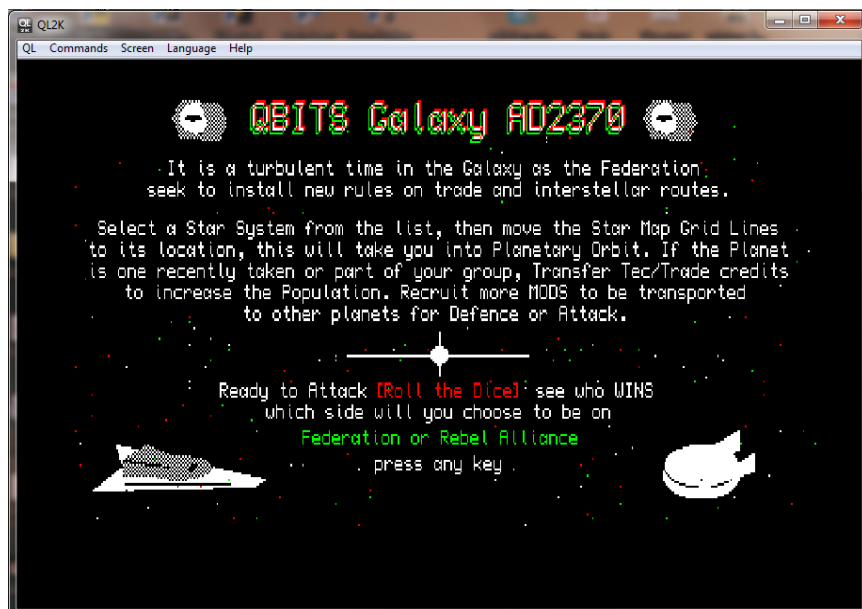


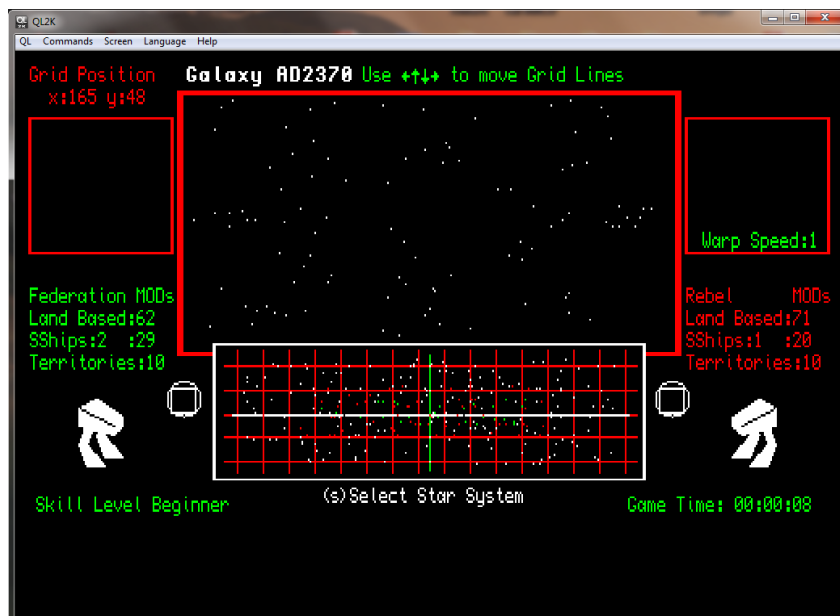
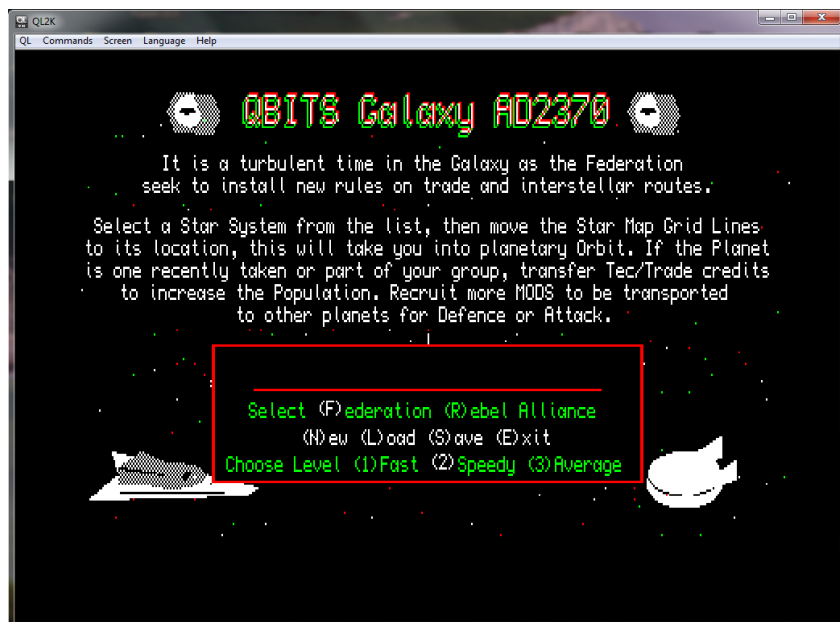


Sinclair QL retro gaming



Sinclair QL retro gaming







QBITS Galaxy AD2370

A Computer Game for me is a bit like writing a novel, moments of creativity generating the plot and sides of conflict, followed by long periods of refinement and then occasionally a flash of inspiration. Designing a Computer Game of course you need a few basic skills in programming. The next question do these meet the needs of creating the imagined environment your Game exists in and to the level of complexity you are aiming to achieve in it function. As to my thoughts behind Galaxy AD2370, the nineteen eights not only landed me with a QL computer and an interest in programming, but it was a time when I read and followed a lot of science fiction.

Early years began with H.G Wells The War of the Worlds, The Time Machine then the TV series of Star Trek and the film Star Wars. This must have energised my early thoughts and now reviving what was lost in the loft for years I can perhaps add Battlestar Galactica and StarGate to my list. Then there was always a keen interest in board games with family members, especially Risk with its later versions such as Risk AD2010.

So how and from where does a Computer Game or any Game idea for that matter begin to germinate. My computer Gaming started with tinkering and exploring the capabilities of simple QL commands. Then at some point manipulating arrays and filtering the outcomes to a desired result. That was where the background work is carried out in any program and very often is the boring and potentially very frustrating bit. What kept me absorbed and committed, back in the eighties, was exploring my creative ability with QL graphics.

Note On Programming Frustration!!!

Well I can remember one of my early programming attempts to load an Array. I had set out the lines of DATA and used RESTORE to set the beginning line number and DIM'ed my array identifier but didn't understand why my FOR loop to READ and load the DATA wasn't working.

After some moments of rant, I found out it was because I had placed my FOR loop in a called PROCEDURE before the line with my DIM array identifier and where the DATA was located. I now always set my DIM arrays at the beginning of a program before calling any PROC's or at least set them within the first line of their relevant PROCEDURE.

QBITS Galaxy AD2370 basic concept

My choice of background to this Game is the Star Systems of a Galaxy (far, far away, really, no I'm just kidding). The opposing forces are a straight choice between The Federation of Planets or Rebel Alliance. The intro page sets the scene and basic instructions as to the nature of the Game. At initialisation, some of the Star Systems are randomly allocated to either The Federation or Rebel Alliance. The other Star Systems are classified as independent civilisations. All are given various levels of Technological development and Trading abilities.

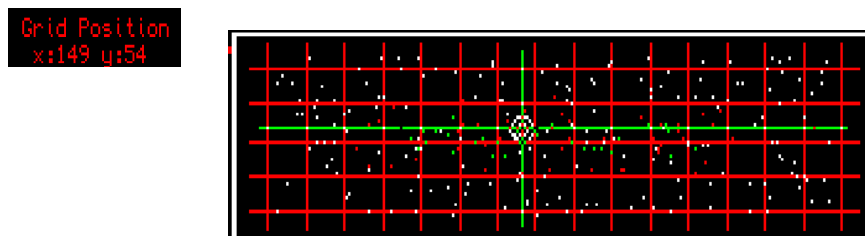
Galaxy AD2370 - The Game

The aim is to annihilate the opposition, namely take all of the Computer Player Star Systems so there are none left from which to recover and retaliate. The Computer will off course try to do likewise, by increasing its number of occupied Star Systems and taking any Star Systems it can from the Game Players. The level of complexity in arriving at such a result is dependent on the strategies employed. Therefore, whoever achieves this primary goal wins the Game.

The Game is reliant on a randomly generated number set 1-8, this is graphically represented as the throw of a dice. A major break at the start of play is how you are dealt your opening hand. 10 Territories are allocated to each side, these Star/Planet systems are randomly sourced with Tec/Trade credits & MODs (Machines Of Destruction) at initialisation. However, the total number of MODs is the same for both sides the difference always being made up as extra StarShip MODs given to the one with the least land based MODs.

Galaxy AD2370 The beginnings

My initial thoughts began with constructing a Galaxy Map, a screen window with random POINTs (stars) and a grid. The x-y coordinates reference being displayed on the screen in conjunction with the movement of vertical and horizontal lines.



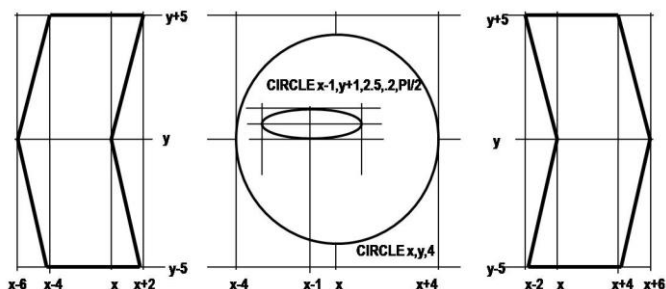
A Star System is identified by a double circle set over the x,y coordinates. The use of the cursor keys acting from within a REPEAT loop move horizontal & vertical Grid Lines to align at the crossover position...

QBITS Galaxy AD2370 StarShips & MODS

A number of ideas to incorporate StarShip screen images and the like was part of my agenda. Creating these can be done in several ways, pixel by pixel, generated from an image or as I do with the use of SuperBASIC commands ARC, CIRCLE, LINE, POINT commands. That is not forgetting the accompanying, INK, FILL, PI and use of SCALE.

Creating a StarShip

The range of designs are numerous, but my aim was to keep it simple. As a Child taking a set of points and linking them up to create a picture kept me busy for hours so employing this technique for designing was not unknown to me.

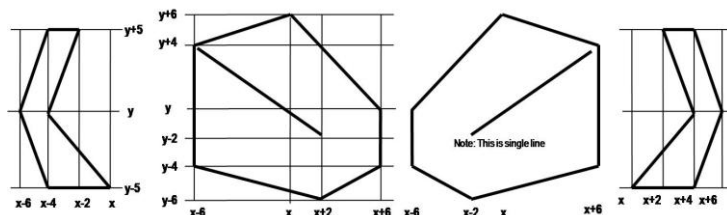


```
LINE x,y TO x+2,y+5 TO x-4,y+5 TO x-6,y TO x-4,y-5 TO x+2,y-5 TO x,y
LINE x,y TO x+2,y+5 TO x+4,y+5 TO x+6,y TO x+4,y-5 TO x-2,y-5 TO x,y
```

Individual elements of an image are drawn separately with their own +/- offsets to global settings and are brought together to create the full image. A PROCedure in which each element LINE or CIRCLE of an image has their own local x,y off-set to a global coordinate is the method I decided to employ (see page 20).

Creating the MODS

The LINE values again are straightforward to work out.



QBITS Galaxy AD2370 Space View Screen

As in any Computer adventure Game there is an expectation of a view screen, in this case the vastness of space depicted by moving through the stars. Here the use of SuperBASIC PAN and SCROLL commands are called in to play to create the illusion of movement. The simple aim was to move Star points away from the centre of the window towards the four corners. The first design was achieved by subdividing the view screen in to four quarters the PAN and SCROLL commands used to move POINTS (Star) directionally out from the centre.

After playing around with the number of generated Star POINTS the code below gave a respectable first result.

```
100 REMark QBGT01 (Test for Space View)
101 :
102 WINDOW 512,256,0,0:PAPER 0:CLS
103 DATA 304,152,98,19
104 DATA 150,75,100,20,150,75,250,20
105 DATA 150,75,100,95,150,75,250,95
106 RESTORE 103
107 :
108 FOR i=3 TO 7:OPEN#i,scr_:READ a,b,c,d:WINDOW#i,a,b,c,d
109 BORDER#3,1,2:INK#3,7
110 FOR i=1 TO 100:POINT#3,RND(10 TO 180),RND(10 TO 90)
111 REPEAT lp
112   FOR i=1 TO 6:POINT#3,RND(20 TO 180),RND(10 TO 90)
113   PAN#4,-2:SCROLL#4,-2:PAN#5,2:SCROLL#5,-2
114   PAN#6,-2:SCROLL#6,2:PAN#7,2:SCROLL#7,2
115   PAUSE 5
116   IF KEYROW(1)=8:FOR i=3 TO 7:CLOSE#i:STOP
117 END REPEAT lp
```



Note: At this point I should add that I develop most of my code using the QL2K emulator on a six years old Desktop running Windows 7. The speed is much faster than even a compiled program running on an original QL hardware. My guess is to run QBITS Galaxy AD 2370 at anything of a respectful speed you are going to need expanded memory and a later version of QL hardware or download and use an emulator such as QL2K or SML.

You had best note for my set up I use the PAUSE command a number of times to slow parts of the program. You might want to change these values to improve your speed.

Galaxy 2370 Planetary Orbit

So I next went exploring ways in which to create views to emulate approaching a Star System. Homing in on a Star I decided could be accomplished by an ever enlarging filled circle. Then coming into orbit above a planet required the curvature of the planet to rise up from the lower part of the screen.

QBITS Galaxy AD2370 Map & Space View Program

```
100 REMark QBGT03 (Test for Galaxy Map - Space View & Orbit)
```

```
104 WINDOW 512,256,0,0:PAPER 0:CLS
```

```
106 DATA 304,152,98,19
```

```
108 DATA 150,75,100,20,150,75,250,20
```

```
110 DATA 150,75,100,95,150,75,250,95
```

```
112 DATA 304,12,98,2
```

```
116 sx=164:sy=48:RESTORE 106:AT#2,18,0:PRINT#2,'QBG03'
```

```
120 InitWin:InitGrid::Main
```

```
124 DEFine PROCedure InitWin
```

```
126 FOR i=3 TO 8:OPEN#i,scr_:READ a,b,c,d:WINDOW#i,a,b,c,d
```

```
128 BORDER#3,1,2:INK#3,7:BORDER#8,1,2:INK#8,7
```

```
130 FOR i=1 TO 100:POINT#3,RND(10 TO 160),RND(10 TO 90)
```

```
132 END DEFine
```

Creating the Galaxy Map with its Stars and Grid was not difficult, moving the Map Grid Lines I used the XOR function of OVER.

```
136 DEFine PROCedure InitGrid
```

```
138 OPEN#13,scr_:WINDOW#13,260,60,120,170
```

```
140 OPEN#14,scr_:WINDOW#14,90,22,8,6
```

```
142 CLS#13:BORDER#13,1,7:INK#13,2:SCALE#13,100,0,0
```

```
144 FOR h=12 TO 96 STEP 18
```

```
146 LINE#13,6,h TO 326,h
```

```
148 END FOR h
```

```
150 FOR i=16 TO 330 STEP 20
```

```
152 LINE#13,i,3 TO i,98
```

```
154 END FOR i
```

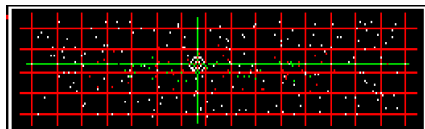
```
156 INK#13,2:FOR i=1 TO 60:POINT#13,RND(60 TO 260),RND(30 TO 70)
```

```
158 INK#13,4:FOR i=1 TO 30:POINT#13,RND(80 TO 240),RND(40 TO 60)
```

```
160 INK#13,7:FOR i=1 TO 180:POINT#13,RND(20 TO 310),RND(10 TO 90)
```

```
162 OVER#13,1:PRINT#13,' use the cursor keys to move grid line':OVER#13,0
```

```
164 END DEFine
```



```
168 DEFine PROCedure grid
```

```
170 IF sx< 10:sx= 10
```

```
172 IF sx>320:sx=320
```

```
174 IF sy< 5:sy= 5
```

```
176 IF sy> 95:sy= 95
```

```
178 OVER#13,-1:INK#13,4
```

```
180 LINE#13,sx,5 TO sx,95:LINE#13,12,sy TO 320,sy:OVER#13,0
```

```
182 CLS#14:INK#14,2:PRINT#14,'Grid Position\' x':sx,' y':sy
```

```
184 END DEFine
```

```
Grid Position
x:149 y:54
```

```

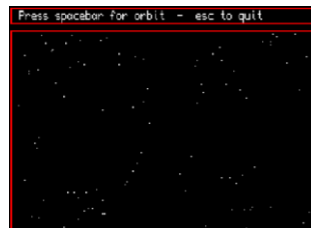
188 DEFine PROCEDURE Main
190 CLS#8:PRINT#8,' Press any key to enter orbit - esc to quit'
192 grid
194 REPEAT lp
196 StarMove
198 IF KEYROW(1)=8:FOR i=3 TO 20:CLOSE#:STOP
200 IF KEYROW(1)=64 :StarView:Eorbit:PAUSE:Lorbit
202 IF KEYROW(1)= 2 :IF sx>5 :grid:sx=sx-5 :grid
204 IF KEYROW(1)= 4 :IF sy<95 :grid:sy=sy+5 :grid
206 IF KEYROW(1)= 16 :IF sx<320 :grid:sx=sx+10 :grid
208 IF KEYROW(1)=128 :IF sy>5 :grid:sy=sy-10 :grid
210 END REPEAT lp
212 END DEFine

```

```

216 DEFine PROCEDURE StarMove
218 FOR i=1 TO 10:POINT#3,RND(10 TO 160),RND(10 TO 90)
220 PAN#4,-2:PAN#5,2:PAN#6,-2:PAN#7,2
222 SCROLL#4,-2:SCROLL#5,-2:SCROLL#6,2:SCROLL#7,2
224 PAUSE 5
226 END DEFine

```

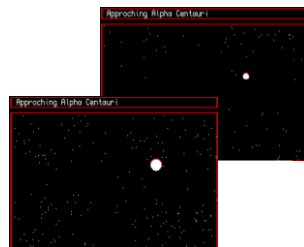


Note: This is the simple use of PAN and SCROLL although order and size of move values are important to get the right balance...

```

230 DEFine PROCEDURE StarView
232 CLS#8:PRINT#8,' Approching Alpha Centauri'
234 FOR a=1 TO 8
236 INK 0:FILL 1:CIRCLE 90,70,a*.6:FILL 0
238 x=90:y=70:m=a*.6:Star
240 FOR b=1 TO 6:POINT#3,RND(10 TO 160),RND(10 TO 90)
242 PAN#4,-1:SCROLL#4,-.5
244 PAN#6,-1:SCROLL#6,.5:PAN#7,1:SCROLL#7,.5
246 PAUSE 10
248 END FOR a
250 END DEFine

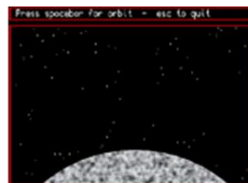
```



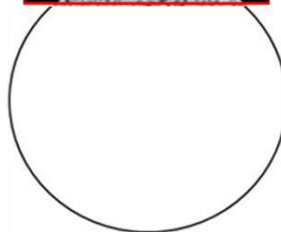
```

254 DEFine PROCEDURE Star
256 INK 2 :FILL 1:CIRCLE x,y,m*.1 :FILL 0
258 INK 241:FILL 1:CIRCLE x,y,m*.95:FILL 0
260 INK 7 :FILL 1:CIRCLE x,y,m*.9 :FILL 0
262 INK 241:FILL 1:CIRCLE x-m*.6,y-m*.4,m*.2,.3,PI/4:FILL 0
264 END DEFine

```



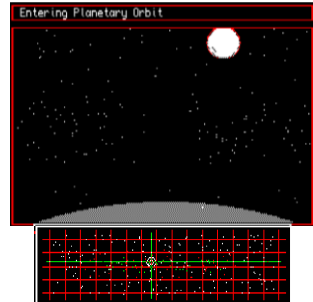
The illusion of a Planet crest rising is achieved basically by a large FILLED CIRCLE, which in SuperBASIC graphics resides off screen until it is SCROLL up to reveal its top.




```

268 DEFine PROCedure Eorbit
270 CLS#8:PRINT#8,' Entering Planetary Orbit'
272 FOR i=1 TO 20
274   SCROLL#3,-1:PAUSE 5
276   POINT#3,RND(10 TO 60),RND(5 TO 95)
278 END FOR i
280 PAUSE 5:INK#3,241
282 FOR p=-240 TO -190 STEP 2
284   FILL#3,1:CIRCLE#3,76,p,200:FILL#3,0:SCROLL#3,-1:PAUSE 5
286   POINT#3,RND(10 TO 160),RND(5 TO 10)
288 END FOR p
290 CLS#8:PRINT#8,' Press any key to leave Orbit'
292 END DEFine

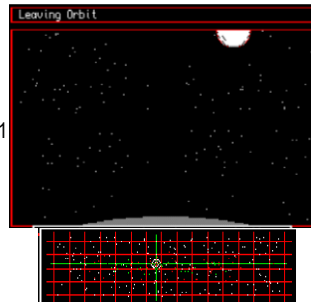
```



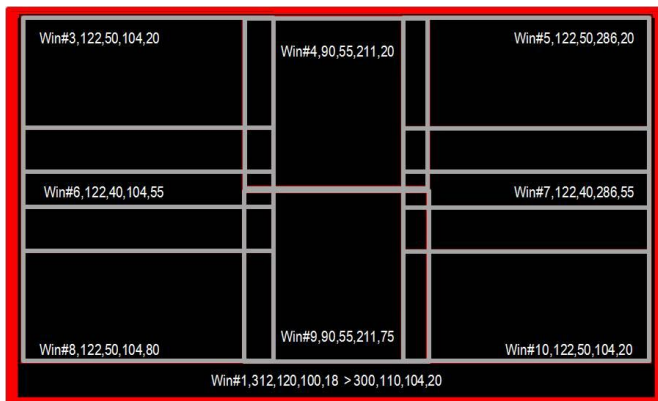
```

296 DEFine PROCedure Lorbit
298 CLS#8:PRINT#8,' Leaving Orbit'
300 CURSOR#3,0,60
302 FOR i=1 TO 10
304   SCROLL#3,2,2:POINT#3,RND(40 TO 120),RND(10 TO 90)
306   SCROLL#3,-1,1:PAN#4,-2:SCROLL#4,-1:PAN#5,2:SCROLL#5,-1
308   PAUSE 5
310 END FOR i
312 CLS#8:PRINT#8,' Press spacebar for orbit - esc to quit'
314 END DEFine

```



These early code investigations gave a starting point to develop further variations and later incorporate them as part of the full Game program. The view screen vertical was shortened and subdivided further into eight WINDOW's with overlaps to give flexibility with SCROLL & PAN and hopefully more realism to the illusion I was trying to create.



QBITS Galaxy AD2370 Intro

The Intro page sets the scene giving some insight to the Games purpose. There's no reason not to embellish it with an imaginative Title and background.

QBITS Title

I use my standard QBITS Graphic Style heading to give a 3D Character effect, only this time playing with various colours and repeating the line of my Title it gave some intriguing different colour arrangements.

```
128 WINDOW#1,512,256,0,0 :PAPER#1,0:CLS#1:CSIZE#1,2,1:OVER -1
130 INK#1,2:FOR i=1 TO 3:CORSOR#1,140+i,20-i:PRINT#1,'QBITS Galaxy AD2370'
132 INK#1,7:FOR i=1 TO 2:CORSOR#1,140+i,20-i:PRINT#1,'QBITS Galaxy AD2370'
134 OVER 0:CSIZE 0,0:
```

Try the code out for yourself, experiment with the INK value and i Offset.

QBITS Galaxy AD2370 Menu

This started out as a return to at End of Game Menu and as the Game progressed an interim access to expected (N)ew (L)oad (S)ave (E)xit options. Then I added the choice of side, Federation or Rebel Alliance. Skill Level options came later.

Note: Device_Filename\$ opens a file 'flp1_QBAD2370_data' to Load (INPUT) or Save (PRINT), which the user can alter for their own use.

Galaxy AD2370 Array values & usage

A String array **astro\$()** for the Star/Planets name. An Integer array **astro()** for the Star/Planet information:- Tec/Trade credits, MODs, Population, Territory, x,y Galaxy Map coordinates, Planet Surface Colour and Orbital Approach.

A integer array **side()** holding each opponents information:-Tec/Trade/MODs.

astro\$(n)	Star/Planet System name	
astro(n,1)	Tec	(1 to 12 credits)
(n,2)	Trade	(1 to 20 credits)
(n,3)	MODs	(0 to 20)
(n,4)	Pop	(0.50 to 10 billion to 2 decimal places)
(n,5)	Independent=0, Federation=1, Rebel Alliance=2	
(n,6)	x coordinate	(10 to 320)
(n,7)	y coordinate	(5 to 95)
(n,8)	Orbital Approach	(1 to 3) left/centre/right
(n,9)	Planet surface colour	(RND 220 to 254)

side(Gp1, 1)	Tec	(1 to 24 credits)	side(Gp2, 1)
(Gp1, 2)	Trade	(1 to 40 credits)	(Gp2, 2)
(Gp1, 3)	MODs	(4 to 100)	(Gp2, 3)
Human Player			Computer Player

Federation=1/ Rebel Alliance=2

It's a simple fact when keeping track of players turns and options it can get confusing. Giving the player the option to be on either the Federation or Rebel Alliance side just doubled the problem. When the human player Gp1 is playing or the computer player Gp2 is playing how to keep tabs on who's who and using what PROCedures.

Gp1= 1 Federation or 2 Rebels visa versa Gp2= 2 Rebels or 1 Federation

Gp2= 1 or 2 **side(Gp2,1/2/3)** Computer Player

The advantage of selecting the two opponents arrays in this way means when you SAVE and then re-LOAD you can if the player wishes swop sides.

At the beginning of a program, listing the String & Integer identifiers is probably good practice. It certainly helps when checking through programming code.

User Defined set user device_filename

Star System Info

Side Info

Delay / Dice enable / (S)ave block

Skill levels

Federation MODs, Territories, SShips

Rebels MODs, Territories, SShips

Game Player who's who / (R)ecruit

Attack!Transfer>Tec/Trade/MODs (R)ecruit

Galaxy Map /Player/Complay coordinates

delete MODs : warp speed : Message\$

Timers Test=1 code checks

QBITS Galaxy AD2370 Stratagems

What enhances a Game is to be able to access information and take decisions, RISK's as to your outcome. Developing a methodology, a stratagem to achieve success is the player's prerogative. My hope is **Galaxy AD2370** offers this...

The Information

My first objective was one of collective positions. A display of each opponents Land and StarShip based MODs and the number of Territories held. The information held in arrays that are initialised at the start then updated as the Game progresses.

```
Federation MODs
Land Based:38
SShips:2 :32
Territories:10
```

PROCedure MODInfo
Delivers this information
to the screen.

```
Rebel MODs
Land Based:50
SShips:1 :20
Territories:10
```

Star System

The Star name and Planetary information uses random numbers to initialise the entries at start of Game. Each Star/Planet system is rated for Technological advance (Tec 1 to 12), Trading abilities (1 to 20) and MODs (1 to 20) for defence. These are the first three entries and there's a reason for their order that which will hopefully become apparent. The next covered is the Population level (0.50 to 10 Billion), this is used in calculating the Tec/Trade credits and MODs status. During the Game these are exchanged in the Transfer phase of play. The fifth entry holds the ownership, Independent=0, Federation=1, Rebel Alliance=2.

The next four are the Galaxy Map x,y coordinates, Planet surface colour and direction of Orbital approach (These do not change during the game).

Game Play

The (s) Select Star System takes you to a display where using the up/down cursor keys you can scroll through all of the fifty Star Systems, providing Map x,y coordinates, Pop, Tec, Trade and MODs status. The Star System is chosen by pressing the spacebar.

```
Lesath
x:178 y:20

Pop :.75
Tec :0
Trade:1
MODs :1
```

PROCedure PLInfo

(Star) Planet 1 to 50 : Tec 1 to 12 : Trade 1 to 20 :MOD's 0 to 20

QBITS Galaxy AA2370 Opponents Status

A player's TEC/Trade credits & MODs are held by the integer array initialised at start of Game. To travel about the Galaxy each player needs a StarShip and to Attack a Star System they will need a number of MODs.

At start of Game 10 Territories each are allocated randomly to the Federation and the Rebel Alliance. The Federation and Rebel land based MODs are then tallied up and any differences compared. If both the same, 20 MODs are allocated to each opponents StarShips. If not the opponent's with the least number of land based MODs has the difference added to their 20 already allocated. Therefore at start of Game both opponents hold the same number of MODs.

Attacking Mode

On entering the Game the player starts by (s) Selecting a Star System. Moving the Galaxy Map Grid Lines to cross at the Star systems x,y coordinates takes you on that Star's approach and then into planetary orbit. If the Planet is governed Independently or by the Opponent side then you are Attacking.

Any key Rolls the Dice and the outcome decides which side loses MODs. The Attacker looses to lower or even scores to that of the defending Planet. If the Attacker has less than 4 MODs the attack is aborted and player sent out of Orbit. If on the other hand the Planet MODs are reduced to 0 then you take over the Star/Planet system. This awards you additional Tec/Trade credits and MODs

```
Federation
Attack
Tec :5
Trade:5
MODs :32
```

```
Federation
↑↓Trasfer + +
> Tec :7
Trade:8
MODs :37
(R)ecruit
```

PROCedure SSInfo

StarShips

1 to 5
Tec 1 to 24
Trade 1 to 40
MODs 4 to 100

Transfer Mode

If the Planet has been recently taken or already under your allegiance then you go into Transfer Mode. Use Cursor keys to select and Transfer Tec/Trade credits and MODs to/from the Planet. You can increase the Population by use of the (R)ecruit key to generate more MODs. These can be Transferred back to your StarShip for transport to other planets, further strengthening their defences or to carry out more Attacks. Be aware this may be your only means of raising MODS after heavy losses and to get back into Attack Mode.

QBITS Galaxy AD2370 Complay

For the computer to be a worthwhile opponent, have I been clever enough to employ a level of Artificial Intelligence. Rest assured I'm not that competent a programmer and my skills are limited. However, this area did require a good amount of forethought.

The obvious route to follow are the actions dictated by the Game. Namely seek out a Star System, Attack and/or Transfer Tec/Trade credits improve the Population and thereby generate and Recruit more MODs. However if this is to be truly representative, then the same fortunes and misfortunes of luck (rolling the dice) must apply as to a human player.

The Pro's and Con's of Conflict

At Game start if the computer is the one with the extra MODs assigned to their StarShip fleet, this is a straight advantage when sallying forth to Attack Star Systems. Yet the fickle hand of fate when rolling the dice can be both fortuitous and equally as quick lead to pending disaster.

Mechanisms to guard against the worst outcomes needs to be provisioned as part of the Computer Game Plan. Therefore, a number of checks need to be carried out before proceeding with an Attack. The number of StarShip MODs should maybe be three or four times larger than the Star Systems MODs. As the Game progress if the Computers StarShips MODs fall low in value, the computer should avoid attacking Star Systems without a definite advantage and seek to recruit more from its own Star/Planet systems.

If on the other hand when the Computer StarShip MODs are significantly high in number, maybe it should seek to prop up its own Planets defences by Transferring MODs to them. Transferring Tec & Trade credits and using the Recruit function built into the Game is the obvious way to improve the numbers of MODs.

Choice of Star Systems

For a Human player this is a relatively easy decision making choice. For the Computer however what becomes the criteria, does it check each and every Star System analysing and weighing the Pro's and Con's to some master plan of action or could a simpler use of random choice be suffice.

At this stage, the task of how and when the Computer Player could interact seemed a little beyond my programming capabilities so I turned to look at it from a different perspective.

QBITS Galaxy AD2370 Simple Turns

How and when to allow the computer to take its turn. I considered a number of options. The first and obvious choice is a straight flip-flop between Human and Computer. This as I might have hinted earlier, with the Game played on random numbers, will put the computer at a disadvantage whereby lesser actions of value are accomplished and lead to a relatively easy win by the human player.

play=1

IF play=1 AND sx=r AND sy=c : **StarView** : **Eorbit** : **GPlayer** : **Lorbit** : play=2

IF play=2 : **Complay** : play=1

sx,sy are the Galaxy Map Grid Line coordinates

r,c are the Star Systems coordinated relative to the Galaxy Map

GPlayer – the ID tag for human Game Player

Complay – the ID tag for my Computer Player

Alternative Turn Arrangements

Here my thoughts were on letting the computer compete for access in the main Game loop controlled by a random number...

IF sx=r AND sy=c : **StarView** : **Eorbit** : **GPlayer** : **Lorbit**

IF RND(1 TO 99)=13 : n=RND(1 TO 50) : **Complay**

My short experimentation down this route, depending on the choice and range of random numbers, proved ineffectual or at times a bit too easy for the computer to block the human player out of play altogether leading to an almost impossible task of ever winning.

It would appear I had been looking at opposite ends of the spectrum for what I was after. In the end, I decided on a simpler choice that of a timer that locked the Computer out of play for a finite time rather than a straight flip-flop or an inconsistent random period.

Variable Level

It followed that the value of delay could be increased or decreased to put pressure of action on the human player giving the pretence of a variable level of skill.

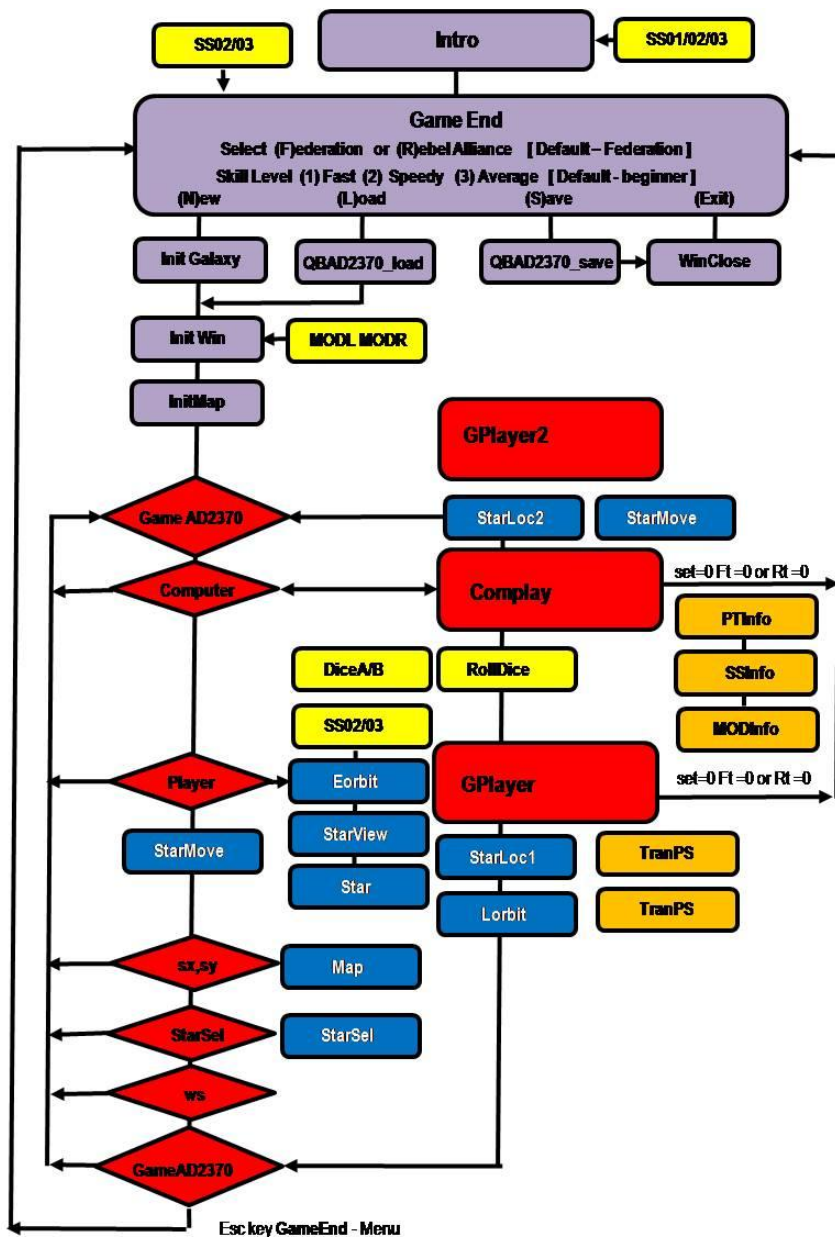
IF DATE>oldclock : **Complay** : oldclock = DATE+tim

Choose Skill Level (1)Fast (2)Speedy (3)Average or default (Beginner)

I think that's most of my early journey now covered it's over to the full program code of **QBITS Galaxy AD2370**. Any problems found, sensible suggestions on improvement or ways to expand the Games potential communicated to me will be most welcome.

ENJOY!

QBITS Galaxy AD2370 Flow Chart



QBIT Galaxy ADA2017 PROCedures

Intro	Introduction to the Game
EndGame	Main Menu (N)ew (L)oad (S)ave (E)xit /Choice of Side /Skill Level
QBAD2370_load	Load previously saved Game
QBAD2370_save	Save current Game for re-Load at a later time
InitGalaxy	Initialise the Star / Planet DATA
InitWin	Initialise the screens (WINDOWS 3 to 18)
WinClose	Close down the screens and clean up...
InitMap	Initialise the grid and stars of Galaxy Map
SS01	StarShip 01
SS02	StarShip 02
SS03	StarShip 03
MODL	MOD Left view
MODR	MOD Right view
RollDice	Dice throw of random numbers 1 to 8
DiceA	Square shot
DiceB	Diamond shot
GameAD2370	Main Game Loop
StarMove	PAN / SCROLL of screens 3 to 10 (Space View)
Map	Moves Grid line on Galaxy Stars Map
StarSel	Scroll through Stars list and select one
StarLoc1	Displays double circle over grid location of Selected Star
StarView	Star Approach
Star	Enlarging Star
Eorbit	PAN / SCROLL screen takes StarShips in to Planetary Orbit
Lorbit	PAN / SCROLL leaves Orbit.
GPlayer	Player decisions over Attack/ Transfer – Tec/Trade/MODs etc.
PTInfo	Planet :- Pop/Tec/Trade/ MODs
SSInfo	StarShip:- Pop/Tec/Trade/ MODs
MODInfo	MODs distribution of Federation and Rebel alliance and Territories
TranPS	Transfer Tec/Trade/MODs Plane t >Starship
TranSP	Transfer Tec/Trade/MODs Starship>Planet
Recruit	Re-Calculate MOD,s after increase in Credit
Victory	Updates Attacking Sides Tec/Trade Credits and MODs
ComPlay	Computer player decisions over Attack/ Transfer Tec/Trade/MODs
StarLoc2	Displays double circle over grid location of Selected Star
GPlayer2	This runs in the guise of a simulated human player to activate press (t)est mode from the Main Menu - checks out various code elements.

Note: GPlayer2 and Complay can play against each other to check strength of Stratagems.

QBIT Galaxy ADA2017 Code

```
100 REMark QBAD2370_Game - QBITS Galxay AD2370 v2.00 2017
101 :
102 REMark Use the Esc key to access Menu (N)ew (L)oad (S)ave (E)xit
103 :
104 device_filename$='flp1_QBAD2370_data': :REMark set user device_filename
105 :
106 DIM astro$(50,20),astro(50,9) :REMark Star/Planet DATA
107 DIM side(2,3) :REMark Players 1-2 DATA
108 :
109 ps=5:en=0:set=1 :REMark delay:Dice enable:block (S)ave
110 sk1=10:sk2=20:sk3=30 :REMark Skill Levels
111 FMOD=0:Ft=10:FSSn=1 :REMark Federation MODs:Territories:SShips
112 RMOD=0:Rt=10:RSSn=1 :REMark Rebel MODs:Territories:SShips
113 Gp=0:Gp1=0:Gp2=0 :REMark Game Player Computer Player
114 s=1:t=1:zr=1 :REMark Attack:Transfer>Tec/Trade/MODs:(R)ecruit
115 :
116 sx=168:sy=48:h1=-9:v1=-9:h2=-9:v2=-9 :REMark Galaxy Map coordinates
117 dm=3:ws=1:Mes$=' ' :REMark delete MODs:warp speed:message
118 Tm1=40:Tm2=5:Test=0 :REMark Timers Test=1 code checks
119 :
120 Intro :REMark Title page - Player Choice
121 :
122 DEFine PROCedure Intro
123 WINDOW#2,488,200,12,10:PAPER#2,0:CLS#2
124 WINDOW#1,512,256,0,0 :PAPER#1,0:CLS#1:CSIZE#1,2,1:OVER -1
125 INK#1,2:FOR i=1 TO 3:CUSOR#1,140+i,20-i:PRINT#1,'QBITs Galaxy AD2370'
126 INK#1,7:FOR i=1 TO 2:CUSOR#1,140+i,20-i:PRINT#1,'QBITs Galaxy AD2370'
127 OVER 0:CSIZE 0,0:SCALE 150,0,0:ink1=7:ink2=241
128 x=60 :y=132:SS01:x=180:y=132:SS01:x=180:y=38:SS02:x=20:y=34:SS03
129 SCALE 100,0,0:For i=1 to 250:ink i:POINT RND(10 to 40),(RND(15 to 90))
130 INK 7
131 CURSOR 90,46 :PRINT 'It is a turbulent time in the Galaxy as the Federation'
132 CURSOR 78,56 :PRINT 'seek to install new rules on trade and interstellar routes.'
133 CURSOR 48,74 :PRINT 'Select a Star System from the list, then move the Star Map Grid Lines'
134 CURSOR 42,84 :PRINT 'to its location, this will take you into Planetary Orbit. If the Planet'
135 CURSOR 42,94 :PRINT 'is one recently taken or part of your group, Transfer Tec/Trade credits'
136 CURSOR 64,104 :PRINT 'to increase the Population. Recruit more MODs to be transported'
137 CURSOR 136,114:PRINT 'to other planets for Defence or Attack.'
138 CURSOR 130,150:PRINT 'Ready to Attack' see who WINS: :INK 2
139 CURSOR 130,150:PRINT '[Roll the Dice].': :INK 7
140 CURSOR 150,160:PRINT 'Which side will you choose to be on:': INK 4
141 CURSOR 172,172:INK 4:PRINT 'Federation or Rebel Alliance':INK 7
142 CURSOR 216,184:PRINT 'press any key'
143 For i=1 to 50:INK i:POINT RND(25 to 125),RND(40 to 50)
144 INK 7:FILL 1:CIRCLE 74,46,1.5,L:FILL
145 LINE 58,46 to 90,46:LINE 74,50 to 74,42::PAUSE:EndGame
146 END DEFine
```



148 DEFine PROCEDURE EndGame

```
149 ch=1:G=0:Gp1=1:Gp2=2:PAPER#ch,0:INK#ch,0:WINDOW#ch,268,62,122,132
150 FOR i=1 TO 20:FILL#ch,1:CIRCLE#ch,160,50,i*3:FILL#ch,0:PAUSE ps/5
151 BORDER#ch,1,2:CLS#ch:INK#ch,7: LINE#ch,30,68 to 300.68: CURSOR#ch,46,8:
152 IF set=0 AND Ft=0
153   INK#ch,7:PRINT#ch, 'Rebel Alliance wins the Game'
154   x=30:y=80:SS02:x=290:y=80:SS02
155   ink1=7:x=30:y=80:SS02:x=290:y=80:SS02:PAUSE 50
156 END IF
157 IF set=0 AND Rt=0
158   INK#ch,7:PRINT#ch, ' Federation wins the Game'
159   x=20:y=80:SS03:x=260:y=80:SS03
160   ink1=7:ink2=241:x=20:y=80:SS03:x=260:y=80:SS03:PAUSE 50
159 END IF
160 INK#ch,2:CURSOR#ch,20,24:PRINT#ch,'Select (F)ederation (R)ebel Alliance'
161 INK#ch,4:CURSOR#ch, 6,24:PRINT#ch,'Choose Level (1)Fast (2)Speedy (3)Average'
162 INK#ch,7:CURSOR#ch,52,36:PRINT#ch,'(N)ew (L)oad (S)ave (E)xit'
163 REPEAT lp
164   k=CODE(INKEY$(-1))
165   SELECT ON k
166     =49:IF Tm1=40:Tm1=sk1:CURSOR#ch, 84,47:PRINT#ch,'(1)' :REMark Fast
167     =50:IF Tm1=40:Tm1=sk2:CURSOR#ch,132,47:PRINT#ch,'(2)' :REMark Mid
168     =51:IF Tm1=40:Tm1=sk3:CURSOR#ch,192,47:PRINT#ch,'(3)' :REMark Slow
169     =70,102: IF G=0:G=1:Gp1=1:Gp2=2:CURSOR#ch, 62,23:PRINT#ch,(F)
170     =82,114: IF G=0:G=1:Gp1=2:Gp2=1:CURSOR#ch,140,23:PRINT#ch,(R)
171     =78,110:set=0 :InitGalaxy:InitWin:InitMap:GameAD2370 :REMark (N)ew
172     =76,108:set=0 :QBAD2370_load:InitWin:InitMap:GameAD2370 :REMark (L)oad
173     =83,115:IF set=0 :QBAD2370_save:WinClose :REMark (S)ave
174     =69,101:WinClose :REMark (E)xit
175     =84,116:AT#1,0,0:IF Test=0:Test=1:PRINT#1,'Test':ELSE Test=0:CLS#1,3
176   END SELECT
177 END REPEAT lp
178 END DEFine
```



Note: G=0/1 Defaults choices for Game Player are Federation and Skill Level - Beginner.

180 DEFine PROCEDURE QBAD2370_load

```
181 AT#2,19,10:PRINT#2,'Loading';
182 OPEN _IN#99,device_filename$
183 FOR a=1 TO 50
184   PRINT#2,'':INPUT#99,astro$(a)
185   FOR b=1 TO 9:INPUT#99,astro(a,b)
186   END FOR a
187 FOR a=1 TO 2192
188   PRINT#2,'':INPUT#99,side(a,b)
189   END FOR a
190 CLOSE#99:AT#2,19,10:CLS#2,4
191 END DEFine
```



193 DEFine PROCEDURE QBAD2370_save

```
194 AT#2,19,10:PRINT#2,'Saving.';
195 DELETE device_filename$
196 OPEN _NEW#99,device_filename$
197 FOR a=1 TO 50
198   PRINT#2,'':PRINT#99,astro$(a)
199   FOR b=1 TO 9:PRINT#99,astro(a,b):END FOR b
200 END FOR a
201 FOR a=1 TO 2
202   FOR b=1 TO 3:PRINT#99,side(a,b) :END FOR b
203 END FOR a
204 CLOSE#99:AT#2,19,10:CLS#2,4
205 END DEFine
```



```

207 DATA 'Vega','Bellatrix','Zibal','Castor','Naos'
208 DATA 'Tabit','Diphda','Elnath','Acrux','Errai'
209 DATA 'Polaris','Homan','Rigel','Sabik','Sarin'
210 DATA 'Furud','Gienah','Hadar','Propus','Izar'
211 DATA 'Enif','Algol','Mirzarm','Libeta','Atria'
212 DATA 'Keid','Kochab','Lesath','Ankaa','Marsic'
213 DATA 'Meissa','Subra','Menkib','Altair','Muscida'
214 DATA 'Cursa','Nashira','Nunki','Ogma','Pollux'
215 DATA 'Sirius','Merack','Serena','Denab','Tureis'
216 DATA 'Beid','Wesen','Yildun','Caph','Zosma'

```

217 DEFINE PROCEDURE InitGalaxy

```

218 RESTORE 206:AT#2,19,8:PRINT#2,'Initilising';
219 FOR n=1 TO 50
220 READ a$:astro$(n)=a$:PRINT#2,'';
221 pop=RND(2 TO 20)/RND(2 TO 8):IF pop<.5:pop=.5
222 astro(n,1)=INT(pop*1.2) :REMark Tec
223 astro(n,2)=INT(pop*2) :REMark Trade
224 astro(n,3)=INT(pop*2) :REMark MODs
225 astro(n,4)=pop :REMark Pop
226 astro(n,5)=0 :REMark Side
227 astro(n,6)=n*6+10 :REMark x coordinate
228 astro(n,7)=RND(10 TO 90) :REMark y coordinate
229 astro(n,8)=RND(220 TO 254) :REMark Planet Colour
230 astro(n,9)=RND(1 TO 3) :REMark Orbit Approach
231 END FOR n
232 REMark astro(n,5) 0=Independent 1=Federation 2=Rebel Alliance
233 FOR i=1 TO 10
234 n=RND(15 TO 35):IF astro(n,5)=0:astro(n,5)=1:ELSE GO TO 234
235 END FOR i
236 FOR i=1 TO 5
237 n=RND(5 TO 20):IF astro(n,5)=0:astro(n,5)=2:ELSE GO TO 237
238 n=RND(35 TO 50):IF astro(n,5)=0:astro(n,5)=2:ELSE GO TO 238
239 END FOR i
240 FOR n=1 TO 50:IF astro(n,5)>0 AND astro(n,3)<3:astro(n,3)=3
241 FOR p=1 TO 2:side(p,1)=5:side(p,2)=5:side(p,3)=0
242 END DEFINE

```



WINDOW#1

Space view screen

WINDOW#2

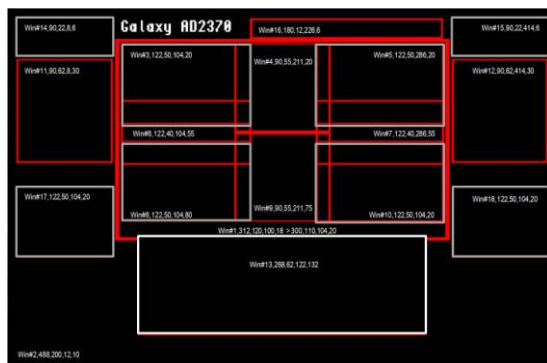
Background screen

WINDOW#3 to 10

Star movement screens

WINDOW#11 to 18

Information screens



```

244 DATA 122,50,104,20,90,55,211,20,120,50,286,20      :REMark win 3-10
245 DATA 122,40,104,55,120,40,286,55
246 DATA 122,50,104,80,90,55,211,75,120,50,286,80
247 DATA 90,62,8,30,90,62,414,30,268,62,122,132      :REMark win 11-13
248 DATA 90,22,8,6,90,22,414,6,192,12,214,6         :REMark win 14-16
249 DATA 90,42,8,106,90,42,414,106                  :REMark win 17-18

```

251 DEFine PROCEDURE InitWin

```

252 WINDOW#1,512,256,0,0:CLS#1:for i=3 to 18:CLOSE#i
253 OPEN#3,scr_112x10a104x6:CSIZE#3,1,0:INK#3,7:OVER#3,1
254 FOR i=1 TO 2:CURSOR#3,i,0:PRINT#3,'Galaxy AD2370'
255 OVER#3,0:CLOSE#3
256 RESTORE 244
257 FOR i=3 TO 18
258 OPEN#i,scr_:READ a,b,c,d:WINDOW#i,a,b,c,d
259 IF i=11 OR i=12:BORDER#i,1,2
260 END FOR i
261 ch=1:WINDOW#ch,312,120,100,18:BORDER#ch,2,2:PAPER#ch,0:CLS
262 WINDOW#ch,300,110,104,20
263 INK#ch,7:FOR i=1 TO 120:POINT#ch,RND(10 TO 200),RND( 5 TO 95)
264 MODL:MODR:l=2.8:dy=26:dx=34:DiceA:dx=146:DiceA:m=1
265 INK#2, 7:CURSOR#2,176,186:PRINT#2,'(s)Select Star System'
266 IF Tm1=10:Tm$='Fast'
267 IF Tm1=20:Tm$='Speedy'
268 IF Tm1=30:Tm$='Average'
269 IF Tm1=40:Tm$='Beginner'
270 INK#2,4:AT#2,19,0:PRINT#2,'Skill Level ',Tm$
271 INK#16,4:PRINT#16,'Use <←→→→> to move Grid Lines'
272 END DEFine

```

Note: Screen display of Skill Level

274 DEFine PROCEDURE WinClose

```

275 CLOSE#99:FOR w=3 TO 18:CLOSE#w
276 WINDOW 512,256,0,0:PAPER 0:CLS
277 WINDOW 490,220,16,8:PRINT#0,'bye...':STOP
278 END DEFine

```

280 DEFine PROCEDURE InitMap

```

281 CLS#13:BORDER#13,1,7:INK#13,2:SCALE#13,100,0,0
282 FOR h=12 TO 96 STEP 18
283 LINE#13,6,h TO 326,h
284 END FOR h
285 FOR v=16 TO 330 STEP 20
286 LINE#13,v,3 TO v,98
287 END FOR v
288 INK#13,2:FOR i=1 TO 60 :POINT#13,RND(60 TO 260),RND(30 TO 70)
289 INK#13,4:FOR i=1 TO 30 :POINT#13,RND(80 TO 240),RND(40 TO 60)
290 INK#13,7:FOR i=1 TO 180:POINT#13,RND(20 TO 310),RND(10 TO 90)
291 CLS#14:INK#14,2:PRINT#14,'Grid Position'
292 END DEFine

```

294 **DEFine PROCedure SS01**

295 ch=1:REMark x=20:y=40

296 x=x-9:y=y+2:INK#ch,241:FILL#ch,1:LINE#ch,x-1,y TO x+1,y+5 TO x-4,y+5 TO x-6,y TO x-4,y-5 TO x+1,y-5 TO x-1,y:FILL#ch,0

297 INK#ch,7:FILL#ch,1:CIRCLE#ch,x,y,4:FILL 0

298 INK#ch,0:FILL#ch,1:CIRCLE#ch,x-1,y,2.5,.2,PI/2:FILL#ch,0

299 x=x+2:y=y-1:INK#ch,241:FILL#ch,1:LINE#ch,x+1,y TO x-1,y+5 TO x+4,y+5 TO x+6,y TO x+4,y-5 TO x-1,y-5 TO x+1,y:FILL#ch,0:INK#ch,7

300 **END DEFINE**



302 **DEFine PROCedure SS02**

303 ch=1:INK#ch,ink1 :REMark x=120:y=40

304 FILL#ch,1:CIRCLE#ch,x,y,10,.6,PI/2 :FILL#ch,0

305 FILL#ch,1:CIRCLE#ch,x,y-2,10,.6,PI/2:FILL#ch,0

306 INK#ch,0:CIRCLE#ch,x,y+1,10,.6,PI/2:INK#ch,ink1

307 x=x-6:y=y:FILL#ch,1:LINE#ch,x,y TO x+16,y+10 TO x+16,y+5 TO x,y:FILL#ch,0

308 x=x+4:y=y-6:FILL#ch,1:LINE#ch,x,y TO x+16,y+10 TO x+16,y+5 TO x,y:FILL#ch,0

309 **END DEFINE**



311 **DEFine PROCedure SS03**

312 ch=1:INK#ch,ink1 :REMark x=50:y=40

313 FILL#ch,1:LINE#ch,x,y TO x+8,y+6 TO x+6,y+12 TO x+45,y+3 TO x+42,y+1 TO x,y:FILL#ch,0:INK#ch,ink2

314 FILL#ch,1:LINE#ch,x+8,y+8 TO x+14,y+12 TO x+24,y+8 TO x+20,y+4 TO x+8,y+4 TO x+8,y+8:FILL#ch,0

315 FILL#ch,1:CIRCLE#ch,x+22,y+6,9,.35,PI/2:FILL#ch,0:INK#ch,0

316 FILL#ch,1:CIRCLE#ch,x+26,y+6,.5,3,PI/4:FILL#ch,0

317 LINE#ch,x+8,y+8 TO x+20,y+6:LINE#ch,x+36,y+1.5 TO x+8,y+1.5:INK#ch,7

318 **END DEFINE**



320 **DEFine PROCedure MODL**

321 ch=2:INK#ch,7:x=20:y=18

322 x=x:y=y:FILL#ch,1:LINE#ch,x,y-5 TO x-4,y-5 TO x-6,y TO x-4,y+5 TO x-2,y+5 TO x-4,y TO x,y-5:FILL#ch,0

323 x=x-4:y=y-2:FILL#ch,1:LINE#ch,x,y-5 TO x-4,y-5 TO x-6,y TO x-4,y+5 TO x-2,y+5 TO x-4,y TO x,y-5:FILL#ch,0

324 x=x-2:y=y+6:FILL#ch,1:LINE#ch,x,y+4 TO x-4,y+2 TO x-4,y-2 TO x+4,y-4 TO x+6,y-2 TO x+6,y TO x,y+4:FILL#ch,0:INK#ch,0:LINE#ch,x+4,y-2 TO x-4,y+2:INK#ch,7

325 **END DEFINE**



327 **DEFine PROCedure MODR**

328 ch=2:INK#ch,7:x=160:y=18

329 x=x:y=y:FILL#ch,1:LINE#ch,x,y-5 TO x+4,y-5 TO x+6,y TO x+4,y+5 TO x+2,y+5 TO x+4,y TO x,y-5:FILL#ch,0

330 x=x+4:y=y-2:FILL#ch,1:LINE#ch,x,y-5 TO x+4,y-5 TO x+6,y TO x+4,y+5 TO x+2,y+5 TO x+4,y TO x,y-5:FILL#ch,0

331 x=x+2:y=y+6:FILL#ch,1:LINE#ch,x,y+4 TO x+4,y+2 TO x+4,y-2 TO x-4,y-4 TO x-6,y-2 TO x-6,y TO x,y+4:FILL#ch,0:INK#ch,0:LINE#ch,x-4,y-2 TO x+4,y+2:INK#ch,7

332 **END DEFINE**



334 DEFine PROCedure RollDice

```

335 ch=2:REMark en=0/n GPlayer en=1/cn Complay en=3 GPlayer2
336 IF en=0 OR en=3:dn=n:ELSE dn=cn
337 IF en=0:PRINT#16,'To Roll Dice press any key': PAUSE
338 FOR a=1 TO RND(4 TO 6)
339   dx= 34:DiceA:PAUSE ps:DiceB
340   dx=146:DiceA:PAUSE ps:DiceB
341   IF en=1:StarMove
342 END FOR a
343 dn1=RND(1 TO 8):dx= 34:DiceA:CURSOR#ch,dx,dy,-3,-4:PRINT#ch,dn1
344 dn2=RND(1 TO 8):dx=146:DiceA:CURSOR#ch,dx,dy,-3,-4:PRINT#ch,dn2
345 IF dn1>dn2:astro(dn,3)=astro(dn,3)-dm
346 IF dn1=dn2:side(Gp,3)=side(Gp,3)-dm
347 IF dn2>dn1:side(Gp,3)=side(Gp,3)-dm
348 IF astro(dn,3)<0:astro(dn,3)=0
349 END DEFine

```



351 DEFine PROCedure DiceA

```

352 INK#ch,0:FILL#ch,1:CIRCLE#ch,dx,dy,4:FILL#ch,0
353 INK#ch,7:CIRCLE#ch,dx,dy,3.9
354 LINE#ch,dx-l,dy+l TO dx+l,dy+l TO dx+l,dy-l TO dx-l,dy-l TO dx-l,dy+l
355 END DEFine

```



357 DEFine PROCedure DiceB

```

358 INK#ch,0:FILL#ch,1:CIRCLE#ch,dx,dy,4:FILL#ch,0
359 INK#ch,7:CIRCLE#ch,dx,dy,3.9
360 LINE#ch,dx,dy+l TO dx+l,dy TO dx,dy-l TO dx-l,dy TO dx,dy+l
361 END DEFine

```



363 DEFine PROCedure GameAD2370

```

364 Sclk=Date:oldclk1=DATE+60:oldclk2=DATE+60:MODInfo
365 h1=-10:v1=-10:n=25:sx=165:sy=48:Map
366 REPEAT lp
367   IF sx=h1 AND sy=v1:StarView:Eorbit:GPlayer:Lorbit
368   IF DATE>oldclk1:Complay:oldclk1=DATE+Tm1
369   IF Test=1 AND DATE>oldclk2:GPlayer2:oldclk2=DATE+Tm2
370   StarMove:Sclk$=Date$(DATE-Sclk)
371   INK#2,4:AT#2,19,61:PRINT#2,'Game Time: ',Sclk$(13 to 20)
372   k=CODE(INKEY$(5))
373   SELECT ON k

```

Note: Game Time addition



```

374   =27 :Tm1=40:EndGame           :REMark (N)ew (L)oad (S)ave (E)xit
375   =32 :ws=ws+1:IF ws>9:ws=1      :REMark Warp speed spacebar
376   =83,115:StarSel:StarLoc1:oldclock=DATE+Tm1 :REMark StarLoc1 Off
377   =192:IF sx>10 :Map:sx=sx-ws :Map :REMark move left
378   =200:IF sx<320 :Map:sx=sx+ws :Map :REMark move right
379   =208:IF sy<95 :Map:sy=sy+ws :Map :REMark move up
380   =216:IF sy>5 :Map:sy=sy-ws :Map :REMark move down
381   =232:IF Test=1 :StarView:Eorbit :PAUSE:Lorbit :REMark F1 - Orbit Checks
382   =236:IF Test=1 : :REMark F2 - Warp Jump
383   END SELECT
384 END REPEAT lp
385 END DEFine

```

387 **DEFine PROCEDURE StarMove**

```
388 hoz=ws:ver=ws:
389 l en<2:CURSOR#12,2,50:PRINT#12, ' Warp Speed:',ws
390 IF ws>0:INK#1,7:FOR w=1 TO ws*2:POINT#1,RND(40 TO 160),RND(30 TO 70)
391 SCROLL#4,-ver:SCROLL#9,ver:PAN#6,-hoz:PAN#7,hoz
392 PAN#3,-hoz:SCROLL#3,-ver:PAN#5,hoz:SCROLL#5,-ver
393 PAN#8,-hoz:SCROLL#8,ver:PAN#10,hoz:SCROLL#10,ver
394 END DEFine
```



396 **DEFine PROCEDURE Map**

```
397 IF sx< 10:sx= 10
398 IF sx>320:sx=320
399 IF sy> 95:sy= 95
400 IF sy< 5:sy= 5
401 ch=13:INK#ch,4
402 OVER#ch,-1:LINE#ch,sx,5 TO sx,95:LINE#ch,12,sy TO 320,sy:OVER#ch,0
403 ch=14:INK#ch,2:CURSOR#ch,12,10:PRINT#ch,'x':sx,' y':sy:CLS#ch,4
404 END DEFine
```



406 **DEFine PROCEDURE StarSel**

```
407 ch=16:CLS#ch:PRINT#ch,'Select Star System ⬆️⬇️':BLOCK#ch,16,4,136,4,4
408 PTInfo:StarLoc1 :REMark StarLoc1 On
409 REPeat ss
410 k=CODE(INKEY$(-1))
411 IF k=208:IF n>1 :n=n-1:PTInfo
412 IF k=216:IF n<50:n=n+1:PTInfo
413 IF k= 32:h1=astro(n,6):v1=astro(n,7):EXIT ss
414 END REPeat ss
415 ch=16:CLS#ch:PRINT#ch,'Match Grid x,y ⇐⇨⬆️⬇️⬆️⇨ Star x,y'
416 END DEFine
```



418 **DEFine PROCEDURE StarLoc1**

```
419 ch=13:INK#ch,7:OVER#ch,-1:CIRCLE#ch,h1,v1,2.5:CIRCLE#ch,h1,v1,6:OVER#ch,0
420 END DEFine
```



422 **DEFine PROCEDURE StarView**

```
423 ch=16:PRINT#ch,'Approching ',astro$(n):ch=1
424 FOR i=1 TO 20
425 INK#ch,0:FILL#ch,1:CIRCLE#ch,100,70,i*.6:FILL#ch,0
426 x=100:y=70:m=i*.5:Star:PAUSE ps
427 END FOR i
428 END DEFine
```



430 **DEFine PROCEDURE Star**

```
431 INK#ch,2 :FILL#ch,1:CIRCLE#ch,x,y,m*.1 :FILL#ch,0
432 INK#ch,241:FILL#ch,1:CIRCLE#ch,x,y,m*.95:FILL#ch,0
433 INK#ch,7 :FILL#ch,1:CIRCLE#ch,x,y,m*.9 :FILL#ch,0
434 INK#ch,241:FILL#ch,1:CIRCLE#ch,x-m*.6,y-m*.4,m*.2,.3,PI/4:FILL#ch,0
435 END DEFine
```



437 DEFine PROCEDURE Eorbit

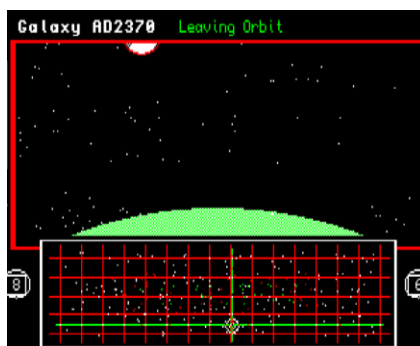
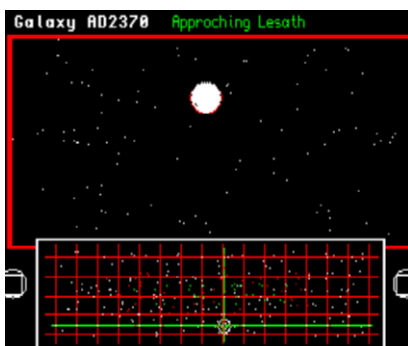
```

438 ch=16:CLS#ch:PRINT#ch,astro$(n);' - Planetary Orbit':PAUSE 10:INK#1,7
439 FOR i=1 TO 20
440 IF astro(n,9)=1:PAN#1,-2:SCROLL#1,-1:POINT#1,RND(180 TO 200),RND(5 TO 95)
441 IF astro(n,9)=3:PAN#1, 2:SCROLL#1,-1:POINT#1,RND(5 TO 20),RND(5 TO 90)
442 IF astro(n,9)=2
443 PAN#3,-2:SCROLL#3,-1:SCROLL#4,-1:PAN #5,2:SCROLL#5,-1:PAN#6,-2
444 PAN#7,2:PAN#8,-2:SCROLL#8,1:SCROLL#9,1:PAN #10,2:SCROLL#10,1
445 POINT#1,RND(40 TO 160),RND(30 TO 70)
446 END IF
447 PAUSE ps
448 END FOR i
449 INK#1,astro(n,8)
450 FOR i=-200 TO -186
451 FILL#1,1:CIRCLE#1,102,i,200:FILL#1,0:PAUSE ps
452 END FOR i
453 ink1=7:ink2=241:IF Gp1=1:x=40:y=50:SS03:ELSE x=160:y=50:SS02
454 END DEFine

```



Note: As an added touch the StarShip graphics for Federation or Rebel Alliance can be dropped in to Orbit above the planet. And removed when leaving orbit.



456 DEFine PROCEDURE Lorbit

```

457 ch=16:CLS#ch:PRINT#ch,'Leaving Orbit':PAUSE 20
458 ink1=0:ink2=0:IF Gp1=1:x=40:y=50:SS03:ELSE x=160:y=50:SS02
459 CURSOR#1,0,60:INK#1,7
460 FOR i=1 TO 20
461 SCROLL#1,-1,1:PAN#6,-1:POINT#1,RND(10 TO 200),RND(10 TO 60)
462 SCROLL#1, 1,2:PAN#7, 1:POINT#1,RND(10 TO 200),RND(10 TO 60)
463 PAUSE ps
464 END FOR i
465 StarLoc1:h1=-9:v1=-9 :REMark StarLoc1 off
466 ws=1:CLS#11:CLS#12:CLS#15:CLS#16
467 END DEFine

```



```

469 DEFiNE PROCeDURE GPlAYER
470 s=1:t=1:zr=1
471 REPeat act
472 IF Ft=0 OR Rt=0:EndGame
473 IF s=1 AND astro(n,5)=Gp1:s=0:Mes$='↑↕ Transfer ⇐⇒ ':SSInfo
474 IF s=1 AND astro(n,3)=0:Victory
475 IF s=1 AND side(Gp1,3)>dm+1
476 Mes$=' Attack' :SSInfo :Gp=Gp1:en=0:RollDice:SSInfo:CLS#16
477 END IF
478 IF s=1 AND side(Gp1,3)<dm+1
479 s=0:ch=16:CLS#ch:PRINT#ch,'Not enough MODs': PAUSE 20:CLS#ch:RETurn
480 END IF
481 IF s=0:ch=11:BLOCK#ch,10,30,0,20,0:AT#ch,t+1,0:PRINT#ch,'>'
482 k=CODE(INKEY$(5))
484 SELeCT ON k
485 =192:TranPS
486 =200:TranSP
487 =208:t=t-1:IF t<1:t=1
488 =216:t=t+1:IF t>3:t=3
489 = 32:RETurn
488 =82,114:IF s=0 AND zr=1:Recruit:zr=0
490 END SELeCT
492 END REPeat act
492 END DEFiNE

```



```

494 DEFiNE PROCeDURE PTInfo
495 ch=12:CLS#15 :side$=' '
496 IF astro(n,5)=1:side$='Federation '
497 IF astro(n,5)=2:side$='Rebel Alliance'
498 PRINT#15,astro$(n)
499 PRINT#15,' x:':astro(n,6):' y:':astro(n,7)
500 CLS#ch:INK#ch,4:Pop$=astro(n,4)
501 PRINT#ch,side$
502 PRINT#ch,' Pop :':Pop$(1 TO 4)
503 PRINT#ch,' Tec :':astro(n,1)
504 PRINT#ch,' Trade:':astro(n,2)
505 PRINT#ch,' MODs :':astro(n,3)
506 END DEFiNE

```

```

Lesath
x:178 y:20

Pop :.75
Tec :0
Trade:1
MODs :1

```



```

508 DEFiNE PROCeDURE SSInfo
509 MODInfo:PTInfo
510 ch=11:CLS#ch:INK#ch,4
511 IF Gp1=1:Gp$='Federation ' :ELSE Gp$='Rebel Alliance'
512 PRINT#ch,Gp$
513 PRINT#ch,Mes$
514 PRINT#ch,' Tec : ':side(Gp1,1)
515 PRINT#ch,' Trade: ':side(Gp1,2)
516 PRINT#ch,' MODs: ':side(Gp1,3)
517 IF s=0 AND zr=1:PRINT#ch,' (R)ecruit '
518 END DEFiNE

```

```

Federation
Attack
Tec :5
Trade:5
MODs :32

```

```

Federation
↑↕Trasfer + +
> Tec :7
Trade:8
MODs :37
(R)ecruit

```



520 DEFine PROCEDURE MODInfo

```
521 FMOD=0:Ft=0:FSSn=0:RMOD=0:Rt=0:RSSn=0
522 FOR i=1 TO 50:IF astro(i,5)=1:Ft=Ft+1:FMOD=FMOD+astro(i,3)
523 FOR i=1 TO 50:IF astro(i,5)=2:Rt=Rt+1:RMOD=RMOD+astro(i,3)
524 IF m=1:MODInit:m=0
525 FSSn=1+INT(side(1,3)/25):RSSn=1+INT(side(2,3)/25):CLS#17:CLS#18
526 IF Gp1=1:Fch=17:Fink=4:Sch=18:Sink=2:ELSE Fch=18:Fink=2:Sch=17:Sink=4
527 INK#Fch,Fink:PRINT#Fch,'Federation MODs'\Land Based:':FMOD'SShips:':FSSn:
':side(1,3)\Territories:':Ft
528 INK#Sch,Sink:PRINT#Sch,'Rebel MODs'\Land Based:':RMOD'SShips:':RSSn:
':side(2,3)\Territories:':Rt
529 END DEFine
```



```
Federation MODs
Land Based:38
SShips:2 :37
Territories:11
```

```
Rebel MODs
Land Based:50
SShips:1 :20
Territories:10
```

531 DEFine PROCEDURE MODInit

```
532 IF FMOD>RMOD:side(2,3)=20+FMOD-RMOD:side(1,3)=20
533 IF RMOD>FMOD:side(1,3)=20+RMOD-FMOD:side(2,3)=20
534 IF FMOD=RMOD:side(1,3)=20:side(2,3)=20
535 END DEFine
```



537 DEFine PROCEDURE TranPS

```
538 REMark Transfer from Planet to Star Ship
539 IF t=1 AND astro(n,1)>1 AND side(Gp1,1)<24
540   astro(n,1)=astro(n,1)-1:side(Gp1,1)=side(Gp1,1)+1
541 END IF
542 IF t=2 AND astro(n,2)>1 AND side(Gp1,2)<40
543   astro(n,2)=astro(n,2)-1:side(Gp1,2)=side(Gp1,2)+1
544 END IF
545 IF t=3 AND astro(n,3)>1 AND side(Gp1,3)<100
546   astro(n,3)=astro(n,3)-1:side(Gp1,3)=side(Gp1,3)+1
547 END IF
548 SSInfo
549 END DEFine
```



551 DEFine PROCEDURE TranSP

```
552 REMark Transfer from Star Ship to Planet
553 IF t=1 AND side(Gp1,1)>1 AND astro(n,1)<12
554   side(Gp1,1)=side(Gp1,1)-1:astro(n,1)=astro(n,1)+1
555 END IF
556 IF t=2 AND side(Gp1,2)>1 AND astro(n,2)<20
557   side(Gp1,2)=side(Gp1,2)-1:astro(n,2)=astro(n,2)+1
558 END IF
559 IF t=3 AND side(Gp1,3)>1 AND astro(n,3)<20
560   side(Gp1,3)=side(Gp1,3)-1:astro(n,3)=astro(n,3)+1
561 END IF
562 SSInfo
563 END DEFine
```



```

565 DEFine PROCedure Recruit
566 REMark Add Tec/Trade credits increase Population & MODs
567 astro(n,4)=astro(n,1)*.5+astro(n,2)*.2
568 astro(n,3)=INT(astro(n,4)/.5)
569 ch=12:AT#ch,5,0:PRINT#ch,' Recruiting':PAUSE 20
570 ch=12:AT#12,5,0:PRINT#12,' '
571 ch=11:AT#ch,5,0:PRINT#ch,' '
572 ch=16:CLS#ch:PRINT#ch,'To leave Orbit press':BLOCK#ch,16,4,130,4,4
573 MODInfo:PTInfo
574 END DEFine

```

Note: This is a once per visit request. Revisiting can gradually increase your Tec/Trade & MODs status and if misfortune reduces your capabilities to disastrous levels maybe your only way of getting back into the Game.

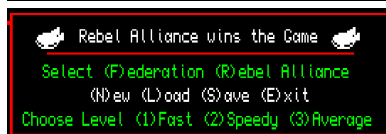
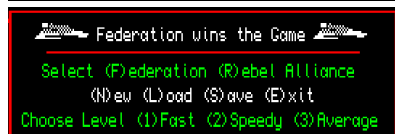
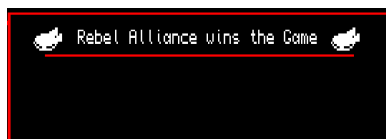
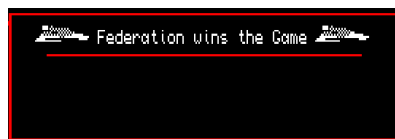
```

576 DEFine PROCedure Victory
577 CLS#16:PRINT#16,'Tec/Trade & Recruit MODs'
578 side(Gp1,1)=side(Gp1,1)+1+INT(astro(n,1)/4)
579 IF side(Gp1,1)>24:side(Gp1,1)=24
580 side(Gp1,2)=side(Gp1,2)+2+INT(astro(n,2)/5)
581 IF side(Gp1,2)>40:side(Gp1,2)=40
582 side(Gp1,3)=side(Gp1,3)+3+INT(astro(n,4)/2)
583 IF side(Gp1,3)>100:side(Gp1,3)=100
584 astro(n,5)=Gp1:s=0:Mes$=' ⬆️⬇️⬇️ Transfer ⇐⇒ '
585 MODInfo:SSInfo
586 END DEFine

```

Note: Taking a Star/Planet System - Victory is where the Tec/Trade credits & MODs rewards are worked out.

The Game Ends when one of the contenders takes all of their opponents territories. The Game returns to the Main Menu which this time in the top section identifies the winning side (Star Ships graphic). During the Game the Main Menu can be called up at or almost any time by pressing the(Esc). At that point if you wish to continue with your present Game your course of action would be to (S)ave it and then re-(L)oad. You can Save the Game at any time and battle on another day. It is possible this conflict will never end.



QBITS Galaxy AD2370 Computer Player

This follows the action of the Game namely selecting a Star System, if not listed as one of its own territories the order of play being to Attack. Once Victory has been secured or if the Star System is already owned then Transfer of Tec & Trade credits are carried out, an increase of Population generated and Recruiting of more MODs taken. If the Computer SShip fleet has less than 100 then these are uploaded for future Attacks or as defence MODs to Transfer to other territories.

This is the code I use for **Complay** the Computer Player.

```
588 DEFine PROCEDURE Complay
589 IF Gp2=1:side$='Federation':ELSE side$='Rebels'
590 IF side(Gp2,3)>=20:cn=RND(1 TO 50)      Note: Star System selected if MODs>=20.
591 IF side(Gp2,3)<20                        If not seek Transfers with own Territories
592   FOR i=1 TO 10
593     cn=RND(1 TO 50):IF astro(cn,3)>10 AND astro(cn,5)=Gp2:EXIT i
594   END FOR i
595 END IF
596 h2=astro(cn,6):v2=astro(cn,7):ws=1:StarLoc2      REMark StarLoc2 On
A 597 IF astro(cn,5)=Gp2
598   ch=16:INK#ch,2:PRINT#ch,' Transfers with ',astro$(cn):StarMove      TRANSFER
599   IF astro(cn,1)< 9:astro(cn,1)=astro(cn,1)+3
600   IF astro(cn,2)<15:astro(cn,2)=astro(cn,2)+5
601   astro(cn,4)=astro(cn,1)*.5+astro(cn,2)*.2
602   astro(cn,3)=INT(astro(cn,4)*2)
603   IF side(Gp2,3)<20 AND astro(cn,3)>3
604     side(Gp2,3)=side(Gp2,3)+astro(cn,3)-3:astro(cn,3)=3
605   END IF
606   IF side(Gp2,3)>50 AND astro(cn,3)<10
607     astro(cn,3)=astro(cn,3)+5:side(Gp2,3)=side(Gp2,3)+5
608   END IF
609 ELSE
610   ch=16:INK#ch,2:PRINT#ch,side$,' Attacks ',astro$(cn)
B 611 IF astro(cn,3)>side(Gp2,3)*3 OR side(Gp2,3)<10:GO TO 619 (C)      ATTACK
612   StarMove:Gp=Gp2:en=1:RollDice:PAUSE 10      SShip MOODs >*3
613   IF astro(cn,3)=0
614     ch=16:INK#ch,2:PRINT#ch,astro$(cn),' Falls'
615     IF side(Gp2,3)<30:side(Gp2,3)=side(Gp2,3)+10
616     IF side(Gp2,3)<60:side(Gp2,3)=side(Gp2,3)+5
617     astro(cn,5)=Gp2:GO TO 595 (A)
618   END IF
619   MODInfo:StarMove:GO TO 609 (B)
620 END IF
C 621 FOR i=1 TO 10:StarMove:PAUSE ps
622   ch=16:INK#ch,4:CLS#ch:MODInfo:StarLoc2:h2=-9:v2=-9      :REMark StarLoc2 Off
623   IF Ft=0 OR Rt=0:EndGame
624 END DEFine

626 DEFine PROCEDURE StarLoc2
627 INK#13,7:OVER#13,-1:CIRCLE#13,h2,v2,2.5:CIRCLE#13,h2,v2,6:OVER#13,0
628 END DEFine
```

QBITS Galaxy AD2370 Game Player2

I didn't take much to realise that **Complay** with a little modification could be used as a simulated human player to check various aspect of **Galaxy AD2370** code. I have left the code so **GPlayer2** & **Complay** can play against each other. To activate press (T)est in the Main Menu. A 60sec delay at start allows checks of PROCedures **Eorbit** and **Lorbit** (F1).

This is the slightly modified code that I used for testing and simulate a human Game Player.

```
630 DEFine PROCEDURE GPlayer2
631 IF Gp1=1:side$='Federation':ELSE side$='Rebels'
632 IF side(Gp1,3)>=20:n=RND(1 TO 50)
633 IF side(Gp1,3)<20
634   FOR i=1 TO 10
635     n=RND(1 TO 50):IF astro(n,3)>10 AND astro(n,5)=Gp1:EXIT i
636   END FOR i
637 END IF
638 h1=astro(n,6):v1=astro(n,7):ws=0:CLS#12:StarLoc1
639 IF astro(n,5)=Gp1
640   Mes$='%4℥Transfer ¼ ½'
641   ch=16:INK#ch,4:PRINT#ch,'Transfers with ',astro$(n)
642   IF astro(n,1)<9:astro(n,1)=astro(n,1)+3
643   IF astro(n,2)<15:astro(n,2)=astro(n,2)+5
644   astro(n,4)=astro(n,1)*.5+astro(n,2)*.2
645   astro(n,3)=INT(astro(n,4)*2)
646   IF side(Gp1,3)<20 AND astro(n,3)>3
647     side(Gp1,3)=side(Gp1,3)+astro(n,3)-3:astro(n,3)=3
648   END IF
649   IF side(Gp1,3)>50 AND astro(n,3)<10
650     astro(n,3)=astro(n,3)+5:side(Gp1,3)=side(Gp1,3)-5
651   END IF
652 ELSE
653   Mes$=' Attack'
654   ch=16:CLS#ch:INK#ch,4:PRINT#ch,side$,' Attacks ',astro$(n)
655 IF astro(n,3)>side(Gp1,3)*3 OR side(Gp1,3)<10:GO TO 665 (C)
656 StarMove:Gp1:en=3:RollDice:PAUSE 10
657 IF astro(n,3)=0
658   ch=16:INK#ch,2:PRINT#ch,astro$(n),' Falls'
659   IF side(Gp1,3)<30:side(Gp1,3)=side(Gp1,3)+10
660   IF side(Gp1,3)<60:side(Gp1,3)=side(Gp1,3)+5
661   astro(n,5)=Gp1:GO TO 639 (A)
662 END IF
663 SSInfo:GO TO 655 (B)
664 END IF
C 665 SSInfo:PAUSE 50:StarLoc1:h1=-9:v1=-9:
666 ws=1:CLS#11:CLS#12:CLS#15:CLS#16:INK#16,7
667 IF Ft=0 OR Rt=0:EndGame
668 END DEFine
```

:REMark StarLoc1 On

REMark StarLoc1 Off

Note: The GO TO command is a very alluring and powerful tool, yet used globally in a program it can easily lead to catastrophic effects. However, used as a simple loop and exit command between a few lines within a PROCEDURE it can be a very utilitarian.

