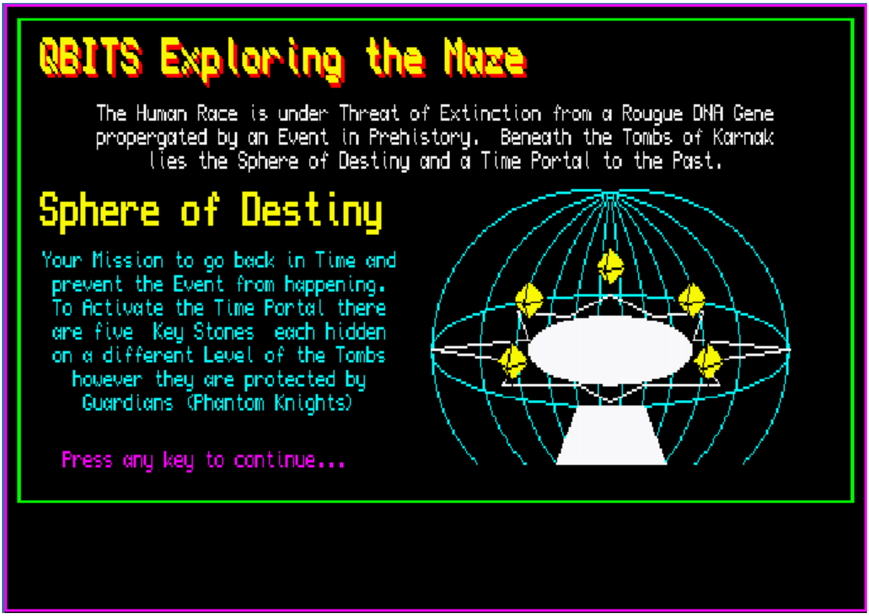
Sinclair QL Retro Computing



QBITS Exploring the Maze

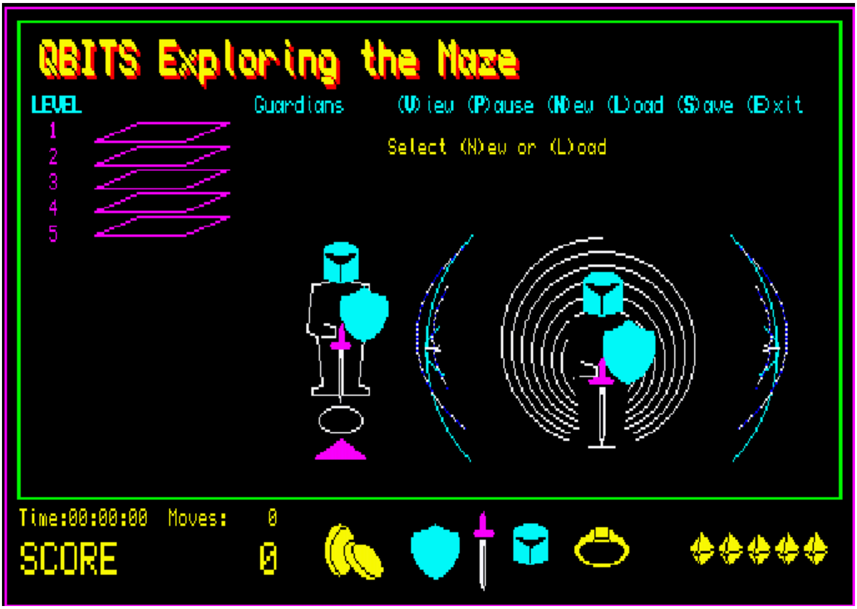
The Human Race is under Threat of Extinction from a Rougue DNA Gene propergated by an Event in Prehistory. Beneath the Tombs of Karnak lies the Sphere of Destiny and a Time Portal to the Past.

Sphere of Destiny

Your Mission to go back in Time and prevent the Event from happening. To Activate the Time Portal there are five Key Stones each hidden on a different Level of the Tombs however they are protected by Guardians (Phantom Knights)

Press any key to continue...

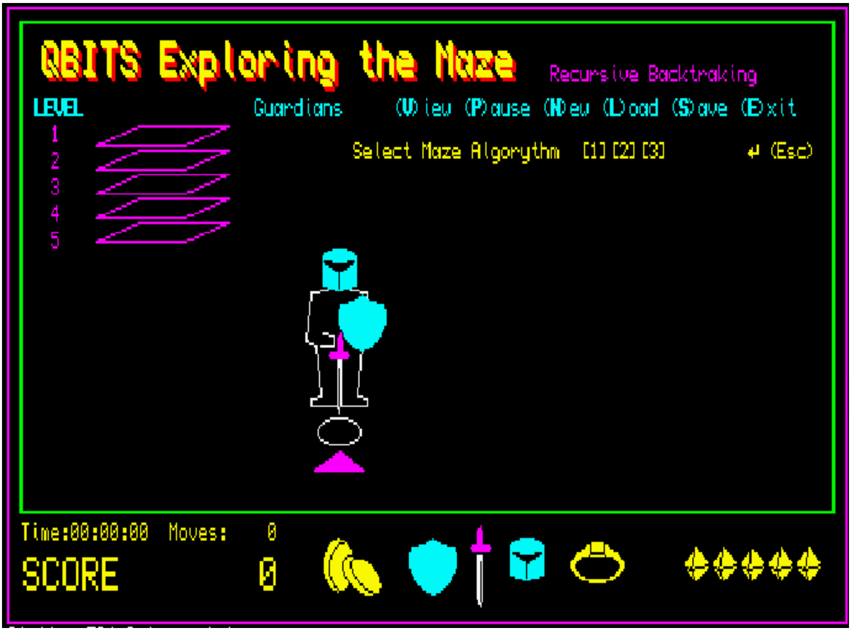
Sinclair QL Retro Computing



Select (N)ew or (L)oad

Time:00:00:00 Moves: 0

SCORE 0



Recursive Backtraking

Select Maze Algorythm [1] [2] [3] ↵ (Esc)

Time:00:00:00 Moves: 0

SCORE 0

QBITS Introduction

The microcomputers released to the home computer market in the mid nineteen eighties came with a ROM resident BASIC Interpreter, which the machine booted directly into. The original BASIC (Beginner' All-Purpose Symbolic Instruction Code) was patterned on FORTRAN with one-to-a-line statements. The syntax evolved among manufactures of home computers to meet the demands of the day. Small enough to fit within the memory constraints, but sophisticated enough be usable by those without much training. Computer magazines of the day published BASIC code lists for games and utilities. All adding to the popularity of BASIC for programming enthusiasts and for a time became the de facto for a standard programming language.

The drawback of those earlier times, computer platforms weren't fast enough to satisfy the growing demands running BASIC code through the built in interpreter. Compiling, writing your program in assembly or machine code gave a faster run time. Recently and by chance, part of a conversation relayed to me implied the possibility of BASIC making something of comeback. Holding that thought I considered the command list of the original SuperBASIC Interpreter launched with the QL machine. The advancements made with SMSQ and the extensions added down the years together with more advanced QL Computers. Not least a number of emulators, running on fast modern computer platforms. The QL2K and QPC2 I run on my desktop has the Super/SBASIC interpreter performance fast enough to negate compiling my programs.

QBITS The Challenge

So having resurrected some of my nineteen-eighties code, tinkered with aspects of the QL Graphics, and explored QL Sound, I now decided to challenge myself to write a Retro style QL Game. Then it was just a matter of coming up with something amazing in concept yet simple to execute, but complex enough to stretch the imagination?

A few years back I wrote a Sci-Fi Trilogy, a group of friend's discover a Rogue DNA Gene that was counting down humanity to extinction. The story evolves around the friends locating five special crystals, which became the Key Stones to unlocking a Time Portal. It is decided one should go back in time to stop the event in prehistory. The Time Portal in my fiction story was called the Sphere of Destiny, which lay beneath the ancient tombs of Karnak. So this was self imposed challenge, develop this theme into a Retro style computer game written in QL Super/SBASIC.

Read on….

QBITS Game Theme

Where to begin! What format could I perhaps base my game around? A dungeon and dragon style quest seemed a likely possibly… that sparked an a-maz-ing idea! Although on reflection many modern computer games are based around them. I'm talking about Mazes, although my earliest experience reminded me of the word obfuscation, the activity of obscuring peoples understanding so as to leave them baffled or bewildered.

The Maze algorithms in today's Gaming World are organized along various lines of Classification involving Dimensions, Topology, Tessellation, Routing Texture, Focus and might use any combination of these.

Two dimensional Mazes traversed by simple compass directions. Three dimensional Mazes with multiple levels, this may be via a stairway or bridge to overlap passageways connecting one area of a Maze to another. Fourth dimensional Mazes, using Portals to transport between past and future areas of the Maze. This all sounded just the ticket.

QBITS Maze Creating Algorithms

So Maze generation is normally defined as a predetermined arrangement of cells most commonly a rectangular grid, but other arrangements are possible. The purpose of the Maze generation algorithm is therefore to fulfil the challenge of finding a route between any two particular cells. Maze generations are often based on random spanning trees algorithms, which simply put means the tree with its branches forms the minimum number of undirected links between all cells.

The **Recursive Backtracking** algorithm is a simple pathway created by randomly selecting an unvisited cell adjacent to one of the sides of the present one. The path is then made by knocking down the walls between. Moving forward into the new cell the random selection continues until there are no surrounding unvisited cells to select. The visited cells of the pathway are then backtracked until one is found to still have an unvisited adjacent cell or cells. The passage way is carved forward again until there are no unvisited cells. Recursive Backtracking continues until all the cells have been visited.

There are a solid handful of algorithms to the above, but for QBITS exploration a version of **Prim's** which instead of recursive backtracking the code, randomly selects a visited cell from the generated list, then continues forward carving a pathway through unvisited cells. A third method sometimes referred to as **Hunt and Kill** purposely moves through the grid selecting a cell and if a randomly chosen side has an unvisited cell will carve a path between the two.

QBITS 2D Maze

So using QL Super/SBASIC my first task was in creating an array of columns and rows each of which identified a walled cell position. Then carve a random pathway between cells with the ability to find a route between any one cell to any other. The displayed graphic would be a simple two dimensional perfect Maze with long and short passageways and some dead ends. By adding further interconnections I labelled halls it then opened up the possibility of multiple routes.

QBITS 3D Maze

By taking the current 2D cell position and by selecting a forward facing direction it is possible to construct a 3D image of the Maze passageway. Using QL Super/SBASIC code to create the roof, floor and side walls, with side gaps for joining passageways. Other possibilities could now be entertained; dead end passageways could be utilised for secret doorways to hidden treasures. The 3D forward view gave the opportunity to place objects blocking further forward movement down the passageway.

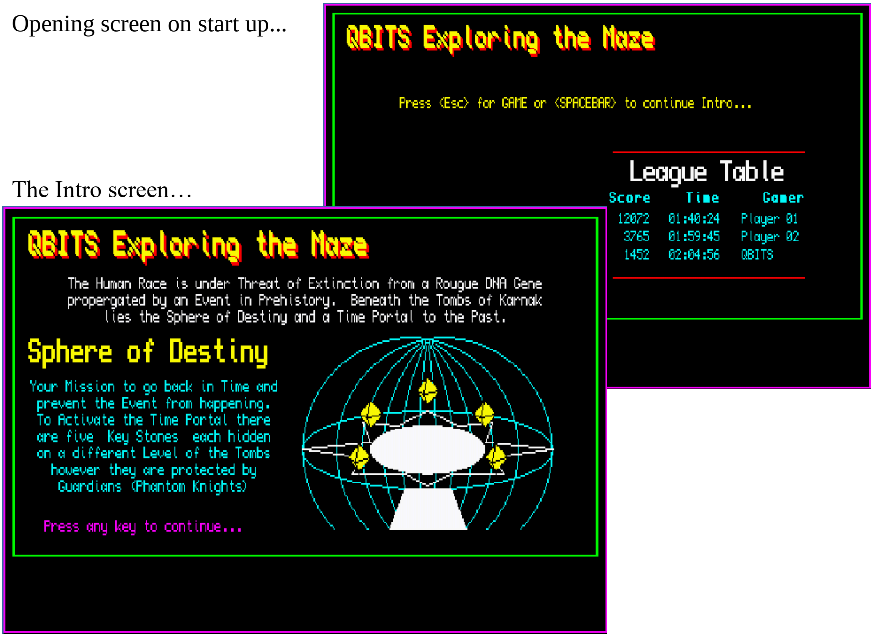
QBITS Maze Design

Already my thoughts were being galvanised into how I would design the screen layout, starting with the Title and Main Menu requirements New, Load, Save, Exit etc.. Then a Score display which included a Game Time clock and a Moves counter as well as a Points counter, and a League Table of best results. I decided on five levels to the Maze before accessing the Sphere of Destiny. An area to display images of the treasures collected and lost. For obstacles to progress I decided on Guardians of the tombs and in the form of Phantom Knights. This all played into multiple uses of the Vector Graphic images I had been developing for the Game. As the Game components evolved I added compass directions North, South, East & West and a forward facing indicator, a numeric display for a Dice throw. Then a tally of defending Guardians deployed on each level and those left for the final confrontation before entering the Sphere of Destiny.

QBITS Maze Screen Layouts

Opening screen on start up...

The Intro screen…



QBITS Exploring the Maze

Press <Esc> for GAME or <SPACEBAR> to continue Intro...

League Table

Score	Time	Gamer
12072	01:40:24	Player 01
3765	01:59:45	Player 02
1452	02:04:56	QBITS

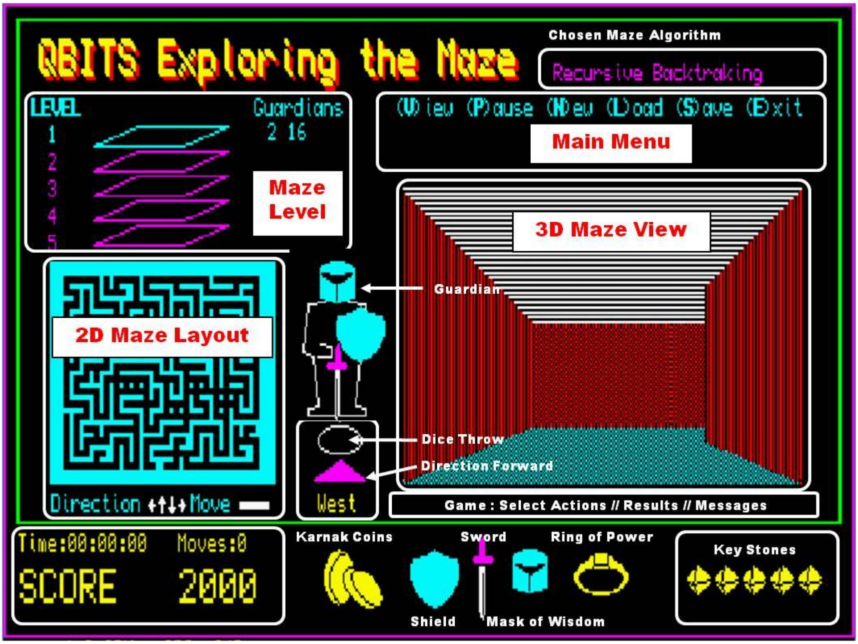
QBITS Exploring the Maze

The Human Race is under Threat of Extinction from a Rougue DNA Gene propergated by an Event in Prehistory. Beneath the Tombs of Karnak lies the Sphere of Destiny and a Time Portal to the Past.

Sphere of Destiny

Your Mission to go back in Time and prevent the Event from happening. To Activate the Time Portal there are five Key Stones each hidden on a different Level of the Tombs however they are protected by Guardians (Phantom Knights)

Press any key to continue...



Game screen…

QBITS Maze - a Walk Through

The opening screen displays the Gamers **League Table** and offers [**Esc**]ape to move directly to the Game or [**SPACEBAR**] to an Introduction Screen in the form of a mission statement. This Intro also displays Graphics of the **Sphere of Destiny**, where the final part of the mission is played out.

Top Left of all the screens is the Title '**QBITS Exploring the Maze**'. Below this on the **Game Screen** are the **Maze Levels** which will show the active level in a different colour and the number of **Defending Guardians**. To the right of these the **Main Menu** with two additions (**V**)iew (**P**)ause then the usual (**N**)ew (**L**)oad (**S**)ave (**E**)xit. At this point the game presents the option to begin a **New** Game or **Load** a previously saved one.

Centre of the screen is a **Guardian** Knight standing over an **Ellipse** within which a random number is displayed when a Player uses the **Sword** option for attack. Below this a **Triangle** with the **Compass Direction** printed below indicating the way forward.

Bottom left shows the **Score Board**, **Game Time**, number of **Moves** and total **Points** collected. For the rest of the lower screen area Game icons are displayed. On the far right the five **Key Stones**, one to be collected from each **level** of the **Maze**. In the middle; **Coin**s of Karnak, **Shield**, **Sword**, **Mask** of Wisdom and **Ring** of Power.

QBITS Maze - (N)ew Game

Here you are first asked to select from one of three slightly differing Maze algorithms. Top right displays the currently chosen algorithm, <**Enter**> sets things in motion. The Maze Generation chosen creates a 2D diagram in the lower left hand part of the screen (Window#3). The first **2D Maze** Level of a **New Game** is drawn with a slight **Pause delay** to show its construction. For other levels and if [**L**]oad is used, depending on the speed of your QL environment there is no Pause delay and the Maze may appear almost instantaneously.

On completion of the **2D Maze** and to the right of the **Guardian** Knight there is now displayed a **3D** view of the present location within in the **Maze** (Window#1). The Maze Level is highlighted upper left and the number of Guardians deployed out of a total of sixteen. Above the 3D screen you are asked to 'Press any key to continue…', doing so will start the **Timer** and the **Game**.

QBITS Maze - Moves

Below the **2D Maze** as shown **Direction** is changed by use of the **CURSOR keys**, this will be shown by a change in the **3D** view and the forward facing direction of **North**, **East**, **West** or **South** printed below the triangle lying between **2D** and **3D** displays. To **Move** forward press the **Spacebar**.

QBITS Maze - Treasures

Moving around the Maze on each Level will uncover a number of Treasures; **Coins** of Karnak, **Mask** of Wisdom, **Ring** of Power and one of the **Key Stones**, collection of which adds Points to the Score. The **Mask** and **Ring** are assets that also aid in defending against or defeating the **Guardians**. The **Key Stone** is required to activate the **Portal** between **Levels** and finally to access the **Sphere of Destiny**.

QBITS Maze - Guardians

Encountering a **Guardian** a Player has four options, use **[1]Shield** which avoids the confrontation by Teleporting to another part of the current Maze level. Use **[2]Sword** to attack, but a six must be thrown to defeat the **Guardian**. If acquired use **[3]Mask** to banish all **Level Guardians** for 120 moves. If acquired use **[4]Ring** to delete all current **Level Guardians**. Each of these choices incurs a loss of Points.

QBITS Maze - Levels

Accepting the **Key Stone** activates the **Portal** and makes the jump to the **next Maze Level**. A jump to the **Sphere of Destiny** can only be made if all remaining **Guardians** have been defeated.

QBITS Maze - Sphere of Destiny

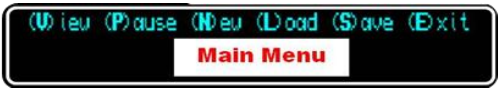
Upon reaching the **Sphere of Destiny** the five acquired **Key Stones** from the Maze Levels have to be aligned to their correct position with those within the Sphere. The prima Key Stone, number Five is given, the other four must be **Matched**. Twenty four different combinations are possible. If successful the **Time Porta**l is open and humanity saved from extinction. If not then hard luck, try another game.

QBITS Maze - Strategy

The aim of the Game is to seek out the **Treasures** of the **Maze** while defending against and defeating the **Guardians** encountered along the way. Apart from the **Key Stone** all Treasures are acquired upon entering the Grid cell containing them; **Coins** of Karnak, **Mask** of Wisdom or **Ring** of Power each adding to the Points scored.

The **Key Stone** location offers a choice of **Y/N** between taking the **Key Stone** or leaving it for a later pickup. Accepting the **Key Stone** will immediately activate the **Portal** to other Levels, apart from when leaving Level five. To reach the **Sphere of Destiny** all remaining **Guardians** have to be defeated first.

Entering the **Sphere of Destiny** and **Match**ing the Key Stones to activate the **Time Portal** can deplete your **Score** dramatically even leading to a failed attempt. The strategy is therefore mostly a balance between gaining as higher **Point count** with minimum **Moves** as possible, while managing to defeat all **Guardians**. So preparing for a heavy sacrifice of Points to activate the **Time Portal** and save **Humanity**.



QBITS Maze - (V)iew

This is an **ON/OFF** toggle switch that displays the location of **Maze Treasures** and the **Current Position** of the player within the 2D Maze. The number of **Points** taken on **each move** depends on the **Level** and if **View** is switched **ON** or **OFF.**

QBITS Maze - (P)ause

The **Game Timer** is halted and time stored (**GTS**). A message is displayed '**Press any key to continue…**', pressing of which will restart the Game Timer and allow continuation of the game.

QBITS Maze - (L)oad

This presents you with a selection of **Device** and Game **Data Filenames**. At this point you can abort Menu choice with <**Esc**>, or continue with <**Enter**> which will make a search, returning '**File NOT found**' or continue with '**Loading…**'. If device is unattached the program will halt with a **QDOS error** message in WINDOW#0.

QBITS Maze - (S)ave

This presents you with a selection of **Device** and Game **Data Filenames**. At this point you can abort Menu choice with <**Esc**>, or continue with <**Enter**> which will **Save** the Game Data to the selected Device/Filename. If device unattached the program will halt with a **QDOS error** message in WINDOW#0.

QBITS Maze - (E)xit

This presents you with '**Exit Game** (**Y/N**)' any key other than 'Y' or 'y' will return to the Game. On exit Windows/channels #3 to #5 will be closed, with windows #0,#1,#2 cleared and restored to default sizes.

QBITS Maze - SCORE

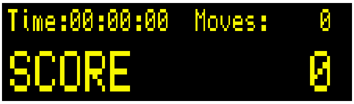
The **SCORE** shows the **Game Duration** in hours, minutes and seconds a count of the **Moves** taken and a **Points** counter. The Timer uses QL Super/SBASIC Commands DATE to set the **Game Clock** (**Gclk**) at start and DATE$ to create an hh:mm:ss display.

ie. **clk$**=DATE$(DATE-**Gclk**+**GTS**) : PRINT **clk$**(13 to 20)

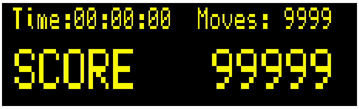
GTS holds the current **Game Time Seconds** for a Game /Pause/Save/Load.

The **Moves** and **Points** are printed using the FILL$ command and with spaces so the counters grow right to left as the number increases.

ie. PRINT ‘SCORE ‘,FILL$(‘ ‘,6-LEN(**snum**))&**snum**



```
Time:00:00:00  Moves:    0
SCORE                    0
```



```
Time:00:00:00  Moves: 9999
SCORE           99999
```

QBITS Maze – Points Table

Points are **Gained** or **Deleted**, these are added or deducted from the variable **snum**.

Maze Moves Calculator **View OFF** **View ON**

On each move **S**core Points are **L**ost: **sl**=**lev** (ie.1 to 5) or **sl**=**lev***5 (ie, 5 to 25)

Maze Treasure Calculator

As you move around the Maze, check the dead end passageways as they may contain a hidden doorway to **Treasure**. Apart from adding valuable **Points** to the Score they may be helpful when dealing with the **Guardians**.

Coins	- Random Selections (100 to 300)	**snum**=**snum**+50+50*RND(2 to 6)
Mask	- Increases with level (100 to 300)	**snum**=**snum**+50+50***lev**
Ring	- Increases with level (1000 to 3000)	**snum**=**snum**+500+500***lev**
KeyStone	- In acquiring Activates the Portal	**snum**=**snum**+2000

Maze Guardian Encounters

Confronted with a **Maze Guardian** you have between two to four choices the **Shield**, **Sword** and if acquired the **Mask** and/or **Ring.** The first two are given at the beginning of the Game the latter two have to be found and acquired on each of the Levels.

[1]Shield	- Portal Jump	**snum**=**snum**-50
[2]Sword	- Each dice thrown if not a 6	**snum**=**snum**-50
	(if dice throw is a 6 Delete Guardian)	**gmax**=**gmax**-1:**glev**=**glev**-1
[3]Mask	- Banish Guardian for 120 moves	**snum**-**snum**-50-50***lev**
[4]Ring	- In Deleting Level Guardians	**gmax**=**gmax**-**glev**:**glev**=0 **snum**=**snum**-500-500***lev**

Maze Sphere of Destiny

For each failed try to Match the KeyStones **snum**=**snum**-500

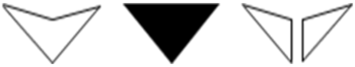
QBITS Maze - Vector Graphics

As computer games developed from the early nineteen-eights it was the graphical displays that most impressed and intrigued, releasing in some cases a rewarding talent of expression. Bitmap images have their place, but with vector graphics you can do so much more. Vectors graphics use lines straight or curved drawn between coordinated points, this makes them easily scalable.

When drawing vector graphics with QL Super/SBASIC you are using the Graphics coordinate system as opposed to Pixel coordinates. This has a couple of idiosyncrasies; the simple one is related to CURSOR coordinates. When attached to a Graphic drawing four coordinates are used, the first pair interpreted as Graphical with the second as a Pixel offset in relation to the first.

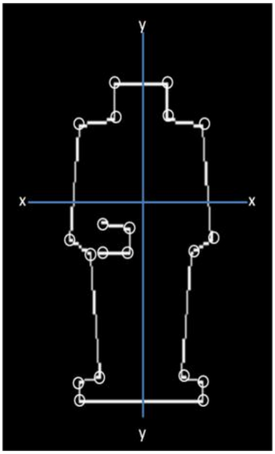
```
CURSOR gx, gy, px, py
```

The second relates to a drawn object as shown below and the use of FILL. When an object is filled with a solid colour QL Super/SBASIC FILLs between min and max line coordinates so it looks a little different to what you might expect. To overcome this two object are required not one. The result can be seen in the Helmets Visor shown here.



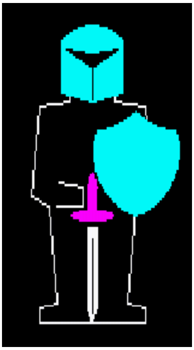
QBITS Maze - Guardian

Vector Graphics is a bit like joining the dots. After setting the scale and location of the x, y zero coordinates; you need to work out the offsets to each position that describes the object. To create the Maze Guardian, I based this on an image of a Knight taken from an old church brass rubbing. The body with the head masked by a helmet, holding a Shield over the left arm and the Sword held with the blade tip down at the feet.



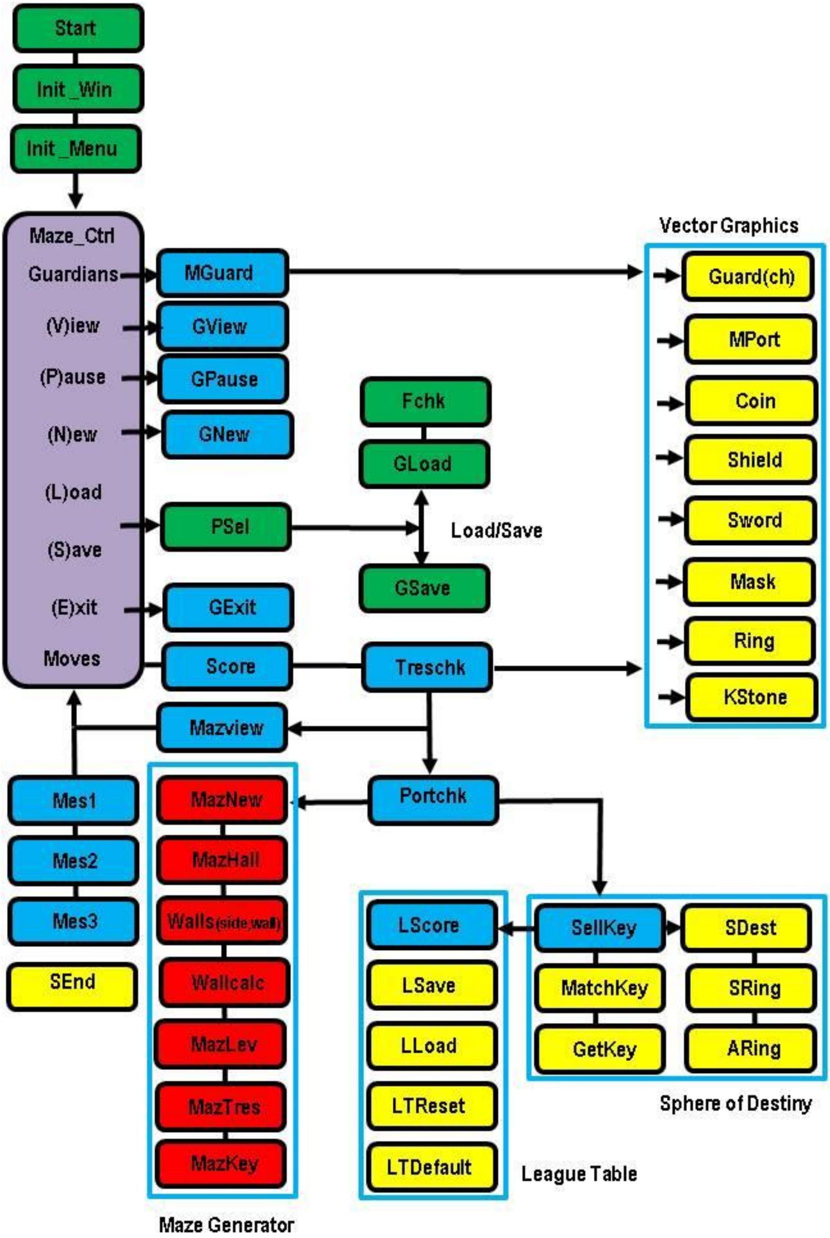
For the Maze Guardian this required first drawing the outline of a body image with the sword arm.

Images for Mask, Shield and Sword are then combined with the Guardian body outline to create the Maze Phantom Knight.

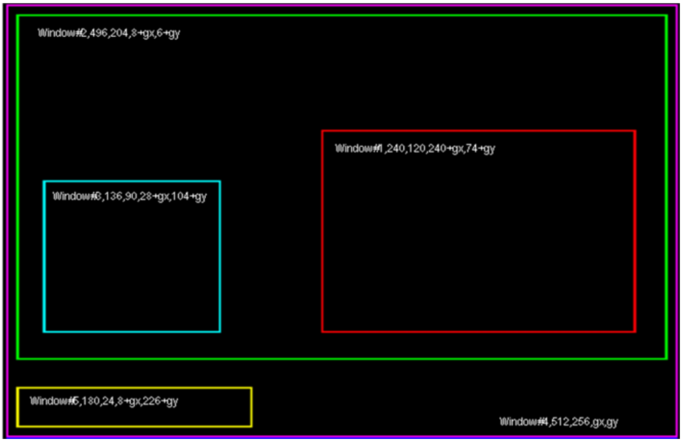


The images of Guardian knight, Shield, Sword and Mask are re-scaled and used in other areas of the screen, the Knight as tomb defender with the others as Maze Treasures.

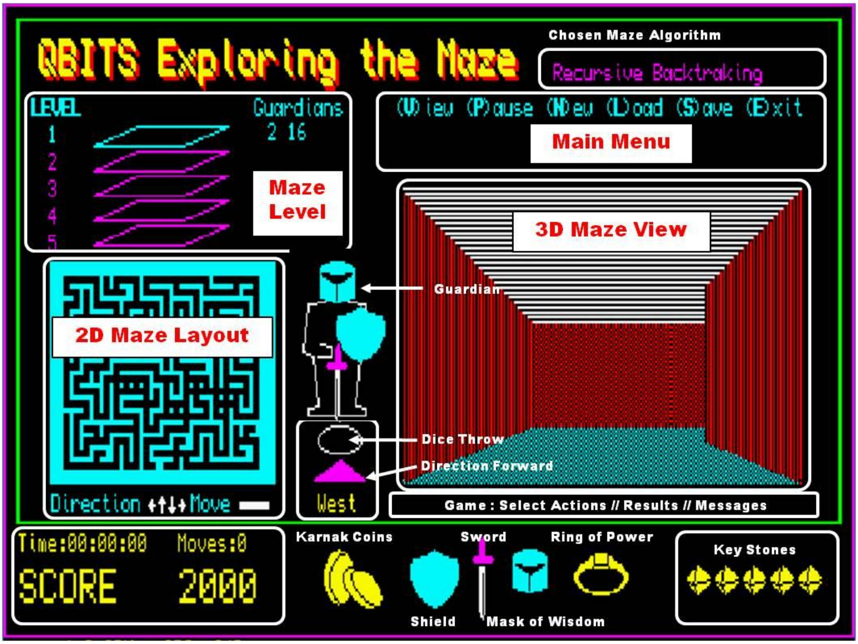
QBITS Maze PROCedures/FuNctios



```
100 REMark QBMazeQPC_v04 (QBITS Maze Game 2019) code review v04Dec

102 REMark arrays
103 DIM drv$(8,5):RESTORE 103:FOR d=1 TO 8:READ drv$(dv)          QL Storage Device
104 DATA 'flp1_','flp2_','win1_','win2_','dos1_','dos2_','nfa1_','nfa2_'
105 DIM GDat$(9,10):FOR f=0 TO 9:GDat$(f)='MazeData_'&f          Data filenames for saving Game
106 DIM dir$(4,5):RESTORE 106:FOR c=1 TO 4:READ dir$(c)          Maze Forward Direction
107 DATA 'West ','East ','North', 'South'
108 DIM Skey(5,3):RESTORE 108:FOR i=1 TO 5:READ Skey(i,1):READ Skey(i,2)
109 DATA -30,28, -36,6, 36,6, 30,28, 0,40
110 DIM grid(21,17), cell(20*16):w=20:h=16                      Main 2D Maze Grid
111 DIM Mkey(5),Tres(12,3),name$(3,10),Grad(3,2):LTDefault

113 REMark variables
114 dv=4:f=0:m=1                  :REMark default drive win2_ MazeDat_(file) Maze Algorithm
115 w=20:h=16:x=0:y=0:cx=0:cy=0:px=0:py=0          :REMark Various Coordinates
116 lev=1:glev=2:gmax=16:gdel=120:sl=lev            :REMark Maze Level Settings
117 Gclk=DATE:GTS=0:sm=0:snum=0           :         :REMark Score Time Moves Points
118 bc=0:sc=0:tc=0:scol=0                           :REMark Various Colours
119 gst=0:gck=0 :fd=3                               :REMark Various Checks/Settings

121 Init_Win:Init_Menu:Maze_Ctrl

123 DEFine PROCedure Init_Win
124 MODE 4:gx=20:gy=30:my=286             :REMark QPC Mode for QL gx=0 gy=0 my=220
125 OPEN#5,scr_10x10a10x10:WINDOW#5,180,24,8+gx,226+gy   :PAPER#5,0
126 OPEN#4,scr_10x10a10x10:WINDOW#4,512,256,gx,gy         :PAPER#4,0
127 OPEN#3,scr_10x10a10x10:WINDOW#3,136,90,28+gx,106+gy :PAPER#3,5
128 CSIZE#5,2,1:INK#5,6:BORDER#4,1,3  :SCALE#4,240,0,0  :CLS#4
129 WINDOW#2,496,204,8+gx,6+gy        :PAPER#2,0 :BORDER#2,1,4
130 WINDOW#1,240,120,240+gx,74+gy     :PAPER#1,0 :SCALE#1,100,-74,-30
131 WINDOW#0,512,30,gx,my :PAPER#0,0:CSIZE#0,0,0:INK#0,7
132 END DEFine
```

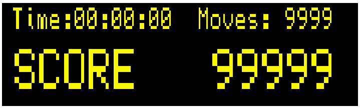

Window#2,496,204,8+gx,6+gy
Window#1,240,120,240+gx,74+gy
Window#3,136,90,28+gx,104+gy
Window#5,180,24,8+gx,226+gy
Window#4,512,256,gx,gy

```
134 DEFine PROCedure Init_Menu
135 CLS#2:CSIZE#2,2,1:OVER#2,1
136 INK#2,2:FOR i=1 TO 4:CURSOR#2,8+i,4+i:PRINT#2,'QBITS Exploring the Maze'
137 INK#2,6:FOR i=1 TO 2:CURSOR#2,8+i,4+i:PRINT#2,'QBITS Exploring the Maze'
138 CSIZE#2,0,0:OVER#2,0:INK#2,6:CURSOR#2,60,54:ch=1:LScore
139 PRINT#2,'Press <Esc> for GAME or <SPACEBAR> to continue Intro...'
140 REPeat Sel_lp
141  k=CODE(INKEY$(-1))
142  SELect ON k
143    =27:CLS:EXIT Sel_lp
144    =32:CLS:Game_Intro:EXIT Sel_lp
145  END SELect
146 END REPeat Sel_lp
147 BLOCK#2,490,40,2,28,0:BLOCK#2,220,136,2,60,0:col=0:lev=6:MazLev
148 OVER#2,1:INK#2,5:FOR i=0 TO 1:CURSOR#2,6+i,30:PRINT#2,'LEVEL'
149 CURSOR#2,224,30 :PRINT#2,'(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit'
150 CURSOR#2,225,30 :PRINT#2,' V     P      N    L      S      E '
151 OVER#2,0:CURSOR#2,140,30:PRINT#2,'Guardians':gst=0 :gck=0:gdel=120
152 ch=4:x=140:y=22:Coin:x=180:y=32:Shield:x=200:y=6:Sword:x=220:y=20:Mask
153 x=250:y=22:Ring:col=7:fil=0:Guard(4):INK#4,7:CIRCLE#4,140,74,9,.6,PI/2
154 INK#2,3:FILL#2,1:LINE#2,65,8 TO 71,12 TO 76,8 TO 65,8:FILL#2,0:Mes1
155 END DEFine
```



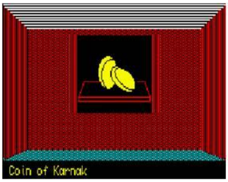
```
157 DEFine PROCedure Maze_Ctrl
158 REPeat Maze_lp
159  IF gst=1:Score:ELSE Gclk=DATE:Score
160  IF gdel=0:MGuard
161  k=CODE(INKEY$(20))
162   SELect ON k
163    =232:km=1:kr=1:MGuard                 :REMark [F1] Activate Guardians
164    =236:PortChk                          :REMark [F2] Activate Portal acts
165    =240: snum=snum+50:Score              :REMark [F3] Plus Points
166    =244:IF snum>50:snum=snum-50:Score    :REMark [F5] Minus Points
167    =248: sc=241:tc=0:GView:tc=3:GView    :REMark [F5] Show Key Stone
168    = 86,118:GView                        :REMark [V]iew ON/OFF
169    = 80,112:GPause:Gclk=DATE             :REMark [P]ause
170    = 78,110:GNew :GPause                 :REMark [N]ew
171    = 76,108:PSel:GLoad:GPause            :REMark [L]oad
172    = 83,115:PSel:GSave                   :REMark [S]ave
173    = 69,101:GExit                        :REMark [E]xit
174    =192:  IF gst=1:fd=1:MazView          :REMark Left   West
175    =200 : IF gst=1:fd=2:MazView          :REMark Right East
176    =208 : IF gst=1:fd=3:MazView          :REMark Up     North
177    =216 : IF gst=1:fd=4:MazView          :REMark Down South
178   ON k=32                                :REMark SpaceBar Forward
179     IF snum<5:Mes2:GO TO 195
180     IF snum<  50+  50*lev:BLOCK#4,30,30,300,216,0:km=0
181     IF snum<500+500*lev:BLOCK#4,50,30,332,216,0:kr=0
182     IF fvn=1
183       INK#2,5:CURSOR#2,236,190:PRINT#2,'Solid Wall!':CLS#2,4
184       BEEP 1000,1,140,190,0,0,0:PAUSE 20
185     ELSE
186       IF fd=1 : px=px-1                            :REMark One cell West
187       IF fd=2 : px=px+1                            :REMark One cell East
188       IF fd=3 : py=py-1                            :REMark One cell North
189       IF fd=4 : py=py+1                            :REMark One cell South
190       BLOCK#3,4,3,2+cx*6,1+cy*5,0   :cx=px:cy=py
191       BLOCK#3,4,3,2+cx*6,1+cy*5,bc           :REMark 2D Maze cell position
192       BEEP 2000,20,40,190,0,0,0:gst=1
193       ofd=fd:gdel=gdel-1:snum=snum-sl:sm=sm+1:MazView
194       Loot=grid(px,py):SELect ON Loot=1,2,4,8:TresChk
195     END IF
196   END SELect
197  END REPeat Maze_lp
198 END DEFine
```

Note: Showing Max possible Moves & Score Points

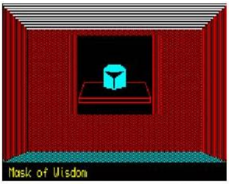


```
Time:00:00:00  Moves: 9999
SCORE    99999
```

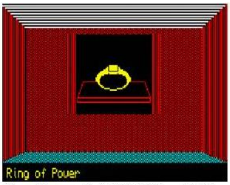
```
200 DEFine PROCedure Score
212 INK#4,6:clk$=DATE$(DATE-Gclk+GTS):CURSOR#4,6,212
202 PRINT#4,'Time:';clk$(13 TO 20);'   Moves:FILL$(4-LEN(sm))&sm
203 PRINT#5,'SCORE ';FILL$(' ',6-LEN(snum))&snum
204 END DEFine
```



Coins of Karnak Gain 50+50xRND(2 to 6) Points



Mask of Wisdom - Gain 50+50xLevel Points

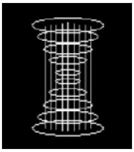


Ring of Power – Gain 500+500xLevel Points

```
206 DEFine PROCedure TresChk
207 FOR i=1 TO 12
208   IF Tres(i,1)=px AND Tres(i,2)=py
209     tn=Tres(i,3):IF tn=0:EXIT i
210     BLOCK 100,60,70,24,0,2,2:FOR j=1 TO 8:BLOCK j*10,60,120-j*5,24,0:PAUSE 5
211     ch=1:INK 2:x=0:y=20:INK#2,6:CURSOR#2,236,190
212     LINE x-24,y-10 TO x-20,y TO x+20,y TO x+24,y-10 TO x-24,y-10
213     LINE x-24,y-10 TO x-24,y-12 TO x+24,y-12 TO x+24,y-10
214     IF Tres(i,3)>1 AND Tres(i,3)<7
215         x=-4:y=26:Coin:PRINT#2,'Coin of Karnak':snum=snum+50*tn
216     END IF
217     IF tn=7
228         ch=4:x=220:y=20:Mask ch=1:x=0:y=20:Mask:km=1
219         PRINT#2,'Mask of Wisdom':snum=snum+50+50*lev
220     END IF
221     IF tn=8
222         ch=4:x=250:y=22:Ring:ch=1:x=0:y=22:Ring  :kr=1
223         PRINT#2,'Ring of Power ':snum=snum+500+500*lev
224      END IF
225     IF tn=9 :ch=1:x=0:y=20:KStone: PortChk:EXIT i
226     Tres(i,3)=0
227  END IF
228 END FOR i
239 END DEFine
```

Note: These **PROC**edure’s check the current grid cell for Actions to be taken.

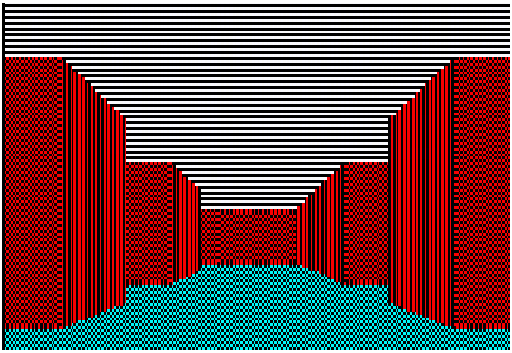
```
231 DEFine PROCedure PortChk
232 CURSOR#2,236,190:PRINT#2,'KeyStone - Portal to Next Level (Y/N)'
233 IF INKEY$(#2,-1)=='Y'
234    IF lev=5
235      IF gmax>0:CURSOR#2,236,190:PRINT#2,’Defeat All Guardians!’:CLS#2,4:RETurn
236      CURSOR#2,236,190:CLS#2,4:KeyStone
237    ELSE
238     lev=lev+1:glev=lev+1:IF glev>gmax OR lev=5::glev=gmax
239      CLS:MPort:PAUSE 20:col=5:MazLev:MazNew:MazHall:MazTres:MazView
240      snum=snum+2000:Score: sl=lev:tc=0:bc=0:
241      px=RND(2 TO 19):py=RND(2 TO 15): km=0:kr=0:gdel=120/RND(2 to 4)
242    END IF
243 ELSE
244    tc=1:GView
245 END IF
246 END DEFine
```



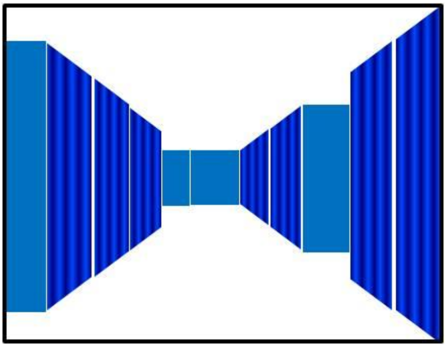
Note: Calculates and Displays the 3D view of Passageways.

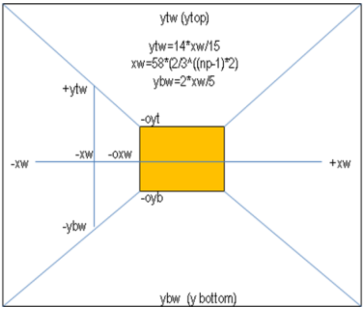
```
248 DEFine PROCedure MazView
249 fvn=0:INK#2,6:CURSOR#2,178,190:PRINT#2,dir$(fd)    :REMark fd forward direction
250 FOR n=1 TO 5
251   fv(n)=0                                            :REMark fv forward view
252   IF fd=1
253     cw=grid(px-n+1,py):IF Walls(8,cw)=1:fv(n)=1
254     IF Walls(4,cw)=1:fv(n)=fv(n)+2                   :REMark cw cell walls
255     IF Walls(1,cw)=0:fvn=n:EXIT n                    :REMark fvn forward view num cells
256   END IF
257   IF fd=2
258     cw=grid(px+n-1,py):IF Walls(8,cw)=1:fv(n)=2
259     IF Walls(4,cw)=1:fv(n)=fv(n)+1
260     IF Walls(2,cw)=0:fvn=n:EXIT n
261   END IF
262   IF fd=3
263     cw=grid(px,py-n+1):IF Walls(1,cw)=1:fv(n)=1
264     IF Walls(2,cw)=1:fv(n)=fv(n)+2
265     IF Walls(4,cw)=0:fvn=n:EXIT n
266   END IF
267   IF fd=4
268     cw=grid(px,py+n-1):IF Walls(1,cw)=1:fv(n)=2
269     IF Walls(2,cw)=1:fv(n)=fv(n)+1
270     IF Walls(8,cw)=0:fvn=n:EXIT n
271   END IF
272 END FOR n
273 vn=fvn:IF fvn=0 :fvn=6:vn=5                          :REMark fvn vn forward view num cells
274 xw=58*(2/3)^((vn-1)*2):ytw=14*xw/15:ybw=-2*xw/5
275 BLOCK 240,82,0,0,7,0,1:BLOCK 240,38,0,82,0,5,3       :REMark Roof & Floor
276 IF fvn=6
277    INK 0,2,1
278    FILL 1:LINE -xw,ytw TO xw,ytw TO xw,ybw TO -xw,ybw TO -xw,ytw:FILL 0
279    GO TO 286
280 END IF
281 INK 0,2,3
282 FILL 1:LINE -xw,ytw TO xw,ytw TO xw,ybw TO -xw,ybw TO -xw,ytw:FILL 0
```

Continues on next page…



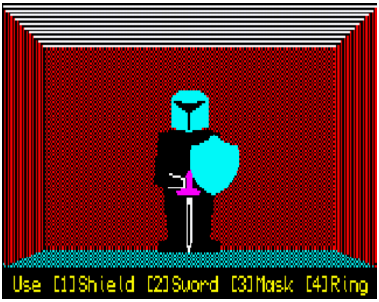
```
284 REPeat sidewalls
285  Wallcalc:cdv=fv(vn):INK 0,2,3
286  IF cdv=2 OR cdv=3
287     FILL 1:LINE oxw,oyt TO oxw,oyb TO xw,oyb TO xw,oyt TO oxw,oyt:FILL 0
288  END IF
289  IF cdv=1 OR cdv=3
290    FILL 1:LINE -oxw,oyt TO -oxw,oyb TO -xw,oyb TO -xw,oyt TO -oxw,oyt:FILL 0
291  END IF
292  INK 0,2,2
293  IF cdv=0 OR cdv=1
294     FILL 1:LINE oxw,oyt TO oxw,oyb TO xw,ybw TO xw,ytw TO oxw,oyt:FILL 0
295  END IF
296  IF cdv=0 OR cdv=2
297     FILL 1:LINE -oxw,oyt TO -oxw,oyb TO -xw,ybw TO -xw,ytw TO -oxw,oyt:FILL 0
298   END IF
299   Wallcalc
300   FILL 1:LINE oxw,oyt TO oxw,oyb TO xw,ybw TO xw,ytw TO oxw,oyt:FILL 0
301   FILL 1:LINE -oxw,oyt TO -oxw,oyb TO -xw,ybw TO -xw,ytw TO -oxw,oyt:FILL 0
302   vn=vn-1:IF vn=0 : EXIT sidewalls
303 END REPeat sidewalls
304 END DEFine
```





```
306 DEFine FuNction Walls(side,wall)
307 ans=0
308 IF side=1:SELect ON wall=1,3,5,7,9,11,13,15   :ans=1
309 IF side=2:SELect ON wall=2,3,6,7,10,11,14,15 :ans=1
310 IF side=4:SELect ON wall=4,5,6,7,12,13,14,15 :ans=1
311 IF side=8 AND wall>7     :ans=1                          :REMark fd forward direction
312 RETurn ans
313 END DEFine

315 DEFine PROCedure Wallcalc
316 oxw=xw:xw=xw*1.5:oyt=ytw:oyb=ybw:ytw=14*xw/15:ybw=-2*xw/5
317 END DEFine
```

```
319 DEFine PROCedure MGuard
320 IF glev=0:RETurn :ELSE ch=1:col=0:fil=1:Guard(1):INK 7:MPort
321 PAUSE 20:MazView:col=0:fil=1:Guard(1):INK#2,6:gdel=120/RND(2 TO 4)
322 REPeat G_lp
323   IF snum<  50:EXIT G_lp
324   CURSOR#2,236,190:PRINT#2,'Use [1]Shield [2]Sword':CLS#2,4
325   IF km=1 AND snum>50+50*lev:CURSOR#2,374,190:PRINT#2,'[3]Mask'
326   IF kr=1  AND snum>500+500*lev:CURSOR#2,422,190:PRINT#2,'[4]Ring'
327    k=CODE(INKEY$(-1))
328   IF k=49:snum=snum-50:px=RND(3 TO 17):py=RND(3 TO 14):CLS:EXIT G_lp
329   IF k=50
330     INK#4,5:FOR i=1 TO 6 :CURSOR#4,197,170:PRINT#4,i:PAUSE 20
331     a=RND(1 TO 6):INK#4,7:CURSOR#4,197,170:PRINT#4,a:PAUSE 20
332     IF a=6 :INK#2,4:gmax=gmax-1:glev=glev-1:EXIT G_lp
333     IF a<>6:INK#2,2:snum=snum-50:Score
334   INK#2,6:CURSOR#2,236,190:PRINT#2,'Try Again':CLS#2,4:PAUSE 30
335   END IF
336   IF k=51 AND snum>50+50*lev:snum=snum-50-50*lev:gdel=120:EXIT G_lp
337   IF k=52 AND snum>500+500*lev
338      snum=snum-500-500*lev:gmax=gmax-glev:glev=0:EXIT G_lp
339   END IF
340 END REPeat G_lp
341 GView:Score:CURSOR#2,236,190:CLS#2,4:INK 7:MPort:PAUSE 20:MazView
342 INK#2,5:CURSOR#2,148,30+10*lev:PRINT#2,glev;'   ';gmax;'   '
343 END DEFine
```



Use [1]Shield [2]Sword [3]Mask [4]Ring

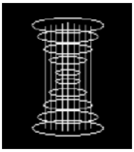
Note: The **MGuard PROC**edure is activated after a number of moves set by a random number. [120 / RND(2 to 4) {ie. 30 40 60 moves}]

Mask - Banishes (**glev**) Level Guardians for 120 moves
Ring – Reduces Guardians (**glev**=0 & **gmax** by **glev**)

```
345 REMark Vector Graphics
```

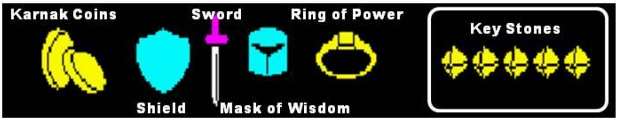
Note: These are draw in **ch**annel's #1, #2, #4

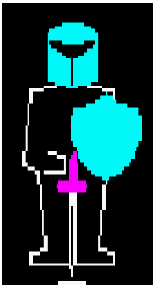
```
347 DEFine PROCedure MPort
348 BEEP 2000,20,40,190,0,0,0:ch=1:x=0:y=20:INK#ch,7
349 FOR i=0 TO 16 STEP 4
350   CIRCLE#ch,0,-20+i*2,25-i,.2,PI/2:CIRCLE#ch,0,50-i*2,25-i,.2,PI/2
351   LINE#ch,-i,-22+i/8 TO -i,52-i/8:LINE#ch,+i,-22+i/8 TO i,52-i/8
352 END FOR i
353 END DEFine
```



```
355 DEFine PROCedure Coin
356 INK#ch,6:FILL#ch,1:CIRCLE#ch,x,y,10,.6,PI:FILL#ch,0
357 INK#ch,0:CIRCLE#ch,x+3,y-1,10,.7,PI
358 INK#ch,6:FILL#ch,1:CIRCLE#ch,x+10,y-4,10,.6,PI/4:FILL#ch,0
359 INK#ch,0:CIRCLE#ch,x+10,y-4,10,.6,PI/4
360 INK#ch,0:CIRCLE#ch,x+12,y-4,9,.5,PI/4
361 END DEFine
```

```
363 DEFine PROCedure Shield
364 FILL#ch,1:INK#ch,5:ARC#ch,x,y TO x-9,y-4, -PI/4
365 ARC#ch,x-9,y-4 TO x,y-22, PI/2:ARC#ch,x,y-22 TO x+9,y-4, PI/2
366 ARC#ch,x+9,y-4 TO x,y, -PI/4:FILL#ch,0
367 END DEFine:

369 DEFine PROCedure Sword
370 FILL#ch,1:INK#ch,7
371 LINE#ch,x,y TO x-1,y+3 TO x-1,y+20 TO x+1,y+20 TO x+1,y+3 TO x,y
372 FILL#ch,0:INK#ch,0:LINE#ch,x,y+2 TO x,y+18:INK#ch,3
373 FILL#ch,1:CIRCLE#ch,x,y+22,5,.2,PI/2:FILL#ch,0
374 FILL#ch,1:CIRCLE#ch,x,y+26,5,.2,PI:FILL#ch,0:CIRCLE#ch,x,y+28,1
375 END DEFine

377 DEFine PROCedure Mask
378 INK#ch,5:FILL#ch,1:ARC#ch,x+7,y+9 TO x-7,y+9,PI/2
379 LINE#ch,x-7,y+9 TO x-7,y-2 TO x,y-4 TO x+7,y-2 TO x+7,y+9:FILL#ch,0
380  INK#ch,0:FILL#ch,1:LINE#ch,x+6,y+7 TO x,y+6 TO x,y+3 TO x+6,y+7:FILL#ch,0
381 FILL#ch,1:LINE#ch,x-6,y+7 TO x,y+6 TO x,y+3 TO x-6,y+7:FILL#ch,0
382 LINE#ch,x,y+4 TO x,y-4
383 END DEFine

385 DEFine PROCedure Ring
386 INK#ch,6:FILL#ch,1:CIRCLE#ch,x,y,11,.6,PI/2 :FILL#ch,0
387 INK#ch,0:FILL#ch,1:CIRCLE#ch,x,y-1,9,.5,PI/2:FILL#ch,0
388 INK#ch,6:FILL#ch,1:CIRCLE#ch,x,y+6,5,.5,PI/2:FILL#ch,0
389 INK#ch,0:LINE#ch,x-3,y+9 TO x+3,y+9 TO x+3,y+5 TO x-3,y+5 TO x-3,y+9
390 END DEFine
```





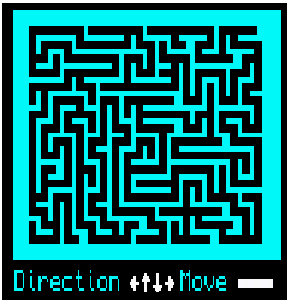
```
392 DEFine PROCedure KStone
393 BEEP 2000,20,40,190,0,0,0:INK#ch,scol:FILL#ch,1
394 LINE#ch,x,y+6 TO x-6,y TO x,y-6 TO x+6,y TO x,y+6:FILL#ch,0
395 INK#ch,0:LINE#ch,x,y+8 TO x-6,y TO x,y-8 TO x+6,y TO x,y+8
396 LINE#ch,x,y+8 TO x-2,y-2 TO x,y-8
397 LINE#ch,x-6,y TO x-2,y-2 TO x+6,y
398 END DEFine

400 DEFine PROCedure Guard(ch)
401 IF ch=1:x=0:y=- 4  :INK#ch,col:FILL#ch,fil
402 IF ch=4:x=140:y=104:INK#ch,col:FILL#ch,fil
403 LINE#ch,x-5,y+32 TO x-5,y+26 TO x-12,y+25 TO x-14,y+6 TO x-10,y+4 TO x-8,y-16 TO x-12,y-17 TO
x-12,y-20 TO x+12,y-20 TO x+12,y-17 TO x+8,y-16 TO x+10,y+4 TO x+14,y+6 TO x+12,y+25 TO x+5,y+26
TO x+5,y+32 TO x-5,y+32:FILL#ch,0
404 INK#ch,7:LINE#ch,x-8,y+4 TO x-2,y+4 TO x-2,y+8 TO x-8,y+9
405 IF ch=1:x=  10:y= 20:Shield:x=  0:y=-24:Sword:x=  0:y=26:Mask
406 IF ch=4:x=150:y=128:Shield:x=140:y= 82:Sword:x=140:y=134:Mask
407 END DEFine
```

```
409 REMark New Game / Level Change

411 DEFine PROCedure MazNew
412 w=20:h=16:DIM grid(w+1,h+1),cell(w*h,2),pm(5),fv(5)
413 CLS#3:INK 7:x=w/2:y=h:cell(0,1)=x:cell(0,2)=y:inc=40:cn=1
414 INK#2,5:CURSOR#2,18,190:PRINT#2,'Direction          Move':INK#2,7
415 CURSOR#2,76,190:PRINT#2,CHR$(188);' ↑ ↓  ← ':BLOCK#2,18,3,136,194,7
416 FOR n=1 TO w*h-1
417   p=0:PAUSE mp                                   :REMark p - Pass / mp - 0.5 Pause delay
418   IF x>1 AND grid(x-1,y)=0 : p=p+1:pm(p)=1       :REMark West wall
419   IF x<w AND grid(x+1,y)=0 : p=p+1:pm(p)=2       :REMark East wall
420   IF y>1 AND grid(x,y-1)=0 : p=p+1:pm(p)=3       :REMark North wall
421   IF y<h AND grid(x,y+1)=0 : p=p+1:pm(p)=4       :REMark South wall
422   IF p=0
423    IF m=1:cn=cn-1:x=cell(cn,1):y=cell(cn,2)
424    IF m=2:cn=0:x=RND(w):y=RND(h):cell(cn,1)=x:cell(cn,2)=y
425    IF m=3:x=x+1:IF x>w : x=1 :y=y+1:IF y>h : y=1
426    IF grid(x,y)=0 : GO TO 444
427    GO TO 439
428   END IF
429   r=pm(RND(1 TO p)):cn=cn+1:cell(cn,1)=x:cell(cn,2)=y
430   IF r=1 : grid(x,y)=grid(x,y)+1:x=x-1:grid(x,y)=2:bx=x*6:by=y*5
431   IF r=2 : grid(x,y)=grid(x,y)+2:bx=x*6:x=x+1:grid(x,y)=1:by=y*5
432   IF r=3 : grid(x,y)=grid(x,y)+4:y=y-1:grid(x,y)=8:bx=x*6:by=y*5
433   IF r=4 : grid(x,y)=grid(x,y)+8:by=y*5:y=y+1:grid(x,y)=4:bx=x*6
434   IF r=1 OR r=2 :BLOCK#3,8,3,2+bx,1+by,0
435   IF r=3 OR r=4 :BLOCK#3,4,8,2+bx,1+by,0
436  END FOR n
437 END DEFine
```

↵ Enter Symbol

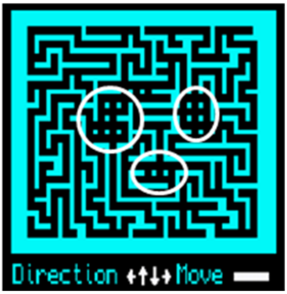
Note: The Maze algorithms

m=1 Recursive backtracking
m=2 Prims Algorithm
m=3 Hunt and Kill Method

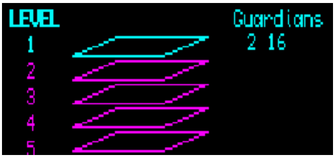


Note: For New Game **mp** set to 0.5 draws the Maze so as to show its construction running on faster QL Hardwar/Software Platforms...

```
439 DEFine PROCedure MazHall
440 REMark grid r row/c col:hw hall width in cells:cw cell wall:sf side facing
441 FOR hall=1 TO 6
442  tx=RND(4 TO 16):ty=RND(4 TO 12):RESTORE 338
443  FOR r=0 TO 1
444   FOR c=0 TO 2
445    BLOCK#3,16,8,2+tx*6,1+ty*5,0
446    BLOCK#3,2,2,6+tx*6,4+ty*5,5 :BLOCK#3,2,2,12+tx*6,4+ty*5,5
447    FOR hw=1 TO 3
448      cw=grid(tx+c,ty+r):READ sf
449      IF Walls(sf,cw)=0:grid(tx+c,ty+r)=grid(tx+c,ty+r)+sf
450    END FOR hw
451   END FOR c
452  END FOR r
453 END FOR hall
454 END DEFine

456 DATA 2,8,8,1,2,8,1,8,8,2,4,4,1,2,4,1,4,4
```



Note: Making alternative routes available by adding further inter-connections.



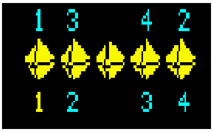
```
458 DEFine PROCedure MazLev
459 BLOCK#2,30,56,148,40,0:BLOCK#4,200,30,300,216,0
460 FOR i=1 TO 5
461   IF i=lev:INK#2,5:ELSE INK#2,3
462   CURSOR#2,16,30+i*11:PRINT#2,i
463   LINE#2,26,84-i*5 TO 46,84-i*5 TO 36,80-i*5 TO 16,80-i*5 TO 26,84-i*5
464 END FOR i
465 IF lev=1:px=10:py=16:ELSE px=RND(3 TO 18):py=RND(2 TO 14)
466 IF lev>1:ch=4:scol=6:y=20:FOR i=1 TO lev-1:x=280+i*12:KStone
467 INK#2,5 : CURSOR#2,148,30+10*lev:PRINT#2,glev;'  ';gmax
468 END DEFine
```

```
470 DEFine PROCedure MazTres
471 DIM Tres(12,3):n=1
472 :
473  REPeat  t_lp
474     IF n>12:n=1:EXIT t_lp
475     tx=RND(1 TO w):ty=RND(1 TO h):tn=grid(tx,ty)
476     FOR i=1 TO n:IF Tres(i,1)=tx AND Tres(i,2)=ty:NEXT t_lp
477     SELect ON tn=1,2,4,8:Tres(n,1)=tx:Tres(n,2)=ty:n=n+1
478  END REPeat t_lp
479  FOR i=1 TO 12:Tres(i,3)=RND(2 TO 6)
480  Tres(3,3)=7:km=0:Tres(11,3)=8:kr=0:Tres(7,3)=9
481 :
482 END DEFine:
```

Note:
Coins of Karnak(2 to 6)
Mask(7) / Ring(8) / KeyStone(9)

Note: Twelve dead ends to passageways are chosen for the Treasure Locations. The Treasures are then allocated one to each of twelve location. Coins of Karnak are distributed to all locations, then three of them are chosen to hold the **Mask**, **Ring** and **Key Stone**.

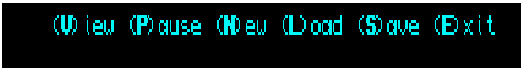
```
484 DEFine PROCedure MazKey
485 DIM Mkey(5):RESTORE 491:ra=RND(24)
486 FOR i=1 TO 24
487   READ a,b,c,d:IF i=ra:Skey(1,3)=a:Skey(2,3)=b:Skey(3,3)=c:Skey(4,3)=d
488 END FOR i
489 END DEFine

491 DATA 1,2,3,4, 1,3,2,4, 2,3,1,4, 2,1,3,4, 3,1,2,4, 3,2,1,4
492 DATA 2,3,4,1, 3,2,4,1, 3,1,4,2, 1,3,4,2, 1,2,4,3, 2,1,4,3
493 DATA 3,4,1,2, 2,4,1,3, 1,4,2,3, 3,4,2,1, 2,4,3,1, 1,4,3,2
494 DATA 4,1,2,3, 4,1,3,2, 4,2,3,1, 4,2,1,3, 4,3,1,2, 4,3,2,1
```



Note: Entering the **Sphere of Destiny** the collected **Key Stones** have to be arranged in the same order as those presented in the Sphere. This requires the correct **matching** of both sets of Key stones. Taking the fifth Stone as already set the other four stones can create 24 different combination sets.

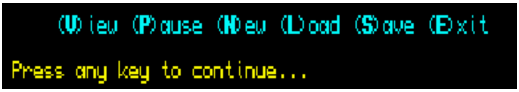
```
496 REMark Menu Commands

498 DEFine PROCedure Mes1
499 BLOCK#2,280,26,200,40,0
500 INK#2,6:CURSOR#2,220,48:PRINT#2,'Select (N)ew or (L)oad':gck=0            :Maze_Ctrl
501 END DEFine

503 DEFine PROCedure GView
504 IF gck=0:RETurn                                               :REMark gck=0 NO Maze_Init
505 IF tc=0 :tc=3:bc=7:sl=lev*5:ELSE tc=0:bc=0:sl=lev
506 FOR n=1 TO  12:IF Tres(n,3)>0:BLOCK#3,4,3,2+Tres(n,1)*6,1+Tres(n,2)*5,tc
507IF sc=241
508   BLOCK#3,4,3,2+Tres(7,1)*6,1+Tres(7,2)*5,sc:PAUSE 10:sc=0
509 END IF
510 BLOCK#3,4,3,2+cx*6,1+cy*5,0:cx=px:cy=py:BLOCK#3,4,3,2+cx*6,1+cy*5,bc
511 END DEFine
```

Note: (V) Toggles **ON/OFF** Highlighted
Cells seen on the 2D Maze Layout



(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit

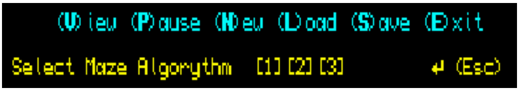
```
513 DEFine PROCedure GPause
514 IF gck=0:RETurn
515 INK#2,6:CURSOR#2,200,48:PRINT#2,'Press any key to continue...'
516 GTS=(DATE-Gclk+GTS):PAUSE:CURSOR#2,200,48:CLS#2,4
517 END DEFine
```



(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit

Press any key to continue...

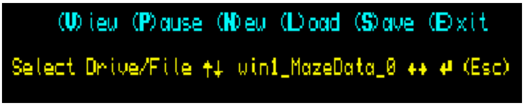
```
519 DEFine PROCedure GNew
520  GTS=(DATE-Gclk+GTS):INK#2,6:CURSOR#2,200,48
521 PRINT#2,'Select Maze Algorithm  [1][2][3]    ← (Esc)'          ↲
522 BLOCK#2,2,4,446,50,6:INK#2,3
523 REPeat New_lp
524  MSel:k=CODE(INKEY$(-1))
525  SELect ON k
526    =49,50,51:m=k-48
527    =27:CURSOR#2,200,48:CLS#2,4:RETurn
528    =10:CURSOR#2,200,48:CLS#2,4:EXIT New_lp
529  END SELect
530 END REPeat New_lp
531 gdel=120/RND(3 TO 4):gmax=16:glev=1:lev=1:col=5:MazLev:MazKey
532 GTS=0:Gclk=DATE: sm=0:snum=2000:Score: gck=1:gst=1
533 w=20:h=16:mp=.5:MazNew:MazHall:TMazTres:MazView:mp=0       :REMark mp Maze PAUSE
534 END DEFine
```



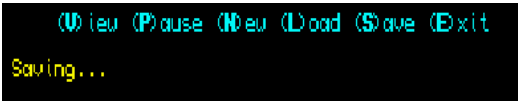
(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit

Select Maze Algorythm [1] [2] [3] ↲ (Esc)

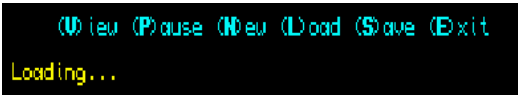
```
536 DEFine PROCedure MSel
537 INK#2,3:IF m=1:CURSOR#2,320,16:PRINT#2,'Recursive Backtraking':CLS#2,4
538 IF m=2:CURSOR#2,320,16:PRINT#2,'Prims Algorithm':CLS#2,4
539 IF m=3:CURSOR#2,320,16:PRINT#2,'Hunt and Kill Method':CLS#2,4
540 END DEFine

542 DEFine PROCedure PSel
543 GTS=(DATE-Gclk+GTS):INK#2,6
544 CURSOR#2,200,48:PRINT#2,'Select Drive/File↑ ↓ ':CLS#2,4
545 CURSOR#2,424,48:PRINT#2,' ↑ ↓ ← (Esc)':BLOCK#2,2,4,448,50,6        ↵
546 REPeat Path_lp
547   CURSOR#2,328,48:PRINT#2,drv$(dv)&GDat$(f)
548   k=CODE(INKEY$(-1))
549   SELect ON k
550     =192:f=f-1:IF f<0:f=9
551     =200:f=f+1:IF f>9:f=0
552     =208:dv=dv-1:IF dv<1:dv=8
553     =216:dv=dv+1:IF dv>8:dv=1
554     = 10:file=1:EXIT Path_lp
555     = 27:file=0:RETurn
556   END SELect
557 END REPeat Path_lp
558 device_filename$=drv$(dv)&GDat$(f):Gf$=GDat$(f)
559 END DEFine
```


(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit
Select Drive/File ↑↓ win1_MazeData_0 ↔ ↵ (Esc)

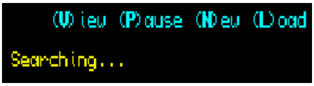
```
561 DEFine PROCedure GSave
562 IF file=0 OR gck=0:CURSOR#2,200,48:CLS#2,4:RETurn
563 DELETE device_filename$
564 CURSOR#2,200,48:PRINT#2,'Saving...';:CLS#2,4
565 OPEN_NEW#99,device_filename$
566 FOR n=1 TO 12:PRINT#2,'.';:PAUSE 1:PRINT#99,Tres(n,3)
567 PRINT#99,m\lev\gmax\glev\km\kr\GTS\sm\snum:CLOSE#99
568 CURSOR#2,200,48:CLS#2,4
569 END DEFine
```


(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit
Saving...

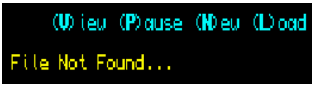
```
571 DEFine PROCedure GLoad
572 IF file=0:CURSOR#2,200,48:CLS#2,4:RETurn
573 FChk:IF file=0:CURSOR#2,200,48:CLS#2,4:RETurn
574 OPEN_IN#99,device_filename$
575 CURSOR#2,200,48:PRINT#2,'Loading...';:CLS#2,4
576 CLS:MazNew:MazHall:MazTres:MazKey
577 CURSOR#2,260,48:FOR n=1 TO 12:PRINT#2,'.';:PAUSE 1:INPUT#99,Tres(n,3)
578 INPUT#99,m\lev\gmax\glev\km;mazkey\kr\GTS\sm\snum:CLOSE#99
579 MSel:MazLev:Score:MazView
580 IF km=1:ch=4:x=220:y=20:Mask
581 IF kr =1 :ch=4:x=250:y=22:Ring
582 CURSOR#2,200,48:CLS#2,4:gdel=120/RND(2 TO 4):gck=1:gst=1
583 END DEFine
```


(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit
Loading...

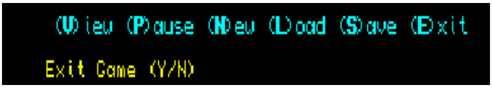
```
585 DEFine PROCedure FChk
586 CURSOR#2,200,48:PRINT#2,'Searching...':CLS#2,4
587 DELETE drv$(dv)&'FList'
588 OPEN_NEW#99,drv$(dv)&'FList':DIR#99,drv$(dv):CLOSE#99
589 OPEN_IN#99,drv$(dv)&'FList'
590 REPeat Dir_lp
591   IF EOF(#99)
592     CLOSE#99:CURSOR#2,200,48:PRINT#2,'File Not Found...'
593     PAUSE  50:CURSOR#2,200,48:CLS#2,4:file=0:slk=1:RETurn
594   END IF
595   INPUT#99,Fchk$:IF Fchk$==Gf$:CLOSE#99:EXIT Dir_lp
596 END REPeat Dir_lp
597 END DEFine
```



```
(V)iew (P)ause (N)ew (L)oad
Searching...
```

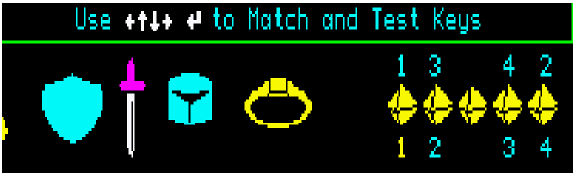


```
(V)iew (P)ause (N)ew (L)oad
File Not Found...
```

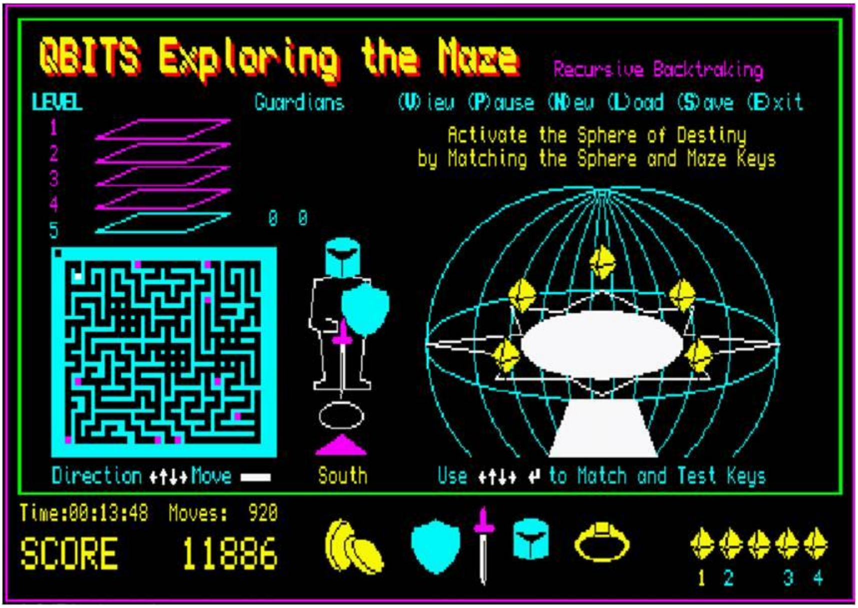


```
(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit
Exit Game (Y/N)
```

```
599 DEFine PROCedure GExit
600 INK#2,6:CURSOR#2,200,48:PRINT#2,'Exit Game (Y/N)':PAUSE
601 IF KEYROW(5)<>64:CURSOR#2,200,48:CLS#2,4:RETurn
602 FOR ch=3 TO 5:CLOSE#ch
603 WINDOW#1,496,204,8+gx,  6+gy:PAPER#1,0:INK#1,7:CLS#1
604 WINDOW#0,496, 40,8+gx,212+gy:PAPER#0,0:INK#0,7:CLS#0
605 INK#2,7:PRINT#0,'Bye...':STOP
606 END DEFine

608 REMark Setting The Sphere of Destiny Keystones

610 DEFine PROCedure KeyStone
611 ch =4:scol=6:x=340:y=20:KStone:SDest:SRing
612 CURSOR#2,256,44:PRINT#2,'Activate the Sphere of Destiny'
613 CURSOR#2,238,54:PRINT#2,'by Matching the Sphere and Maze Keys'
614 INK#2,5:CURSOR#2,250,190:PRINT#2,'Use              to Match and Test Keys'
615 INK#2,7:CURSOR#2,274,190:PRINT#2,' ←↑↓→  ←':BLOCK#2,2,4,310,192,7   ↵ Enter
616 check=0:col=0:FOR kp=1 TO 4:ks=kp:GetKey
617 REPeat key_lp
618   IF snum<500:snum=0:Score:Mes2:EXIT key_lp
619   GetKey:k=CODE(INKEY$(-1))
620   SELect ON k
621     =192:kp=kp -1:IF kp<1:kp=4
622     =200:kp=kp+1:IF kp>4:kp=1
623     =208:ks=ks+1:IF ks>4:ks=1
624     =216:ks=ks -1:IF ks<1:ks=4
625     = 10:MatchKey:IF check<5: snum=snum-500:Score: ELSE  Mes3:EXIT key_lp
626     =244:IF col=0:col=5:ELSE col=0
627    END SELect
628 END REPeat key_lp
629  gst=0:Maze_Ctrl
630 END DEFine
```

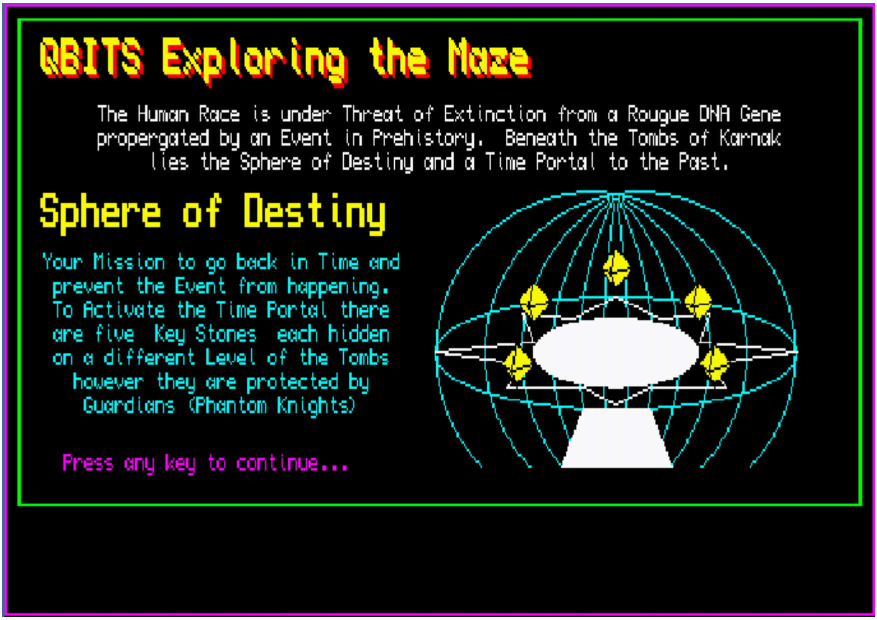


Use ←↑↓→ ↵ to Match and Test Keys



Activate the Sphere of Destiny
by Matching the Sphere and Maze Keys

```
632 DEFine PROCedure MatchKey
633 check=1:FOR i=1 TO 4:IF Skey(i,3)=Mkey(i):check=check+1
634 END DEFine

636 DEFine PROCedure GetKey
637 IF kp=1:Mkey(1)=ks:c=414
638 IF kp=2:Mkey(2)=ks:c=430
639 IF kp=3:Mkey(3)=ks:c=466
640 IF kp=4:Mkey(4)=ks;c=485
641 RESTORE 644:INK#4,col:For i=1 to 4:READ a:CURSOR#4,a,212:PRINT#4,a,212:PRINT#4,SKey(i,3)
642 RESTORE 644:INK#4,5   :For i=1 to 4:READ a:CURSOR#4,a,240:PRINT#4,a,212:PRINT#4,MKey(i)
643 INK#4,6       :CURSOR#4,c,240:PRINT#4,MKey(kp)
644 DATA 414,430,466,484
645 END DEFine

647 DEFine PROCedure Mes2
648 If gst=0:RETurn
649 INK#2,6:CURSOR#2,236,190:PRINT#2,'Hard Luck You FAILED - Try a New Game '
650 CLS:CLS#3:lev=1:glev=2:MazLev::SEnd:col=0:fil=1:Guard(1):Mes1
651 END DEFine

653 DEFine PROCedure Mes3
654 INK#2,6:CURSOR#2,236,190:PRINT#2,'The Past has Changed - Humanity Saved '
655 CLS:CLS#3:lev=1:glev=2:MazLev:SEnd:ch=1:LName:Mes1
656 END DEFine
```



QBITS Exploring the Maze

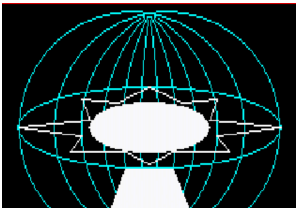
The Human Race is under Threat of Extinction from a Rougue DNA Gene
propergated by an Event in Prehistory. Beneath the Tombs of Karnak
lies the Sphere of Destiny and a Time Portal to the Past.

Sphere of Destiny

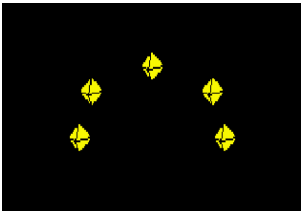
Your Mission to go back in Time and
prevent the Event from happening.
To Activate the Time Portal there
are five Key Stones each hidden
on a different Level of the Tombs
however they are protected by
Guardians (Phantom Knights)

Press any key to continue...

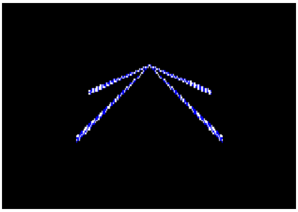
```
658 DEFine PROCedure Game_Intro
659 DIM S$(3,70),M$(7,40)
660 S$(1)='The Human Race is under Threat of Extinction from a Rougue DNA Gene'
661 S$(2)='propergated by an Event in Prehistory.  Beneath the Tombs of Karnak'
662 S$(3)='     lies the Sphere of Destiny and a Time Portal to the Past.'
663 M$(1)='Your Mission to go back in Time and'
664 M$(2)=' prevent the Event from happening.'
665 M$(3)=' To Activate the Time Portal there'
666 M$(4)=' are five  Key Stones  each hidden'
667 M$(5)=' on a different Level of the Tombs'
668 M$(6)='   however they are protected by'
669 M$(7)='    Guardians (Phantom Knights)'
670 INK#2,7:FOR i=1 TO 3:CURSOR#2,44,24+i*10:PRINT#2,S$(i)
671 CSIZE#2,2,1:OVER#2,1
672INK#2,6:FOR i=1 TO 2:CURSOR#2,8+i,70:PRINT#2,'Sphere of Destiny'
673 CSIZE#2,0,0:OVER#2,0
674  INK#2,5:FOR i=1 TO 7:CURSOR#2,12,86+i*10:PRINT#2,M$(i)
675 INK#2,3:CURSOR#2,24,180:PRINT#2,'Press any key to continue...'
676 SDest:SRing:PAUSE:ARing:SEnd:col=0:fil=1:Guard(1)
677 END DEFine
```



```
679 DEFine PROCedure SDest
680 col=5:ss=8:x=0:y=10:INK 7:FILL 0
681 REPeat sphere_lp
682  FOR i=0 TO 1.1  STEP .1
683    ARC x,y+ss TO x,y-ss,PI*i
684    ARC x,y+ss TO x,y-ss,-PI*i
685    INK col:IF col=5:col=0:ELSE col=5
686  END FOR i
687  BEEP 2000,8,20,-8,0,0,0:ss=ss+8:IF ss>56:EXIT sphere_lp
688  PAUSE 5:INK 0:FILL 1:CIRCLE x,y,36+ss,ss*2/100,PI:FILL 0
689 END REPeat sphere_lp
690 INK 5:CIRCLE x,y,66,.3,PI/2:y=20:INK 7
691 LINE x,y+10 TO x-16,y+2 TO x-34,y+4 TO x-30,y-6 TO x-66,y-10 TO x-34,y-14 TO x-40,y-22 TO x-12,y-22 TO x,y-28 TO x+12,y-22 TO x+40,y-22 TO x+34,y-14 TO x+66,y-10 TO x+30,y-6 TO x+34,y+4 TO x+16,y+2 TO x,y+10
692 FILL 1:LINE x-12,y-30 TO x-20,y-50 TO x+20,y-50 TO x+12,y-30 TO x-12,y-30:FILL 0
693 FILL 1:CIRCLE x,y-10,30,.4,PI/2:FILL 0:INK 0:PAUSE 10
694 END DEFine
```

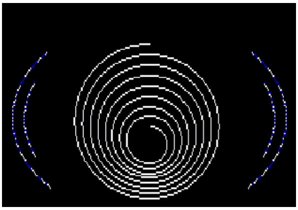


```
696 DEFine PROCedure SRing
697 ch=1:scol=6:FOR i=1 TO 5:x=Skey(i,1):y=Skey(i,2):KStone:PAUSE 10
698 END DEFine
```

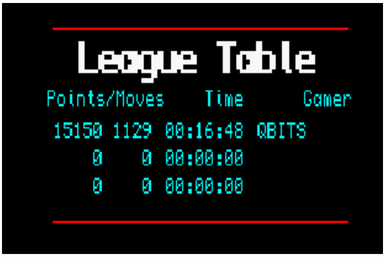


```
700 DEFine PROCedure ARing
701 INK 241:x1=Skey(i,1):y1=Skey(i,2)
702 FOR i=1 TO 4
703   x2=Skey(i,1):y2=Skey(i,2)
704   FILL 1:LINE x1,y1 TO x2,y2 TO x2,y2-2 TO x1,y1:FILL 0
705 END FOR i
706 END DEFine
```

```
708 DEFine PROCedure SEnd
709 FOR i=1 TO 24 STEP 2
710    INK 241:CIRCLE 0,14,i*3,.7,PI/2:BEEP 2000,40,120,90,0,0,0:PAUSE 5
711 END FOR i
712 INK 0:FILL 1:CIRCLE,0,14,60:FILL 0
713 BEEP 30000,1,250,90,-8,15,15:INK 7
714 FOR i=50 TO 15 STEP -5
715    ARC 0,i TO 0,-i/2,PI:ARC 0,-i/2 TO 0,i-5,PI:PAUSE i/5
716 END FOR i
717 BEEP 10000,4,200,190,0,0,0:PAUSE 20
718 END DEFine
```



```
720 REMark League Table
```



```
League Table
Points/Moves    Time      Gamer
15150 1129 00:16:48 QBITS
    0    0 00:00:00
    0    0 00:00:00
```

```
722 DEFine PROCedure LScore
723 FOR i=1 TO 7
724   PAUSE 5:BLOCK180,12*i,30,66-i*6,0
725   BLOCK 180,1,30,66-i*6,2:BLOCK180,1,30,66+i*6,2
726 END FOR i
727 OVER 1:CSIZE 2,1:INK 7
728 FOR i=1 TO 2:CURSOR 44+i,28:PRINT 'League Table'
729 OVER 0:CSIZE 1,0:INK 5
730 CURSOR 26+i,50:PRINT'Score/Moves    Time       Gamer'
731 FOR a=1 TO 3
732  HS1=Grad(a,1):HS2=Grad(a.2):HST$=DATE$(Grad(a,3))
733  CURSOR  24,52+a*12:PRINT FILL$(' ',5-LEN(HS1))&HS1
734  CURSOR  66,52+a*12:PRINT FILL$(' ',4-LEN(HS2))&HS2
735  CURSOR  98,52+a*12:PRINT HST$(13 TO 20)
736  CURSOR 154,52+a*12:PRINT name$(a)
737 END FOR a
738 END DEFine

740 DEFine PROCedure LName
741 GTS=DATE-Gclk+GTS:Gclk$=DATE$(GTS)
742 FOR i=1 TO 3
743   IF Grad(i,1)<snum:Gmr=i:EXIT i:ELSE Gmr=0
744 END FOR I
745 IF Gmr=0:
746   LScore:RETurn
747  ELSE
748     Grad(Gmr,1)=snum:Grad(Gmr,2)=GTS:LScore
749     ch=6:OPEN#ch,con_10x10a0x0_10:WINDOW#ch,60,10,390+gx,126+gy+Gmr*12
750     PAPER#ch,0:CLS#ch:INK#ch,6:INPUT#ch,name$(Gmr)
751     CLOSE#ch:LSave
752  END IF
753 END DEFine

755 DEFine PROCedure LSave
756 DELETE drv$(dv)&'QBMazeDLT':OPEN_NEW#99,drv$(dv)&'QBMazeLT'
757 FOR a=1 TO 3:PRINT#99,name$(a)\Grad(a,1)\Grad(a,2)
758 CLOSE#99
759 END DEFine

761 DEFine PROCedure LLoad
762 OPEN_IN#99,drv$(dv)&'QBMazeLT'
763 FOR a=1 TO 3:INPUT#99,name$(a)\Grad(a,1)\Grad(a,2)
764 CLOSE#99
765 END DEFine
```

```
767 DEFine PROCedure LTDefault
768 REMark Score League Table
769 name$(1)='QBITS     ':Grad(1,1)=1730:Grad(1,2)=1072
770 name$(2)='          ':Grad(2,1)=   0:Grad(2,2)=0
771 name$(3)='          ':Grad(3,1)=   0:Grad(3,2)=0
772 REMark LSave
773 END DEFine

775 DEFine PROCedure LTReset
776 REMark Resets League Table
777 name$(1)='          ':Grad(1,1)= 0:Grad(1,2)=0
778 name$(2)='          ':Grad(2,1)= 0:Grad(2,2)=0
779 name$(3)='          ':Grad(3,1)= 0:Grad(3,2)=0
780 REMark LSave
781 END DEFine
```

Note: Load QBMazeQPC_v04dec then call **LTDefault & LTReset** as and when required.

QBITS Maze – Prog Checks

My programs versions, the first draft of the code, then adding Intro and Menus, followed a full code revision. This would be removing the test checks and maybe some renaming variable etc. Early run errors are the usual typos or missing variables, and possible arithmetic overflows, division by zero etc. Hopefully the parsing of the interpreter will identify some or most of these. Where program code is not performing as expected it is useful to locate the area generating the problem. For this I place a number of PAUSE commands in the code. Possibly add PRINT statements identifying the changes to variables and array information.

QBITS Exploring the Maze

For the final Game checks I utilise hidden commands. These are the programmers so called cheats that Gamers seek to uncover.

F1 Activates **Guardian** with **[1]Shield [2]Sword [3]Mask [4]Ring** action choices.

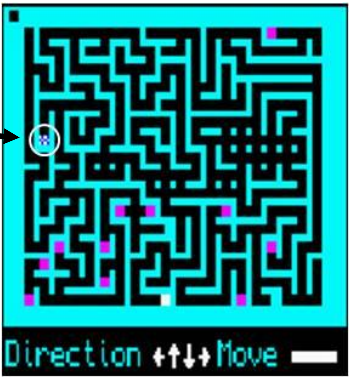
F2 Activates the **Portal** (**Y/N**) jump to the next level.

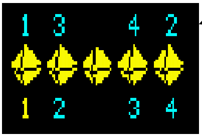
F3 adds 50 to **snum**

F4 subtracts 50 from **snum**

F5 For **Maze Levels** this flashes the **Key Stone** Location in the **2D Maze**

For the **Sphere of Destiny** it identifies the order of the **Sphere Key Stones**.



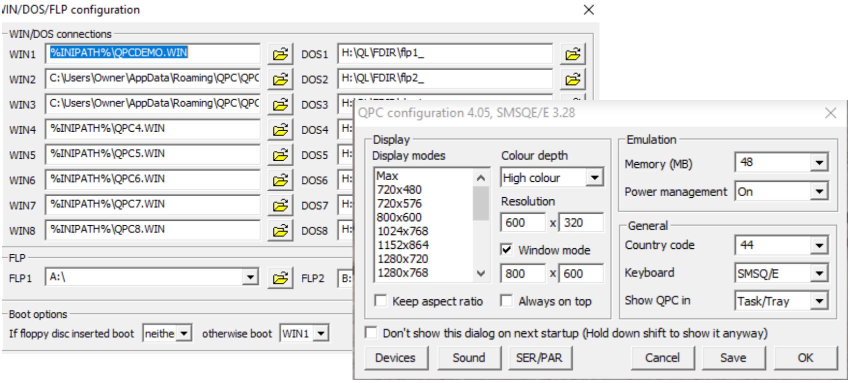


Emulators QL2K & QPC2:

Having downloaded a copy of the **QBITS Maze S**uper/SBASIC code and loaded it into a recognised QL device. Use the QDOS command LRUN as shown:-

```
        LRUN flp1_QBMazeQPC_v04
or      LRUN flp1_QBMazeQL_v04
```

QPC2 Emulator - In WIN/DOS connections attach folder references to the dos drivers.



/IN/DOS/FLP configuration

WIN/DOS connections

WIN1 %INIPATH%\QPCDEMO.WIN — DOS1 H:\QL\FDIR\flp1_
WIN2 C:\Users\Owner\AppData\Roaming\QPC\QPC — DOS2 H:\QL\FDIR\flp2_
WIN3 C:\Users\Owner\AppData\Roaming\QPC\QPC — DOS3 H:
WIN4 %INIPATH%\QPC4.WIN — DOS4 H:
WIN5 %INIPATH%\QPC5.WIN — DOS5 H:
WIN6 %INIPATH%\QPC6.WIN — DOS6 H:
WIN7 %INIPATH%\QPC7.WIN — DOS7 H:
WIN8 %INIPATH%\QPC8.WIN — DOS8 H:

FLP

FLP1 A:\ — FLP2 B:

Boot options

If floppy disc inserted boot neithe — otherwise boot WIN1

QPC configuration 4.05, SMSQ/E 3.28

Display

Display modes: Max, 720x480, 720x576, 800x600, 1024x768, 1152x864, 1280x720, 1280x768

Colour depth: High colour

Resolution: 600 x 320

☑ Window mode

800 x 600

□ Keep aspect ratio □ Always on top

Emulation

Memory (MB) 48

Power management On

General

Country code 44

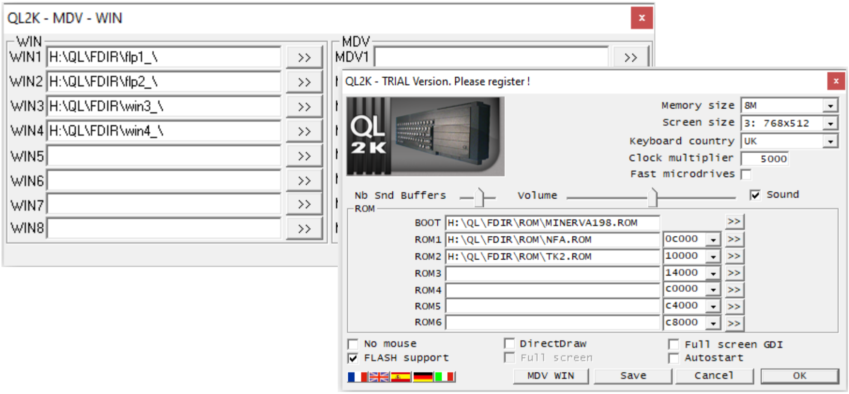
Keyboard SMSQ/E

Show QPC in Task/Tray

□ Don't show this dialog on next startup (Hold down shift to show it anyway)

QL2K Emulator

For windows this uses an application called **QLAYT-86.EXE or QLAY-X64.EXE** downloaded with **QL2K** that creates a QDOS Director**y** file and used to append or delete files in it.



QL2K - MDV - WIN

WIN

WIN1 H:\QL\FDIR\flp1_\
WIN2 H:\QL\FDIR\flp2_\
WIN3 H:\QL\FDIR\win3_\
WIN4 H:\QL\FDIR\win4_\
WIN5
WIN6
WIN7
WIN8

MDV

MDV1

QL2K - TRIAL Version. Please register !

Memory size 8M
Screen size 3: 768x512
Keyboard country UK
Clock multiplier 5000
Fast microdrives □

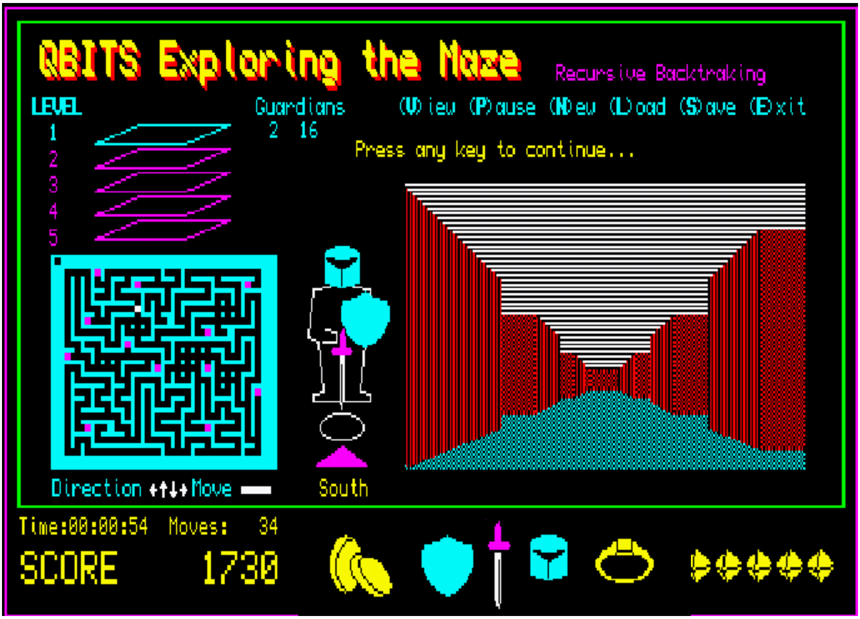
Nb Snd Buffers — Volume — ☑ Sound

ROM

BOOT H:\QL\FDIR\ROM\MINERVA198.ROM
ROM1 H:\QL\FDIR\ROM\NFA.ROM 0c000
ROM2 H:\QL\FDIR\ROM\TK2.ROM 10000
ROM3 14000
ROM4 c0000
ROM5 c4000
ROM6 c8000

□ No mouse □ DirectDraw □ Full screen GDI
☑ FLASH support □ Full screen □ Autostart

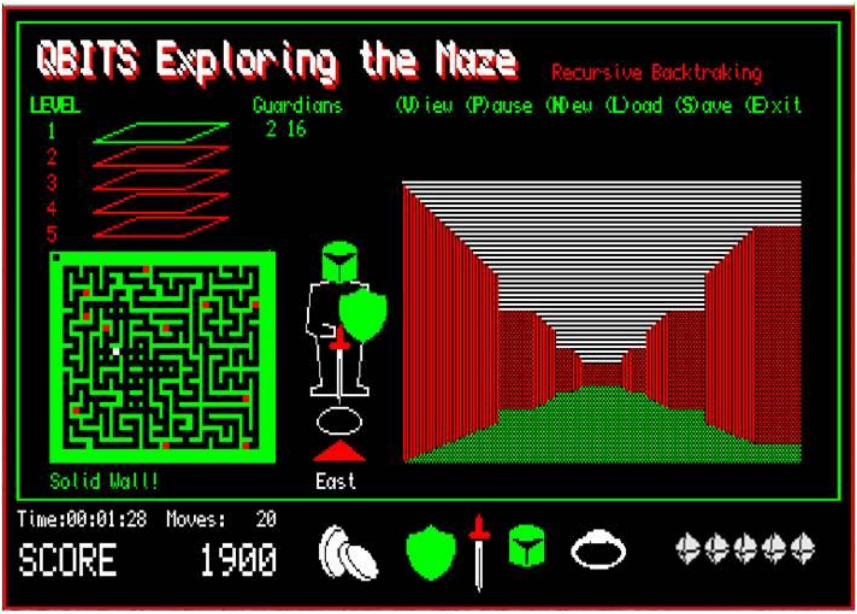
Note: Files will not **LOAD** or **RUN** if not compatible with the **QDOS** operating s**y**stem **y**ou are using. This even applies to QL software that does not work with or onl**y** works with certain versions of QL ROM's or with added Toolkit e**x**tensions.



Press any key to continue...

Time:00:00:54 Moves: 34

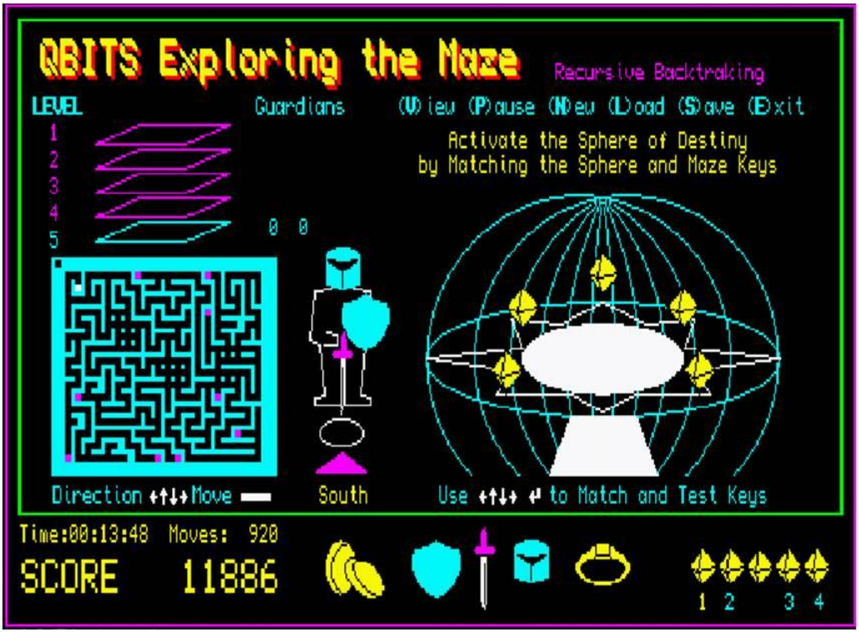
SCORE 1730



Solid Wall!!

Time:00:01:28 Moves: 20

SCORE 1900

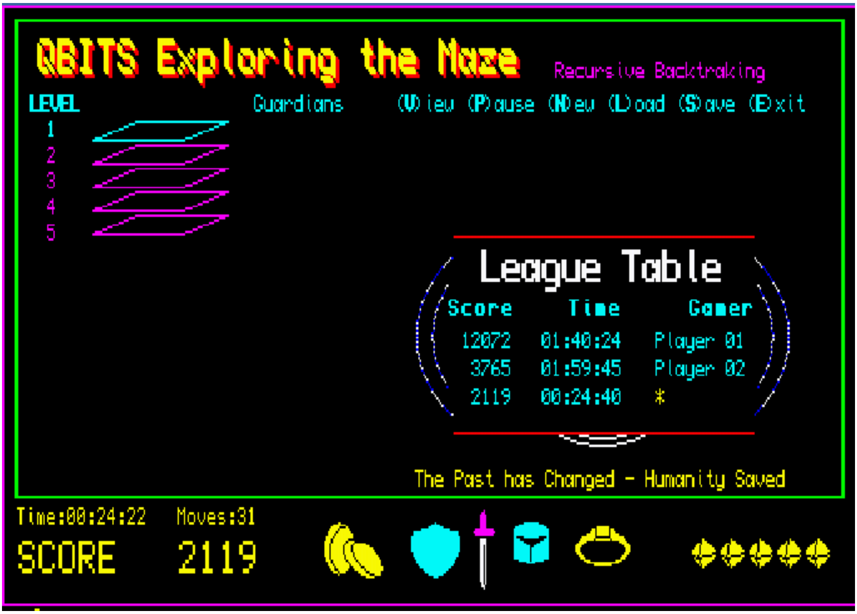


Activate the Sphere of Destiny
by Matching the Sphere and Maze Keys

Use ←↑↓→ ↵ to Match and Test Keys

Time:00:13:48 Moves: 920

SCORE 11886



League Table

Score	Time	Gamer
12072	01:40:24	Player 01
3765	01:59:45	Player 02
2119	00:24:40	*

The Past has Changed - Humanity Saved

Time:00:24:22 Moves:31

SCORE 2119