



Introduction

Back in the day I attended a business course at London University on Company Finance. Assigned to groups as Company Directors we were tasked to run a business, 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. At the end of the course the crux of it all, success or failure was dictated by a Stock Market Simulation run over 3-days representing a three-year period in the Trading life of a group of Companies.

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 gain rewards through Company Dividends and the Maverick method, riding the rise and fall of Share Prices, making snap judgments when to Buy and Sell. The latter statistically riskier and 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 to screen. This began with a Stock list of Companies with Share pricing and volumes on offer. Some start-up capital to buy a Portfolio of selected Stock. Then 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 a 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'

The Intro screen gives a quick overview and then asks you to Select a **Currency (D) (E) (P) (Y)** from the **Intro Screen**.

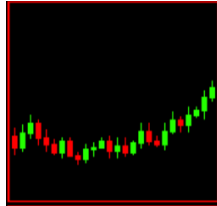


The '**£**' is set as the **Default**.

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**.



Sym	Last Chg	Vol
RIO	12.85	12 200
BA	22.87	6 300
U	32.92	8 200
EDM	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
SPC	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
EDM	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 difference between previous and current weeks **Last Share Price**.

Ask & Bid values are calculated for current week against **Last Share price**.

Vol (Volume) is the number of **Shares** 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**', 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
EDM	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
ECN	13.43 4%	100 12.91
CSC	5.87 2%	200 5.73

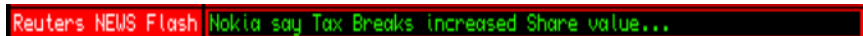
Press '**S**' and use Cursor keys to change the presented '**Ask**' and '**Vol**' values. **Enter** actions your choice, **Spacebar** aborts. Your '**Ask**' Price may or may not be accepted. If not try again.



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.



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

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.



GAME END
Performance Review
8000 Plus - Expert
4000 Plus - Skillful
2500 Plus - Average
1000 Plus - Acceptable
500 Plus - Survivable
Your Fired

QBITS Trader 'GAME END'

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



QBITS Trader 'Title Strings'

Back in the early days with QL SuperBASIC I experimented printing character strings to screen. The CSIZE command with the use of OVER can generate the appearance of **Bold** characters with a horizontal pixel offset. Repeat this in the vertical it can produce a **3D** Style for program Headings.

Title is used for headings, **ch** Windows channel ID, **ci** the character INK, CSIZE **tw, td** then **ts** for 3D size, **tx, ty** the pixel coordinates, and, **str\$** to hold character string.

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



```
1355 DEFINE PROCEDURE Title(ch,bi,ci,tw,td,tx,ty,cstr$)
1356 CSIZE#ch,tw,td:OVER#ch,1
1357 INK#ch,bi:FOR i=0 TO ts:CUSOR#ch,tx+i,ty:PRINT#ch,cstr$
1357 INK#ch,ci:FOR i=0 TO ts:CUSOR#ch,tx+i,ty:PRINT#ch,cstr$
1358 CSIZE#ch,0,0:OVER#ch,0:INK#ch,7
1359 END DEFINE
```

Note: Changes in CSIZE can cause problems with following PRINT statements and OVER printing with CSIZE 0,0 blurs the characters **TBold** uses a simple space offset.



```
1180 DEFINE PROCEDURE TBold(ch,col,cs,cx,cy,cstr$)
1181 INK#ch,col:OVER#ch,1
1182 FOR a=1 TO LEN(str$)
1183   FOR b=0 TO cs:CUSOR#ch,cx+b+a*(6+cs),cy:PRINT#ch,str$(a)
1184 END FOR a:OVER#ch,0
1185 END DEFINE
```

TBold 2,5,1,32,4,'MARKET'




TBold 2,5,1,382,4,'PORTFOLIO'

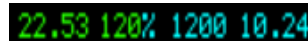
QBITS Trader 'Currency'

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

```
1187 DEFINE PROCEDURE DRJ(ch,cx,cy,dp,sp,num)
1188 str$=ABS(INT(num)):sl=LEN(str$)
1189 IF dp>0 AND sl>dp:str$=str$(1 TO sl-2)& '.'&str$(sl-1 TO sl)
1190 IF dp>0 AND sl=dp:str$='.'&str$
1191 IF dp>0 AND sl<dp:str$='0'&str$
1192 CUSOR#ch,cx,cy:PRINT#ch,FILL$(' ',sp-LEN(str$))&str$
1193 END DEFINE
```

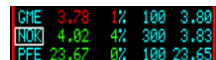
Decimal Right Justified

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
NOV	4.02	4%	300	3.83
PFE	23.67	0%	100	23.65

```
1055 DEFINE PROCEDURE HGL(ch,w,d,x,y)
1056 OVER#ch,-1:LINE#ch,x,y TO x,y+td TO x+w,y+d TO x+w,y TO x,y:OVER#ch,0
1057 END DEFINE
```

HiGHLight

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



Ask	Bid	Vol
12.89	12.60	400
↑	↓	↑
	5040	
↑	↓	↑

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 a 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 **End of Week Share Prices** for all 40 **MARKET** Companies.

```
1245 DEFINE PROCEDURE Trend_Set(wn)
1246 CLS#0:CURSOR#0,60,10:PRINT#0,'Calculating Stock Trends':CLS#0,4
1247 IF pm>0:n=Asset(RND(1 TO pm),1):rc=RND(1 TO 8):PRINT#6,COS(n);';RNew$(rc)
1248 FOR a=1 TO 40
1249   CURSOR#0,200+a*6,10:PRINT#0,' '
1250   Trend(a,wn,1)=100*RND(1 TO 8):ry=RND(2 TO 12):mf=RND(1 TO 2)
1251   IF wn>1:oy=Trend(a,wn-1,4):ELSE oy=84
1252   IF pm>0
1253     IF rc=1 AND a=n:mf=1:ry=12 :REMark Shares ↑
1254     IF rc=2 AND a=n:mf=2:ry=12 :REMark Shares ↓
1255     IF rc=3 AND a=n:Stock(n,2)=RND(8 TO 16) :REMark Div ↑
1256     IF rc=4 AND a=n:Stock(n,2)=RND(0 TO 4) :REMark Div ↓
1257     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.

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 covered 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**

```
1373 DEFINE PROCEDURE Init_Stocks
1374 CURSOR#0,76,10:PRINT#0,'Initialising Market':CLS#0,4:RESTORE 1418
1475 FOR a=1 TO 40
1376 READ CN$(a),Stock$(a):FOR b=1 TO 6:READ Stock(a,b):END FOR b
1377 CURSOR#0,184+a*6,10:PRINT#0,':PAUSE 1
1378 END FOR a
Note: Data Lines Follow for the forty Entries
```

```
1215 DEFINE PROCEDURE MStock(n,my) :REMark Market Stock
1216 INK#3,5:CURSOR#3,2,my:PRINT#3,Stock$(n) :REMark Stock Sym
1217 IF Trend(n,wn,4)>Trend(n,wn-1,4):INK#3,4:INK#7,4:ELSE INK#3,2:INK#7,2
1218 DRJ 3,26,my,2,5,Stock(n,5)+Trend(n,wn,4) :REMark Stock Last
1219 DRJ 3,60,my,0,3,Trend(n,wn,4)-Trend(n,wn-1,4) :REMark Stock Chg
1220 INK#7,5:DRJ 3,82,my,0,4,Trend(n,wn,1) :REMark Stock Vol
1221 END DEFINE
```

Having forty entries which are not all shown to screen at the same time required the need to scroll Up and Down the list within its own window space. 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
1024 = 9:IF schk=0:schk=1:ELSE schk=0 [Tab Key]
1059 DEFINE PROCEDURE M_Up
1060 IF mr<1 AND sn>0:SCROLL#3,10:MStock sn,0
1061 IF mr<1:mr=1
1062 IF sn<1:sn=1
1063 END DEFINE
1065 DEFINE PROCEDURE M_Dn
1066 IF mr>18 AND sn<41:SCROLL#3,-10:MStock sn,170
1067 IF mr>18:mr=18
1068 IF sn>40:sn=40
1069 END DEFINE
```



Sym	Last	Chg	Vol
RIO	12.85	12	200
BA	22.87	6	300
U	32.92	8	200
EDM	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

The **Sym** of a Company entry is highlighted with **HGL** and the use of **M_Up** and **M_dn** to scroll the other entries of the list to screen. The **<Tab>** key switches between **MARKET** and **PORTFOLIO** lists

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 with <Spacebar>.


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 also lower future **Dividend** pay-outs.

PORTFOLIO					
Sym	Last	Chg	Stock	Price	
EDM	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	
ECI	13.43	4%	100	12.91	
CSC	5.87	2%	200	5.73	

```

1127 DEFINE PROCEDURE S_Add(pr,snum)
1128 Asset(pr,1)=sn:Asset(pr,2)=INT(Asset(pr,2)+vol):Asset(pr,3)=cost
1129 sval=sval+snum:sfee=sfee+5+INT(vol/100):cash=cash-snum
1130 Trend(sn,wn,4)=cost-Stock(sn,5):PStock sn,pr:CStock sn
1131 END DEFINE

```

Ask	Bid	Vol	BUY
6.31	6.10	200	
↑ ↓	↑ ↓	↑ ↓	← →

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.

```

1145 DEFINE PROCEDURE S_Del(pr,snum)
1146 Asset(pr,2)=Asset(pr,2)-vol:Asset(pr,3)=cost
1147 sval=sval-snum:sfee=sfee+5+INT(vol/100):cash=cash+snum
1148 Trend(sn,wn,4)=cost-Stock(pn,5):PStock pn,pr:CStock sn
1149 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.

```

1151 DEFINE PROCEDURE S_Sort(pr,pm)
1152 IF pr<pm
1153   FOR row=pr TO pm-1
1154     FOR c=1 TO 3:Asset(row,c)=Asset(row+1,c)
1155   END FOR row
1156 END IF
1157 FOR c=1 TO 3:Asset(pm,c)=0
1158 IF pr=1:SCROLL#5,-10
1159 IF pr>1:CUSOR#5,0,(pr-2)*10:SCROLL#5,-10,2
1160 pm=pm-1:IF pr>pm:pr=pm
1151 END DEFINE

```

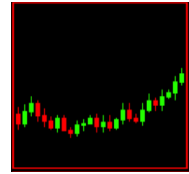
QBITS Trader 'Profuts & Dividens'

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'

AStock Account Info [Variables **sval sdiv 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**]



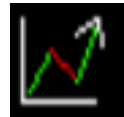
Sym	Div	Yld	P/E	Open	High	Low	Last	Chg	Ask	Bid	Vol
EOM	8%	20	30	6.35	6.93	5.89	6.28	7	6.41	6.15	500
Company Evaluation: BUY					Reckoner:		↑ ↓ ↑ ↓ ← → ← —				

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**.

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

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.



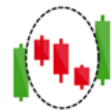
The volume of turnover can indicate the stability of a company to weather any sudden market depressions. It may also be due to trying to maintain Market Share leading to falling profits. It may depend on materials or parts provided by other suppliers. Other disrupting factors can be the weather, industrial 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
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 **PAUSE's** the program, Return will likely jump the display to the next **WEEK**.

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	HiGh Lights 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

QBITS Trader Coding

1000 REMark **QBITS_Trader_bas** [QBITS Trader 2024 QL40th - QPC2] vM30

1002 dev\$='win1_':MODE 4:gx=0:gy=0 :REMark Basic Settings

1004 **WHEN ERROr :CONTINUE:END WHEN**

1006 REMark **Import QBITSConfig Settings - QPC2**

1007 OPEN _IN#9,dev\$&'QBITSConfig':INPUT#9,gx\gy\dn\$:CLOSE#9

1010 **REMark Arrays**

1011 DIM Cn\$(40,20),Stock\$(40,3),Stock(40,6),RNew\$(8,40),Trend(40,156,4)

1012 Tdel=40 :REMark Trade week delay 40sec

1014 **Intro_Trader:Init_Trader:Init_Stocks:Init_RNews:QBITS_Trader**

1016 **DEFine PROCEDURE QBITS_Trader**

1017 DIM Asset(8,4),Audit(130)

1018 pm=0:FOR wn=1 TO 26:DRJ 8,0,0,3,wn:Trend_Set wn:CLS#0

1019 sn=1:FOR mr=1 TO 18:MStock mr,(mr*10)-10

1020 sn=1:mr=1:pm=0:pr=1:pn=1:n=1:cash=10000:sval=0:sdiv=0:sfee=0:sinv=0

1021 schk=0:tmck=0:TMck=0:AStock:THelp:GTm=DATE+Tdel

1022 **REPEAT Loop**

1023 **SMWeek**:IF Asset(pr,1)=0:schk=0

1024 IF schk=0:INK#3,7:HGL 3,16,9,0,181-mr*10:n=sn

1025 IF schk=1:INK#5,7:HGL 5,16,9,0, 81-pr*10:n=Asset(pr,1)

1026 k=CODE(INKEY\$(50)):BLOCK#7,240,10,0,32,0

1027 IF schk=0:HGL 3,16,9,0,181-mr*10

1028 IF schk=1:HGL 5,16,9,0, 81-pr*10

1029 **SElect ON k**

1030 =208:IF schk=0:mr=mr-1:sn=sn-1:M_Up:ELSE pr=pr-1:IF pr<1:pr=1

1031 =216:IF schk=0:mr=mr+1:sn=sn+1:M_Dn:ELSE pr=pr+1:IF pr>pm:pr=pm

1032 =66, 98:IF schk=0:S_Buy sn:AStock:CLS#0:THelp :REMark (B)UY

1033 =83,115:IF schk=1:S_Sell pr:AStock:CLS#0:THelp :REMark (S)ELL

1034 =81,113:QExit:BLOCK#0,20,10,400,14,0 :REMark (Q)uit

1035 =78,110:CLS#5:CLS#1:CLS#0:QBITS_Trader :REMark (N)ew Game

1036 =84,116:TReview :REMark (T)rader Review

1037 = 32:CStock n:STrend n,wn :REMark View Info / Trends

1038 = 9:IF schk=0:schk=1:ELSE schk=0 :REMark TAB Stock<>Portfolio

1039 =35:IF TMck=0:TMck=1:ELSE TMck=0 :REMark TestMode On/Off

1040 =232,236,240,244,248:IF TMck=1:TestMode :REMark Test F1/F2/F3/F4/F5

1041 **END SElect**

1042 **END REPEAT Loop**

1043 **END DEFine**

1045 **DEFine PROCEDURE QExit**

1046 CURSOR#0,400,14:PRINT#0,'Y/N':PAUSE:IF KEYROW(5)=64:L RUN dn\$:STOP

1047 **END DEFine**

1049 **DEFine PROCEDURE THelp**

1050 TBold 0,5,1,64,4,'(N)ew (B)UY MARKET <<Tab>> PORTFOLIO (S)ELL (Q)uit'

1051 TBold 0,5,0,118,14, '(↑ ↓)Set Price & Volume (←→) (Rtn or ←Act)'

1052 BLOCK#0,12,3,314,18,5:BLOCK#0,2,4,352,16,5 **Note Spacebar and Enter tail**

1053 **END DEFine**

```

1055 DEFINE PROCEDURE HGL(ch,w,d,x,y)
1056 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
1057 END DEFINE

```

:REMark HighLight

```

1059 DEFine PROCEDURE M_Up
1060 IF mr<1 AND sn>0:SCROLL#3,10:MStock sn,0
1061 IF mr<1:mr=1
1062 IF sn<1:sn=1
1063 END DEFine

1065 DEFine PROCEDURE M_Dn
1066 IF mr>18 AND sn<41:SCROLL#3,-10:MStock sn,170
1067 IF mr>18:mr=18
1068 IF sn>40:sn=40
1069 END DEFine

```

Symbol	Last	Chg	Stock	Price
EOM	7.81	1%	200	7.84
GME	3.78	1%	100	3.80
NOK	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

Symbol	Last	Chg	Vol
RIO	12.44	3	100
BA	23.40	12	300
U	32.90	4	200
EOM	6.68	4	700
CCL	3.70	3	300
T	17.20	5	400
DIS	5.20	5	300
F	4.78	3	600
CME	2.82	5	500
SPG	32.26	12	300
GE	18.99	4	300
JNJ	13.26	9	800
NOK	3.58	5	300
KO	6.09	4	800
PFE	23.04	4	800
NEE	17.69	2	200
WFC	4.59	5	600
RR	24.23	4	800

```

1071 DEFINE PROCEDURE SMWeek
1072 del=GTm-DATE:IF del<0:del=0
1073 BLOCK#4,4,42-del,28,10+del,4:BLOCK#4,4,42-del,28,70,2
1074 IF GTm<DATE:BLOCK#4,4,48,28,4,0:BLOCK#4,4,48,28,68,0
1075 IF GTm<DATE:Stock_Update wn:GTm=DATE+Tdel
1076 END DEFINE

```



Weeks Passed

```

1078 DEFine PROCEDURE Stock_Update(wn)
1079 IF wn<156:wn=wn+1:ELSE TReview:RETurn
1080 Trend_Set wn:CStock n:STrend n,wn:sval=0:dnum=0
1081 FOR rs=18 TO 1 STEP -1:MStock (sn-mr)+rs,(rs*10)-10
1082 FOR rp=1 TO pm
1083 m=Asset(rp,1):pdiv=Stock(Asset(rp,1),2)
1084 snum=INT(Asset(rp,2)*(Stock(m,5)+Trend(m,wn,4))/100)
1085 IF wn MOD 13=0 AND pdiv=0
1086 dnum=INT(snum*pdiv/400):sdiv=sdiv+dnum:cash=cash+dnum
1087 END IF
1088 sval=sval+snun:PStock Asset(rp,1),rp
1089 END FOR rp
1090 AStock:PAUSE 10:CLS#0:THelp
1091 END DEFine

```

Bars Extend
Up/Down
End of Week

```

1093 DEFine PROCEDURE TReview
1094 INK 7: IF pm=0:RETurn :ELSE CLS#1
1095 CURSOR 36,38:PRINT 'Performance Review: Audit(0)=10000
1096 LINE 18,20 TO 18,90:INK 248:LINE 18,40 TO 160,40:INK 4
1097 FOR i=1 TO wn-26
1098 x1=23+i:x2=24+i:y1=Audit(i-1)/250:y2=Audit(i)/250:LINE x1,y1 TO x2,y2
1099 END FOR i
1100 INK 5:CURSOR 32,110:PRINT 'SpaceBar to Return'
1101 IF wn=156
1102 CURSOR 0,0:Title 1,7,2,1,1,42,10,'GAME END':BLOCK 120,10,32,110,0
1103 score=Audit(130)-10000:cx=48:str$='Your Fired'
1104 IF score> 500:cx=48:str$='Survivable'
1105 IF score>1000:cx=48:str$='Acceptable'
1106 IF score>2500:cx=58:str$='Average'
1107 IF score>4000:cx=58:str$='Skilful'
1108 IF score>8000:cx=60:str$='Expert'
1109 TBold 1,7,1,cx,108,str$:CURSOR 32,110:PRINT '(N)ew (Q)uit'
1110 END IF
1111 END DEFine


```



```

1113 DEFINE PROCEDURE S_Buy(sn)
1114 TBuy 7,131,11:TBold 7,4,2,344,0,'BUY':CStock sn:STrend sn,wn
1115 vn=Trend(sn,wn,1):vm=vn:cn=Stock(sn,5)+Trend(sn,wn,4)
1116 S_Vol 102.5,286,2.5,vn,vm,cn:IF k=32:RETURN
1117 IF cost<Stock(sn,5)+Trend(sn,wn,4)+RND(6 TO 12)
1118 CURSOR#7,2,32:PRINT#7,'Unsuccessful Bid - Try Again':PAUSE 20:RETURN
1119 END IF
1120 snum=INT(vol*cost)/100 :REMark cost=Bid
1121 IF snum+20>cash:CURSOR#7,2,32:PRINT#7,'Insufficient Funds':PAUSE 20:RETURN
1122 FOR pr=1 TO pm:IF Asset(pr,1)=sn:S_Add pr,snum:RETURN
1123 IF pm<8:pm=pm+1:S_Add pm,snum:RETURN
1124 CURSOR#7,2,32:PRINT#7,'Sell Some Shares'
1125 END DEFINE

```

Ask	Bid	Vol	BUY
6.31	6.10	200	
↑ ↓	↑ ↓	↑ ↓	← =

```

1127 DEFINE PROCEDURE S_Add(pr,snum)
1128 Asset(pr,1)=sn:Asset(pr,2)=INT(Asset(pr,2)+vol):Asset(pr,3)=cost
1129 sval=sval+snum:sfee=sfee+5+INT(vol/100):cash=cash-snum
1130 Trend(sn,wn,4)=cost-Stock(sn,5):PStock sn,pr:CStock sn
1131 END DEFINE

```

```

1133 DEFINE PROCEDURE S_Sell(pr)
1134 pn=Asset(pr,1):IF pn<=0:RETURN
1135 TSell 7,133,10:TBold 7,2,2,340,0,'SELL'
1136 STrend pn,wn:CStock pn:vm=Asset(pr,2):IF vm>900:vm=900
1137 vn=vm:S_Vol 88.5,248,2.5,vn,vm,Stock(pn,5)+Trend(pn,wn,4):IF k=32:RETURN
1138 IF cost>Stock(pn,5)+Trend(pn,wn,4)+RND(0 TO 12)
1139 CURSOR#7,0,32:PRINT#7,'Unsuccessful Sale - Try Again':PAUSE 20:RETURN
1140 END IF
1141 snum=INT(vol*cost)/100 REMark cost=Ask
1142 S_Del pr,snum:IF Asset(pr,2)=0:S_Sort pr,pm
1143 END DEFINE

```

Ask	Bid	Vol	SELL
4.20	3.93	200	
↑ ↓	↑ ↓	↑ ↓	← =

```

1145 DEFINE PROCEDURE S_Del(pr,snum)
1146 Asset(pr,2)=Asset(pr,2)-vol:Asset(pr,3)=cost
1147 sval=sval-snum:sfee=sfee+5+INT(vol/100):cash=cash+snun
1148 Trend(sn,wn,4)=cost-Stock(pn,5):PStock pn,pr:CStock sn
1149 END DEFINE

```

```

1151 DEFINE PROCEDURE S_Sort(pr,pm)
1152 IF pr<pm
1153 FOR row=pr TO pm-1
1154 FOR c=1 TO 3:Asset(row,c)=Asset(row+1,c)
1155 END FOR row
1156 END IF
1157 FOR c=1 TO 3:Asset(pm,c)=0
1158 IF pr=1:SCROLL#5,-10
1159 IF pr>1:CURSOR#5,0,(pr-2)*10:SCROLL#5,-10,2
1160 pm=pm-1:IF pr>pm:pr=pm
1161 END DEFINE

```

```

1163 DEFINE PROCEDURE S_Vol(x,cx,dp,sp,vn,vm,cn)
1164 INK#7,5:HGL 7,12,15,x,0:HGL 7,7,4,15,116,4,0
1165 REPEAT Vol_Ip
1166 DRJ 7,324,12,0,3,vn:DRJ 7,cx,12,2,5,cn
1167 DRJ 7,cx,22,0,5,INT((cn*vn)/100)
1168 k=CODE(INKEY$(-1))
1169 IF k=192:vn=vn-10:IF vn< 10:vn= 10
1170 IF k=200:vn=vn+10:IF vn> vm:vn= vm
1171 IF k=208:cn=cn+ 1 :IF cn>9990:cn=9990
1172 IF k=216:cn=cn - 1 :IF cn< 1:cn= 1
1173 IF k= 32:EXIT Vol_Ip
1174 IF k= 10:vol=vn:cost=cn:EXIT Vol_Ip
1175 END REPEAT Vol_Ip
1176 INK#7,5:HGL 7,12,15,x,0:HGL 7,7,4,15,116,4,0
1177 BLOCK#7,30,12,cx,22,0:BLOCK#7,34,32,348,0,0
1178 END DEFINE

```

Last Chg	Ask	Bid	Vol	BUY
6.10	3	6.31	6.10	200
Reckoner :		1220		
↑ ↓	↑ ↓	↑ ↓	↑ ↓	← →

:REMark Volume of Shares

:REMark Ask/Bid Price

Last Chg	Ask	Bid	Vol	SELL
4.10	9	4.20	3.93	200
Reckoner :		840		
↑ ↓	↑ ↓	↑ ↓	↑ ↓	← →

```

1180 DEFINE PROCEDURE Tbold(ch,col,cs,cx,cy,ctr$)
1181 INK#ch,col:OVER#ch,1
1182 FOR a=1 TO LEN(str$)
1183 FOR b=0 TO cs:CURSOR#ch,cx+b+a*(6+cs),cy:PRINT#ch,str$(a)
1184 END FOR a:OVER#ch,0
1185 END DEFINE

```

1187 DEFINE PROCEDURE DRJ(ch,cx,cy,dp,sp,num) :REMark Decimal Right Justified

```

1188 str$=ABS(INT(num)):sl=LEN(str$)
1189 IF dp>0 AND sl>dp:str$=str$(1 TO sl-2)&'.'&str$(sl-1 TO sl)
1190 IF dp>0 AND sl=dp:str$='.'&str$
1191 IF dp>0 AND sl<dp:str$='.0'&str$
1192 CURSOR#ch,cx,cy:PRINT#ch,FILL$(' ',sp-LEN(str$))&str$
1193 END DEFINE

```

Variuos outputs with/without

decimal point

22.53 120% 1200 10.24

```

1195 DEFINE PROCEDURE AStock :REMark Account Stock Info
1196 DRJ 2,428,122,0,5,sval :REMark Share Value 1-8 Int((stock*last)/100)
1197 DRJ 2,428,132,0,5,sdiv :REMark WK13 Dividend 1-8 Int((sval*sdiv)/400)
1198 DRJ 2,428,142,0,5,sfee :REMark Fees each Transaction vol/100
1199 DRJ 2,428,152,0,5,cash+sdiv-sfee :REMark Cash+Dividens-Fees
1200 Audit(wn-26)=sval+cash+sdiv-sfee:sinv=(cash+sval)-10000
1201 IF sinv<0:INK#2,2:ELSE INK#2,4
1202 DRJ 2,428,112,0,5,sinv:INK#2,5
1203 END DEFINE

```

```

1205 DEFINE PROCEDURE PStock(n,pr) :REMark Portfolio Stock Info
1206 INK#5,5:py=(pr-1)*10:CURSOR#5,2,py:PRINT#5,Stock$(n) :REMark Stock Sym
1207 sChg=Stock(n,5)+Trend(n,wn,4)-Asset(pr,3):CURSOR#5,78,py:PRINT#5, '%'
1208 IF sChg<0:INK#5,2:ELSE INK#5,4
1209 DRJ 5,26,py,2,5,Stock(n,5)+Trend(n,wn,4) :REMark Last
1210 DRJ 5,60,py,0,3,(sChg/Asset(pr,3))*100:INK#5,5 :REMark Chg
1211 DRJ 5,90,py,0,4,Asset(pr,2) :REMark Stock
1212 DRJ 5,119,py,2,5,Asset(pr,3) :REMark Price
1213 END DEFINE

```

PORTFOLIO				
Sum	Last Chg	Stock	Price	
EDM	7.01	1%	200	7.04
CME	3.78	1%	100	3.80
MMO	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

```

1215 DEFine PROCEDURE MStock(n,my) :REMark Market Stock Info
1216 INK#3,5:CURSOR#3,2,my:PRINT#3,Stock$(n) :REMark Stock Sym
1217 IF Trend(n,wn,4)>Trend(n,wn-1,4):INK#3,4:INK#7,4:ELSE INK#3,2:INK#7,2
1218 DRJ 3,26,my,2,5,Stock(n,5)+Trend(n,wn,4) :REMark Stock Last
1219 DRJ 3,60,my,0,3,Trend(n,wn,4)-Trend(n,wn-1,4) :REMark Stock Chg
1220 INK#7,5:DRJ 3,82,my,0,4,Trend(n,wn,1) :REMark Stock Vol
1221 END DEFine

```



Symbol	Last	Chg	Vol
RIO	12.85	12	200
BA	22.87	6	300
U	32.92	8	200
EDM	6.28	7	500

```

1223 DEFine PROCEDURE CStock(n) :REMark Company Stock Info
1224 INK#7,5:CURSOR#7,0,12:PRINT#7,Stock$(n);' %' :REMark Sym
1225 REMark DRJ 7, 24,10,0,4,Stock(n,1) :REMark Unused at this time
1226 DRJ 7, 24,12,0,2,Stock(n,2) :REMark Dividend
1227 DRJ 7, 52,12,0,2,Stock(n,3) :REMark Yield
1228 DRJ 7, 74,12,0,2,Stock(n,4) :REMark Price/Expense
1229 ops=Stock(n,5):ols=Trend(n,wn-1,4):nws=Trend(n,wn,4)
1230 DRJ 7,92,12,2,5,ops+ols :REMark Open
1231 DRJ 7,125,12,2,5,ops+ols+Trend(n,wn,2) :REMark High
1232 DRJ 7,158,12,2,5,ops-Trend(n,wn,3)-ols :REMark Low
1233 IF nws>ols:INK#3,4:INK#7,4:ELSE INK#3,2:INK#7,2
1234 DRJ 7,191,12,2,5,ops+nws :REMark Last
1235 DRJ 7,224,12,0,3,nws-ols :INK#7,5 :REMark Change
1236 DRJ 7,324,12,0,3,Trend(n,wn,1) :REMark Volume
1237 DRJ 7,286,12,2,5,ops+nws-RND(10 TO 20) :REMark Bid
1238 DRJ 7,248,12,2,5,ops+nws-RND(10 TO 40) :REMark Ask
1239 wk1=Trend(n,wn-13,4)*Stock(n,4)*Stock(n,3)/100
1240 wk2=Trend(n,wn,4)*Stock(n,4)*Stock(n,3)/100
1241 IF wk1>wk2:str$='BUY':ELSE str$='SELL' :REMark 13wk Trend
1242 CURSOR#7,122,22:PRINT#7,str$
1243 END DEFine

```



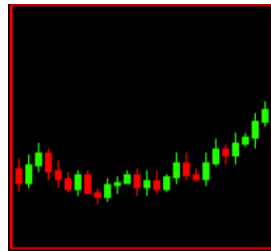
Sym	Div	Yld	P/E	Open	High	Low	Last	Chg	Ask	Bid	Vol
EDM	8%	20	30	6.35	6.93	5.89	6.28	7	6.41	6.15	500

Company Evaluation: BUY Reckoner:

```

1245 DEFine PROCEDURE STrend(n,en) :REMark Stock Trend
1246 CURSOR#2,178,28:PRINT#2,FILL$(' ',20-LEN(CN$(n)))&CN$(n)
1247 INK#4,7:DRJ 4,2,56,2,5,Stock(n,5)+Trend(n,en,4)
1248 DRJ 8,0,0,3,en:CLS:INK#4,7:cw=1:CLS
1249 FOR i=en-25 TO en
1250 IF i>1:oy=Trend(n,i-1,4):ELSE oy=84
1251 hy=Trend(n,i,2):ly=Trend(n,i,3):cy=Trend(n,i,4)
1252 IF cy>oy:col=4:y1=cy:y2=oy:ELSE col=2:y1=oy:y2=cy
1253 Wick 1,col,177,hy,ly,y1,y2
1254 END FOR i
1255 IF y1>84:SCROLL#1,(y1-84)*.6:ELSE SCROLL#1,-(84-y1)*.6
1256 END DEFine

```



```

1258 DEFine PROCEDURE Wick(ch,col,wx,hy,ly,y1,y2)
1259 PAN-7:x1=wx-2:x2=wx+2:INK#ch,col:LINE#ch,col:wx,hy TO wx,ly
1260 FILL#ch,1:LINE#ch,x1,y1 TO x2,y1 TO x2,y2 TO x1,y2 TO x1,y1:FILL#ch,0
1261 END DEFine

```



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

```

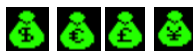
1263 DEFINE PROCEDURE Trend_Set(wn)
1264 CLS#0:CURSOR#0,60,10:PRINT#0,'Calculating Stock Trends':CLS#0,4
1265 IF pm>0:n=Asset(RND(1 TO pm),1):rk=RND(1 TO 8):PRINT#6,CN$(n),'':RNew$(rk)
1266 FOR a=1 TO 40
1267   CURSOR#0,200+a*6,10:PRINT#0,','
1268   Trend(a,wn,1)=100*RND(1 TO 8):ry=RND(2 TO 12):mf=RND(1 TO 2)
1269   IF wn>1:oy=Trend(a,wn-1,4):ELSE oy=84
1270   IF pm>0
1271     IF rk=1 AND a=n:mf=1:ry=12 :REMark Shares↑ Bull
1272     IF rk=2 AND a=n:mf=2:ry=12 :REMark Shares↓ Bearl
1273     IF rk=3 AND a=n:Stock(n,2)=RND(8 TO 16) :REMark Div ↑
1274     IF rk=4 AND a=n:Stock(n,2)=RND(0 TO 4) :REMark Div ↓
1275     IF rk=5 AND a=n:mf=1:ry=6 :REMark Tax ↑
1276     IF rk=6 AND a=n:mf=2:ry=6 :REMark Tax ↓
1277     IF rk=7:mf=1:ry=8 :REMark Bull ↑
1278     IF rk=8:mf=2:ry=8 :REMark ↓
1279   END IF
1280   IF wn MOD 13=0 AND Stock(a,2)>0:mf=2:ry=2+INT(Stock(a,2)/4)
1281   IF mf=1:cy=oy+ry:IF cy>148:cy=oy-RND(2 TO 6)
1282   IF mf=2:cy=oy-ry:IF cy<20:cy=oy+RND(2 TO 6)
1283   IF cy>oy:hy=cy+RND(2 TO 7):ly=oy-RND(2 TO 7)
1284   IF cy<oy:hy=oy+RND(2 TO 7):ly=cy-RND(2 TO 7)
1285   Trend(a,wn,2)=hy:Trend(a,wn,3)=ly:Trend(a,wn,4)=cy
1286 END FOR a
1287 END DEFINE

```

1289 REMark Intro Screen

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

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



```

1291 DEFINE PROCEDURE Intro_Trader
1292 OPEN#9,scl_:_WINDOW#9,512,256,gx,gy:PAPER#9,0:BORDER#9,1,3:CLS#9
1293 BLOCK#9,508,1,0,222,3:SCALE#9,100,0,0
1294 Title 9,2,3,1,3,155,19,'QBITS TRADER':Title 9,6,3,1,2,158,20,'QBITS TRADER'
1295 Mrket 9,36,88:Chart 9,60,48:Money 9,86,48,288,142,'£':Assets 9,112,88
1296 Bull 9,36,46:Title 9,4,3,1,1, 90,144,'↑':Title 9,4,1,0,2,110,156,'BULL'
1297 Bear 9,112,46:Title 9,2,3,1,1,400,144,'↓':Title 9,2,1,0,2,364,156,'BEAR'
1298 INK#9,6:Pillar 9,12,60:Pillar 9,134,60:INK#9,5
1299 RESTORE 1300:FOR i=1 TO 7:READ str$:CURSOR#9,98,40+i*10:PRINT#9,str$
1300 DATA "Stock Market" Portfolio," "
1301 DATA "The Game runs a Simulated Stock Market Trader's Desk"
1302 DATA "where you Manage a Portfolio of Shares for 126 weeks"
1303 DATA "of a three year period. Check Share Gains and Losses"
1304 DATA " Increase your Investment through Company Dividends"
1305 DATA " or by simply Buying and Selling Shares."
1306 TBuy 9,58,34:Trader 9,74,40:TSell 9,92,34:CURSOR#9,98,190
1307 INK#9,3:PRINT#9,'Select Currency: $( )ollar μ( )uro `( )ound ž( )en"
1308 RESTORE 1309:INK#9,7:FOR i=1 TO 4:READ x,c$:CURSOR#9,x,190:PRINT#9,c$
1309 DATA 224,'D',284,'E',332,'P',386,'Y'
1310 k=CODE(INKEY$(-1)):c$="":CLS#9
1311 SElect ON k=68,100:c$='£' :REMark Shift4
1312 SElect ON k=69,101:c$='€' :REMark CtrlShiftU or CtrlShiftK ?
1314 SElect ON k=80,112:c$='¥' :REMark Shift3
1315 END DEFINE

```


1317 REMark Trader Desk

1319 DEFine PROCEDURE Init_Trader

```

1320 WINDOW#0,512,32,gx,gy+224 :PAPER#0,0:INK#0,5:BORDER#0,1,3
1321 WINDOW#1,186,125,120+gx,40+gy:PAPER#1,0:INK#1,7:BORDER#1,1,2
1322 WINDOW#2,500,222,gx+6,gy+2 :PAPER#2,0:INK#2,7:RESTORE 1325
1323 FOR i=3 TO 8:OPEN#i,scr_:READ a,b,c,d:WINDOW#i,a,b,c+gx,d+gy:BORDER#i,1,2
1324 INK#6,7:SCALE#4,80,0,0:SCALE#2,100,0,0:SCALE#1,164,0,0
1325 DATA 112,193, 6, 17, 42,125,308, 40, 154,93,352,17 :REMark Win 3/4/5
1326 DATA 386, 12,120,211, 386, 44,120,166, 42,22,308,17 :REMark Win 6/7/8
1327 Title 2,2,2,1,1,134,3,'QBITS TRADER':Title 2,6,2,1,1,136,4,'QBITS TRADER'
1328 PAPER#8,7:INK#8,0:CSIZE#8,2,1:CLS#8
1329 BLOCK#7,346,10,0,0,1:INK#7,7 :SCALE#7,20,0,0:OVER#7,1
1330 RESTORE 1331:FOR i=1 TO 16:READ x,y,str$:CURSOR#7,x,y:PRINT#7,str$
1331 DATA 0,0,'Sym',24,0,'Div',48,0,'Yld',72,0,'P/E',96,0,'Open',129,0,'High'
1332 DATA 164,0,'Low',194,0,'Last',224,0,'Chg',254,0,'Ask',292,0,'Bid'
1333 DATA 325,0,'Vol',254,32,'↑ ↓',292,32,'↑ ↓',325,32,'← →',366,32,'← →'
1334 BLOCK#7,2,4,372,34,7:BLOCK#7,10,3,352,36,7 :OVER#7,0
1335 INK#7,7:CURSOR#7,2,22:PRINT#7,'Company Evaluation: Reckoner:'
1336 Chart 2,38,89:Trader 2,161,34:Money 2,161,48,476,122,c$:BLOCK#9,112,12,4,210,2
1337 STRIP#9,2:INK#9,0:CURSOR#9,5,211:PRINT#9,'Reuters NEWS Flash':STRIP#9,0
1338 Title 2,5,1,0,2,380,5,'PORTFOLIO':Assets 2,162,99
1339 INK#5,7:STRIP#5,1:PRINT#5,'Sym Last Chg Stock Price':STRIP#5,0
1340 WINDOW#5,150,80,354+gx,29+gy:SCALE#5,80,0,0
1341 INK#4,4:FOR i=1 TO 3:CURSOR#4,4,i*14:PRINT#4,'↑'
1342 Bull 4,8,76:FOR i=1 TO 6:BLOCK#4,8,1,14,14+i*6,4
1343 INK#4,2:FOR i=1 TO 3:CURSOR#4,4,54+i*14:PRINT#4,'↓'
1344 Bear 4,8,7:FOR i=1 TO 6:BLOCK#4,8,1,14,64+i*6,2
1345 Title 2,5,1,0,2,32,5,'MARKET':Market 2,34,99
1346 INK#3,7:STRIP#3,1:PRINT#3,'Sym Last Chg Vol':STRIP#3,0
1347 WINDOW#3,108,180,8+gx,29+gy:SCALE#3,180,0,0 :REMark Market
1348 RESTORE 1349:FOR i=1 TO 9:READ x,y,str$:CURSOR#2,x,y:PRINT#2,str$
1349 DATA 2,4,'↑ ↓',346,4,'↑ ↓',132,28,'Trends',310,5,'Week'
1350 DATA 378,122,'Shares:',360,132,'Dividends:',354,142,'Tax & Fees:'
1351 DATA 372,112,'Profits:',372,152,'Credits:'
1352 BLOCK#2,10,3,10,9,7:BLOCK#2,10,3,354,9,7
1353 END DEFine

```

1355 DEFine PROCEDURE Title(ch,col,w,d,t,tx,ty,str\$)

```

1356 CSIZE#ch,w,d:OVER#ch,1:INK#ch,col
1357 FOR i=0 TO t:CURSOR#ch,tx+i,ty:PRINT#ch,str$
1358 CSIZE#ch,0,0:OVER#ch,0:INK#ch,7
1359 END DEFine

```



Note: Title set with horizontal vertical offset to mimic 3D affect.

1361 DEFine PROCEDURE Init_RNews

```

1362 RESTORE 1363:FOR i=1 TO 8:READ RNew$(i)
1363 DATA 'announce release of New Shares...'
1364 DATA 'say bad Sales reduced Share value...'
1365 DATA 'announces Increase in their Dividends...'
1366 DATA 'announce Lower Dividends this quarter...'
1367 DATA 'say Tax Breaks increased Share value...'
1368 DATA 'say Tax Penalties reduced Share value...'
1369 DATA '- Forecasts a Shares Price Rise...'
1370 DATA '- Market Depression Reduces Share value.'

```



```

1371 END DEFine
1373 DEFine PROCEDURE Init_Stocks
1374 CURSOR#0,76,10:PRINT#0,'Initialising Market':CLS#0,4:RESTORE 1380
1375 FOR a=1 TO 40
1376   READ CNS$(a),Stock$(a):FOR b=1 TO 6:READ Stock(a,b):END FOR b
1377   CURSOR#0,184+a*6,10:PRINT#0,':':PAUSE 1
1378 END FOR a
1379 REMark Info>> Company Name:Sym:Sector,Status,Div,Yld,P/E,Last,Chg
1380 DATA 'Rio Tinto','RIO',0,5,20,30,1200,12
1381 DATA 'Boeing Co.','BA',0,0,20,30,2200,-5
1382 DATA 'Visa','V',0,10,12,30,3220,18
1383 DATA 'Exxon Mobil','EOM',0,8,20,30,580,-6
1384 DATA 'Carnival Corp.','CCL',0,5,20,30,230,-8
1385 DATA 'AT&T Inc.','T',0,12,20,30,1620,-5
1386 DATA 'Walt Disney Co.','DIS',0,0,20,30,490,2
1387 DATA 'Ford Motors','F',0,4,20,30,370,-10
1388 DATA 'GameStop','GME',0,12,20,30,240,2
1389 DATA 'Simon Property Grp','SPG',0,0,20,30,3200,2
1390 DATA 'General Electric','GE',0,8,20,30,1820,2
1391 DATA 'Johnson & Johnson','JNJ',0,0,20,30,1230,2
1392 DATA 'Nokia','NOK',0,12,20,30,280,2
1393 DATA 'Coca-Cola Co.','KO',0,0,20,30,470,2
1394 DATA 'Pfizer Inc.','PFE',0,10,20,30,2220,2
1395 DATA 'NextEra Enery','NEE',0,0,20,30,1670,2
1396 DATA 'Wells Fargo','WFC',2000,12,20,30,370,2
1397 DATA 'Rolls Royce Holdings','RR',0,10,20,30,2390,2
1398 DATA 'New Concept Energy','GBR',0,0,20,30,3200,2
1399 DATA 'Drax','DRX',12,10,20,30,1310,2
1400 DATA 'SilverCrest Metals','U1L',0,10,20,30,3200,2
1401 DATA 'Invinity Enery','IES',5,10,20,30,1260,2
1402 DATA 'Ecolab','ECL',0,12,20,30,1250,2
1403 DATA 'Locheed Martin Corp.','LMT',0,4,20,30,3290,2
1404 DATA 'PayPal Holdings','PYP',0,0,20,30,1200,2
1405 DATA 'Kinder Morgan','KMI',0,5,20,30,1530,2
1406 DATA 'Vista Gold Grp','B1B',0,10,20,30,4200,2
1407 DATA 'Computer Systems Co.','CSC',0,5,20,30,490,2
1408 DATA 'Intel Corp.','INT',0,5,20,30,385,2
1409 DATA 'Procter & Gamble Co.','PG',0,12,20,30,890,2
1410 DATA 'Walmart','WMT',0,5,20,30,470,2
1411 DATA 'Exelon Corp.','EXC',0,12,20,30,780,2
1412 DATA 'Tesco','TSC',0,5,20,30,380,2
1413 DATA 'Greggs','GRG',0,0,20,30,230,2
1414 DATA 'Hunting','HTG',0,0,20,30,320,2
1415 DATA 'Centrica','CNA',0,5,20,30,670,2
1416 DATA 'Abingdon Health','ABD',0,0,20,30,1200,2
1417 DATA 'Medica Group P','MGP',0,5,20,30,1370,2
1418 DATA 'Clarkson','CKN',0,0,20,30,200,2
1419 DATA 'Rank','RNK',0,10,20,30,680,2
1420 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

1422 REMark Trader Graphics

1424 DEFINE PROCEDURE TBuy(ch,x,y)

```
1425 INK#ch,6:FILL#ch,1:ARC#ch,x-4,y TO x+4,y,-PI/2 LINE#ch TO x+4,y-3:
1426 ARC#ch TO x-4,y-3,-PI/2:LINE#ch TO x-4,y : FILL#ch,0
1427 INK#ch,0:ARC#ch,x-3,y-6 TO x+4,y-6,PI/2
1428 ARC#ch,x-4,y-1.4 TO x+3.6,y-1.4,PI/2:ARC#ch,x-4,y-2.5 TO x+4,y-2.5,PI/2
1429 END DEFINE
```



1431 DEFINE PROCEDURE TSell(ch,x,y)

```
1432 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
1433 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
1434 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
1435 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
1436 END DEFINE
```



1438 DEFINE PROCEDURE Market(ch,x,y)

```
1439 INK#ch,7:FILL#ch,1
1440 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
1441 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
1442 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
1443 LINE#ch,x,y-2.2 TO x,y-4.6 :LINE#ch,x-3,y-5 TO x+3,y-5
1444 END DEFINE
```



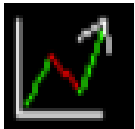
1446 DEFINE PROCEDURE Assets(ch,x,y)

```
1447 INK#ch,7:FILL#ch,1:LINE#ch,x+3,y-1 TO x-1.8,y-1
1448 LINE#ch TO x-3.8,y-4 TO x+1,y-4 TO x+3.2,y-1:FILL#ch,0
1449 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
1450 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
1451 END DEFINE
```



1453 DEFINE PROCEDURE Chart(ch,x,y)

```
1454 INK#ch,7:LINE#ch,x,y TO x,y-5 TO x+5,y-5:INK#ch,4
1455 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
1456 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
1457 END DEFINE
```



1459 DEFINE PROCEDURE Money(ch,x,y,c\$)

```
1460 INK#ch,4:FILL#ch,1:CIRCLE#ch,x,y,1.6,.6,PI/2:FILL#ch,0
1461 INK#ch,4:FILL#ch,1:CIRCLE#ch,x,y-4,3:FILL#ch,0
1462 INK#ch,4:FILL#ch,1:CIRCLE#ch,x,y-5,4,.6,PI/2:FILL#ch,0
1463 INK#ch,0:LINE#ch,x-1,y-1 TO x,y-2 TO x+1,y-1
1464 STRIP#ch,4:INK#ch,0:CSIZE#ch,2,0
1465 CURSOR#ch,x,y,-7,+5:PRINT#ch,c$:STRIP#ch,0:CSIZE#ch,0,0
1466 END DEFINE
```



1468 DEFINE PROCEDURE Trader(ch,x,y)

```
1469 INK#ch,7:FILL#ch,1:CIRCLE#ch,x,y+.5,2.2:FILL#ch,0
1470 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
1471 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
1472 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
1473 INK#ch,0:CIRCLE#ch,x-1,y+.6,1,.5,PI/2:CIRCLE#ch,x+1,y+.6,1,.5,PI/2
1474 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
1475 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
1476 END DEFINE
```



```

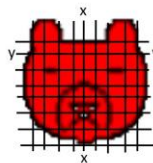
1478 DEFine PROCEDURE Bear(ch,x,y)
1479 INK#ch,2:FILL#ch,1:LINE#ch,x-3,y+1 TO x+3,y+1
1480 ARC#ch TO x+3,y-5,-PI/2 TO x-3,y-5,-PI/3 TO x-3.2,y+1,-PI/2:FILL#ch,0
1481 FILL#ch,1:CIRCLE#ch,x-3,y+1,1:FILL#ch,0
1482 FILL#ch,1:CIRCLE#ch,x+3,y+1,1:FILL#ch,0 : INK#ch,0
1483 LINE#ch,x-1.2,y-5 TO x+1,y-3.5 TO x-1,y-3.5 TO x+1.2,y-5
1484 LINE#ch,x-2,y-1 TO x-1.4,y-1:LINE#ch,x+1.4,y-1 TO x+2,y-1
1485 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)

```

1487 DEFine PROCEDURE Bull(ch,x,y)
1488 INK#ch,4:FILL#ch,1:ARC#ch,x-1,y+1 TO x+1,y+1,PI/4
1489 ARC#ch TO x+2.8,y, PI/2 TO x+.8,y-6,-PI/4 TO x-.8,y-6,PI/4
1490 ARC#ch TO x-2.8,y,-PI/4 TO x-1,y+1,PI/2:FILL#ch,0
1491 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
1492 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
1493 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
1494 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
1495 CIRCLE#ch,x-.8,y-4.8,.4:CIRCLE#ch,x+.8,y-4.8,.4
1496 END DEFine

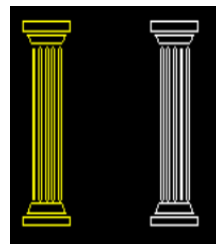
```



```

1498 DEFine PROCEDURE Pillar(ch,x,y)
1499 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
1500 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
1501 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
1502 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
1503 FOR c=1 TO 5
1504 x1=x-4+c*1.5-.3:x2=x-4+c*1.5+.3:y1=y+17:y2=y-17
1505 ARC#ch,x1,y1 TO x2,y1,-PI:LINE#ch TO x2,y2
1506 ARC#ch,x2,y2 TO x1,y2,-PI:LINE#ch TO x1,y1
1507 END FOR c
1508 END DEFine

```



```

1510 DEFine PROCEDURE TestMode

```

```

1511 SElect ON k

```

```

1512 =232:Stock_Update 25:wn=26

```

:REMark Resets to Start

F1 Reset to Wk

```

25

```

```

1513 =236:Stock_Update wn

```

:REMark Increments Weeks

F2 Add a Week

```

1514 =240:Wick 1,col,177,90,70,85,75:IF col=4:col=2:ELSE col=4

```

F3 Check Candle

```

Wick

```

```

1515 =244:FOR i=1 TO 8:PRINT#6,CN$(i);" ";RNew$(i):PAUSE 50:CLS#6

```

F4 Reuters News

```

Flash

```

```

1516 =248:Audit(0)=10000

```

```

1517 FOR i=1 TO 130:Audit(i)=Audit(i-1)+i*RND(-3 TO 7):END FOR I

```

F5 Game End

```

1518 wn=156:pm=1:TReview:wn=25:pm=0

```

```

1519 END SElect

```

```

1520 END DEFine

```