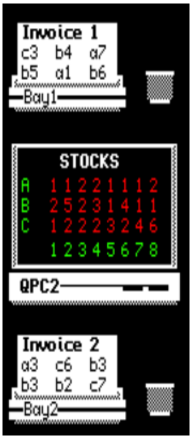
```
1185 DEFine PROCedure Truck_Pick
1186 rt=r:ct=c:l=lev                    :REMark rt,ct,lt temp hold of r,c,lev
1187 IF p=208:r=r-1                     :REMark Pick Up
1188 IF p=216:r=r+1                     :REM Pick Down
1189 IF c>5 AND c<14 AND Stock(r,c,l)>=1 AND Stock(r,c,l)<=24
1190    Aud(Stock(r,c,l))=Aud(Stock(r,c,l))-1
1191    stk=Stock(r,c,l):Stock(r,c,l)=32:Stock_prn stk:pick=1
1192    Bin_cls:Bin_prn:Bin_id:box=stk:box$=stk$
1193 END IF
1194 IF p=192:c=c-1                     :REMark Pick Left
1195 IF p=200:c=c+1                     :REMark Pick Right
1196 IF c<5 AND Stock(r,c,1)<25
1197   stk=Stock(r,c,1):Stock(r,c,1)=0:Bin_cls:Bin_prn:Bin_id:box=stk:box$=stk$
1198 END IF
1199 r=rt:c=ct:Bin_prn:IF box<>0:Bin_lev
1200 END DEFine
```

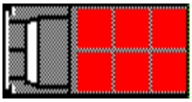


```
1202 DEFine PROCedure Truck_Drop
1203 rt=r:ct=c:l=lev
1204 IF p=208:r=r-1                     :REMark Drop Up
1205 IF p=216:r=r+1                     :REMark Drop Down
1206 IF c>5 AND c<14 AND Stock(r,c,l)=32
1207    Stock(r,c,l)=box
1208    Aud(Stock(r,c,l))=Aud(Stock(r,c,l))+1:Stock_prn box
1209    stk$=box$:Bin_prn:box=0:box$='  ':pick=0
1210 END IF
1211 IF p=192:c=c-1                     :REMark Drop Left
1212 IF p=200:c=c+1                     :REMark lDrop Right
1213 IF c<5 AND Stock(r,c,1)=0
1214   Stock(r,c,1)=box:box=0:stk$=box$:Bin_prn:box$='  '
1215 END IF
1216 r=rt:c=ct:IF box=0:ch=4:BLOCK#ch,12,6,4,36-(8*lev),0
1217 END DEFine
```

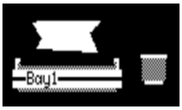


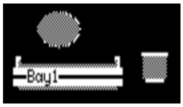
```
1219 DEFine PROCedure Arrival(bay)
1220 IF bay=1:py=16:ly=71:rl=2:ELSE py=171:ly=131:rl=5
1221 ar=r:ac=c:Prn_in:Lorry_in:n=0:ch=5:WINDOW#ch,352,192,gx+18,gy+31
1222 FOR r=rl TO rl+1
1223  FOR c=2 TO 4
1224   Stock(r,c,1)=0:Bin_cls
1225   IF Del=bay
1226    n=n+1:stk=SReq(n):Stock(r,c,l)=stk:Bin_id:Bin_prn
1227   END IF
1228  END FOR c
1229 END FOR r
1230 r=ar:c=ac:IF Del=bay:Din=1
1231 IF bay=1 AND bay2=1:dlay=300*Skill:ELSE dlay=0
1232 IF bay=1:ty=25 :Dtime1=dlay+Clk+(1200*Skill):Dept$=DATE$(Dtime1)
1233 IF bay=2 AND bay1=1:dlay=300*Skill:ELSE dlay=0
1234 IF bay=2:ty=152:Dtime2=dlay+Clk+(1200*Skill):Dept$=DATE$(Dtime2)
1235 STRIP#ch,5:INK#ch,0:CURSOR#ch,84,ty:PRINT#ch,Dept$(13 TO 17)
1236 END DEFine

1238 DEFine PROCedure Prn_in
1239 ch=5:WINDOW#ch,70,30,gx+386,gy+py:PAPER#ch,0:CLS#ch
1240 n=0:PAPER#ch,7:INK#ch,0:OVER#ch,1:SCROLL#ch,-10
1241 IF Del=bay
1242  String$='Delivery':CLS#8      :REMark clear Stock Request Window
1243 ELSE
1244  String$='Invoice '&Inv:Inv=Inv+1
1245 END IF
1246 FOR i=0 TO 1:CURSOR#ch,2+i,20:PRINT#ch,String$
1247 FOR d=1 TO 2
1248  PAUSE 2:SCROLL #ch,-10
1249  FOR a=1 TO 3
1250   n=n+1:IF Del=bay:stk=SReq(n):ELSE stk=RND(1 TO 24):InB(bay,n)=stk
1251   Bin_id:CURSOR#ch,-22+a*24,20:PRINT#ch,stk$:stk$='  '
1252  END FOR a
1253 END FOR d
1254 END DEFine

1256 DEFine PROCedure Lorry_in
1257 ch=5:WINDOW#ch,120,40,gx+18,gy+ly:PAPER#ch,248
1258 FOR i=1 TO 19:PAN#ch,4:PAUSE 2
1259 BLOCK#ch,2,40,0,0,0
1260 FOR i=1 TO 5
1261  PAN#ch,4:BLOCK#ch,4,40,0,0,248:BLOCK#ch,4,2,0,4,0:BLOCK#ch,4,2,0,36,0
1262 END FOR i
1263 BLOCK#ch,2,32,0,4,0
1264 FOR i=1 TO 2
1265  PAN#ch,4:BLOCK#ch,4,2,0,4+i,0:BLOCK#ch,4,29-i,0,6+i,7:BLOCK#ch,4,2,0,36-i,0
1266 END FOR i
1267 BLOCK#ch,2,28,0,6,0
1268 FOR i=1 TO 3
1269  PAN#ch,4:BLOCK #ch,4,2,0,6,0:BLOCK#ch,4,2,0,19,0:BLOCK#ch,4,2,0,32,0
1270 END FOR i
1271 BLOCK#ch,6,4,0,0,7 :BLOCK#ch,4,4,0,0,0 :BLOCK#ch,6,4,0,36,7
1272 BLOCK#ch,4,4,0,36,0:BLOCK#ch,2,36,0,2,7:PAPER#ch,0:PAN#ch,4
1273 END DEFine
```

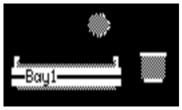


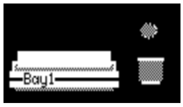


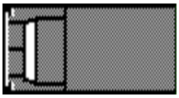
```
1275 DEFine PROCedure Depart(bay)
1276 IF bay=1:lr=2:py=16:ly=71:ty=21:ELSE lr=5:py=171:ly=131:ty=148
1277 dr=r:dc=c:n=0
1278 FOR r=lr TO lr+1
1279  FOR c=2 TO 4
1280   IF Del=bay
1281     IF Stock(r,c,1)=0:Sales=Sales-1:END IF
1282   ELSE
1283     n=n+1:IF Stock(r,c,1)=InB(bay,n):Sales=Sales+2:END IF
1284     IF Stock(r,c,1)>=1 AND Stock(r,c,1)<=24:Sales=Sales+1:END IF
1285   END IF
1286   Stock(r,c,1)=255
1287  END FOR c
1288 END FOR r
1289 IF Del=bay AND Din=1
1290    Del=0:Din=0
1291 ELSE
1292    Stock_aud:Score:ASRep(Ino,1)=Asset:ASRep(Ino,2)=Sales:Ino=Ino+1
1293 END IF
1294 ch=5:WINDOW#ch,352,192,gx+18,gy+31:BLOCK#ch,30,10,84,ty,5  :REMark Clr DTime
1295 r=dr:c=dc:Lorry_out:Prn_out:Stock_aud:Score
1296 END DEFine

1298 DEFine PROCedure Prn_out
1299 ch=5:WINDOW#ch,70,30,gx+386,gy+py:PAPER#ch,0:CLS#ch:INK#ch,7:FILL#ch,1
1300 LINE#ch,10,90 TO 170,90 TO 150,60 TO 170,15 TO 10,20 TO 25,50 TO 10,90
1301 PAUSE 8:CLS#ch:FILL#ch,1
1302 LINE#ch,30,80 TO 150,80 TO 130,60 TO 145,25 TO 30,30 TO 40,50 TO 20,80
1303 PAUSE 5:CLS#ch:INK#ch,248
1304 FILL#ch,1:CIRCLE#ch,70,60,40,.8,PI/2:PAUSE 5:CLS#ch
1305 FILL#ch,1:CIRCLE#ch,150,70,20:PAUSE 10:CLS#ch:BLOCK#ch,70,10,0,20,7
1306 ch=5:WINDOW#ch,20,30,gx+476,gy+py-6:PAPER#ch,0:CLS#ch:INK#ch,248
1307 FILL#ch,1:CIRCLE#ch,20,70,15:PAUSE 5:FOR i=1 TO 6:SCROLL#ch,5:PAUSE 3
1308 END DEFine

1310 DEFine PROCedure Lorry_out
1311 ch=5:WINDOW#ch,120,40,gx+18,gy+ly:PAPER#ch,0:BLOCK#ch,72,40,48,0,248
1312 FOR i=1 TO 30:PAN#ch,-4:PAUSE 1
1313 IF c<5
1314   IF bay=1 AND r<4:c=5:r=4:Bin_cls:Truck_Pos 192
1315   IF bay=2 AND r>4:c=5:r=4:Bin_cls:Truck_Pos 192
1316 END IF
1317 END DEFine

1319 DEFine PROCedure Stock_loss
1320 IF Asset<36:RETurn::ch=8:CLS#ch:INK#ch,7:OVER#ch,1
1321 CURSOR#ch,48,5:PRINT#ch,'Attention Stock LOSS!':att=RND(1 TO 7)
1322 IF att=1:FOR i=8 TO RND(9 TO 11):Stock(1,i,1)=32:Stock(7,i,1)=32
1323 IF att=3:FOR i=1 TO RND(2 TO   3):Stock(3,i,3)=32:Stock(5,i,3)=32
1324 IF att=5:FOR i=7 TO RND(8 TO 10):Stock(1,i,1)=32:Stock(7,i,1)=32
1325 IF att=7:FOR i=4 TO RND(5 TO   7):Stock(3,i,3)=32:Stock(5,i,3)=32
1326 Stock_aud:Score:FOR stk=1 TO 24:Stock_prn stk
1327 END DEFine
```



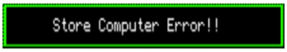
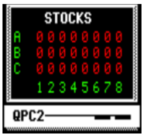




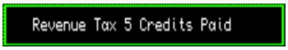




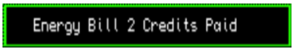
Attention Stock LOSS!


```
Store Computer Error!!
```

```
   STOCKS
A 00000000
B 00000000
C 00000000
  12345678
QPC2
```

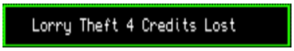
```
1329 DEFine PROCedure Store_err
1330 ch=8:CLS#ch:INK#ch,7: CURSOR#ch,36,5:PRINT#ch,'Store Computer Error!';
1331 FOR stk=1 TO 24:Aud(stk)=0:Stock_prn(stk)
1332 FOR t=1 TO 5:PAUSE 30:Store_Clk:ch=8:PRINT#ch,'!';
1333 CLS#ch:Stock_aud:FOR stk=1 TO 24:Stock_prn stk
1334 END DEFine
```


```
Revenue Tax 5 Credits Paid
```

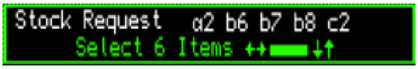
```
1336 DEFine PROCedure Asset_Tax
1337 IF Sales<42:RETurn
1338 ch=8:CLS#ch:INK#ch,7:Tr=INT(Sales/10):Sales=Sales-Tr
1339 CURSOR#ch,20,5:PRINT#ch,'Revenue Tax ';Tr;' credits paid':Score
1340 END DEFine
```


```
Energy Bill 2 Credits Paid
```

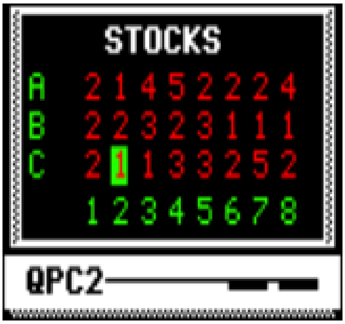
```
1342 DEFine PROCedure Energy_Bill
1343 IF Sales<36:RETurn
1344 ch=8:CLS#ch:INK#ch,7:Ec=INT(Asset/20):Sales=Sales-Ec
1345 CURSOR#ch,20,5:PRINT#ch,'Energy Bill ';Ec;' credits paid':Score
1346 END DEFine
```

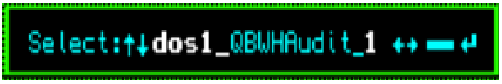

```
Lorry Theft 4 Credits Lost
```

```
1348 DEFine PROCedure Stolen_Stock
1349 IF Sales<24:RETurn
1350 ch=8:CLS#ch:INK#ch,7:Ls=RND(2 TO 6):Sales=Sales-Ls
1351 CURSOR#ch,20,5:PRINT#ch,'Lorry Theft ';Ls;' credits lost':Score
1352 END DEFine
```


```
Stock Request   a2 b6 b7 b8 c2
        Select 6 Items ↔▬ ↓↑
```

```
1354 DEFine PROCedure Stock_Request
1355 IF Del>0 OR Sales<12:RETurn
1356 ch=8:CLS #ch:x=1:y=0:n=1:s=0
1357 INK #ch,7:CURSOR #ch,4,0  :PRINT #ch,'Stock Request'
1358 INK #ch,4:CURSOR #ch,36,10:PRINT #ch,'Select 6 Items ← →      ↑↓'
1359 BLOCK #ch,20,4,140,14,4                              :REMark SpaceBar
1360 Stock_aud:FOR stk=1 TO 24:Stock_prn stk
1361 REPeat Stock_lp
1362  Store_Clk : IF n>6:Del=RND(1 TO 2):RETurn
1363  ch=7:STRIP #ch,4:INK #ch,0:Stock_prn x+8*y
1364  k$=INKEY$(#1,20):k=CODE(k$)
1365  ch=7:STRIP #ch,0:INK #ch,2:Stock_prn x+8*y
1366  SELect ON k
1367    =192:IF x>1:x=x-1:ELSE x=1
1368    =200:IF x<8:x=x+1:ELSE x=8
1369    =208:IF y>0:y=y-1:ELSE y=0
1370    =216:IF y<2:y=y+1:ELSE y=2
1371    = 32:IF n<=6
1372          ch=8:Ink#ch,7:stk=x+y*8:SReq(n)=stk:Bin_id
1373         CURSOR #ch,80+(n*18),1:PRINT#ch,stk$;n=n+1:str$=' '
1374        END IF
1375  END SELect
1376 END REPeat Stock_lp
1377 END DEFine
```

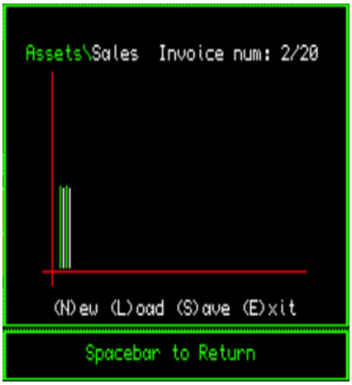

```
      STOCKS
A  2 1 4 5 2 2 2 4
B  2 2 3 2 3 1 1 1
C  2 1 1 3 3 2 5 2
   1 2 3 4 5 6 7 8
QPC2
```



```
Select:↑↓dos1_QBWHAudit_1 ↔ ▬ ↵
```

```
1379 DEFine PROCedure SelPath
1380 INK#ch,5:CURSOR#ch,8,6:PRINT#ch,'Select: ⬆⬇      ';SD$;' ⬅ ➡ ↵'
1381 BLOCK#ch,12,3,186,10,5:BLOCK#ch,2,4,206,8,5:OVER#ch,0:INK#ch,7
1382 REPeat Path_lp
1383  CURSOR#ch,68,6:PRINT#ch,drv$(dn%):CURSOR#ch,156,6:PRINT#ch,fnum
1384  k=CODE(INKEY$(-1))
1385  SELect ON k
1386   =216:dn%=dn%-1  :IF dn%<0:dn%=dm%
1387   =208:dn%=dn%+1  :IF dn%>dm%:dn%=0
1388   =200:fnum=fnum+1:IF fnum>9:fnum=9
1389   =192:fnum=fnum-1:IF fnum<0:fnum=0
1390   = 10:pck=1:EXIT Path_lp
1391   = 32:pck=0:EXIT Path_lp
1392  END SELect
1393 END REPeat Path_lp
1394 END DEFine

1396 DEFine PROCedure FCheck
1397 CLS#ch:CURSOR#ch,56,6:PRINT#ch,'Searching...'
1398 PAUSE 20:DELETE drv$(dn%)&'FList'
1399 OPEN_NEW#99,drv$(dn%)&'FList':DIR#99,drv$(dn%):CLOSE#99
1400 OPEN_IN#99,drv$(dn%)&'FList'
1401 REPeat Dir_lp
1402   IF EOF(#99):CLOSE#99:CLS#ch:pck=0:EXIT Dir_lp
1403   INPUT#99,Fchk$:IF Fchk$==SD$&fnum:CLOSE#99:pck=1:EXIT Dir_lp
1404 END REPeat Dir_lp
1405 END DEFine

1407 DEFine PROCedure Load_Audit
1408 ch=8:CLS#ch:SelPath:IF pck=0:CLS#ch:RETurn :ELSE FCheck
1409 IF pck=0 OR eck=1
1410    CLS#ch:eck=0:CURSOR#ch,56,6:PRINT#ch,'File Not Found...'
1411    PAUSE 50:CLS#ch:RETurn
1412 END IF
1413 IF bay1>0:Depart 1:bay1=0
1414 IF bay2>0:Depart 2:bay2=0
1415 ch=8:CLS#ch:CURSOR#ch,56,6:PRINT#ch,'Loading...';
1416 OPEN_IN#99,drv$(dn%)&SD$&fnum
1417 FOR l=1 TO 4
1418   FOR r=1 TO 7
1419     FOR c=1 TO 14:INPUT#99,Stock(r,c,l)
1420     CURSOR#ch,104+r*6,6:PRINT#ch,'.';:PAUSE 2
1421   END FOR r
1422 END FOR l
1423 FOR n=1 TO 40:INPUT#99,ASRep(n,1)\ASRep(n,2)
1424 INPUT#99,Inv\Ino\Invmax\Sales\Sk$(sk):CLOSE#99:CLS#ch
1425 Init_Layout:Gs=2:BLOCK#3,60,11,276,7,5:QBold 0,0,276,8,Sk$(sk)
1426 END DEFine
```

```
1428 DEFine PROCedure Save_Audit
1429 ch=8:CLS#ch:SelPath:IF pck=0:CLS#ch:RETurn :ELSE FCheck
1430 IF eck=1
1431    eck=0:CLS#ch:CURSOR#ch,56,6:PRINT#ch,'DEVICE ERROR'
1432    PAUSE 50:CLS#ch:RETurn
1433 END IF
1434 IF pck=1
1435    CLS#ch:CURSOR#ch,56,6:PRINT#ch,'Overwrite y/n':PAUSE
1436    IF KEYROW(5)<>64:CLS#ch:RETurn
1437 END IF
1438 FOR r=2 TO 3:FOR c=2 TO 4:Stock(r,c,1)=255:END FOR c:END FOR r
1439 FOR r=5 TO 6:FOR c=2 TO 4:Stock(r,c,1)=255:END FOR c:END FOR r
1440 DELETE drv$(dn%)&SD$&fnum:OPEN_NEW#99,drv$(dn%)&SD$&fnum
1441 CLS#ch:CURSOR#ch,56,6:PRINT#ch,'Saving...';:CLS#ch,2
1442 FOR l=1 TO 4
1443  FOR r=1 TO 7
1444   FOR c=1 TO 14:PRINT#99,Stock(r,c,l)
1445   CURSOR#ch,104+r*6,6:PRINT#ch,'.';:PAUSE 2
1446  END FOR r
1447 END FOR l
1448 FOR n=1 TO 40:PRINT#99,ASRep(n,1)\ASRep(n,2)
1449 Inv=Ino:PRINT#99,Inv\Ino\Invmax\Sales\Sk$(sk):CLOSE#99:CLS#8
1450 END DEFine

1452 DEFine PROCedure ASReport
1453 INK#ch,2:LINE#ch,12,16 TO 100,16:LINE#ch,15,12 TO 15,80
1454 INK#ch,4:CURSOR#ch,12,16:PRINT#ch,'Assets\';
1455 INK#ch,7:PRINT#ch,'Sales  Invoice num: ';Ino-1;'/';Invmax
1456 x=16:y=18:z=2:IF Asset>120 OR Sales>120:z=4
1457 FOR n=2 TO 80 STEP 2
1458   INK#ch,4:LINE#ch,x+n,y TO x+n,y+ASRep(n/2,1)/z
1459   INK#ch,7:LINE#ch,x+n+1,y TO x+n+1,y+ASRep(n/2,2)/z
1460 END FOR n
1461 END DEFine

1463 DEFine PROCedure Init_Stock
1464 RESTORE 1474
1465 FOR r=1 TO 7
1466   FOR c=1 TO 14
1467     READ Stock(r,c,1)
1468     IF c>=6 AND c<=13
1469       n=RND(1 TO 32):IF n>24:n=32
1470       Stock(r,c,3)=n:Stock(r,c,2)=32:Stock(r,c,4)=32
1471     END IF
1472   END FOR c
1473 END FOR r
1474 DATA 225,225,225,225,255,1,32,3,4,5,6,7,8,255
1475 DATA 225,225,225,225,0,0,0,0,0,0,0,0,0,255
1476 DATA 225,225,225,225,0,9,10,32,11,12,13,15,16,255
1477 DATA 225,225,225,225,0,255,255,255,255,255,255,255,255,255
1478 DATA 225,225,225,225,0,17,18,3,4,21,21,23,24,255
1479 DATA 225,225,225,225,0,0,0,0,0,0,0,0,0,255
1480 DATA 225,225,225,225,255,19,10,11,32,20,22,23,24,255
1481 END DEFine
```



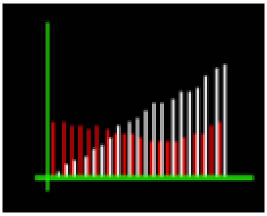
```
Assets\Sales  Invoice num: 2/20

(N)ew  (L)oad  (S)ave  (E)xit

Spacebar to Return
```

```
1483 DEFine PROCedure QBWH_Init
1484 CLS#2:F1=0:stk=0:r=4:c=5:l=1:p=192:bay1=0:bay2=0
1485 REMark ** Store Layout **
1486 ch=3:OPEN#ch,scr_:WINDOW#ch,356,194,gx+16,gy+26
1487 BORDER#ch,1,2:PAPER#ch,32:CLS#ch
1488 BLOCK#ch,120,50,0,38,0:BLOCK#ch,120,50,0,98,0       :REMark Drive Bay 1&2
1489 REMark ** QBITS Title **
1490 ch=3:WINDOW#ch,220,41,gx+138,gy+6:BORDER#ch,1,4:PAPER#ch,0:CLS#ch
1491 OVER#ch,1:CSIZE#ch,2,1
1492 INK#ch,2:FOR i=0 TO 3:CURSOR#ch,14+i,4:PRINT#ch,'QBITS Warehouse'
1493 INK#ch,6:FOR i=0 TO 1:CURSOR#ch,16+i,5:PRINT#ch,'QBITS Warehouse'
1494 OVER#ch,0:BLOCK#ch,212,1,2,25,4
1495 REMark ** Headings **
1496 ch=3:WINDOW#ch,352,192,gx+18,gy+27:CSIZE#ch,0,0
1497 BLOCK#ch, 74,11,124   ,7,7:QBold 0,1,156,8,'TIME'
1498 BLOCK#ch, 60,11,276,   7,5:QBold 7,1,208,8,'(M)enu'
1499 BLOCK#ch,106,12, 12, 24,5:QBold 0,0,12, 25,'Departure:'
1500 BLOCK#ch,106,12, 12,151,5:QBold 0,0,12,152,'Departure:'
1501 REMark ** PickUp **
1502 ch=4:OPEN#ch,scr_:WINDOW#ch,80,42,gx+30,gy+6:BORDER#ch,1,2:PAPER#ch,0
1503 CLS#ch:Init_PickUp:QBold 7,1,-6,0,'Level':WINDOW#ch,30,40,gx+74,gy+7
1504 REMark ** Printers Bay 1 & 2 **
1505 ch=5:OPEN#ch,scr_
1506 FOR b=1 TO 2
1507    IF b=1:y=36:ELSE y=191
1508    WINDOW#ch,82,24,gx+380,gy+y                        :REMark Printer
1509    BLOCK#ch,78,16,2,6,7:BLOCK#ch,74,16,4, 8,248:BLOCK#ch,82,10,0,12,7
1510    BLOCK#ch,82,2,0,16,0:BLOCK#ch,70,10,6, 0,  7
1511    STRIP#ch,7:INK#ch,0 :CURSOR#ch,10,12:PRINT#ch,'Bay'&b
1512    WINDOW#ch,20,16,gx+476,gy+y+4                      :REMark Waste basket
1513    BLOCK#ch,20,2,0,0,7 :BLOCK#ch,14, 2,3,14,  7:BLOCK#ch,18,12,1,2,248
1514 END FOR b
1515 REMark ** Score **
1516 ch=6:OPEN#ch,scr_:WINDOW#ch,80,24,gx+30,gy+193:BORDER#ch,1,2:PAPER#ch,0
1517 CLS#ch:QBold 7,1,0,1,'Sales':QBold 4,1,0,12,'Assets'
1518 REMark ** QL2K PC **
1519 ch=7:OPEN#ch,scr_:WINDOW#ch,120,80,gx+380,gy+76:PAPER#ch,0:CLS#ch
1520 BLOCK#ch,116,62,2,0,7 :BLOCK#ch,114,60,3,1,248:BLOCK#ch,110,60,5,1,7
1521 BLOCK#ch,108,58,6,2,0 :BLOCK#ch,120,14,0,64,7 :BLOCK#ch,76,1,36,70,0
1522 BLOCK#ch,14,2,80,71,0 :BLOCK#ch,14,2,98,71,0  :BLOCK#ch,116,2,2,78,248
1523 QBold 0,1,0,66,'QPC2' :WINDOW#ch,100,56,gx+388,gy+80:PAPER#ch,0:CLS#ch
1524 QBold 7,1,20,0,'STOCKS':INK#ch,4:CURSOR#ch,0,12:PRINT#ch,'A'\'B'\'C'
1525 FOR i=1 TO 8:CURSOR#ch,10+(i*10),44:PRINT#ch,i
1526 REMark ** Requests & Messages **
1527 ch=8:OPEN#ch,scr_:WINDOW#ch,220,22,gx+138,gy+195:BORDER#ch,1,4:PAPER#ch,0
1528 END DEFine

1530 DEFine PROCedure QBold(col,cs,cx,cy,str$)
1531 INK#ch,col:OVER#ch,1
1532 FOR a=1 TO LEN(str$)
1533   FOR b=0 TO cs:CURSOR#ch,cx+b+a*(cs+6),cy:PRINT#ch,str$(a)
1534 END FOR a:OVER#ch,0
1535 END DEFine
```

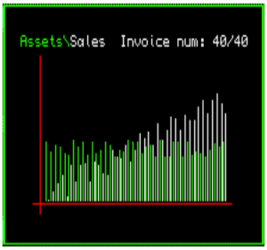
WareHouse & Walls

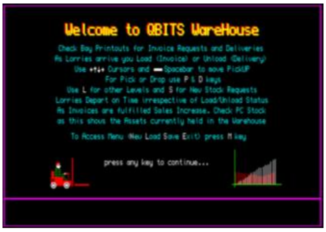
```
1537 DEFine PROCedure Init_Layout
1538 ch=3:WINDOW#ch,352,192,gx+18,gy+27:BLOCK#ch,222,148,120,20,32
1539 BLOCK#ch,196,1,142, 23,4:BLOCK#ch,2,143,338,23,4:BLOCK#ch,1,60,144, 65,4
1540 BLOCK#ch,196,1,142,165,4:BLOCK#ch,2, 20,142,23,4:BLOCK#ch,2,20,142,145,4
1541 BLOCK#ch,196,20,144,84,4 :BLOCK#ch,192,16,146,86,0:INK#ch,7:OVER#ch,1
1542 CURSOR#ch,152,89:PRINT#ch,'←↑    ↓→':BLOCK#ch,12,4,168,92,7
1543 CURSOR#ch,200,89:PRINT#ch,'P    D    L     S':INK#ch,4
1544 CURSOR#ch,200,89:PRINT#ch,' ick  rop  evel  tocks':OVER#ch,0
1545 FOR d=1 TO 7
1546   FOR a=1 TO 14:IF Stock(d,a,1)<200:BLOCK#ch,22,18,2+a*24,5+d*20,2
1547 END FOR d
1548 END DEFine

1550 DEFine PROCedure Init_PickUp
1551 INK#ch,32:FILL#ch,1:CIRCLE#ch,46,32,5,3,-.8:FILL#ch,0               :REMark Legs
1552 INK#ch,32:FILL#ch,1:CIRCLE#ch,28,44,18,.6,PI:FILL#ch,0              :REMark Body
1553 INK#ch, 7:FILL#ch,1:CIRCLE#ch,30,65,5:FILL#ch,0                     :REMark Head
1554 BLOCK#ch,2,1,16,13,0:BLOCK#ch,2,3,12,11,2:BLOCK#ch,6,2,13,10,2  :REMark Hair
1555 INK#ch, 7:FILL#ch,1:CIRCLE#ch,50,42,4,.5,PI/2:FILL#ch,0             :REMark Hand
1556 BLOCK#ch,1,8,32,22,2  :BLOCK#ch,4,2,28,22,2: FILL#ch,1:INK#ch,2
1557 LINE#ch,8,8 TO 68,8 TO 68,20 TO 44,20 TO 44,30 TO 16,30             :REMark PickUp
1558 LINE#ch TO 16,50 TO 9,50 TO 9,30 TO 8,30 TO 8,8:FILL#ch,0
1559 INK#ch,0:LINE#ch,9,21 TO 18,21:LINE#ch,9,26 TO 18,26                :REMark Vents
1560 INK#ch,2:LINE#ch,72,4 TO 72,90:LINE#ch,74,4 TO 74,90                :REMark Lift bar
1561 INK#ch,7:FILL#ch,1:CIRCLE#ch,16,7,8:FILL#ch,0                       :REMark Left Wheel
1562 INK#ch,7:FILL#ch,1:CIRCLE#ch,55,7,8:FILL#ch,0                       :REMark Right Wheel
1563 INK#ch,0:CIRCLE#ch,16,7,5:CIRCLE#ch,55,7,5                          :REMark Wheel Hubs
1564 END DEFine
```

```
1566 DEFine PROCedure AS_Demo1
1567 INK#ch,4:LINE#ch,10,20 TO 110,20:LINE#ch,15,15 TO 15,90
1568 a=50:s=0:x=16:y=22:demo=2
1569 FOR n=2 TO 80 STEP 4
1570   o=INT(RND(1 TO 24)/RND(4 TO 8)):d=RND(5 TO 6)
1571   IF a>RND(8 TO 24):s=s+o*3:a=a-o
1572   IF s>28 AND RND(1 TO 24)=7:a=a+d:s=s-d
1573   IF n>40 AND RND(1 TO 3)=3:a=a+6
1574   INK#ch,2:LINE#ch,x+n,y TO x+n,y+a/2
1575   INK#ch,7:LINE#ch,x+n+2,y TO x+n+2,y+s/2
1576 END FOR n
1577 END DEFine
```



```
1579 DEFine PROCedure AS_Demo2
1580 FOR n=1 TO 40
1581  ASRep(n,1)=(50+RND(-12 TO 2)):ASRep(n,2)=(n*2+RND(-6 TO 18))
1582 END FOR n
1583 Ino=41:Invmax=40:ASReport
1584 Ino=1:Invmax=20:FOR n=1 TO 40:ASRep(n,1)=0:ASRep(n,2)=0
1585 END DEFine
```





Welcome to QBITS WareHouse
Check Bay Printouts for Invoice Requests and Deliveries
As Lorries arrive you Load (Invoice) or Unload (Delivery)
Use ←↑↓→ Cursors and ▬ Spacebar to move PickUP
For Pick or Drop use P & D keys
Use L for other Levels and S for New Stock Requests
Lorries Depart on Time irrespective of Load/Unload Status
As Invoices are fulfilled Sales Increase. Check PC Stock
as this shows the Assets currently held in the Warehouse
To Access Menu (New Load Save Exit) press M key
press any key to continue...

```
1587 DEFine PROCedure QBWH_Intro
1588 WINDOW#2,512,224,gx,gy    :PAPER#2,0:BORDER#2,1,3:CLS#2
1589 WINDOW#0,512, 32,gx,gy+224:PAPER#0,0:BORDER#0,1,3:CLS#0
1590 CSIZE#2,2,1:OVER#2,1:str$='Welcome to QBITS WareHouse'
1591 INK#2,2:FOR i=0 TO 1 :CURSOR#2,94+i,17:PRINT#2,str$
1592 INK#2,6:FOR i=0 TO 1 :CURSOR#2,96+i,18:PRINT#2,str$
1593 CSIZE#2,0,0:OVER#2,0 :INK#2,5:RESTORE 1596

1595 FOR i=1 TO 9:READ x,y,str$:CURSOR#2,x,y:PRINT#2,str$
1596 DATA  86, 44,'Check Bay Printouts for Invoice Requests and Deliveries'
1597 DATA  80, 56,'As Lorries Arrive you Load (Invoice) or UnLoad (Delivery)'
1598 DATA  82, 68,'Use Cursors     to Move PickUP and     Spacebar to Rotate'
1599 DATA 156, 80,'Use   for Pick and   for Drop'
1600 DATA  98, 92,'Use   for other Levels and   for New Stock Requests'
1601 DATA  82,104,'Lorries Depart on Time Irrespective of Load/UnLoad Status'
1602 DATA  82,116,'Check PC Stock as this shows the Assets currently held in'
1603 DATA  86,128,'the WareHouse. As Invoices are Fulfilled Sales Increase'
1604 DATA 106,146,'Press   Key to Access Menu [ ew  oad  ave  uit]'

1606 INK#2,7:FOR i=1 TO 9:READ x,y,str$:CURSOR#2,x+i,y:PRINT#2,str$
1607 DATA 180,80,'P',270,80,'D',122,92,'L',258,92,'S'
1608 DATA 138,146,'M',268,146,'N',292,146,'L',320,146,'S',348,146,'Q'
1609 CURSOR#2,160,176:PRINT#2,'press any key to continue...'
1610 INK#2,7:CURSOR#2,136,68:PRINT#2,'←↑↓ →':BLOCK#2,14,3,240,72,7
1611 WINDOW#1,80,40,gx+72,gy+172:ch=1:Init_PickUp:BLOCK#2,24,2,112,206,2
1612 WINDOW#1,100,60,gx+360,gy+160   :AS_Demo1   :PAUSE
1613 END DEFine
```

QBITS WareHouse Test Mode

To review QBITS Warehouse code Test Mode checks various Procedures, Invoices, Delivery, Printouts, Lorry movements, Store Clock, Stock Request, Game End and Hazards, Stock Loss, Computer Error and Sales credit deducted for Tax Revenues, Energy Bill payments and Stolen Lorry Stock. F1 key halts other actions of the main program to return press F1 key again after completing test selected.

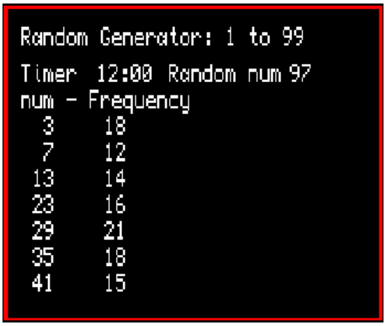
```
1615 DEFine PROCedure Test_Mode
1616 F1=1:ATemp=Asset:STemp=Sales:Sales=50
1617 REPeat Test
1618  ch=8:CLS#ch:PRINT#ch,'Test Mode':Score:tk=CODE(INKEY$(-1))
1619  SELect ON tk
1620   =49:bay1=1:Arrival 1                :REMark (1)Bay1_in Print & Lorry
1621   =50:Depart 1:bay1=0                 :REMark (2)Bay1_out Print & Lorry
1622   =51:bay2=1:Arrival 2                :REMark (3)Bay2_in Print & Lorry
1623   =52:Depart 2:bay2=0                 :REMark (4)Bay2_out Print & Lorry
1624   =53:Store_err                       :REMark (5)Compute Crash
1625   =54:Stock_loss  :PAUSE              :REMark (6)Assets Lost
1626   =55:Asset_Tax   :PAUSE              :REMark (7)Loss of Sales credits
1627   =56:Energy_Bill :PAUSE              :REMark (8)Loss of Sales credits
1628   =57:Stolen_Stock:PAUSE              :REMark (9)loss of Sales credits
1629   =97:Sales=Sales+6:Stock_aud         :REMark (a)Increase Sales
1630   =65:Sales=Sales-6:Stock_aud         :REMark (A)Decrease Sales
1631   =67:Tim=Tim+3600 :Store_Clk         :REMark (C)Clock 1hr increments
1632   =71:Gs=4:QBWH_Menu                  :REMark (G)Game End
1633   =83:Stock_Request                   :REMark (S s) Note:Sales +12
1634   =232:F1=0:CLS#8:Asset=ATemp:Sales=STemp:Score:EXIT Test
1635  END SELect
1636 END REPeat Test
1637 END DEFine
```

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 Bills 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 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
```



```
Random Generator: 1 to 99
Timer  12:00  Random num 97
num - Frequency
  3     18
  7     12
 13     14
 23     16
 29     21
 35     18
 41     15
```

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 deductions are made.