

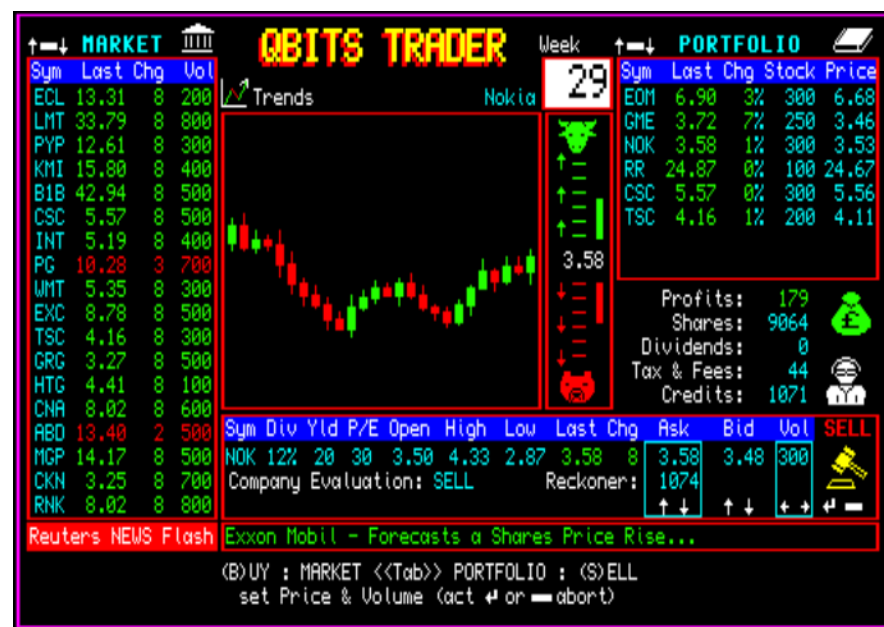
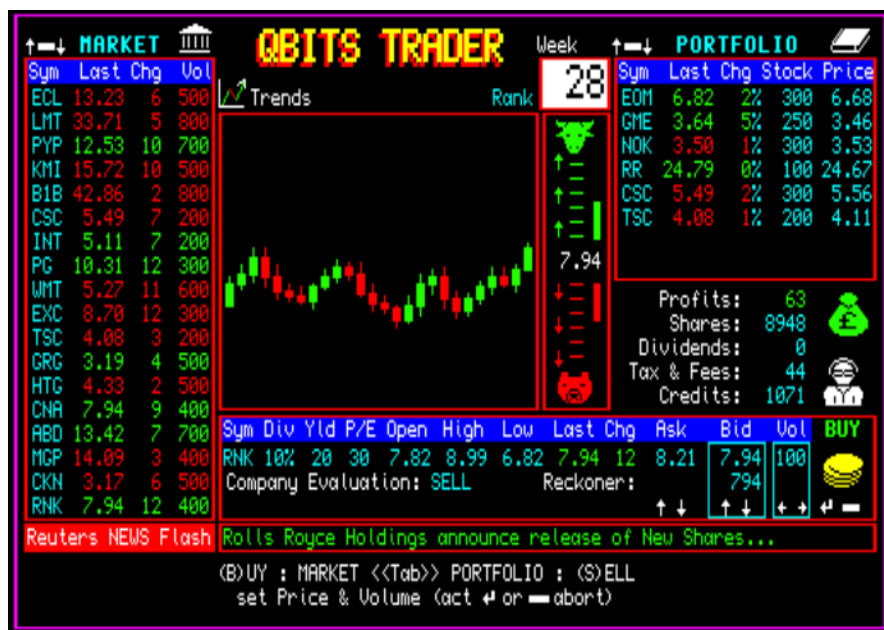


Sinclair QL Retro Gaming



Sinclair QL Retro Gaming







Introduction

Since the earliest times our ancestors have engaged in the art of bartering. As communities grew and met up with others Trading became an instinctive part of being human. Be it in goods or services it provided a means to furthering expectations and fulfilment of our dreams.

Back in the day I with a group of others attended a business course at London University on Company Finance. A short while after finishing the course we were invited back. I was intrigued to know why? There was the hit of being part of some experiment. Assigned to groups we played as a Company board of Directors, our task to invent marketable products, devise advertising and sales campaigns, manage factory production and deliver goods to imaginary customers. Today that may sound a bit like the theme for the Apprentice, the crux of it all was raising our start-up capital by a Shares issue. Success or failure was dictated by a Stock Market Simulation run over three days judging the progress of our companies for a pretend three-year period. As you would expect just to liven up the process were the dictates of the course lecturers, acting sometimes favourably to our requests and enquiries, but more often than not raising problems on our company stratagems, marketing directions and expectations.

QBITS Trader ‘Concept’

I guess those early memories became the thoughts for a QBITS Stock Market Game. It began with a bit of research reviewing setups of Stock Market Trader screens and investing my time looking at Board Games with a similar theme. I imagined two ways to play the Stock Market the long haul where your investments reward you through Company Dividend pay-outs and the Maverick method, riding the rise and fall of Share Prices judging when to Buy and Sell to accumulate your wealth. The second method riskier with the potential to fail dramatically.

As with most of QBITS coding the first steps were thoughts on layout, what elements would be incorporated and the best way to represent them on screen. To begin a Stock list of Companies with Share pricing and volumes on offer. Some start-up capital to buy a Portfolio of selected Stock. Then there was how to control the Buying and Selling and display Gains and Losses.

Design of a Game involves developing those parts that provide a repeated familiarity, and those random elements that will be different each time the Game is played. The common repeated elements help promote the development of stratagems to advance a player’s ability to do well. The random elements just make the Game a touch more worthwhile Challenge.

QBITS Trader ‘Challenge’

To manage a Stock Market Trader Desk and make as large as possible the return from an initial start-up fund. Initialised with Stocks and Share Price Trends, the Game is to build a Shares Portfolio, which is further adapted to meet the Price changes of a Simulated Stock Market. The time limit being set as a Three-Year period. Accumulative Dividend pay-outs and Profits from Buying and Selling to provide the End Game Score.

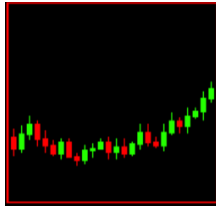
QBITS Trader 'Intro'

Select **Currency (D) (E) (P) (Y)** from the **Intro Screen**, any other key will Select '**£**' set as the **Default**.



The Trader Screen is displayed and **Market** entries Initialised. The Company Share **Trends** are then Calculated for the first half year (twenty-six weeks). The beginning Phase of the game is to build a **Portfolio** of Company Shares using the opening **10000 Credits**.

↑↓	MARKET	
Sym	Last Chg	Vol
R10	12.85	12 200
BA	22.87	6 300
U	32.92	8 200
EOM	6.28	7 500
CCL	2.83	3 200
T	17.29	5 800
DIS	6.13	9 800
F	4.14	9 400
GME	3.20	4 200
SPG	32.74	3 400
GE	18.84	9 300
JNJ	13.38	12 400



As the game progresses the **Bull & Bear** Status window show vertically extending bars giving notice of next **End of Week Share Price** changes.

Trends are shown using **CandleWick** Graphics.

Scroll the **MARKET** list of Companies with **Up/Down** cursor keys and select with **SpaceBar** to reveal Company **Information** and display Share **Trends** for the preceding twenty-six weeks.



QBITS Trader 'Company Info'

Sym is the abbreviation/index of a Company's Name.

Sym	Div	Yld	P/E	Open	High	Low	Last Chg	Ask	Bid	Vol
EOM	8%	20	30	6.35	6.93	5.89	6.28	7	6.41	6.15 500
Company Evaluation: BUY					Reckoner:					
					↑ ↓		↑ ↓		+ +	

Div (Dividend) is a declared % of Face Value (FV) of a Share based on annual Company profits. A range between 3% to 8% is considered healthy. Irrespective of Market value they are paid out to shareholders quarterly (every 13Wks).

Yld (Yield) is a rate of **return** calculated by subtracting the starting value of the investment from its final value and then dividing the figure by the starting value before multiplying by 100.

P/E (Price/Earnings) represents the Market value of Stock compared to the Company's earnings. It shows what the market is willing to pay based on past or forecasted future earnings. This is an indicator to the viability for future investment.

Open, High, Low, Last are used in analysing a changing **Share Price** and are represented by a **Candle** for the **Open to Last** or Closing price range and **Wicks** for the **High** and **Low** of price movement, the **Colour** reveals the direction. A **Bull** is shown in **Green** with **Red** for a **Bear**.



Review of patterns formed by the **CandleWick** graphics can indicate market opportunities - provide insight into the balance between **Buying** and **Selling** pressures, a steady continuation or market indecision.

The **Chg** (Change) is the difference between the previous and current weeks **Last Share Price**.

Ask & Bid values are calculated from the current weeks **Last Share Price**.

Vol is the Shares Volume being offered.

QBITS Trader 'Account'

The account details provide a visual status as to how well things are going. As the Game Progresses **Profits** will vary with Current **Share Price** but hopefully climb. The **Dividends** shown are added to the **Credits** and can be used to buy more Shares, potentially more expensive ones.



Profits:	614	
Shares:	10035	
Dividends:	474	
Tax & Fees:	51	
Credits:	1002	

QBITS Trader 'BUY'

Press '**B**' and use Cursor keys to change the presented '**Bid**' and '**Vol**' values. **Enter** actions your choice, **Spacebar** aborts. Your '**Bid**' may or may not be accepted. It will be rejected if lack of funds '**Credits**'



Last Chg	Ask	Bid	Vol	BUY
7.94	12	8.21	7.94	100
Reckoner:				794
	↑ ↓	↑ ↓	↑ ↓	← →

QBITS Trader 'Fees'

Each Transaction incurs a '**Fee**', payment of which is deducted from the **Credits**.

QBITS Trader 'SELL'

Once the **PORTFOLIO** has an entry or entries use <<Tab>> to switch between **Market** and **PortFolio** then Scroll selection with **Up/Dow** Cursor keys to highlight an entry. **Chg** is % difference between **PortFolio Price** shown and **Last Share Price**.



Sum	Last Chg	Stock Price
EOM	6.56	7% 300 6.09
GME	3.62	10% 500 3.28
NOK	3.83	5% 100 3.99
RR	24.90	1% 60 24.52
ECM	13.43	4% 100 12.91
OSC	5.87	2% 200 5.73

Press '**S**' and use Cursor keys to change the presented '**Ask**' and '**Vol**' values. **Enter** actions your choice, **Spacebar** aborts. Your '**Ask**' Price may or may not be accepted. If not try again.



Last Chg	Ask	Bid	Vol	SELL
3.58	8	3.58	3.48	300
Reckoner:				1074
	↑ ↓	↑ ↓	↑ ↓	← →

QBITS Trader 'Reuters NEWS Flash'

Once a **PortFolio** has been started as part of the **End of Week Share Price** a **Reuters NEWS Flash** is displayed. This will change one or more of the Company's whose Shares are held and the calculation of **Share Price** or **Dividends** posted.



QBITS Trader 'Dividends'

The first method of acquiring returns on Stock is to choose those that have **Dividends**. Of the Company Stocks held in the **PortFolio**, those that post a dividend make a pay-out every 13 weeks. To thwart the possibility of buying shares just before a dividend pay-out and reselling to make a quick profit, a Company's **Share Price** is reduced after a dividend pay-out.

QBITS Trader 'GAME END'

After 156 Weeks the **GAME ENDS** the **Profits** and **Dividends** are added together to identify **Total Assets**. The Performance Review shows bands to describe levels of achievement.

Press **N** for (N)ew Game or **Q** to (Q)uit.



GAME END	
Performance Review	
Total Assets:	0
8000 Plus	- Excellent
6000 Plus	- Very Good
4000 Plus	- Acceptable
2000 Plus	- Survivable
Much Less	- Bad Luck
(N)ew	(Q)uit

QBITS Trader 'Strings'

In some ways this took me back to my early days with QL SuperBASIC, printing character strings to screen. The CSIZE command gives scope to creating differing character sets and with the use of OVER can generate the appearance of **Bold** characters with a horizontal pixel offset. Extend this to offsets in the vertical and it can produce a **3D Style**. My **Title** code used with CSIZE 2,1 is adopted for program Headings.

Title is used for **Bold** headings, CSIZE **w, d** for size, **t** for the offset, ch Windows channel id, tx, ty as pixel coordinates, and col the character INK, str\$ to hold character string.

1373 Title 2,2,2,1,1,134,2,'QBITS TRADER' x-offset : Title 2,6,2,1,1,136,0,'QBITS TRADER' repeated with y-offset

```
1393 DEFine PROCEDURE Title(ch,col,w,d,t,tx,ty,str$)
1394 CSIZE#ch,w,d:OVER#ch,1:INK#ch,col
1395 FOR i=0 TO t:CURSOR#ch,tx+i,ty:PRINT#ch,str$
1396 CSIZE#ch,0,0:OVER#ch,0:INK#ch,7
1397 END DEFine
```

Title 2,5,1,0,2,32,4,'MARKET'

Title 2,5,1,0,1,382,4,'PORTFOLIO'

Using **Title** for **Bold** Characters with CSIZE 0,0 merges the characters. Re-CSIZE within program loops cause errors in following PRINT lines.

Title	TBold
BUY	BUY

```
1160 DEFine PROCEDURE TBold(ch,col,cs,cx,cy,str$)
1161 INK#ch,col:OVER#ch,1
1162 FOR a=1 TO LEN(str$)
1163   FOR b=0 TO cs:CURSOR#ch,cx+b+a*(6+cs),cy:PRINT#ch,str$(a)
1164 END FOR a:OVER#ch,0
1165 END DEFine
```

QBITS Trader 'Currency'

For Currency & Numbers you need to **Right Justify**, add a **decimal point (dp)** and apply **spaces (sp)**. QL SuperBASIC allows coercion of unsuitable data to a type that will allow a specified operation to proceed. This converts numeric character strings to floating point integers and vice versa. This allows the **num** variable in **DRJ** to accept an integer or numeric string.

```
1167 DEFine PROCEDURE DRJ(ch,cx,cy,dp,sp,num)
1168 str$=ABS(INT(num)):sl=LEN(str$)
1169 IF dp>0 AND sl>dp:str$=str$(1 TO sl-2)&'.'&str$(sl-1 TO sl)
1170 IF dp>0 AND sl=dp:str$='.'&str$
1171 IF dp>0 AND sl<dp:str$='.0'&str$
1172 CURSOR#ch,cx,cy:PRINT#ch,FILL$(' ',sp-LEN(str$))&str$
1173 END DEFine
```

Decimal Right Justified

Variuos outputs with/without decimal point

QBITS Trader 'Highlight'

This uses LINE and the Graphics coordinates system to draw a box. The SCALE is made the same as the Windows vertical pixel size.

GME	3.78	1%	100	3.80
NOK	4.02	4%	300	3.83
PFE	23.67	0%	100	23.65

```
1043 DEFine PROCEDURE HGL(ch,w,d,x,y)
1044 OVER#ch,-1:LINE#ch,x,y TO x,y+d TO x+w,y+d TO x+w,y TO x,y:OVER#ch,0
1045 END DEFine
```

HiGhLight

Ask	Bid	Vol
12.89	12.60	400
↑ ↓	5040	↑ ↓

This is also used to highlight the **Ask**, **Bid** and **Vol** areas
HGL 7,12,15,88,5,0 : HGL 7,12,15,102,0 : HGL 7,7,4,15,116,4,0

QBITS Trader 'Share Pricing'

The Game opening Initialises the Market entries setting Company Name, Symbol, Dividends, Yields, Price/Earnings. Trends are then Calculated for the first 26 weeks generating a range of changes for Prices and Volume of Shares Offered.

As **QBITS Trader** Game progresses and a **PortFolio** is built, continued changes are made to the **Share Price**, **Volume** and **Dividends**. Future releases may also include changes that reflect **Yield** and **Price/Earnings**.

Altering Stock Prices involves making changes to the integer that represents the Share value and is also used to Calculate the Graphic coordinates and builds for the **CandleWick** display. For each Week of Trading the display requires four values **Open (oy)**, **High (hy)**, **Low (ly)** and **Last (cy)** close 'y' coordinates. Open (**oy**) is based on the previous Weeks Last (**cy**) entry.

The array **Trend(sn,wn,1-4)** identifies the Company place in the Stock Market List (1 to 40, the week number (**wn**) and stores integers for ,1) **Volume** of Shares ,2) **High** ,3) **Low** ,4) **Last**.

Trend_Set(wn) generates the **End of Week Share Price** changes for all 40 **MARKET** entries.

```
1242 DEFine PROCedure Trend_Set(wn)
1243 CLS#0:CURSOR#0,60,10:PRINT#0,'Calculating Stock Trends':CLS#0,4
1244 IF pm>0:n=Asset(RND(1 TO pm),1):rc=RND(1 TO 8):PRINT#6,CO$(n),' :RNew$(rc)
1245 FOR a=1 TO 40
1246   CURSOR#0,200+a*6,10:PRINT#0,','
1247   Trend(a,wn,1)=100*RND(1 TO 8):ry=RND(2 TO 12):mf=RND(1 TO 2)
1248   IF wn>1:oy=Trend(a,wn-1,4):ELSE oy=84
1249   IF pm>0
1250     IF rc=1 AND a=n:mf=1:ry=12 :REMark Shares ↑
1251     IF rc=2 AND a=n:mf=2:ry=12 :REMark Shares ↓
1252     IF rc=3 AND a=n:Stock(n,2)=RND(8 TO 16) :REMark Div ↑
1253     IF rc=4 AND a=n:Stock(n,2)=RND(0 TO 4) :REMark Div ↓
1254     IF rc=5 AND a=n:mf=1:ry=6 :REMark Tax ↑
1255     IF rc=6 AND a=n:mf=2:ry=6 :REMark Tax ↓
1256     IF rc=7:mf=1:ry=8 :REMark Bull ↑
1257     IF rc=8:mf=2:ry=8 :REMark Bear ↓
1258   END IF
1259   IF wn MOD 13=0 AND Stock(a,2)>0:mf=2:ry=2+INT(Stock(a,2)/4)
1260   IF mf=1:cy=oy+ry:IF cy>148:cy=oy-RND(2 TO 6)
1261   IF mf=2:cy=oy-ry:IF cy< 20:cy=oy+RND(2 TO 6)
1262   IF cy>oy:hy=cy+RND(2 TO 7):ly=oy-RND(2 TO 7)
1263   IF cy<oy:hy=oy+RND(2 TO 7):ly=cy-RND(2 TO 7)
1264   Trend(a,wn,2)=hy:Trend(a,wn,3)=ly:Trend(a,wn,4)=cy
1265 END FOR a
1266 END DEFine
```

[Variables: **pm** portfolio max number of entries : **rc** row company : **ry** RND y : **mf** +/- math function]

Once the **PortFolio** holds one or more entries at the end of each **Trading Week** there is a selected **Reuters NEWS Flash** of Company held Shares. This adds further elements to the calculation of changes to the **Share Price** and/or **Dividend** % posted pay-outs.

Note: Further Share Price manipulation is not ruled out for Future QBITS Trader Versions.

QBITS Trader 'MARKET'

To create a Market for **QBITS Trader** each Company name, Symbol, and basic Information was obtained from reading Stock Market Lists. This is based on Sectors covering Energy suppliers, Consumer goods, Financial Institutes, Health Care, Industrial Manufactures, Information Technology, Mining/Material Production, Real Estate and Utilities.

QBITS Trader 'Stocks'

Information is held in four Arrays, **CN\$(40)** the Company Name, **Stock\$(40)** Company **Sym**
Stock(40,6) 1) unused, 2) **Div**, 3) **Yld**, 4) **P/E**, 5) **Price**, 6) **Chg**
Trend(40,156,3) 1) **Vol**, 2) **High**, 3) **Low**, 4) **Last**

```
1411 DEFine PROCedure Init_Stocks
1412 CURSOR#0,76,10:PRINT#0,'Initialising Market':CLS#0,4:RESTORE 1418
1413 FOR a=1 TO 40
1414 READ CN$(a),Stock$(a):FOR b=1 TO 6:READ Stock(a,b):END FOR b
1415 CURSOR#0,184+a*6,10:PRINT#0,':PAUSE 1
1416 END FOR a
```

Display of Market Stock

```
1194 DEFine PROCedure MStock(n,my) :REMark Market Stock
1195 INK#3,5:CURSOR#3,2,my:PRINT#3,Stock$(n) :REMark Stock Sym
1196 IF Trend(n,wn,4)>Trend(n,wn-1,4):INK#3,4:INK#7,4:ELSE INK#3,2:INK#7,2
1197 DRJ 3,26,my,2,5,Stock(n,5)+Trend(n,wn,4) :REMark Stock Last
1198 DRJ 3,60,my,0,3,Trend(n,wn,4)-Trend(n,wn-1,4) :REMark Stock Chg
1199 INK#7,5:DRJ 3,82,my,0,4,Trend(n,wn,1) :REMark Stock Vol
1200 END DEFine
```

The list has forty entries which are not all shown to screen at the same time so the need to scroll the list Up and Down within its own window space was required. Part of the main program loop includes actions following a key press (**k**) in this case the **Up/Down** cursors.

```
1014 k=CODE(INKEY$(50)):BLOCK#7,240,10,0,32,0
1018 =208:IF schk=0:mr=mr-1:sn=sn-1:M_Up:ELSE pr=pr-1:IF pr<1:pr=1
1019 =216:IF schk=0:mr=mr+1:sn=sn+1:M_Dn:ELSE pr=pr+1:IF pr>pm:pr=pm
```

HGL highlights the **Sym** of a Company entry and with help of **M_Up** and **M_dn** scrolls the other entries of the list to screen. Added was the **<Tab>** key option to switch between **MARKET** and **PORTFOLIO** lists.

```
1024 = 9:IF schk=0:schk=1:ELSE schk=0 [Tab Key]
1047 DEFine PROCedure M_Up
1048 IF mr<1 AND sn>0:SCROLL#3,10:MStock sn,0
1049 IF mr<1:mr=1
1050 IF sn<1:sn=1
1051 END DEFine
1053 DEFine PROCedure M_Dn
1054 IF mr>18 AND sn<41:SCROLL#3,-10:MStock sn,170
1055 IF mr>18:mr=18
1056 IF sn>40:sn=40
1057 END DEFine
```



Sym	Last	Chg	Vol
R10	12.85	12	200
BA	22.87	6	300
U	32.92	8	200
EOM	6.28	7	500
CCL	2.83	3	200
T	17.29	5	800
DIS	6.13	9	800
F	4.14	9	400
GME	3.20	4	200
SPG	32.74	3	400
GE	18.84	9	300
JNJ	13.38	12	400

QBITS Trader 'PORTFOLIO'


Up to eight (8) Portfolio entries can be held at any one time. Use, the <Tab> key to switch between **Market** and **PorFolio** and cursor keys to scroll Up/Down to Highlight and Select one of the Entries.

If you **BUY** Shares from the **Market** of a Company already in the **Portfolio** list these will be added to the **Stocks** held and the **Price** will be updated to the new purchase price. The one disadvantage to this is if the new price is lower than the previous it can lower future **Dividend** pay-outs.



Sym	Last	Chg	Stock	Price
EOM	6.56	7%	300	6.09
GME	3.62	10%	500	3.28
NOK	3.83	5%	100	3.99
RR	24.90	1%	60	24.52
ETL	13.43	4%	100	12.91
CSC	5.87	2%	200	5.73

```
1107 DEFine PROCEDURE S_Add(pr,snum)
1108 Asset(pr,1)=sn:Asset(pr,2)=INT(Asset(pr,2)+vol):Asset(pr,3)=cost
1109 sval=sval+snum:sfee=sfee+5+INT(vol/100):cash=cash-snum
1110 Trend(sn,wn,4)=cost-Stock(sn,5):PStock sn,pr:CStock sn
1111 END DEFine
```



Ask	Bid	Vol	BUY
6.31	6.10	200	
	1220		

If you **SELL** Shares from the Portfolio the chosen Volume will be deducted from the Stocks held. If you SELL all of the Stocks held the Company entry will be deleted.

```
1125 DEFine PROCEDURE S_Del(pr,snum)
1126 Asset(pr,2)=Asset(pr,2)-vol:Asset(pr,3)=cost
1127 sval=sval-snum:sfee=sfee+5+INT(vol/100):cash=cash+snum
1128 Trend(sn,wn,4)=cost-Stock(pn,5):PStock pn,pr:CStock sn
1129 END DEFine
```



Ask	Bid	Vol	SELL
4.20	3.93	200	
840			

Deleting a **PortFolio** entry requires a change to the **Asset** array held information and the screen row of characters to be removed. PROCEDURE **S_Sort** addresses this requirement by moving held information for the three possible **Asset** list positions, First row Middle rows and Last row. By calculating and setting the **CURSOR** position and using **SCROLL** -10 with option 2 the screen area below the cursor moves upward and clears the screen row/entry being deleted.

```
1131 DEFine PROCEDURE S_Sort(pr,pm)
1132 IF pr<pm
1133   FOR row=pr TO pm-1
1134     FOR c=1 TO 3:Asset(row,c)=Asset(row+1,c)
1135   END FOR row
1136 END IF
1137 FOR c=1 TO 3:Asset(pm,c)=0
1138 IF pr=1:SCROLL#5,-10
1139 IF pr>1:CURSOR#5,0,(pr-2)*10:SCROLL#5,-10,2
1140 pm=pm-1:IF pr>pm:pr=pm
1141 END DEFine
```

QBITS Trader 'Profits & Dividends'

Profits are the difference between the Price when Shares were bought and the Last Share Price and multiplied by the Stock Volumes held. Company Stocks held by the Portfolio who post a div% receive Dividends which are accumulative and pay-out every quarter (13weeks).

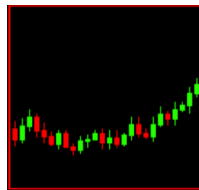


Profits:	614	
Shares:	10035	
Dividends:	474	
Tax & Fees:	51	
Credits:	1002	

QBITS Trader 'Stock UpDates'

The **END** of **WEEK** Triggers a new set of Trend Calculations and the following Updates to the Trader **Accounts**, **MARKET**, **PORTFOLIO**, Company **INFO** and **Trends** displays and a new **Reuters NEWS Flash**.

AStock	Account Info	[Variables sval sdv sfee sinv cash]
PStock n, pm	Portfolio Info	[Variables wn py]
MStock n, my	Market Info	[Variables wn]
CStock n	Company Info	[Variables str\$ ops ols nws wk1 wk2]
STrend n en	CandleWicks	[Variables n en]



Reuters NEWS Flash Nokia say Tax Breaks increased Share value...

Sym	Div	Yld	P/E	Open	High	Low	Last	Chg	Ask	Bid	Vol
EOM	8%	20	30	6.35	6.93	5.89	6.28	7	6.41	6.15	500
Company Evaluation: BUY							Reckoner:				

↑ ↓ ↑ ↓ + + ← =

QBITS Trader 'Trend Analysis'

The Buying and Selling Price of Shares varies over a period of time and depends on a number of factors. This essentially comes down to marketable goods or services and maintaining Market Share over other competitors.



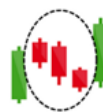
The volume of turnover can indicate the stability of a company to weather any sudden market depressions. It may also be due to trying to maintain Market Share leading to falling profits. It may depend on materials or parts supplied from other suppliers. Other disrupting factors can be the weather, industrial disputes, interest rate changes, government taxes, new restrictions on certain materials or products, border controls, conflicts etc.

QBITS Trader 'Trading Strategy'

This is based on summarising the Trends with movement displayed as Candle & Wicks. Small **CandleWicks** indicate little price movement and represent consolidation. Large **CandleWicks** show strong buying or selling pressure. These identify the highs and lows of the Trading Share Price in **Bull** and **Bear** Markets.



The Time to **Buy**: a **GREEN** **CandleWick** showing a lower opening price rising to a high and falling back to close above the opening price. The time to Sell a **RED** **CandleWick** with wick pointing upwards indicates an opening price rising and then falling to a lower closing price.



Trading Volumes, Price Movements, should be viewed with a Company's strengths and weaknesses, derived from the Dividends, Yields and Price/Earnings.

At the end of the three years (**156 weeks**) the **Game Ends**. Results will reveal if you have made any significant gains. This will be the combination of Dividends and Profits. For the best rewards divide Stock into those with high Dividend pay-outs and those that add Quick Profits by Buying and Selling at the right time.

QBITS Trader 'Summary'

Name	Company Name	
Sym	Symbol	1-3 Capital Letters identifying the Company
Iss	Issue	Not Used in V2: IPO Initial Public Shares Offer
Div	Dividend	Yearly return as a % paid on each share.
Yld	Yeild	Annual Dividend divided by current stock price
P/E	Price/Earnings	Ratio between stock Price and Company's Earning
Open	Price	Opening Share Price
High	Price	Trade High between Open - Close
Low	Price	Trade Low between Open - Close
Last	Last Price	Stock price at end of Trading Week
Chg	Net Change	Change between previous Last and current Last.
Ask	Price	Share Price Requested
Bid	Price	Share Price Offered
Vol	Volume	The number of Shares Offered for Exchange.

QBITS Trader 'Reuter NEWS Flash'

Announcements with potential to affect Share Price or Dividends.

BUYing Company Shares:

- Check 1 If Insufficient Funds or Bid to low Purchase Rejected.
- Check 2 If Company Already in Portfolio Add Shares to existing.
- Check 3 If Company Not already in Portfolio Add New Entry to next free slot

SELLing Company shares:

- Check 1 If Company shares sold and Stock reduced to zero Delete slot.

Stock Trends:

This monitors the Stock Share movement over a period of time, displays 26 Weeks.

QBITS Trader 'Challenge:'

Starting with Credits of 10000 - Buy/Sell Shares to make Gains either through Dividend payments and/or by taking advantage of Share Price Rises.

NOTE: If you need a break! Activating the **BUY** or **SELL** suspends **End of Week** Timer.
In effect pausing the program, returning will likely jump the display to the next End of Week.

QBITS Trader 'Development'

Version 1: Screen layouts, Graphics and scrolling Stock entries

Version 2: Trend Calculations, CandleWicks, Reuters NEWS : Accompanying pdf

QBITS Trader 'Future Enhancements'

Maybe extend Game End to include a League Table and Save & Load functions.
Include Yields and Price/Earnings as part of Trend Calculations.

QBITS Trader PROCedures

Intro	Opening Introduction to QBTS Trader and Selection of Currency
Init_Trader	Opens windows and sets Trader screen layout
Title	Presents BOLD and 3D Character strings
Init_RNews	The NEWS statements used in the Game
Init_Stocks	Set up the Games Stock Market
QBITS_Trader	Main Program Loop
THelp	Help on Key usage
HGL	HiGhL ights to Aid Game Play
M_Up	Market Scroll Up
M_Dn	Market Scroll Down
SMWeek	Print Stock Market Week Number
Game_End	Review of Assets & Performance
S_Buy	BUY Shares
S_Add	Add Shares to Portfolio
S_Sell	SELL Shares
S_Del	Delete Shares in Portfolio
S_Sort	Sort Portfolio List
S_Vol	Display Change Ask Bid, Volume and manual changes
TBold	Display Bold Charters
DRJ	Draw Right Justified + decimal point
AStock	Display Account Changes
PStock	Display PortFolio Stock entries
MStock	Display Market Stock entries
CStock	Display Company Info of Stock selected
STrend	Display 26 weeks of Price Changes of Company selected
Wick	Display Price changes in Candle Wick format
Trend_Set	Calculate Week changes for the Market Stock
Graphics	
TBuy	Display Buy icon and heading
TSell	Display SELL icon and heading
Market	Market icon
Asset	Asset icon
Chart	Simple Bull/Bear Chart
Money	Coins Display
Trader	Head & shoulders of Trader with headphones
Bear	Bear view full face
Bull	Bull view full face
Pillar	Display Fluted Column

1000 REMark QBTrader_v2 (QBITS Trader v2 2021) QPC2 & BBQL

1002 gx=20:gy=30:Tdel=30 :REMark **BBQL** gx=0 gy=0 ; Tdel Time delay
 1003 DIM Asset(8,4),Stock(40,6),Trend(40,156,4)
 1004 DIM Cn\$(40,20),Stock\$(40,3),RNew\$(8,40)
 1005 Intro:Init_Trader:Init_RNews:Init_Stocks:QBITS_Trader

1007 DEFINE PROCEDURE QBITS_Trader
 1008 sval=0:sdiv=0:sfee=0:sinv=0:cash=10000:DRJ 8,0,0,0,3,26
 1009 n=1:sn=1:mr=1:pn=1:pr=1:pm=0:schk=0: GTm=DATE+Tdel:AStock:THelp
 1010 REPEAT Loop
 1011 SMWeekK:IF Asset(pr,1)=0:schk=0
 1012 IF schk=0:INK#3,7:HGL 3,16,9,0,181-mr*10:n=sn
 1013 IF schk=1:INK#5,7:HGL 5,16,9,0, 81-pr*10:n=Asset(pr,1)
 1014 k=CODE(INKEY\$(50)):BLOCK#7,240,10,0,32,0
 1015 IF schk=0:HGL 3,16,9,0,181-mr*10
 1016 IF schk=1:HGL 5,16,9,0, 81-pr*10
 1017 SELECT ON k
 1018 =208:IF schk=0:mr=mr-1:sn=sn -1:M_Up:ELSE pr=pr-1:IF pr<1:pr=1
 1019 =216:IF schk=0:mr=mr+1:sn=sn+1:M_Dn:ELSE pr=pr+1:IF pr>pm:pr=pm
 1020 =66, 98:IF schk=0:S_Buy sn:AStock:THelp :REMark (B)UY
 1021 =83,115:IF schk=1:S_Sell pr:AStock:THelp :REMark (S)ELL
 1022 =81,113:EXIT Loop :REMark (Q)uit
 1023 =78,110:IF wn>=156:Stock_Update 25 :REMark (N)ew
 1024 = 32:CStock n:STrend n,wn :REMark View Info / Trends
 1025 = 9:IF schk=0:schk=1:ELSE schk=0

1026 to 1030 lines used for Testing Phase (see page 20)

1031 END SELECT
 1032 END REPEAT Loop
 1033 CLS#9:FOR i=3 TO 9:CLOSE#i
 1034 INK#2,7:INK#0,7:CLS#0:PRINT#0,'Bye...'
 1035 END DEFINE

1037 DEFINE PROCEDURE THelp
 1038 CLS#0:CUSOR#0,112, 4:PRINT#0,'(B)UY : MARKET <<Tab>> PORTFOLIO : (S)ELL'
 1039 CUSOR#0,124,14:PRINT#0,'set Price & Volume (act ¼ or abort)'
 1040 BLOCK#0,2,4,274,16,7:BLOCK#0,12,3,296,18,7
 1041 END DEFINE

1043 DEFINE PROCEDURE HGL(ch,w,d,x,y) :REMark HiGhLight
 1044 OVER#ch,-1:LINE#ch,x,y TO x,y+d TO x+w,y+d TO x+w,y TO x,y:OVER#ch,0
 1045 END DEFINE

1047 DEFINE PROCEDURE M_Up
 1048 IF mr<1 AND sn>0:SCROLL#3,10:MStock sn,0
 1049 IF mr<1:mr=1
 1050 IF sn<1:sn=1
 1051 END DEFINE

1053 DEFINE PROCEDURE M_Dn
 1054 IF mr>18 AND sn<41:SCROLL#3,-10:MStock sn,170
 1055 IF mr>18:mr=18
 1056 IF sn>40:sn=40
 1057 END DEFINE

MARKET			
Sum	Last	Chg	Vol
RIO	12.44	3	100
BA	23.40	12	300
U	32.90	4	200
EDM	6.68	4	700
CCL	3.70	3	300
T	17.20	5	400
DIS	5.20	5	300
F	4.78	3	600
CHE	2.82	5	500
SPG	32.26	12	300
GE	18.99	4	300
JNJ	13.26	9	800
NOK	3.58	5	300
KD	6.09	4	800
NEE	23.04	4	800
WFC	4.59	5	600
RR	24.23	4	800

PORTFOLIO			
Sum	Last	Chg	Stock Price
EDM	7.01	1%	200 7.04
CHE	3.78	1%	100 3.88
DIS	4.82	4%	300 3.83
PFE	23.67	8%	100 23.65
RR	24.64	8%	100 24.57
TSC	5.22	2%	200 5.07
CSC	6.30	3%	200 6.18

```

1059 DEFine PROCEDURE SMWeek
1060 del=GTm-DATE:IF del<0:del=0
1061 BLOCK#4,4,42-del,28,10+del,4:BLOCK#4,4,42-del,28,70,2
1062 IF GTm<DATE:BLOCK#4,4,48,28,4,0:BLOCK#4,4,48,28,68,0
1063 IF GTm<DATE:Stock_Update wn:GTm=DATE+Tdel
1064 END DEFine

```

```

1066 DEFine PROCEDURE Stock_Update(wn)
1067 IF wn<156:wn=wn+1:ELSE Game_End:RETurn
1068 Trend_Set wn:CStock n:STrend n,wn:sval=0:snum=0:dnum=0
1069 FOR rs=18 TO 1 STEP -1 :MStock (sn-mr)+rs,(rs*10)-10
1070 FOR rp=1 TO pm
1071   m=Asset(rp,1):pdiv=Stock(Asset(rp,1),2)
1072   snum=INT(Asset(rp,2)*(Stock(m,5)+Trend(rn,wn,4))/100)
1073   IF wn MOD 13=0 AND pdiv>0
1074     dnum=INT(snum*pdiv/400):sdiv=sdiv+dnum:cash=cash+dnum
1075   END IF
1076   sval=sval+snum:PStock Asset(rp,1),rp
1077 END FOR rp
1078 AStock:PAUSE 10:CLS#0:THelp
1079 END DEFine

```



Weeks Passed

↑
Bars Extend
Up/Down
End of Week



```

1081 DEFine PROCEDURE Game_End
1082 CLS#1:Title 1,7,2,1,1,42,6,'GAME END'
1083 RESTORE 1084:FOR i=1 TO 9:READ x,y,str$:CURSOR#1,x,y:PRINT#1,str$
1084 DATA 36,28,'Performance Review',30,40,'Total Assets:'
1085 DATA 30,56,'8000 Plus - Excellent',30,66,'6000 Plus - Very Good'
1086 DATA 30,66,'6000 Plus - Very Good',30,76,'4000 Plus - Acceptable'
1087 DATA 30,86,'2000 Plus - Survivable',30,96,'Much Less - Bad Luck'
1088 DATA 48,110,(')ew ( )uite'
1089 TBold 1,7,1,46,110,'N':TBold 1,7,1,100,110,'Q'
1090 DRJ 1,120,42,0,5,sinv+sdiv-sfee
1091 END DEFine

```



```

1093 DEFine PROCEDURE S_Buy(sn)
1094 TBuy 7,131,11:TBold 7,4,346,0,'BUY':CStock sn:STrend sn,wn
1095 vn=Trend(sn,wn,1):vm=vn:cn=Stock(sn,5)+Trend(sn,wn,4)
1096 S_Vol 102.5,286,2,5,vn,vm,cn:IF k=32:RETurn
1097 IF cost<Stock(sn,5)+Trend(sn,wn,4)-RND(6 TO 12)
1098   CURSOR#7,2,32:PRINT #7,'Unsuccessful Bid - Try Again':RETurn
1099 END IF
1100 snum=INT(vol*cost)/100 :REMark cost=Bid
1101 IF snum+20>cash:CURSOR#7,2,32:PRINT#7,'Insufficient Funds':RETurn
1102 FOR pr=1 TO pm:IF Asset(pr,1)=sn:S_Add pr,snum:RETurn
1103 IF pm<8:pm=pm+1:S_Add pm,snum:RETurn
1104 CURSOR#7,2,32:PRINT#7,'Sell Some Shares'
1105 END DEFine

```



```

1107 DEFine PROCEDURE S_Add(pr,snum)
1108 Asset(pr,1)=sn:Asset(pr,2)=INT(Asset(pr,2)+vol):Asset(pr,3)=cost
1109 sval=sval+snum:sfee=sfee+5+INT(vol/100):cash=cash-snum
1110 Trend(sn,wn,4)=cost-Stock(sn,5):PStock sn,pr:CStock sn
1111 END DEFine

```

```

1113 DEFine PROCEDURE S_Sell(pr)
1114 pn=Asset(pr,1):IF pn<=0:RETurn
1115 TSell 7,133,10:TBold 7,2,344,0,'SELL'
1116 STrend pn,wn:CStock pn:vm=Asset(pr,2):IF vm>900:vm=900
1117 vn=vm:S_Vol 88.5,248,2,5,vn,vm,Stock(pn,5)+Trend(pn,wn,4):IF k=32:RETurn
1118 IF cost>Stock(pn,5)+Trend(pn,wn,4)+RND(0 TO 12)
1119 CURSOR#7,0,32:PRINT#7,'Unsuccesful Sale - Try Again':RETurn
1120 END IF
1121 snum=INT(vol*cost)/100 :REMark cost=Ask
1122 S_Del pr,snun:IF Asset(pr,2)=0:S_Sort pr,pm
1123 END DEFine

```

Ask	Bid	Vol	SELL
4.20	3.93	200	
840			
↑ ↓	↑ ↓	← →	← →

```

1125 DEFine PROCEDURE S_Del(pr,snun)
1126 Asset(pr,2)=Asset(pr,2)-vol:Asset(pr,3)=cost
1127 sval=sval-snum:sfee=sfee+5+INT(vol/100):cash=cash+snum
1128 Trend(sn,wn,4)=cost-Stock(pn,5):PStock pn,pr:CStock sn
1129 END DEFine

```

```

1131 DEFine PROCEDURE S_Sort(pr,pm)
1132 IF pr<pm
1133 FOR row=pr TO pm-1
1134 FOR c=1 TO 3:Asset(row,c)=Asset(row+1,c)
1135 END FOR row
1136 END IF
1137 FOR c=1 TO 3:Asset(pm,c)=0
1138 IF pr=1:SCROLL#5,-10
1139 IF pr>1:CURSOR#5,0,(pr-2)*10:SCROLL#5,-10,2
1140 pm=pm-1:IF pr>pm:pr=pm
1141 END DEFine

```

```

1143 DEFine PROCEDURE S_Vol(x,cx,dp,sp,vn,vm,cn)
1144 INK#7,5:HGL 7,12,15,x,0:HGL 7,7,4,15,116.4,0
1145 REPEAT Vol_Ip
1146 DRJ 7,324,12,0,3,vn:DRJ 7,cx,12,2,5,cn
1147 DRJ 7,cx,22,0,5,INT((cn*vn)/100)
1148 k=CODE(INKEY$(-1))
1149 IF k=192:vn=vn-10:IF vn< 10:vn= 10
1150 IF k=200:vn=vn+10:IF vn> vm:vn= vm
1151 IF k=208:cn=cn+1 :IF cn>9990:cn=9990
1152 IF k=216:cn=cn-1 :IF cn< 1:cn= 1
1153 IF k= 32:EXIT Vol_Ip
1154 IF k= 10:vol=vn:cost=cn:EXIT Vol_Ip
1155 END REPEAT Vol_Ip
1156 INK#7,5:HGL 7,12,15,x,0:HGL 7,7,4,15,116.4,0
1157 BLOCK#7,30,12,cx,22,0:BLOCK#7,34,32,348,0,0
1158 END DEFine

```

Last Chg	Ask	Bid	Vol	BUY
6.10	6.31	6.10	200	
Reckoner:		1220		
↑ ↓	↑ ↓	↑ ↓	← →	← →

:REMark Volume of Shares

:REMark Ask/Bid Price

Last Chg	Ask	Bid	Vol	SELL
4.10	4.20	3.93	200	
Reckoner:	840			
↑ ↓	↑ ↓	↑ ↓	← →	← →

```

1160 DEFine PROCEDURE TBold(ch,col,cs,cx,cy,ctr$)
1161 INK#ch,col:OVER#ch,1
1162 FOR a=1 TO LEN(str$)
1163 FOR b=0 TO cs:CURSOR#ch,cx+b+a*(6+cs),cy:PRINT#ch,str$(a)
1164 END FOR a:OVER#ch,0
1165 END DEFine

```



```

1167 DEFINE PROCEDURE DRJ(ch,cx,cy,dp,sp,num) :REMark Decimal Right Justified
1168 str$=ABS(INT(num)):sl=LEN(str$)
1169 IF dp>0 AND sl>dp:str$=str$(1 TO sl-2)&'&str$(sl-1 TO sl)
1170 IF dp>0 AND sl=dp:str$='&str$
1171 IF dp>0 AND sl<dp:str$=!'0&str$
1172 CURSOR#ch,cx,cy:PRINT#ch,FILL$(' ',sp-LEN(str$))&str$
1173 END DEFINE

```

Variuos outputs with/without decimal point

22.53 120% 1200 10.24

```

1175 DEFINE PROCEDURE AStock :REMark Account Stock Info
1176 DRJ 2,428,120,0,5,sval :REMark Share Value 1-8 Int((stock*last)/100)
1177 DRJ 2,428,130,0,5,sdiv :REMark WK13 Dividend 1-8 Int((sval*sdiv)/400)
1178 DRJ 2,428,140,0,5,sfee :REMark Fees each Transaction vol/100
1179 DRJ 2,428,150,0,5,cash+sdiv-sfee :REMark Cash -Buys-fees+Sales+Dividends
1180 sinv=(cash+sval)-10000:IF sinv<0:INK#2,2:ELSE INK#2,4
1181 DRJ 2,428,110,0,5,sinv :INK#2,5 :REMark Profit
1182 END DEFINE

```

```

Profits: 128
Shares: 10121
Dividends: 0
Tax & Fees: 47
Credits: 40

```

```

1184 DEFINE PROCEDURE PStock(n,pr) :REMark Portfolio Stock Info
1185 INK#5,5:py=(pr-1)*10:CURSOR#5,2,py:PRINT#5,Stock$(n) :REMark Stock Sym
1186 sChg=Stock(n,5)+Trend(n,wn,4)-Asset(pr,3):CURSOR#5,78,py:PRINT#5,'% '
1187 IF sChg<0:INK#5,2:ELSE INK#5,4
1188 DRJ 5,26,py,2,5,Stock(n,5)+Trend(n,wn,4) :REMark Last
1189 DRJ 5,60,py,0,3,(sChg/Asset(pr,3))*100:INK#5,5 :REMark Chg
1190 DRJ 5,90,py,0,4,Asset(pr,2) :REMark Stock
1191 DRJ 5,119,py,2,5,Asset(pr,3) :REMark Price
1192 END DEFINE

```

PORTFOLIO

Sym	Last	Chg	Stock	Price
EOM	7.01	1%	200	7.04
GME	3.78	1%	100	3.80
INX	4.02	4%	300	3.83
PFE	23.67	0%	100	23.65
RR	24.64	0%	100	24.57
TSC	5.22	2%	200	5.07
CSC	6.30	3%	200	6.10

```

1194 DEFINE PROCEDURE MStock(n,my) :REMark Market Stock Info
1195 INK#3,5:CURSOR#3,2,my:PRINT#3,Stock$(n) :REMark Stock Sym
1196 IF Trend(n,wn,4)>Trend(n,wn-1,4):INK#3,4:INK#7,4:ELSE INK#3,2:INK#7,2
1197 DRJ 3,26,my,2,5,Stock(n,5)+Trend(n,wn,4) :REMark Stock Last
1198 DRJ 3,60,my,0,3,Trend(n,wn,4)-Trend(n,wn-1,4) :REMark Stock Chg
1199 INK#7,5:DRJ 3,82,my,0,4,Trend(n,wn,1) :REMark Stock Vol
1200 END DEFINE

```

MARKET

Sym	Last	Chg	Vol
RIO	12.85	12	200
BA	22.87	6	300
V	32.92	8	200
EOM	6.28	7	500

```

1202 DEFINE PROCEDURE CStock(n) :REMark Company Stock Info
1203 INK#7,5:CURSOR#7,0,12:PRINT#7,Stock$(n),' %' :REMark Sym
1204 REMark DRJ 7, 24,10,0,4,Stock(n,1) :REMark Unused
1205 DRJ 7, 24,12,0,2,Stock(n,2) :REMark Dividend
1206 DRJ 7, 52,12,0,2,Stock(n,3) :REMark Yield
1207 DRJ 7, 74,12,0,2,Stock(n,4) :REMark Price/Expense
1208 ops=Stock(n,5):ols=Trend(n,wn-1,4):nws=Trend(n,wn,4)
1209 DRJ 7,92,12,2,5,ops+ols :REMark Open
1210 DRJ 7,125,12,2,5,ops+ols+Trend(n,wn,2) :REMark High
1211 DRJ 7,158,12,2,5,ops-(Trend(n,wn,3)-ols) :REMark Low
1212 IF nws>ols:INK#3,4:INK#7,4:ELSE INK#3,2:INK#7,2
1213 DRJ 7,191,12,2,5,ops+nws :REMark Last
1214 DRJ 7,224,12,0,3,nws-ols :INK#7,5 :REMark Change
1215 DRJ 7,324,12,0,3,Trend(n,wn,1) :REMark Volume
1216 DRJ 7,286,12,2,5,ops+nws-RND(10 TO 20) :REMark Bid
1217 DRJ 7,248,12,2,5,ops+nws+RND(10 TO 40) :REMark Ask
1218 wk1=Trend(n,wn-13,4)*Stock(n,4)*Stock(n,3)/100
1219 wk2=Trend(n,wn,4)*Stock(n,4)*Stock(n,3)/100
1220 IF wk1>wk2:str$='BUY 'ELSE str$='SELL'
1221 CURSOR#7,122,22:PRINT#7,str$
1222 END DEFINE

```

Sym	Div	Yld	P/E	Open	High	Low	Last	Chg	Ask	Bid	Vol
EOM	8%	20	30	6.35	6.93	5.89	6.28	7	6.41	6.15	500

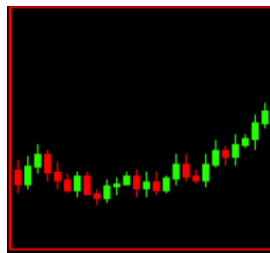
Company Evaluation: BUY Reckoner:

```

1224 DEFINE PROCEDURE STrend(n,en)
1225 CURSOR#2,178,26:PRINT#2,FILL$(' ',20-LEN(CN$(n)))&CN$(n)
1226 INK#4,7:DRJ 4,2,56,2,5,Stock(n,5)+Trend(n,en,4)
1227 DRJ 8,0,0,3,en:CLS:INK#4,7:cw=1:CLS
1228 FOR i=en-25 TO en
1229 IF i>1:oy=Trend(n,i-1,4):ELSE oy=84
1230 hy=Trend(n,i,2):ly=Trend(n,i,3):cy=Trend(n,i,4)
1231 IF cy>oy:col=4:y1=cy:y2=oy:ELSE col=2:y1=oy:y2=cy
1232 Wick 1,col,179,hy,ly,y1,y2
1233 END FOR i
1234 IF y1>84:SCROLL#1,(y1-84)*.6:ELSE SCROLL#1,-(84-y1)*.6
1235 END DEFINE

```

:REMark Stock Trend



```

1237 DEFINE PROCEDURE Wick(ch,col,wx,hy,ly,y1,y2)
1238 PAN-7:x1=wx-2:x2=wx+2:INK#ch,col:LINE#ch,wx,hy TO wx,ly
1239 FILL#ch,1:LINE#ch,x1,y1 TO x2,y1 TO x2,y2 TO x1,y2 TO x1,y1:FILL#ch,0
1240 END DEFINE

```



Variables: n - stock num : en - end week : ch - channel : col - colour : wx hy ly y1 y2 Graphic coordinates

Note: To simulate a Shares **MARKET**, Random numbers are used to generate the rise and fall of prices to all Company Shares. For those Company shares that make up the Players **Portfolio** an additional eight changes can take place. A Random choice can apply a **Bear** or **Bull** share price to an individual entry (1-2) or all entries (7-8). A selected entry can have the Company **Dividends** rise or fall reducing or increasing Quarterly (week 13) pay-outs. For choice (5-6) again share sales can be increased or decreased due to Company **TAX** payments. These directed choices are the result of a Random selected **Reuters NEWS Flash**.

```

1242 DEFINE PROCEDURE Trend_Set(wn)
1243 CLS#0:CURSOR#0,60,10:PRINT#0,'Calculating Stock Trends':CLS#0,4
1244 IF pm>0:n=Asset(RND(1 TO pm),1):rk=RND(1 TO 8):PRINT#6,CN$(n),' :RNew$(rk)
1245 FOR a=1 TO 40
1246 CURSOR#0,200+a*6,10:PRINT#0,','
1247 Trend(a,wn,1)=100*RND(1 TO 8):ry=RND(2 TO 12):mf=RND(1 TO 2)
1248 IF wn>1:oy=Trend(a,wn-1,4):ELSE oy=84
1249 IF pm>0
1250 IF rk=1 AND a=n:mf=1:ry=12 :REMark Shares ↑
1251 IF rk=2 AND a=n:mf=2:ry=12 :REMark Shares ↓
1252 IF rk=3 AND a=n:Stock(n,2)=RND(8 TO 16) :REMark Div ↑
1253 IF rk=4 AND a=n:Stock(n,2)=RND(0 TO 4) :REMark Div ↓
1254 IF rk=5 AND a=n:mf=1:ry=6 :REMark Tax ↑
1255 IF rk=6 AND a=n:mf=2:ry=6 :REMark Tax ↓
1256 IF rk=7:mf=1:ry=8 :REMark Bull ↑
1257 IF rk=8:mf=2:ry=8 :REMark Bear ↓
1258 END IF
1259 IF wn MOD 13=0 AND Stock(a,2)>0:mf=2:ry=2+INT(Stock(a,2)/4)
1260 IF mf=1:cy=oy+ry:IF cy>148:cy=oy-RND(2 TO 6)
1261 IF mf=2:cy=oy-ry:IF cy< 20:cy=oy+RND(2 TO 6)
1262 IF cy>oy:hy=cy+RND(2 TO 7):ly=oy-RND(2 TO 7)
1263 IF cy<oy:hy=oy+RND(2 TO 7):ly=cy-RND(2 TO 7)
1264 Trend(a,wn,2)=hy:Trend(a,wn,3)=ly:Trend(a,wn,4)=cy
1265 END FOR a
1266 END DEFINE

```

Variables: rk - Reuters key : ry - RND y offset : mf - maths function +/-
wn - week num : oy - open : hy - high : ly - low : cy - close (Last)

1268 REMark Trader Graphics

1270 DEFine PROCEDURE TBuy(ch,x,y)

```
1271 INK#ch,6:FILL#ch,1:ARC#ch,x-3.5,y TO x+4,y,-PI/2
1272 LINE#ch TO x+3.4,y-3:ARC#ch TO x-4,y-3,-PI/2:LINE#ch,x-4,y-3 TO x-4,y
1273 FILL#ch,0:INK#ch,0:ARC#ch,x-3,y-.6 TO x+4,y-.6,PI/2
1274 ARC#ch,x-4,y-1.6 TO x+3.6,y-1.6,PI/2:ARC#ch,x-4,y-2.5 TO x+3.6,y-2.5,PI/2
1275 END DEFine
```



1277 DEFine PROCEDURE TSell(ch,x,y)

```
1278 INK#ch,6:LINE#ch,x-5,y-2.5 TO x-1,y-2.5 TO x,y-4 TO x-6,y-4 TO x-5,y-2.5
1279 FILL#ch,1:LINE#ch,x-5,y TO x-2,y+4 TO x,y+2.8 TO x-3,y-1 TO x-5,y
1280 FILL#ch,0:LINE#ch,x-2,y+1.8 TO x+2.4,y-1 TO x+1.8,y-2 TO x-2,y+.6
1281 INK#ch,0:LINE#ch,x-5,y+1 TO x-2,y-.8:LINE#ch,x-3,y+3.6 TO x,y+1.8
1282 END DEFine
```



1284 DEFine PROCEDURE Market(ch,x,y)

```
1285 INK#ch,7:FILL#ch,1
1286 LINE#ch,x-3,y-1.5 TO x,y TO x+3,y-1.5 TO x-3,y-1.5:FILL#ch,0
1287 LINE#ch,x-1.3,y-2.2 TO x-1.3,y-4.6:LINE#ch,x-2.5,y-2.2 TO x-2.5,y-4.6
1288 LINE#ch,x+1.3,y-2.2 TO x+1.3,y-4.6:LINE#ch,x+2.5,y-2.2 TO x+2.5,y-4.6
1289 LINE#ch,x,y-2.2 TO x,y-4.6 :LINE#ch,x-3,y-5 TO x+3,y-5
1290 END DEFine
```



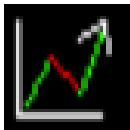
1292 DEFine PROCEDURE Assets(ch,x,y)

```
1293 INK#ch,7:FILL#ch,1:LINE#ch,x+3,y-1 TO x-1.8,y-1
1294 LINE#ch TO x-3.8,y-4 TO x+1,y-4 TO x+3.2,y-1:FILL#ch,0
1295 LINE#ch,x-2.5,y-1 TO x-4.5,y-4:LINE#ch,x+1.2,y-5 TO x+3.5,y-2
1296 ARC#ch,x-4,y-4 TO x-4,y-5,PI:LINE#ch TO x+1,y-5:ARC#ch TO x+1,y-4,-PI
1297 END DEFine
```



1299 DEFine PROCEDURE Chart(ch,x,y)

```
1300 INK#ch,7:LINE#ch,x,y TO x,y-5 TO x+5,y-5:INK#ch,4
1301 LINE#ch,x+5,y-4.5 TO x+2,y-2:INK#ch,2:LINE#ch TO x+3.5,y-3.8:INK#ch,4
1302 LINE#ch TO x+5,y :INK#ch,7:LINE#ch,x+3.6,y-1 TO x+5,y TO x+5.2,y-2
1303 END DEFine
```



1305 DEFine PROCEDURE Money(ch,x,y,c\$)

```
1306 INK#ch,4:FILL#ch,1:CIRCLE#ch,x,y,1.6,.6,PI/2:FILL#ch,0
1307 INK#ch,4:FILL#ch,1:CIRCLE#ch,x,y-4,3:FILL#ch,0
1308 INK#ch,4:FILL#ch,1:CIRCLE#ch,x,y-5,4,.6,PI/2:FILL#ch,0
1309 INK#ch,0:LINE#ch,x-1,y-1 TO x,y-2 TO x+1,y-1
1310 STRIP#ch,4:INK#ch,0:CSIZE#ch,2,0
1311 CURSOR#ch,x,y,-7,+5:PRINT#ch,c$:STRIP#ch,0:CSIZE#ch,0,0
1312 END DEFine
```



1314 DEFine PROCEDURE Trader(ch,x,y)

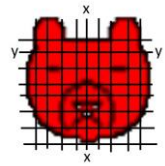
```
1315 INK#ch,7:FILL#ch,1:CIRCLE#ch,x,y+.5,2.2:FILL#ch,0
1316 ARC#ch,x+2.8,y+.6 TO x-2.8,y+.6,PI:INK#ch,0:LINE#ch,x-5,y-1 TO x+5,y-1
1317 INK#ch,7:FILL#ch,1:ARC#ch,x-4,y-4 TO x-1,y-3,-PI:LINE#ch TO x+1,y-3
1318 ARC#ch TO x+4,y-4,-PI:LINE#ch TO x+4,y-6 TO x-4,y-6 TO x-4,y-4:FILL#ch,0
1319 INK#ch,0:CIRCLE#ch,x-1,y+.6,1,.5,PI/2:CIRCLE#ch,x+1,y+.6,1,.5,PI/2
1320 LINE#ch,x-1.8,y-2 TO x,y-4 TO x+1.8,y-2:LINE#ch,x,y-4 TO x,y-6
1321 LINE#ch,x-2.4,y-4.5 TO x-2.4,y-6:LINE#ch,x+2.4,y-4.5 TO x+2.4,y-6
1322 END DEFine
```



Graphics Design

These are first designed using a Grid to work out x and y offsets to a defined location within a screen window. It then becomes a trial-and-error process to present a fit by use of varying the Graphics SCALE of the window and degrees of scale for use with the Graphic Commands ARC CIRCLE LINE.

ch - Windows channel : x,y - Graphics coordinates (deg) : c\$ - currency (string variable)



1324 DEFine PROCEDURE Bear(ch,x,y)

```
1325 INK#ch,2:FILL#ch,1:ARC#ch,x-3,y+1 TO x-1.6,y+1,-80:FILL#ch,0
1326 LINE#ch,x-3,y+1 TO x+3,y+1:FILL#ch,1:ARC#ch,x+1.6,y+1 TO x+3,y+1,-80
1327 ARC#ch TO x+3,y-5,-PI/2 TO x-3,y-5,-PI/3 TO x-3.2,y+1,-PI/2:FILL#ch,0
1328 INK#ch,0:ARC#ch,x-1.5,y-5.4 TO x+1.5,y-5.4,-80
1329 LINE#ch,x-1.2,y-5 TO x+1,y-3.5 TO x-1,y-3.5 TO x+1.2,y-5
1330 LINE#ch,x-2,y-1 TO x-1.4,y-1:LINE#ch,x+1.4,y-1 TO x+2,y-1
1331 END DEFine
```



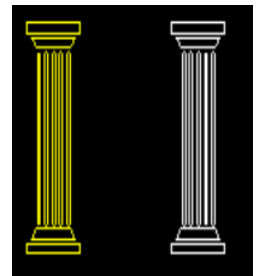
1333 DEFine PROCEDURE Bull(ch,x,y)

```
1334 INK#ch,4:FILL#ch,1:ARC#ch,x-1,y+1 TO x+1,y+1,PI/4
1335 ARC#ch TO x+2.8,y, PI/2 TO x+.8,y-6,-PI/4 TO x-.8,y-6,PI/4
1336 ARC#ch TO x-2.8,y,-PI/4 TO x-1,y+1,PI/2:FILL#ch,0
1337 FILL#ch,1:ARC#ch,x-2,y+1 TO x-4,y+3,-PI/2 TO x-2,y+1, PI:FILL#ch,0
1338 FILL#ch,1:ARC#ch,x+2,y+1 TO x+4,y+3, PI/2 TO x+2,y+1,-PI:FILL#ch,0
1339 LINE#ch,x-3,y TO x-5,y-1 TO x-3,y-1:LINE#ch,x+3,y TO x+5,y-1 TO x+3,y-1
1340 INK#ch,0:LINE#ch,x-1.8,y-1.5 TO x-1,y-1.5:LINE#ch,x+1.8,y-1.5 TO x+1,y-1.5
1341 CIRCLE#ch,x-.8,y-4.8,.4:CIRCLE#ch,x+.8,y-4.8,.4
1342 END DEFine
```



1344 DEFine PROCEDURE Pillar(ch,x,y)

```
1345 LINE#ch, x-5,y+21 TO x+6,y+21 TO x+6,y+23 TO x-5,y+23 TO x-5,y+21
1346 LINE#ch, x-3,y+18 TO x+4,y+18 TO x+5,y+20 TO x-4,y+20 TO x-3,y+18
1347 LINE#ch, x-3,y-18 TO x+4,y-18 TO x+5,y-20 TO x-4,y-20 TO x-3,y-18
1348 LINE#ch, x-5,y-21 TO x+6,y-21 TO x+6,y-23 TO x-5,y-23 TO x-5,y-21
1349 FOR c=1 TO 5
1350   x1=x-4+c*1.5-.3:x2=x-4+c*1.5+.3:y1=y+17:y2=y-17
1351   ARC#ch,x1,y1 TO x2,y1,-PI:LINE#ch TO x2,y2
1352   ARC#ch,x2,y2 TO x1,y2,-PI:LINE#ch TO x1,y1
1353 END FOR c
1354 END DEFine
```



1356 REMark **Set-Up Trader Desk**

[BBQL Mode 4 Colour]

1358 DEFine PROCEDURE Init_Trader

1359 WINDOW#0,500,30,6+gx,224+gy :PAPER#0,0:INK#0,7:RESTORE 1364

1360 WINDOW#1,186,125,120+gx,41+gy:PAPER#1,0:INK#1,7:BORDER#1,1,2

1361 FOR i=3 TO 8

1362 OPEN#i,scr_6x8a0x0:READ a,b,c,d:WINDOW#i,a,b,c+gx,d+gy:BORDER#i,1,2

1363 END FOR i

1364 DATA 112,193,6,18, 42,125,308,41, 154,93,352,18 :REMark Win 3/4/5

1365 DATA 386,12,120,212, 386,44,120,167, 42,22,308,18 :REMark win 6/7/8

1366 PAPER#8,7:INK#8,0:CSIZE#8,2,1:CLS#8:OVER#7,1:INK#7,7:RESTORE 1368

1367 **BLOCK#7,346,10,0,0,1**:FOR i=1 TO 16:READ x,y,str\$:CURSOR#7,x,y:PRINT#7,str\$ **[BLOCK#7,346,10,0,0,2]**

1368 DATA 0,0,'Sym',24,0,'Div',48,0,'Yld',72,0,'P/E',96,0,'Open',129,0,'High'

1369 DATA 164,0,'Low',194,0,'Last',224,0,'Chg',254,0,'Ask',292,0,'Bid'

1370 DATA 325,0,'Vol',254,32,'↑ ↓',292,32,'↑ ↓',325,32,'← →',350,32,'← →'

1371 BLOCK#7,2,4,356,34,7:BLOCK#7,10,3,364,36,7:OVER#7,0

1372 SCALE#7,20,0,0:SCALE#4,80,0,0:SCALE#2,100,0,0:SCALE#1,164,0,0

1373 Title 2,2,2,1,1,134,2,'QBITS TRADER':Title 2,6,2,1,1,136,0,'QBITS TRADER'

[INK#5,0:STRIP#5,2]

1374 **INK#5,7:STRIP#5,1**:PRINT#5,'Sym Last Chg Stock Price':STRIP#5,0

1375 WINDOW#5,150,80,354+gx,30+gy:SCALE#5,80,0,0

1376 **INK#3,7:STRIP#3,1**:PRINT#3,'Sym Last Chg Vol':STRIP#3,0

[INK#3,0:STRIP#3,2]

1377 WINDOW#3,108,180,8+gx,30+gy:SCALE#3,180,0,0

1378 INK#4,4:FOR i=1 TO 3:CURSOR#4,4,i*14:PRINT#4,'↑'

1379 **Bull 4,8,76**:FOR i=1 TO 6:BLOCK#4,8,1,14,14+i*6,4

1380 INK#4,2:FOR i=1 TO 3:CURSOR#4,4,54+i*14:PRINT#4,'↓'

1381 **Bear 4,8,7**:FOR i=1 TO 6:BLOCK#4,8,1,14,64+i*6,2

1382 Title 2,5,1,0,2,32,4,'MARKET':Title 2,5,1,0,2,382,4,'PORTFOLIO':INK#2,7

1383 **RESTORE 1384**:FOR i=1 TO 9:READ x,y,str\$:CURSOR#2,x,y:PRINT#2,str\$

1384 DATA 0,4,'↑ ↓',344,4,'↑ ↓',132,26,'Trends',300,4,'Week'

1385 DATA 378,120,'Shares:',360,130,'Dividends:',354,140,'Tax & Fees:'

1386 DATA 372,110,'Profits:',372,150,'Credits:'

1387 BLOCK#2,10,3,8,8,7:Market 2,35,100:BLOCK#2,10,3,352,8,7:Assets 2,170,100

1388 **Chart 2,40,89:Trader 2,168,31:Money 2,169,46,c\$**:BLOCK#9,112,12,4,211,2

1389 STRIP#9,2:CURSOR#9,5,212:PRINT#9,'Reuters NEWS Flash':STRIP#9,0:INK#2,5

1390 INK#7,7:CURSOR#7,2,22:PRINT#7,'Company Evaluation: Reckoner:'

1391 END DEFine

1393 DEFine PROCEDURE Title(ch,col,w,d,t,tx,ty,str\$)

1394 CSIZE#ch,w,d:OVER#ch,1:INK#ch,col

1395 FOR i=0 TO t:CURSOR#ch,tx+i,ty:PRINT#ch,str\$

1396 CSIZE#ch,0,0:OVER#ch,0:INK#ch,7

1397 END DEFine

QBITS TRADER

Note: Title x2 both with horizontal offset and second with a vertical offset to mimic 3D affect.

1399 DEFine PROCEDURE Init_RNews

1400 **RESTORE 1401**:FOR i=1 TO 8:READ Rnew\$(i)

1401 DATA 'announce release of New Shares...'

1402 DATA 'say bad Sales reduced Share value...'

1403 DATA 'announces Increase in their Dividends...'

1404 DATA 'announce Lower Dividends this quarter...'

1405 DATA 'say Tax Breaks increased Share value...'

1406 DATA 'say Tax Penalties reduced Share value...'

1407 DATA '- Forecasts a Shares Price Rise...'

1408 DATA '- Market Depression reduces Share value.'

1409 END DEFine

Reuters NEWS Flash Nokia say Tax Breaks increased Share value...

```

1411 DEFINE PROCEDURE Init_Stocks
1412 CURSOR#0,76,10:PRINT#0,'Initialising Market':CLS#0,4:RESTORE 1418
1413 FOR a=1 TO 40
1414   READ CNS(a),Stock$(a):FOR b=1 TO 6:READ Stock(a,b):END FOR b
1415   CURSOR#0,184+a*6,10:PRINT#0,':':PAUSE 1
1416 END FOR a
1417 REMark Info>> Company Name : Sym : Status,Div,Yld,P/E,Price,Chg
1418 DATA 'Rio Tinto','RIO',0,5,20,30,1200,12
1419 DATA 'Boeing Co.','BA',0,0,20,30,2200,-5
1420 DATA 'Visa','V ',0,10,12,30,3220,18
1421 DATA 'Exxon Mobil','EOM',0,8,20,30,580,-6
1422 DATA 'Carnival Corp.','CCL',0,5,20,30,230,-8
1423 DATA 'AT&T Inc.','T ',0,12,20,30,1620,-5
1424 DATA 'Walt Disney Co.','DIS',0,0,20,30,490,2
1425 DATA 'Ford Motors','F ',0,4,20,30,370,-10
1426 DATA 'GameStop','GME',0,12,20,30,240, 2
1427 DATA 'Simon Property Grp','SPG',0,0,20,30,3200, 2
1428 DATA 'General Electric','GE ',0,8,20,30,1820, 2
1429 DATA 'Johnson & Johnson','JNJ',0,0,20,30,1230, 2
1430 DATA 'Nokia','NOK',0,12,20,30,280, 2
1431 DATA 'Coca-Cola Co.','KO ',0,0,20,30,470, 2
1432 DATA 'Pfizer Inc','PFE',0,10,20,30,2220, 2
1433 DATA 'NextEra Enery','NEE',0,0,20,30,1670, 2
1434 DATA 'Wells Fargo','WFC',2000,12,20,30,370, 2
1435 DATA 'Rolls Royce Holdings','RR',0,10,20,30,2390, 2
1436 DATA 'New Concept Energy','GBR',0,0,20,30,3200, 2
1437 DATA 'Drax','DRX',12,10,20,30,1310, 2
1438 DATA 'SilverCrest Metals','U1L',0,10,20,30,3200, 2
1439 DATA 'Invinity Enery','IES',5,10,20,30,1260, 2
1440 DATA 'Ecolab','ECL',0,12,20,30,1250, 2
1441 DATA 'Locheed Martin Corp.','LMT',0,4,20,30,3290, 2
1442 DATA 'PayPal Holdings','PYP',0,0,20,30,1200, 2
1443 DATA 'Kinder Morgan','KMI',0,5,20,30,1530, 2
1444 DATA 'Vista Gold Grp','B1B',0,10,20,30,4200, 2
1445 DATA 'Computer Systems Co.','CSC',0,5,20,30,490, 2
1446 DATA 'Intel Corp.','INT',0,5,20,30,385, 2
1447 DATA 'Procter & Gamble Co.','PG ',0,12,20,30,890, 2
1448 DATA 'Walmart','WMT',0,5,20,30,470, 2
1449 DATA 'Exelon Corp.','EXC',0,12,20,30,780, 2
1450 DATA 'Tesco','TSC',0,5,20,30,380, 2
1451 DATA 'Greggs','GRG',0,0,20,30,230, 2
1452 DATA 'Hunting','HTG',0,0,20,30,320, 2
1453 DATA 'Centrica','CNA',0,5,20,30,670, 2
1454 DATA 'Abingdon Health','ABD',0,0,20,30,1200, 2
1455 DATA 'Medica Group P','MGP',0,5,20,30,1370, 2
1456 DATA 'Clarkson','CKN',0,0,20,30,200, 2
1457 DATA 'Rank','RNK',0,10,20,30,680, 2
1458 :
1459 pm=0:FOR wn=1 TO 26:DRJ 8,0,0,3,wn:Trend_Set wn:PAUSE 1
1460 sn= 1:FOR mr=1 TO 18:MStock mr,(mr*10)-10
1461 END DEFINE

```

MARKET			
Sym	Last	Chg	Vol
RIO	12.35	3	300
BA	22.82	12	500
V	32.78	4	200
EOM	6.38	4	800
CCL	2.93	3	100
T	16.68	5	600
DIS	6.25	11	100
F	4.79	3	300
GME	3.40	5	200
SPG	32.84	9	500
GE	18.61	4	700
JNJ	13.49	8	500
NOK	3.90	5	500
KO	6.16	3	300
PFE	23.03	4	300
NEE	17.72	10	100
WFC	4.93	5	800
RR	24.94	4	500
GBR	32.71	9	800
DRX	13.61	6	700
U1L	32.28	3	600
IES	13.34	4	700
ECL	13.21	4	700
LMT	34.02	9	600
PYP	12.57	12	600
KMI	15.70	6	800
B1B	42.61	5	600
CSC	5.21	11	700
INT	4.79	2	300
PG	9.66	4	300
WMT	5.08	2	500
EXC	8.84	6	200
TSC	5.14	8	200
GRG	3.38	8	700
HTG	3.72	3	100
CNA	6.93	5	100
ABD	13.12	10	400
MGP	14.03	2	200
CKN	3.22	2	300
RNK	7.87	5	300

Variables pm - Portfolio max slots : wn - week num : sn - stock num : mr - Market row

1463 REMark Intro Screen

Note: Select currency [D] [E] [P] [Y]



any other key will set Default Currency c\$ = £



1465 DEFine PROCedure Intro

```

1466 OPEN#9,scr_10x10a10x10:SCALE#9,80,0,0
1467 WINDOW#9,512,256,gx,gy:PAPER#9,0:BORDER#9,1,3:CLS#9
1468 WINDOW#2,500,212,6+gx,4+gy:PAPER#2,0:CLS#9
1469 Title 2,2,3,1,2,153,26,'QBITS TRADER':Title 2,6,3,1,2,155,24,'QBITS TRADER'
1470 Market 9,30,70:Chart 9,48,32:Money 2,100,30,'£':Assets 9,89,70
1471 Bull 9,29,37:Title 2,4,3,1,1, 90,147,'↑':Title 9,4,1,0,2,110,160,'BULL'
1472 Bear 9,90,37:Title 2,2,3,1,1,396,147,'↓':Title 9,2,1,0,2,364,160,'BEAR'
1473 INK#9,6:Pillar 9,10,46:RESTORE 1475
1474 INK#9,5:FOR i=1 TO 7:READ str$:CURSOR#9,98,40+i*10:PRINT#9,str$
1475 DATA "Stock Market" PortFolio", " "
1476 DATA "The Game runs a Simulated Stock Market Trader's Desk"
1477 DATA "where you Manage a Portfolio of Shares for 126 weeks"
1478 DATA "of a three year period. Check Share Gains and Losses"
1479 DATA "Increase your Investment through Company Dividends"
1480 DATA "or by simply Buying and Selling Shares."
1481 TBuy 2,60,20:Trader 9,59,30:TSell 2,116,20:CURSOR#9,98,190
1482 INK#9,3:PRINT#9,"Select Currency: $( )ollar €( )uro £( )ound ¥( )en"
1483 RESTORE 1484:INK#9,7:FOR i=1 TO 4:READ x,c$:CURSOR#9,x,190:PRINT#9,c$
1484 DATA 224,'D',284,'E',332,'P',386,'Y'
1485 k=CODE(INKEY$(-1)):c$='k':CLS#9
1486 SELECT ON k=68,100:c$='$( )' :REMark Shift4
1487 SELECT ON k=69,101:c$='€' :REMark CtrlShiftU or CtrlShiftK ?
1488 SELECT ON k=80,112:c$='£' :REMark Shift3
1489 SELECT ON k=89,121:c$='¥' :REMark CtrlShiftI.
1490 END Define

```

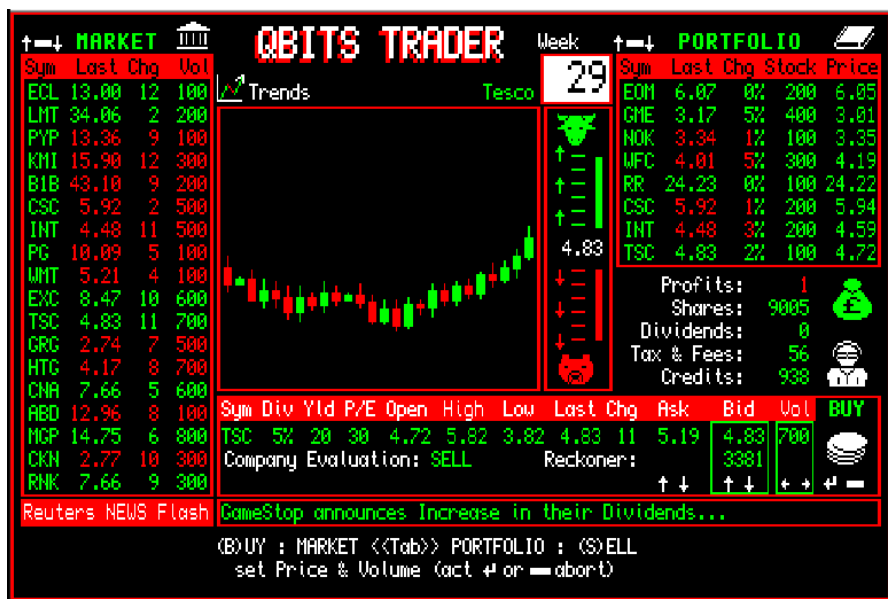
QBITS Trader 'Testing

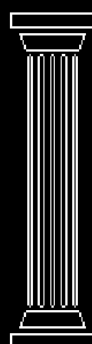
Missing Line numbers: The F1 to F5 keys used for Testing elements of the Program code.

```

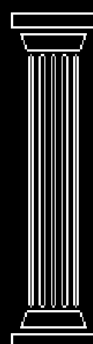
1026 =232:Stock_UpDate 25:wn=26 :REMark F1 Reset to Week 26
1027 =236:Stock_UpDate wn :REMark F2 Advance one week
1028 =248:Sock_UpDate156:Game_End :REMark F3
1029 =244:Wick 1,co.,179,90,70,85.75:IF col=4:col=2:ELSE col=4 F4 Check Candle Wick
1030 =240:FOR i=1 to 8:PRINT#6:CO$(i);";RNew$(i):PAUSE 50:CLS#6 F5 Check RNew$

```



QBITS TRADER



Stock Market

PortFolio

The Game runs a Simulated Stock Market Trader's Desk where you Manage a Portfolio of Shares for 126 weeks of a three year period. Check Share Gains and Losses Increase your Investment through Company Dividends or by simply Buying and Selling Shares.



Select Currency: \$(Dollar) €(Euro) £(Pound) ¥(Yen)

MARKET



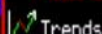
QBITS TRADER

Week

PORTFOLIO



Sym	Last	Chg	Vol
NEE	17.83	4	500
WFC	4.40	5	800
RR	24.90	11	600
GBR	32.79	11	300
DRX	13.87	4	200
UIL	32.64	4	700
IES	13.81	4	700
ECL	13.43	4	100
LMT	33.19	3	500
PYP	13.18	5	400
KMI	16.64	3	800
B1B	43.37	4	800
CSC	5.87	2	300
INT	4.09	3	200
PG	9.24	5	400
WMT	5.85	3	200
EXC	8.85	5	600
TSC	4.45	3	500



Trends

Cisco Systems

156

GAME END

Performance Review

Total Assets: 8855

8000 Plus - Excellent

6000 Plus - Very Good

4000 Plus - Acceptable

2000 Plus - Survivable

Much Less - Bad Luck

(N)ew (Q)uit



Sym	Last	Chg	Stock	Price
EOM	6.56	7%	300	6.09
GME	3.62	10%	500	3.28
NOK	3.83	5%	100	3.99
RR	24.90	1%	60	24.52
UIL	13.43	4%	100	12.91
CSC	5.87	2%	200	5.73

Profits: 4647
 Shares: 8172
 Dividends: 4250
 Tax & Fees: 42
 Credits: 10683



Sym	Div	Yld	P/E	Open	High	Low	Last	Chg	Ask	Bid	Vol
CSC	3%	20	30	5.89	6.90	4.94	5.87	2	6.09	5.77	300

Company Evaluation: BUY Reckoner:

Reuters NEWS Flash

Cisco Systems announce release of New Shares...

(B)UY : MARKET <<Tab>> PORTFOLIO : (S)ELL
 set Price & Volume (act ↑ or = abort)