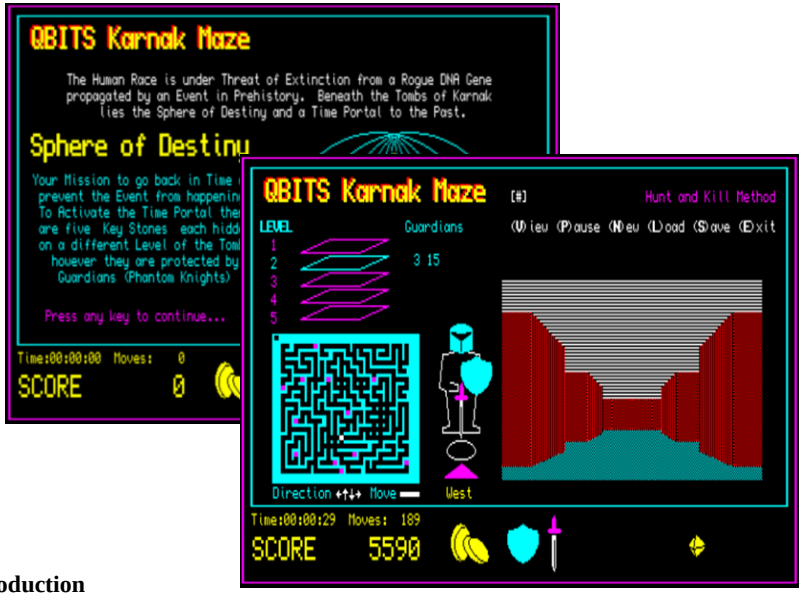


QBITS Introduction

Maze algorithms in today's Gaming World are organized as various Classifications involving Dimensions, Topology, Tessellation, Routing Texture, Focus and might use any combination. Two-dimensional Mazes are traversed by use of cardinal directions. Three dimensional Mazes use stairways or bridges with overlapping passageways connecting one area of a Maze to another. Fourth dimensional Mazes, use Portals to transport between past and future areas of the Maze.

Maze Creating Algorithms

Maze generation is usually a predetermined arrangement of cells, commonly a rectangular grid, but other arrangements are possible. A Maze Generation Algorithm is to fulfil the challenge of finding a route between any two particular cells. They are usually in the form of a random spanning tree, which simply put means a tree structure with branches that form the minimum number of undirected links between all cells.

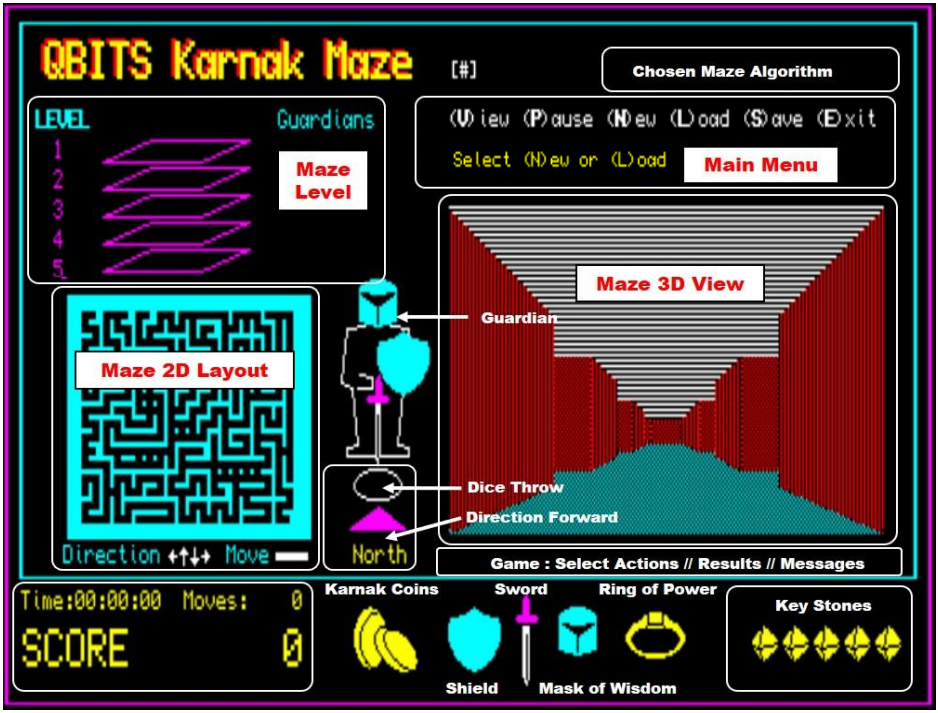
The **Recursive Backtracking** algorithm is a pathway created by randomly selecting an unvisited cell adjacent to one of the sides of the present one. The path is then made by knocking down the wall between. Moving into the new cell, the random selection continues until there are no surrounding unvisited 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.

There are a solid handful of algorithms to the above, a version called **Prim's** which 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 Karnak Maze - a Walk Through

The Intro screen displays a **Mission Statement** and the graphics of the **Sphere of Destiny** where keystones in the final part of the mission have to be aligned in the correct order to open a Portal to the psst.

The opening screen shows top left the Title '**QBITS Karnak Maze**'. Below this the **Maze Levels** which will show the active level in a different colour and the number of **Defending Guardians** to defeat. To the right of these the **Main Menu** with (**V**)iew (**P**)ause in addition to 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.



Blow on the left the Maze is shown in 2D with Direction keys, cursors for compass and spacebar to move. Centre of the screen is displayed a **Phantom 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. To the right is a 3DView of the Maze in the direction of travel. Check out blocked ends as they may reveal a hidden recess containing treasures.

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. In the middle; **Coin**s of Karnak, **Shield**, **Sword**, **Mask** of Wisdom and **Ring** of Power. On the far right the five **KeyStones**, one to be collected from each **level** of the **Maze**

QBITS Karnak Maze - Menu

QBITS Karnak Maze - (V)iew
This is an **ON/OFF** toggle 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 Karnak 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 Karnak Maze - (N)ew Game
You are first asked to select from one of three presented Maze algorithms. Top right of screen displays the 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 a **3D** view is shown 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**.

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 Karnak Maze - (L)oad
This presents you with a selection of **Device** and Game **Data Filenames**. At this point you can abort Menu choice with <**Spacebar**>, or continue with <**Enter**> which will make a search, returning '**File NOT found**' or continues with '**Loading…**'.

QBITS Karnak Maze - (S)ave
This presents you with a selection of **Device** and Game **Data Filenames**. At this point you can abort Menu choice with <**Spacebar**>, or continue with <**Enter**> if device is unavailable program will display 'DEVICE ERROR' or if already exists Overwrite Y/N. If choice is to continue then program displays 'Saving…' if not then returns to menu without saving.

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.

QBITS Karnak Maze – Phantom Guardians
Encountering a **Phantom 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 Karnak Maze - Treasures
Moving around the Maze Levels will uncover a number of Treasures; **Coins** of Karnak, **Mask** of Wisdom, **Ring** of Power and a **Key Stones**, collection of each adds Points to the **Score**. The **Mask** and **Ring** are assets that also aid in defending against or defeating the **Guardians**.

QBITS Karnak Maze - Levels
Accepting the **KeyStone** activates a **Portal** and makes a jump to the next **Level**. A jump to the **Sphere of Destiny** can only be made if all the **Phantom Guardians** have been defeated.

QBITS Karnak Maze - Sphere of Destiny
To reach the **Sphere of Destiny** the five **KeyStones** 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** use Left & Right Cursors to highlight one of four numbered Stones, use Up/Down to change number, then Enter to Test a Match.

Twenty-four different combinations are possible. Each attempt will cost points deleted from those acquired. If successful before losing all your wealth the **Time Portal** is opened and humanity saved from extinction. If not, then hard luck, try another game.

QBITS Karnak 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. All Treasures are hidden behind a false section of the wall of a dead-end passageway. If you try to go through the wall sliding doors expose a recess with **Coins** of Karnak, **Mask** of Wisdom or **Ring** of Power each adding to the Points scored or a Keystone.

The **KeyStone** location offers a choice of **Y/N** between taking the **KeyStone** or leaving it for a later pickup. Accepting the **KeyStone** will immediately activate a **Portal** to the next Level, down to Level five. Level Five's Keystone is used to reach the **Sphere of Destiny,** but all remaining **Phantom 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 at the same time managing to defeat all of the **Phantom Guardians**. So, depending on your luck or skill in choosing the Keystones arrangement, be prepared to sacrifice heavy the Points accrued to activate the **Time Portal** to go back and save **Humanity**.

Best of Luck!

QBITS Karnak Maze - SCORE
The **SCORE** shows the **Game Duration** in hours, minutes and seconds. A tally of the **Moves** taken and **Points** (wealth) accrued. 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**

QBITS Karnak Maze – Points Table
When **Points** are **Gained** or **Deleted**, these are changes are ascribed to 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 doorways to **Treasure**. Apart from adding valuable **Points** to the Score they may be helpful when dealing with the **Phantom 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 **Phantom Guardian** you have four possible choices, use the **Shield**, **Sword** and if acquired the **Mask** and/or **Ring.** The first two are given at the beginning of each level the **Mask** and **Ring** 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**

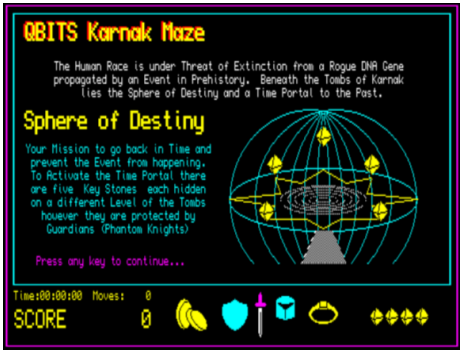
Maze Sphere of Destiny
For each failed try to Match the KeyStones **snum**=**snum**-500

QBITS KarnakMaze Procedures

Init_Win	Setup Main Windows Size and Screen Location
Init_Maze	Display Intro Screen – Game Background and Mission Statement.
QBITS_Maze	Main Menu
Score	Update of Points and movements
MazeView	Calculates Graphics to Draw 3D Maze View
Walls, Wallcalc	
Mes1	Select (N)ew or (L)oad
Mes2	Game End – 'Hard Luck You Failed - Try New Game'
Mes3	Game End – 'The Past has Changed – Humanity is Saved'
TresChk	Checks Maze Location for revealing Treasure
PortChk	KeyStone Portal Transfer to next level
Mguard	Phantom Guardians **a**ppearance and Choices to defeat
Vector Graphics	**Guard, MPort, Coin, Shield, Sword, Mask, Ring, KStone**
MazNew	Construct a New Maze from Selected algorithm
MazHall	Increases route through Maze by adding further inter-connections
MazLev	Displays Active level **of** maze
MazTres	Randon allocation of location and selected Treasure Stored**.**
MazKey	Randon selection of possible Sphere Ring Combination.
MSel	Select of Maze Algorithm **[1 2 3]**
GView	Maze 3D Forward view of Selected Direction
GPause	Pause Game and Suspend Clock
GNew	Start a New Game
SelPath	Select Device and Filename
FCheck	**Cries out search for selected Filename**
MLoad	Load Previously Stored Maze Settings
MSave	Save Maze Settings
QExit	Leave Program
KeyStone	
MatchKey	Check Keystone Settings Align with Selected arrangement**.**
GetKay	Displays Sphere of Destiny - Keystone Order (!F5 Cheat!
SDest, SRing,, SEnd	Sphere of Destany, KeyStones, Then Time Tunnel Graphics
LScore	League Table Scores
LName	New Player Entry
LSave	Save LScore
LLoad	Load LScore
LDefault	Set League Table with Default valuses
LResset	Reset League Table

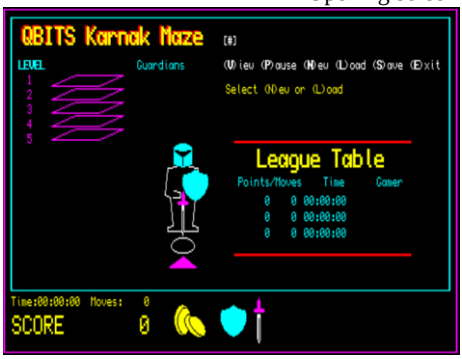
QBITS Karnak Maze Coding

```
1000 REMark QBITS_KarnakMaze_bas [QBITS Maze Game 2023 Review - QPC2]

1002 dev$='dos7_':MODE 4:gx=0:gy=0   :REMark Basic Settings

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

1006 REMark Import QBITSConfig Settings - QPC2
1007 OPEN_IN#9,dev$&'QBITSConfig':INPUT#9,gx\gy\dn$\dev$\dn%\dm%
1008 DIM drv$(15,5):FOR d=0 TO 15:INPUT#9,drv$(d):END FOR d:CLOSE#9

1011 REMark Array SetUp
1012 SD$='MazeData_':LLoad
1013 DIM dir$(4,5):RESTORE 1014:FOR c=1 TO 4:READ dir$(c)
1014 DATA 'West ','East ','North', 'South'
1015 DIM SKey(5,3):RESTORE 1016:FOR i=1 TO 5:READ SKey(i,1):READ SKey(i,2)
1016 DATA -34,32, -40,3, 40,3, 34,32, 0,50
1017 DIM grid(21,17),cell(20*16):w=20:h=16
1018 DIM Mkey(5),Tres(12,3),name$(3,10),Grad(3,3)

1020 REMark Variable Settings
1021 w=20:h=16:x=0:y=0:cx=0:cy=0:px=0:py=0   :REMark Various Coordinates
1022 lev=1:glev=2:gmax=16:gdel=120           :REMark Maze Level/Guardians
1023 Gclk=DATE:GTS=0:sm=0:snum=0:sl=1        :REMark Score/Time/Moves/Points
1024 bc=0:sc=0:tc=0:scol=0:col=0             :REMark Various Colours
1025 gst=0:fd=3                             :REMark Maze Moves/Direction
1026 eck=0:pck=0:mck=0:TM=0                 :REMark Checks:ERR/FilePath/MazeGen/Test Mode

1028 f=0:m=1:Init_Win:Init_Maze:LScore:Mes1:QBITS_Maze
```



Intro Screen

Opening screen



```
1030 DEFine PROCedure QBITS_Maze
1031 REPeat Maze_lp
1032  IF gst=1:Score:ELSE Gclk=DATE:Score
1033  IF gdel=0:MGuard
1034  k=CODE(INKEY$(20))
1035  SELect ON k
1036    = 61:IF TM=0:TM=1:ELSE TM=0          :REMark [#]  Test Mode
1037    =232:IF TM=1:km=1:kr=1:MGuard        :REMark [F1] Guardians
1038    =236:IF TM=1:PortChk:GView           :REMark [F2] Open Portal
1039    =240:IF TM=1:snum=snum+50:Score      :REMark [F3] Score +
1040    =244:IF TM=1:snum=snum -50:Score     :REMark [F4] Score -
1041    =248:IF TM=1:tc=0:GView:tc=3:GView   :REMark [F5] Key Stone
1042    = 86,118:GView                       :REMark [V]iew ON/OFF
1043    = 80,112:GPause:Gclk=DATE            :REMark [P]ause
1044    = 78,110:GNew  :mck=1:GPause         :REMark [N]ew (Maze Check)
1045    = 76,108:MLoad:mck=1:GPause          :REMark [L]oad (Maze check)
1046    = 83,115:IF mck=1:MSave              :REMark [S]ave
1047    = 69,101:QExit                       :REMark [E]xit
1048    =192:IF gst=1:fd=1:MazView           :REMark Left  West
1049    =200:IF gst=1:fd=2:MazView           :REMark Right East
1050    =208:IF gst=1:fd=3:MazView           :REMark Up    North
1051    =216:IF gst=1:fd=4:MazView           :REMark Down  South
1052    ON k=32                              :REMark SpaceBar Forward
1053      IF snum<5:Mes 2:GO TO 1069
1054      IF snum<  50+  50*lev:BLOCK#4,30,30,300,216,0:km=0
1055      IF snum<500+500*lev:BLOCK#4,50,30,332,216,0:kr=0
1056      IF fvn=1
1057        INK#2,5:CURSOR#2,236,190:PRINT#2,'Solid Wall!':CLS#2,4
1058        BEEP 1000,1,140,190,0,0,0:PAUSE 20
1059      ELSE
1060        IF fd=1 : px=px-1                :REMark One Cell West
1061        IF fd=2 : px=px+1                :REMark One Cell East
1062        IF fd=3 : py=py-1                :REMark One cell North
1063        IF fd=4 : py=py+1                :REMark One cell South
1064        BLOCK#3,4,3,2+cx*6,1+cy*5,0  :cx=px:cy=py
1065        BLOCK#3,4,3,2+cx*6,1+cy*5,bc     :REMark 2D Maze cell position
1066        BEEP 2000,20,40,190,0,0,0:gst=1
1067        ofd=fd:gdel=gdel-1:snum=snum-sl:sm=sm+1:MazView
1068        Loot=grid(px,py):SELect ON Loot=1,2,4,8:TresChk
1069      END IF
1070  END SELect
1071  END REPeat Maze_lp
1072 END DEFine
```

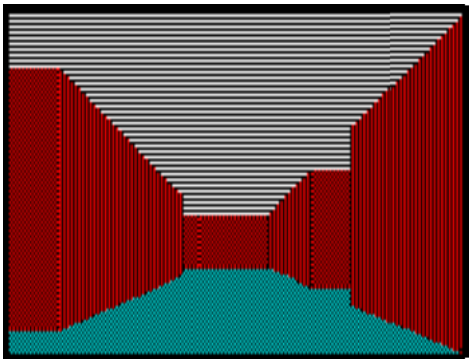

```
1074 DEFine PROCedure Score
1075 INK#4,6:clk$=DATE$(DATE-Gclk+GTS):CURSOR#4,6,212
1076 PRINT#4,'Time:';clk$(13 TO 20);'  Moves: ';FILL$(' ',4-LEN(sm))&sm
1077 PRINT#5,'SCORE  ';FILL$(' ',6-LEN(snum))&snum
1078 END DEFine
```

Note: Spaces fill the blanks right to left for Move and Score until they increase to fulfil their digital span.

```
1080 DEFine PROCedure MazView
1081 fvn=0:INK#2,6:CURSOR#2,178,190:PRINT#2,dir$(fd):CLS#2,4
1082 FOR n=1 TO 5
1083   fv(n)=0                                        :REMark fv forward view
1084   IF fd=1
1085     cw=grid(px-n+1,py):IF Walls(8,cw)=1:fv(n)=1
1086     IF Walls(4,cw)=1:fv(n)=fv(n)+2               :REMark cw cell walls
1087     IF Walls(1,cw)=0:fvn=n:EXIT n                :REMark fvn forward view num cell
1088   END IF
1089   IF fd=2
1090     cw=grid(px+n-1,py):IF Walls(8,cw)=1:fv(n)=2
1091     IF Walls(4,cw)=1:fv(n)=fv(n)+1
1092     IF Walls(2,cw)=0:fvn=n:EXIT n
1093   END IF
1094   IF fd=3
1095     cw=grid(px,py-n+1):IF Walls(1,cw)=1:fv(n)=1
1096     IF Walls(2,cw)=1:fv(n)=fv(n)+2
1097     IF Walls(4,cw)=0:fvn=n:EXIT n
1098   END IF
1099   IF fd=4
1100     cw=grid(px,py+n-1):IF Walls(1,cw)=1:fv(n)=2
1101     IF Walls(2,cw)=1:fv(n)=fv(n)+1
1102     IF Walls(8,cw)=0:fvn=n:EXIT n
1103   END IF
1104 END FOR n
1105 vn=fvn:IF fvn=0 :fvn=6:vn=5                      :REMark fvn vn forward view num cells
1106 xw=58*(2/3)^((vn-1)*2):ytw=14*xw/15:ybw=-2*xw/5
1107 BLOCK 240,82,0,0,7,0,1:BLOCK 240,38,0,82,0,5,3  :REMark Roof & Floor
1108 IF fvn=6
1109   INK 0,2,1
1110   FILL 1:LINE -xw,ytw TO xw,ytw TO xw,ybw TO -xw,ybw TO -xw,ytw:FILL 0
1111   GO TO 1115
1112 END IF
1113 INK 0,2,3
1114 FILL 1:LINE -xw,ytw TO xw,ytw TO xw,ybw TO -xw,ybw TO -xw,ytw:FILL 0
```

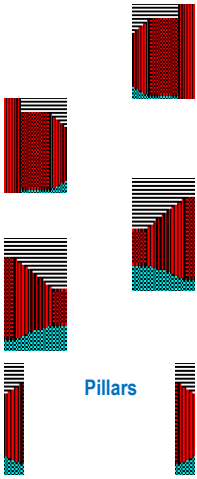
Code Continues on next page…

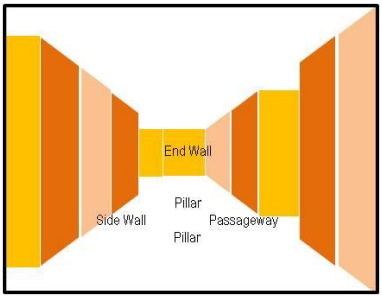
Note: As computer games developed from the early nineteen-eights it was the graphical displays that most impressed and intrigued. Vectors graphics use lines straight or curved, drawn between coordinated points, this makes them easily scalable. QL Super/SBASIC uses the Graphics coordinate system as opposed to the Pixel coordinates used with bitmaps.

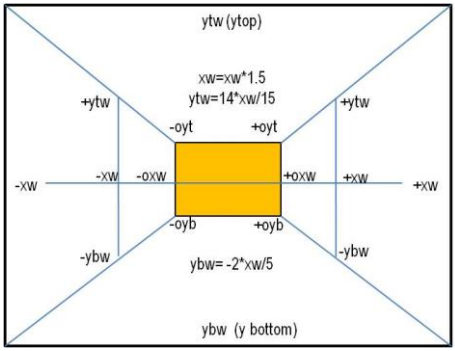


Note: Vector Graphics are used to Display the 3D View of Passageways.

```
1115 REPeat sidewalls
1116  Wallcalc:cdv=fv(vn):INK 0,2,3
1117  IF cdv=2 OR cdv=3
1118     FILL 1:LINE oxw,oyt TO oxw,oyb TO xw,oyb TO xw,oyt TO oxw,oyt:FILL 0
1119  END IF
1120  IF cdv=1 OR cdv=3
1121     FILL 1:LINE -oxw,oyt TO -oxw,oyb TO -xw,oyb TO -xw,oyt TO -oxw,oyt:FILL 0
1122  END IF
1123  INK 0,2,2
1124  IF cdv=0 OR cdv=1
1125     FILL 1:LINE oxw,oyt TO oxw,oyb TO xw,ybw TO xw,ytw TO oxw,oyt:FILL 0
1126  END IF
1127  IF cdv=0 OR cdv=2
1128     FILL 1:LINE -oxw,oyt TO -oxw,oyb TO -xw,ybw TO -xw,ytw TO -oxw,oyt:FILL 0
1129  END IF
1130  Wallcalc
1131  FILL 1:LINE oxw,oyt TO oxw,oyb TO xw,ybw TO xw,ytw TO oxw,oyt:FILL 0
1132  FILL 1:LINE -oxw,oyt TO -oxw,oyb TO -xw,ybw TO -xw,ytw TO -oxw,oyt:FILL 0
1133  vn=vn-1:IF vn=0 : EXIT sidewalls
1134 END REPeat sidewalls
1135 END DEFine
```

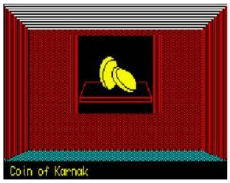




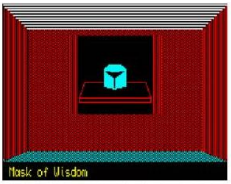


Note: The Maze Passageways consist of an End Wall, Gaps with Horizontal Walls for Side Entrances, or Blocked by a Perspective Side Wall in between Pillars.

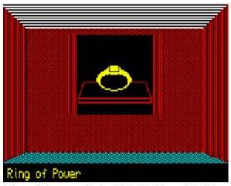
```
1137 DEFine FuNction Walls(side,wall)
1138 ans=0
1139 IF side=1:SELect ON wall=1,3,5,7,9,11,13,15  :ans=1
1140 IF side=2:SELect ON wall=2,3,6,7,10,11,14,15 :ans=1
1141 IF side=4:SELect ON wall=4,5,6,7,12,13,14,15 :ans=1
1142 IF side=8 AND wall>7 :ans=1
1143 RETurn ans                                   :REMark identify fd (forward direction)
1144 END DEFine

1146 DEFine PROCedure Wallcalc
1147 oxw=xw:xw=xw*1.5:oyt=ytw:oyb=ybw:ytw=14*xw/15:ybw=-2*xw/5
1148 END DEFine
```



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



Mask of Wisdom - Gain 50+50xLevel Points

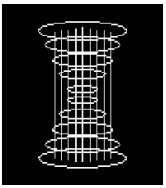


Ring of Power – Gain 500+500xLevel Points

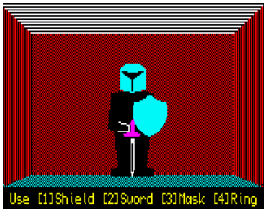
```
1150 DEFine PROCedure TresChk
1151 FOR i=1 TO 12
1152  IF Tres(i,1)=px AND Tres(i,2)=py
1153   tn=Tres(i,3):IF tn=0:EXIT i
1154   BLOCK 100,60,70,24,0,2,2:FOR j=1 TO 8:BLOCK j*10,60,120-j*5,24,0:PAUSE 5
1155   ch=1:INK 2:x=0:y=20:INK#2,6:CURSOR#2,236,190
1156   LINE x-24,y-10 TO x-20,y TO x+20,y TO x+24,y-10 TO x-24,y-10
1157   LINE x-24,y-10 TO x-24,y-12 TO x+24,y-12 TO x+24,y-10
1158   IF tn>1 AND tn<7
1159     Coin 1,-4,26:PRINT#2,'Coin of Karnak':snum=snum+50*tn
1160   END IF
1161   IF tn=7
1162     Mask 4,220,20:Mask 1,0,20:km=1
1163     PRINT#2,'Mask of Wisdom':snum=snum+50+50*lev
1164   END IF
1165   IF tn=8
1166     Ring 4,250,22:Ring 1,0,22:kr=1
1167     PRINT#2,'Ring of Power ':snum=snum+500+500*lev
1168   END IF
1169   IF tn=9:KStone 1,0,20:PortChk:EXIT i
1170   Tres(i,3)=0
1171  END IF
1172 END FOR i
1173 END DEFine
```

Note: These **PROC**edure's check for Actions to be taken.

```
1175 DEFine PROCedure PortChk
1176 CURSOR#2,236,190:PRINT#2,'Key Stone - Portal to Next Level (Y/N)'
1177 IF INKEY$(#2,-1)=='Y'
1178   IF lev=5
1179     IF gmax>0:CURSOR#2,236,190:PRINT#2,'Defeat All Guardians':CLS#2,4:RETurn
1180     CURSOR#2,236,190:CLS#2,4:CLS:KeyStone
1181   ELSE
1182     lev=lev+1:glev=lev+1:IF glev>gmax OR lev=5:glev=gmax
1183     CLS:MPort:PAUSE 20:col=5:MazLev:MazNew:MazHall:MazTres:MazView
1184     snum=snum+2000:tc=0:bc=0:km=0:kr=0:gdel=120/RND(2 TO 4)
1185     px=RND(2 TO 19):py=RND(2 TO 15):CURSOR#2,220,190:CLS#2,4
1186   END IF
1187 ELSE
1188   tc=1:BLOCK#2,240,10,230,190,0
1189 END IF
1190 END DEFine
```



```
1192 DEFine PROCedure MGuard
1193 IF glev=0:RETurn :ELSE Guard 1:INK 7:MPort
1194 PAUSE 20:MazView:Guard 1:INK#2,6:gdel=120/RND(2 TO 4)
1195 REPeat G_lp
1196  IF snum<  50:EXIT G_lp
1197  CURSOR#2,236,190:PRINT#2,'Use [1]Shield [2]Sword':CLS#2,4
1198  IF km=1 AND snum>50+50*lev:CURSOR#2,374,190:PRINT#2,'[3]Mask'
1199  IF kr=1 AND snum>500+500*lev:CURSOR#2,422,190:PRINT#2,'[4]Ring'
1200  k=CODE(INKEY$(-1))
1201  IF k=49:snum=snum-50:px=RND(3 TO 17):py=RND(3 TO 14):CLS:EXIT G_lp
1202  IF k=50
1203     INK#4,5:FOR i=1 TO 6 :CURSOR#4,197,170:PRINT#4,i:PAUSE 20
1204     a=RND(1 TO 6):INK#4,7:CURSOR#4,197,170:PRINT#4,a:PAUSE 20
1205     IF a=6 :INK#2,4:gmax=gmax-1:glev=glev-1:EXIT G_lp
1206     IF a<>6:INK#2,2:snum=snum-50:Score
1207  INK#2,6:CURSOR#2,236,190:PRINT#2,'Try Again':CLS#2,4:PAUSE 30
1208  END IF
1209  IF k=51 AND snum>50+50*lev:snum=snum-50-50*lev:gdel=120:EXIT G_lp
1210  IF k=52 AND snum>500+500*lev
1211     snum=snum-500-500*lev:gmax=gmax-glev:glev=0:EXIT G_lp
1212  END IF
1213 END REPeat G_lp
1214 GView:Score:CURSOR#2,236,190:CLS#2,4:INK 7:MPort:PAUSE 20:MazView
1215 INK#2,5:CURSOR#2,148,30+10*lev:PRINT#2,glev;'  ';gmax;'  '
1216 END DEFine
```

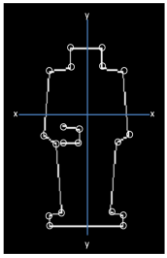


Note: The **MGuard PROC**edure activates after a set number of moves; [120 / RND (2 to 4) {i.e. 30 40 60 moves}]

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

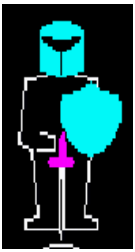
```
1218 REMark Vector Graphics
```

```
1220 DEFine PROCedure Guard(ch)
1221 IF ch=1:x=0:y=- 4  :INK#ch,0:FILL#ch,1
1222 IF ch=4:x=140:y=104:INK#ch,7:FILL#ch,0
1223 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
1224 LINE#ch 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
1225 LINE#ch 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
1226 LINE#ch TO x+5,y+32 TO x-5,y+32:FILL#ch,0
1227 INK#ch,7:LINE#ch,x-8,y+4 TO x-2,y+4 TO x-2,y+8 TO x-8,y+9
1228 IF ch=1:Shield 1, 10, 20:Sword 1, 0,-24:Mask 1,  0, 26
1229 IF ch=4:Shield 4,150,128:Sword 4,140,82:Mask 4,140,134
1230 END DEFine
```



Note: The Phantom Guardians

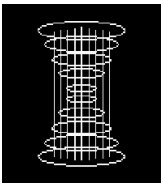
Vector Graphics is like joining the dots. After setting the scale and location of the x, y zero coordinates; work out the offsets to each position that describes the object. The Maze Guardian was based on an image of a Knight taken from an old Church brass rubbing.



The body with head Masked by a helmet, then a Shield over the left arm and a Sword held with the blade tip down at the feet.

First the outline of the body is drawn with the sword arm. Then Images created for the Helmet, Shield and Sword are then overlayed to create the Maze Phantom Knight.

```
1232 DEFine PROCedure MPort
1233 BEEP 2000,20,40,190,0,0,0:ch=1:x=0:y=20:INK#ch,7
1234 FOR i=0 TO 16 STEP 4
1235   CIRCLE#ch,0,-20+i*2,25-i,.2,PI/2:CIRCLE#ch,0,50-i*2,25-i,.2,PI/2
1236   LINE#ch,-i,-22+i/8 TO -i,52-i/8:LINE#ch,+i,-22+i/8 TO i,52-i/8
1237 END FOR i
1238 END DEFine
```



```
1240 DEFine PROCedure Coin(ch,x,y)
1241 INK#ch,6:FILL#ch,1:CIRCLE#ch,x,y,10,.6,PI:FILL#ch,0
1242 INK#ch,0:CIRCLE#ch,x+3,y-1,10,.7,PI
1243 INK#ch,6:FILL#ch,1:CIRCLE#ch,x+10,y-4,10,.6,PI/4:FILL#ch,0
1244 INK#ch,0:CIRCLE#ch,x+10,y-4,10,.6,PI/4
1245 INK#ch,0:CIRCLE#ch,x+12,y-4,9,.5,PI/4
1246 END DEFine
```


```
1248 DEFine PROCedure Shield(ch,x,y)
1249 FILL#ch,1:INK#ch,5:ARC#ch,x,y TO x-9,y-4, -PI/4
1250 ARC#ch,x-9,y-4 TO x,y-22, PI/2:ARC#ch,x,y-22 TO x+9,y-4, PI/2
1251 ARC#ch,x+9,y-4 TO x,y, -PI/4:FILL#ch,0
1252 END DEFine
```


```
1254 DEFine PROCedure Sword(ch,x,y)
1255 FILL#ch,1:INK#ch,7
1256 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
1257 FILL#ch,0:INK#ch,0:LINE#ch,x,y+2 TO x,y+18:INK#ch,3
1258 FILL#ch,1:CIRCLE#ch,x,y+22,5,.2,PI/2:FILL#ch,0
1259 FILL#ch,1:CIRCLE#ch,x,y+26,5,.2,PI:FILL#ch,0:CIRCLE#ch,x,y+28,1
1260 END DEFine
```

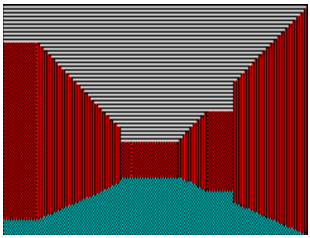

```
1262 DEFine PROCedure Mask(ch,x,y)
1263 INK#ch,5:FILL#ch,1:ARC#ch,x+7,y+9 TO x-7,y+9,PI/2
1264 LINE#ch,x-7,y+9 TO x-7,y-2 TO x,y-4 TO x+7,y-2 TO x+7,y+9:FILL#ch,0
1265 INK#ch,0:FILL#ch,1:LINE#ch,x+6,y+7 TO x,y+6 TO x,y+3 TO x+6,y+7:FILL#ch,0
1266 FILL#ch,1:LINE#ch,x-6,y+7 TO x,y+6 TO x,y+3 TO x-6,y+7:FILL#ch,0
1267 LINE#ch,x,y+4 TO x,y-4
1268 END DEFine
```

Note: When an object is filled with a solid colour QL Super/SBASIC FILLs between min and max line coordinates. The visor opening for the helmet therefore has to be drawn as to two objects.

```
1270 DEFine PROCedure Ring(ch,x,y)
1271 INK#ch,6:FILL#ch,1:CIRCLE#ch,x,y,11,.6,PI/2 :FILL#ch,0
1272 INK#ch,0:FILL#ch,1:CIRCLE#ch,x,y-1,9,.5,PI/2:FILL#ch,0
1273 INK#ch,6:FILL#ch,1:CIRCLE#ch,x,y+6,5,.5,PI/2:FILL#ch,0
1274 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
1275 END DEFine
```

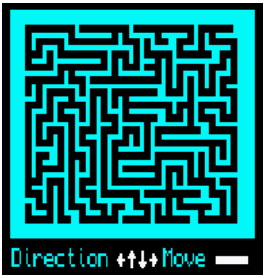

```
1277 DEFine PROCedure KStone(ch,x,y)
1278 BEEP 2000,20,40,190,0,0,0:INK#ch,6:FILL#ch,1
1279 LINE#ch,x,y+6 TO x-6,y TO x,y-6 TO x+6,y TO x,y+6:FILL#ch,0
1280 INK#ch,0:LINE#ch,x,y+8 TO x-6,y TO x,y-8 TO x+6,y TO x,y+8
1281 LINE#ch,x,y+8 TO x-2,y-2 TO x,y-8
1282 LINE#ch,x-6,y TO x-2,y-2 TO x+6,y
1283 END DEFine:
```

1285 REMark **New Game / Level Change**



```
1287 DEFine PROCedure MazNew
1288 w=20:h=16:DIM grid(w+1,h+1),cell(w*h,2),pm(5),fv(5)
1289 CLS#3:INK 7:x=w/2:y=h:cell(0,1)=x:cell(0,2)=y:inc=40:cn=1
1290 INK#2,5:CURSOR#2,18,190:PRINT#2,'Direction      Move':INK#2,7
1291 CURSOR#2,76,190:PRINT#2,'←↑↓→':BLOCK#2,18,3,136,194,7
1292 FOR n=1 TO w*h-1
1293   p=0:PAUSE mp                          :REMark p - Pass / mp - 0.5 Pause Delay
1294   IF x>1 AND grid(x-1,y)=0 : p=p+1:pm(p)=1   :REMark West wall
1295   IF x<w AND grid(x+1,y)=0 : p=p+1:pm(p)=2   :REMark East wall
1296   IF y>1 AND grid(x,y-1)=0 : p=p+1:pm(p)=3   :REMark North wall
1297   IF y<h AND grid(x,y+1)=0 : p=p+1:pm(p)=4   :REMark South wall
1298   IF p=0
1299     IF m=1:cn=cn-1:x=cell(cn,1):y=cell(cn,2)
1300     IF m=2:cn=0:x=RND(w):y=RND(h):cell(cn,1)=x:cell(cn,2)=y
1301     IF m=3:x=x+1:IF x>w : x=1 :y=y+1:IF y>h : y=1
1302     IF grid(x,y)=0 : GO TO 1299
1303     GO TO 1294
1304   END IF
1305   r=pm(RND(1 TO p)):cn=cn+1:cell(cn,1)=x:cell(cn,2)=y
1306   IF r=1 : grid(x,y)=grid(x,y)+1:x=x-1:grid(x,y)=2:bx=x*6:by=y*5
1307   IF r=2 : grid(x,y)=grid(x,y)+2:bx=x*6:x=x+1:grid(x,y)=1:by=y*5
1308   IF r=3 : grid(x,y)=grid(x,y)+4:y=y-1:grid(x,y)=8:bx=x*6:by=y*5
1309   IF r=4 : grid(x,y)=grid(x,y)+8:by=y*5:y=y+1:grid(x,y)=4:bx=x*6
1310   IF r=1 OR r=2 :BLOCK#3,8,3,2+bx,1+by,0
1311   IF r=3 OR r=4 :BLOCK#3,4,8,2+bx,1+by,0
1312 END FOR n
1313 END DEFine
```

Note: The Maze algorithms
m=1 Recursive backtracking
m=2 Prims Algorithm
m=3 Hunt and Kill Method



Note: For faster QL Hardwar/Software Platforms... mp set to 0.5 so Maze construction is slowed.


```
1315 DEFine PROCedure MazHall
1316 REMark grid r row/c col:hw hall width:cw cell wall:sf side facing
1317 FOR hall=1 TO 6
1318   tx=RND(4 TO 16):ty=RND(4 TO 12):RESTORE 1330
1319   FOR r=0 TO 1
1320     FOR c=0 TO 2
1321       BLOCK#3,16,8,2+tx*6,1+ty*5,0
1322       BLOCK#3,2,2,6+tx*6,4+ty*5,5 :BLOCK#3,2,2,12+tx*6,4+ty*5,5
1323       FOR hw=1 TO 3
1324         cw=grid(tx+c,ty+r):READ sf
1325         IF Walls(sf,cw)=0:grid(tx+c,ty+r)=grid(tx+c,ty+r)+sf
1326       END FOR hw
1327     END FOR c
1328 END FOR r
1329 END FOR hall
1330 DATA 2,8,8,1,2,8,1,8,8,2,4,4,1,2,4,1,4,4
1331 END DEFine
```

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


```
1333 DEFine PROCedure MazLev
1334 BLOCK#2,50,58,132,40,0:BLOCK#4,200,40,300,212,0
1335 FOR i=1 TO 5
1336   IF i=lev:INK#2,5:ELSE INK#2,3
1337   CURSOR#2,16,30+i*11:PRINT#2,i
1338   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
1339   IF i<lev:KStone 4,280+i*12,20
1340 END FOR i
1341 IF lev=1:px=10:py=16:ELSE px=RND(3 TO 18):py=RND(2 TO 14)
1342 INK#2,col:CURSOR#2,148,30+10*lev:PRINT#2,glev;' ';gmax
1343 END DEFine

1345 DEFine PROCedure MazTres
1346 DIM Tres(12,3):n=1
1347 REPeat t_lp
1348   IF n>12:n=1:EXIT t_lp
1349   tx=RND(1 TO w):ty=RND(1 TO h):tn=grid(tx,ty)
1350   FOR i=1 TO n:IF Tres(i,1)=tx AND Tres(i,2)=ty:NEXT t_lp
1351   SELect ON tn=1,2,4,8:Tres(n,1)=tx:Tres(n,2)=ty:n=n+1
1352 END REPeat t_lp
1353 FOR i=1 TO 12:Tres(i,3)=RND(2 TO 6)
1354 Tres(3,3)=7:km=0:Tres(11,3)=8:kr=0:Tres(7,3)=9
1355 END DEFine
```

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

```
1357 DEFine PROCedure MazKey
1358 DIM Mkey(5):RESTORE 1362:ra=RND(24)
1359 FOR i=1 TO 24
1360   READ a,b,c,d:IF i=ra:SKey(1,3)=a:SKey(2,3)=b:SKey(3,3)=c:SKey(4,3)=d
1361 END FOR i
1362 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
1363 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
1364 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
1365 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
1366 END DEFine
```

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 create 24 different combinations.

```
1368 REMark Menu Commands

1370 DEFine PROCedure Mes1
1371 BLOCK#2,200,26,240,40,0
1372 INK#2,6:CURSOR#2,240,48:PRINT#2,'Select (N)ew or (L)oad':gck=0
1373 END DEFine
```

```
1375 DEFine PROCedure MSel
1376 IF m=1:Maz$='Recursive Backtraking'
1377 IF m=2:Maz$='Prims Algorithm'
1378 IF m=3:Maz$='Hunt and Kill Method'
1379 INK#2,3:CURSOR#2,332,12:PRINT#2,FILL$(' ',25-LEN(Maz$))&Maz$
1380 END DEFine
```


```
1382 DEFine PROCedure GView
1383 IF gck=0:RETurn
1384 IF tc=0:tc=3:bc=7:sl=lev*5:ELSE tc=0:bc=0:sl=lev
1385 IF TM=1:BLOCK#3,4,3,2+Tres(7,1)*6,1+Tres(7,2)*5,241:PAUSE 30
1386 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
1387 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
1388 END DEFine
```

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


```
1390 DEFine PROCedure GPause
1391 IF gck=0:RETurn
1392 INK#2,6:CURSOR#2,240,48:PRINT#2,'Press any key to continue...'
1393 GTS=(DATE-Gclk+GTS):PAUSE:BLOCK#2,250,10,240,48,0
1394 END DEFine
```

```
1396 DEFine PROCedure GNew
1397 GTS=(DATE-Gclk+GTS):INK#2,6:CURSOR#2,240,48
1398 PRINT#2,'Select Maze Algorythm [1][2][3] ←(Esc)'
1399 BLOCK#2,12,3,460,52,6:BLOCK#2,2,4,480,50,6:INK#2,3:om=m
1400 REPeat New_lp
1401  MSel:k=CODE(INKEY$(-1))
1402  SELect ON k
1403    =49,50,51:m=k-48
1404    =32:BLOCK#2,250,10,240,48,0:m=om:MSel:RETurn
1405    =10:BLOCK#2,250,10,240,48,0:CLS:EXIT New_lp
1406  END SELect
1407 END REPeat New_lp
1408 gdel=120/RND(3 TO 4):gmax=16:glev=2:lev=1:col=5:MazLev:MazKey
1409 GTS=0:Gclk=DATE:sm=0:snum=2000:Score:gck=1:gst=1
1410 w=20:h=16:mp=.5:MazNew:MazHall:MazTres:MazView:mp=0
1411 END DEFine
```



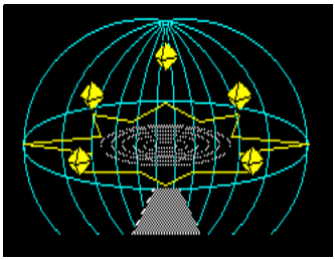
```
1413 DEFine PROCedure SelPath
1414 REMark TS=(DATE-Gclk+GTS):INK#2,6
1415 INK#2,6:CURSOR#2,240,48:PRINT#2,'Select: ↑↓ ';SD$;' ←→'←
1416 BLOCK#2,12,3,412,52,6:BLOCK#2,2,4,432,50,6
1417 REPeat Path_lp
1418  CURSOR#2,300,48:PRINT#2,drv$(dn%):CURSOR#2,384,48:PRINT#2,fnum
1419  k=CODE(INKEY$(-1))
1420  SELect ON k
1421    =192:fnum=fnum -1:IF fnum<0:fnum=9
1422    =200:fnum=fnum+1:IF fnum>9:fnum=0
1423    =208:dn%=dn%   -1:IF dn%<0:dn%=dm%
1424    =216:dn%=dn%  +1:IF dn%>dm%:dn%=0
1425    = 10:pck=1:EXIT Path_lp
1426    = 32:pck=0:EXIT Path_lp
1427  END SELect
1428 END REPeat Path_lp
1429 END DEFine
```

Enter symbol

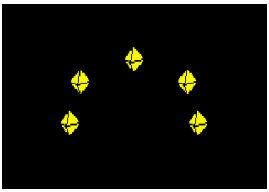
```
1431 DEFine PROCedure FCheck
1432 CURSOR#2,240,48:PRINT#2,'Searching...':CLS#2,4
1433 PAUSE 20:DELETE drv$(dn%)&'FList'
1434 OPEN_NEW#99,drv$(dn%)&'FList':DIR#99,drv$(dn%):CLOSE#99
1435 OPEN_IN#99,drv$(dn%)&'FList'
1436 REPeat Dir_lp
1437   IF EOF(#99):CLOSE#99:BLOCK#2,250,10,240,48,0:pck=0:EXIT Dir_lp
1438   INPUT#99,Fchk$:IF Fchk$==SD$&fnum:CLOSE#99:pck=1:EXIT Dir_lp
1439 END REPeat Dir_lp
1440 END DEFine

1442 DEFine PROCedure MLoad
1443 SelPath:IF pck=0:BLOCK#2,250,10,240,48,0:RETurn :ELSE FCheck
1444 IF pck=0 OR eck=1
1445   CURSOR#2,240,48:PRINT#2,'File Not Found...':CLS#2,4
1446   PAUSE 20:BLOCK#2,250,10,240,48,0:eck=0:RETurn
1447 END IF
1448 OPEN_IN#99,drv$(dn%)&SD$&fnum:CURSOR#2,240,48:PRINT#2,'Loading...';
1449 FOR n=1 TO 12:INPUT#99,Tres(n,3):PRINT#2,'.';:PAUSE 2
1450 INPUT#99,m\lev\gmax\glev\km\kr\GTS\sm\snum:CLOSE#99
1451 CLS:MazNew:MazHall:MazTres:MazKey
1452 MSel:col=5:MazLev:Score:MazView
1453 IF km=1:Mask 4,220,20
1454 IF kr  =1:Ring 4,250,22
1455 BLOCK#2,250,10,240,48,0:gdel=120/RND(2 TO 4):gst=1:gck=1
1456 END DEFine

1458 DEFine PROCedure MSave
1459 SelPath:IF pck=0 OR gck=0:BLOCK#2,250,10,240,48,0:RETurn :ELSE FCheck
1460 IF eck=1
1461   CURSOR#2,240,48:PRINT#2,'DEVICE ERROR...':CLS#2,4
1462   PAUSE 20:BLOCK#2,250,10,240,48,0:eck=0:RETurn
1463 END IF
1464 IF pck=1
1465   CURSOR#2,240,48:PRINT#2,'Overwrite y/n':PAUSE
1466   IF KEYROW(5)<>64:BLOCK#2,250,10,240,48,0:RETurn
1467 END IF
1468 DELETE drv$(dn%)&SD$&fnum:OPEN_NEW#99,drv$(dn%)&SD$&fnum
1469 CURSOR#2,240,48:PRINT#2,'Saving...';
1470 FOR n=1 TO 12:PRINT#99,Tres(n,3):PRINT#2,'.';:PAUSE 2
1471 PRINT#99,m\lev\gmax\glev\km\kr\GTS\sm\snum:CLOSE#99
1472 BLOCK#2,250,10,240,48,0
1473 END DEFine

1475 DEFine PROCedure QExit
1476 INK#2,6:CURSOR#2,240,48:PRINT#2,'Exit Game (Y/N)':CLS#2,4:PAUSE
1477 IF KEYROW(5)=64:LRUN dn$:STOP:ELSE CURSOR#2,200,48:CLS#2,4
1478 END DEFine
```

```
1480 REMark Sphere of Destiny - Matching Keystones

1482 DEFine PROCedure KeyStone
1483 KStone 4,340,20:scol=6:SDest:SRing
1484 CURSOR#2,258,44:PRINT#2,'Activate the Sphere of Destiny'
1485 CURSOR#2,240,54:PRINT#2,'by Matching the Sphere and Maze Keys'
1486 INK#2,5:CURSOR#2,250,190:PRINT#2,'Use       to Match and Test Keys'
1487 INK#2,7:CURSOR#2,274,190:PRINT#2,'←↑↓→ ←':BLOCK#2,2,4,310,192,7 ← Enter Symbol
1488 check=0:col=0:FOR kp=1 TO 4:ks=kp:GetKey
1489 REPeat key_lp
1490  IF snum<500:snum=0:Score:Mes2:EXIT key_lp
1491  GetKey:k=CODE(INKEY$(-1))
1492  SELect ON k
1493    =192:kp=kp -1:IF kp<1:kp=4
1494    =200:kp=kp+1:IF kp>4:kp=1
1495    =208:ks=ks +1:IF ks>4:ks=1
1496    =216:ks=ks -1:IF ks<1:ks=4
1497    = 10:MatchKey:IF schk<5:snum=snum-500:Score:ELSE Mes3:EXIT key_lp
1498    =248:IF col=0:col=5:ELSE col=0
1499  END SELect
1500 END REPeat key_lp
1501 gst=0:BLOCK#2,250,30,240,40,0:Mes1:QBITS_Maze
1502 END DEFine

1504 DEFine PROCedure MatchKey
1505 schk=1:FOR i=1 TO 4:IF SKey(i,3)=Mkey(i):schk=schk+1
1506 END DEFine

1508 DEFine PROCedure GetKey
1509 IF kp=1:Mkey(1)=ks:c=414
1510 IF kp=2:Mkey(2)=ks:c=430
1511 IF kp=3:Mkey(3)=ks:c=466
1512 IF kp=4:Mkey(4)=ks:c=484
1513 RESTORE 1516:INK#4,col:FOR i=1 TO 4:READ a:CURSOR#4,a,212:PRINT#4,SKey(i,3)
1514 RESTORE 1516:INK#4,5  :FOR i=1 TO 4:READ a:CURSOR#4,a,240:PRINT#4,Mkey(i)
1515 INK#4,6:CURSOR#4,c,240:PRINT#4,Mkey(kp)
1516 DATA 414,430,466,484
1517 END DEFine

1519 DEFine PROCedure Mes2
1520 IF gst=0:RETurn
1521 INK#2,6:CURSOR#2,240,190:PRINT#2,'Hard Luck You FAILED - Try a New Game '
1522 CLS:CLS#3:lev=1:glev=2:MazLev:SEnd:col=0:fil=1:Guard(1):Mes1
1523 END DEFine

1525 DEFine PROCedure Mes3
1526 INK#2,6:CURSOR#2,240,190:PRINT#2,'The Past has Changed - Humanity Saved '
1527 CLS:CLS#3:lev=1:glev=2:MazLev:SEnd:BLOCK#2,250,30,240,40,0:LName:Mes1
1528 END DEFine
:
```

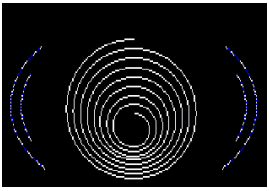
```
1530 DEFine PROCedure SDest
1531 col=5:ss=8:x=0:y=10:INK 7:FILL 0
1532 REPeat sphere_lp
1533  FOR i=0 TO 1.1  STEP .1
1534    ARC x,y+ss TO x,y-ss,PI*i:ARC x,y+ss TO x,y-ss,-PI*i
1535    INK col:IF col=5:col=0:ELSE col=5
1536  END FOR i
1537  BEEP 2000,8,20,-8,0,0,0:ss=ss+8:IF ss>56:EXIT sphere_lp
1538  PAUSE 5:INK 0:FILL 1:CIRCLE x,y,36+ss,ss*2/100,PI:FILL 0
1539 END REPeat sphere_lp
1540 INK 5:CIRCLE x,y,66,.3,PI/2:y=20:INK 6::FILL 0
1541 LINE x,y+9 TO x-16,y TO x-34,y+4 TO x-30,y-6 TO x-66,y-10
1542 LINE TO x-34,y-14 TO x-40,y-22 TO x-12,y-22 TO x,y-28 TO x+12,y-22
1543 LINE 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
1544 LINE TO x+16,y TO x,y+9:FILL 0:INK 248
1545 FILL 1:LINE x-6,y-30 TO x-16,y-50 TO x+16,y-50 TO x+6,y-30
1546 LINE TO x-6,y-30:FILL 0
1547 FOR i=1 TO 6:CIRCLE x,y-10,i*5,.3,PI/2:PAUSE 2
1548 END DEFine
```



```
1550 DEFine PROCedure SRing
1551 ch=1:scol=6:FOR i=1 TO 5:KStone 1,SKey(i,1),SKey(i,2):PAUSE 5
1552 END DEFine
```



```
1554 DEFine PROCedure SEnd
1555 FOR i=1 TO 24 STEP 2
1556    INK 241:CIRCLE 0,14,i*3,.7,PI/2:BEEP 2000,40,120,90,0,0,0:PAUSE 2
1557 END FOR i
1558 INK 0:FILL 1:CIRCLE,0,14,60:FILL 0:BEEP 30000,1,250,90,-8,15,15:INK 7
1559 FOR i=50 TO 15 STEP -5
1560    ARC 0,i TO 0,-i/2,PI:ARC 0,-i/2 TO 0,i-5,PI:PAUSE i/5
1561 END FOR i
1562 BEEP 10000,4,200,190,0,0,0:PAUSE 20
1563 END DEFine
```



QBITS Maze – Game Checks

The hidden commands. These are the cheats that Gamers seek to uncover. Press **[#]** to toggle ON/OFF.

- **F1** Activates **Guardian** with **[1]Shield [2]Sword [3]Mask [4]Ring** action choices.
- **F2** Activates the **Portal** (**Y/N) to** jump to the next level.
- **F3** Adds 50 to **snum**
- **F4** Subtracts 50 from **snum**
- **F5** For the **Maze Levels** this Highlights the **KeyStone** Location shown in Green.

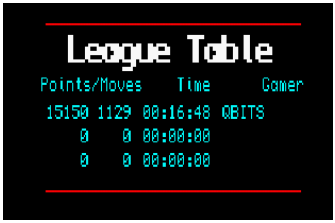
For the **Sphere of Destiny,** it identifies the order of the **Key Stones**

1565 REMark **QBITS Maze Screen Setup**

1567 **DEFine PROCedure Init_Win**
1568 OPEN#5,scr_:WINDOW#5,180,24,8+gx,226+gy :PAPER#5,0
1569 OPEN#4,scr_:WINDOW#4,512,256,gx,gy :PAPER#4,0
1570 OPEN#3,scr_:WINDOW#3,136,90,28+gx,106+gy:PAPER#3,5
1571 CSIZE#5,2,1:INK#5,6:BORDER#4,1,3:CLS#4:SCALE#4,240,0,0
1572 WINDOW#2,496,204,8+gx,6+gy :PAPER#2,0:BORDER#2,1,5
1573 WINDOW#1,240,120,240+gx,74+gy :PAPER#1,0:SCALE#1,100,-74,-30
1574 WINDOW#0,496,32,gx+8,gy+220 :PAPER#0,0:CSIZE#0,0,0:INK#0,7
1575 **END DEFine**

1577 **DEFine PROCedure Init_Maze**
1578 DIM S$(3,70),M$(7,40):CLS#2
1579 S$(1)='The Human Race is under Threat of Extinction from a Rogue DNA Gene'
1580 S$(2)='propagated by an Event in Prehistory. Beneath the Tombs of Karnak'
1581 S$(3)=' lies the Sphere of Destiny and a Time Portal to the Past.'
1582 M$(1)='Your Mission to go back in Time and'
1583 M$(2)=' prevent the Event from happening.'
1584 M$(3)=' To Activate the Time Portal there'
1585 M$(4)=' are five Key Stones each hidden'
1586 M$(5)=' on a different Level of the Tombs'
1587 M$(6)=' however they are protected by'
1588 M$(7)=' Guardians (Phantom Knights)'
1589 CSIZE#2,2,1:OVER#2,1
1590 INK#2,2:FOR i=1 TO 2:CURSOR#2,7+i,3:PRINT#2,'QBITS Karnak Maze'
1591 INK#2,6:FOR i=1 TO 2:CURSOR#2,9+i,4:PRINT#2,'QBITS Karnak Maze'
1592 INK#2,6:FOR i=1 TO 2:CURSOR#2,8+i,70:PRINT#2,'Sphere of Destiny'
1593 CSIZE#2,0,0:OVER#2,0:INK#2,7
1594 INK#2,7:FOR i=1 TO 3:CURSOR#2,44,24+i*10:PRINT#2,S$(i)
1595 INK#2,5:FOR i=1 TO 7:CURSOR#2,12,86+i*10:PRINT#2,M$(i)
1596 **Coin 4,140,22:Shield 4,180,32:Sword 4,200,6:Mask 4,220,22:Ring 4,250,22**
1597 **Score:**FOR i=1 TO 4:**KStone 4,280+i*12,20**
1598 **SDest:SRing**:INK#2,3
1599 CURSOR#2,24,180:PRINT#2,'Press any key to continue...':PAUSE
1600 CLS:BLOCK#2,490,40,2,28,0:BLOCK#2,220,136,2,60,0
1601 OVER#2,1:col=0:lev=0:**MazLev**:INK#2,7:Guard(4)
1602 CURSOR#2,236,12:PRINT#2,'[=]'
1603 CURSOR#2,236,30:PRINT#2,'(V)iew (P)ause (N)ew (L)oad (S)ave (E)xit '
1604 CURSOR#2,237,30:PRINT#2,' V P N L S E '
1605 INK#2,5:FOR i=0 TO 1:CURSOR#2,6+i,30:PRINT#2,'LEVEL'
1606 OVER#2,0:CURSOR#2,140,30:PRINT#2,'Guardians':gst=0:gck=0:gdel=120:pck=0
1607 INK#4,7:CIRCLE#4,140,74,9,.6,PI/2
1608 INK#2,3:FILL#2,1:LINE#2,65,8 TO 71,12 TO 76,8 TO 65,8:FILL#2,0
1609 **END DEFine**

```
1611 REMark League Table

1613 DEFine PROCedure LScore
1614 FOR i=1 TO 7
1615   PAUSE 5:BLOCK 200,12*i,20,66-i*6,0
1616   BLOCK 200,2,20,66-i*6,2:BLOCK 200,2,20,66+i*6,2
1617 END FOR i
1618 INK 6:OVER 1:CSIZE 2,1:FOR i=0 TO 1:CURSOR 44+i,28:PRINT 'League Table'
1619 INK 5:OVER 0:CSIZE 0,0:CURSOR 24,50:PRINT 'Points/Moves    Time      Gamer'
1620 FOR a=1 TO 3
1621  CURSOR  30,52+a*12:PRINT FILL$(' ',5-LEN(Grad(a,1)))&Grad(a,1)
1622  CURSOR  66,52+a*12:PRINT FILL$(' ',4-LEN(Grad(a,2)))&Grad(a,2)
1623  CURSOR 154,52+a*12:PRINT name$(a) :HST$=DATE$(Grad(a,3))
1624  CURSOR  98,52+a*12:PRINT HST$(13 TO 20)
1625 END FOR a
1626 END DEFine

1628 DEFine PROCedure LName
1629 GTS=DATE-Gclk+GTS:Gclk$=DATE$(GTS)
1630 FOR i=1 TO 3: IF Grad(i,1)<snum:Gmr=i:EXIT i:ELSE Gmr=0
1631 IF Gmr=0:LScore:RETurn
1632 IF Gmr>0
1633   Grad(Gmr,1)=snum:Grad(Gmr,2)=sm:Grad(Gmr,3)=GTS:LScore
1634   ch=6:OPEN#ch,con_:WINDOW#ch,60,10,394+gx,126+gy+Gmr*12
1635   PAPER#ch,0:CLS#ch:INK#ch,6:INPUT#ch,name$(Gmr):CLOSE#ch:LSave
1636 END IF
1637 END DEFine

1639 DEFine PROCedure LSave
1640 DELETE dev$&'QBMazeLT':OPEN_NEW#99,dev$&'QBMazeLT'
1641 FOR a=1 TO 3:PRINT#99,name$(a)\Grad(a,1)\Grad(a,2)\Grad(a,3)
1642 CLOSE#99
1643 END DEFine

1645 DEFine PROCedure LLoad
1646 OPEN_IN#99,dev$&'QBMazeLT'
1647 FOR a=1 TO 3:INPUT#99,name$(a)\Grad(a,1)\Grad(a,2)\Grad(a,3)
1648 CLOSE#99
1649 END DEFine

1651 DEFine PROCedure LTDefault
1652 REMark League Table Score
1653 name$(1)='QBITS    ':Grad(1,1)=1730:Grad(1,2)=756:Grad(1,3)=1072
1654 name$(2)='         ':Grad(2,1)=   0:Grad(2,2)=  0:Grad(2,3)=0
1655 name$(3)='         ':Grad(3,1)=   0:Grad(3,2)=  0:Grad(3,3)=0:LSave
1656 END DEFine

1658 DEFine PROCedure LTReset
1659 REMark League Table Reset
1660 name$(1)='         ':Grad(1,1)=0:Grad(1,2)=0:Grad(1,3)=0
1661 name$(2)='         ':Grad(2,1)=0:Grad(2,2)=0:Grad(2,3)=0
1662 name$(3)='         ':Grad(3,1)=0:Grad(3,2)=0:Grad(3,3)=0:LSave
1663 END DEFine
```



Note: Load **QBMaze_v04** [un-REMark **LSave]**, then call **LTDefault** or **LTReset** as required.