



QBITS Introduction

Maze algorithms in today's Gaming World are organized in 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 uses a predetermined arrangement of cells, commonly in a rectangular grid, but other designs are possible. A Maze Generation Algorithm is to fulfil the challenge of finding a route between any two 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.

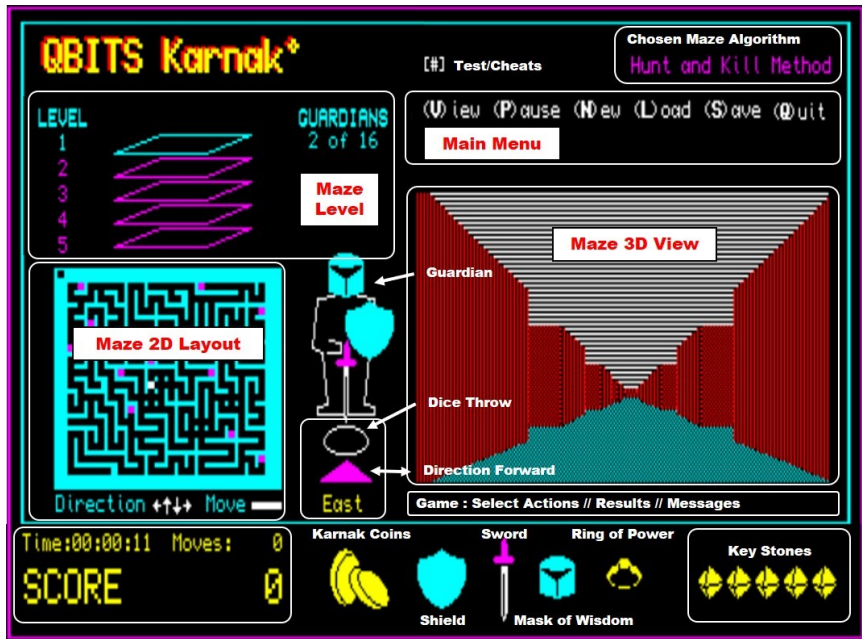
The **Recursive Backtracking** algorithm is a pathway created by randomly selecting an unvisited cell adjacent to one of the sides of the present one. The path is then made by knocking down the wall between. Moving into the new cell, the random selection continues until there are no surrounding unvisited 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** moves purposely through the grid selecting a cell and again 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 games are aligned in the correct order to open the Time Portal to the past.

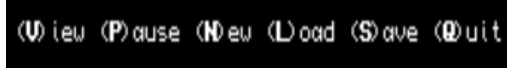
The game screen shows top left the Title '**QBITS Karnak Maze**'. Below this the **Maze Levels** which will show the active level in a different colour and the number of **Defending Guardians** to defeat. To the right of these the **Main Menu** with (V)iew (P)ause in addition to the usual (N)ew (L)oad (S)ave (Q)uit. At this point the game presents the option to Start a **New Game** or **Load** a previously saved one.



Below on the left the Maze is shown in 2D with Cursor keys for compass Direction and Spacebar to move. Centre left 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 passage ends as they may open up a hidden recess containing treasures Coins, Mask, Ring or a Keystone.

Bottom left shows the **Score Board**, **Game Time**, number of **Moves** and total **Points** collected. Displayed in the rest of the lower screen area **Coins** 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** paid for **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 '**DEVICE ERROR**' will be shown 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 - (Q)uit

This presents you with '**Quit 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. The program then returns to QBITS Menu if set to do so.

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 jumps 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

For 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 a successful match is made before losing all your wealth the **Time Portal** is opened and humanity saved from extinction.



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 or **Keystone** each adding to the Points scored.

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 **Matching** 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**. Finally, having entered the **Sphere of Destiny** depending on your luck or skill in choosing the right **Keystones** arrangement, will activate the **Time Portal** to go back and save **Humanity**.

Best of Luck!

QBITS Karnak Maze - SCORE

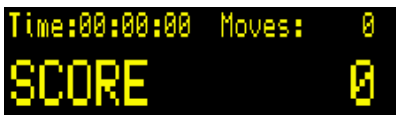
The **SCORE** shows the **Game Duration** in hours, minutes and seconds. A tally of the **Moves** taken and **Points** (wealth) accrued. The Timer uses QL Super/SBASIC Commands **DATE** to set the **Game Clock (Gclk)** at start and **DATES** to create an hh:mm:ss display.

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 changes are ascribed to variable **snum**.

Maze Moves Calculator

[View OFF](#)

[View ON](#)

On each move Score Points are Lost: **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 Selection (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 (if dice throw is a 6 Delete Guardian)	snum=snum-50 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 appearance 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 values
LResset	Reset League Table

QBITS Karnak Maze Coding

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

1002 dev\$='win1_':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

1010 REMark **Array SetUp**

1011 DIM brg\$(4,5):RESTORE 1011:FOR c=1 TO 4:READ brg\$(c)

1012 DATA 'West ','East ','North', 'South'

1013 DIM SKey(5,3):RESTORE 1013:FOR i=1 TO 5:READ SKey(i,1):READ SKey(i,2)

1014 DATA -34,32, -44,2, 44,2, 34,32, 0,48

1015 DIM grid(21,17),cell(20*16):w=20:h=16

1016 DIM Mkey(5),Tres(12,3),name\$(3,10),Grad(3,3)

1018 REMark **Variable Settings**

1019 w=20:h=16:x=0:y=0:cx=0:cy=0:px=0:py=0 :REMark Various Coordinates

1020 lev=1:glev=2:gmax=16:gdel=120 :REMark Maze Level/Guardians

1021 Gclk=DATE:GTS=0:sm=0:snum=0:sl=1 :REMark Score/Time/Moves/Points

1022 bc=0:sc=0:tc=0:scol=0:col=0 :REMark Various Colours

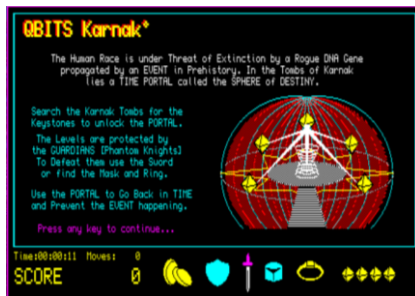
1023 gst=0:fd=3 :REMark Maze Moves/Direction

1024 eck=0:pck=0:mck=0:TM=0 :REMark Checks:ERR/FilePath/MazeGen/Test Mode

1025 SD\$='QBKMazeData_' :REMark Data file : Load Legue Table

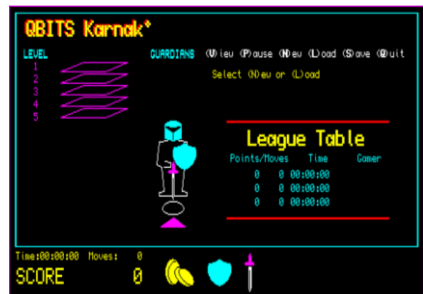
1026 f=0:m=1

1028 Init_Win:Init_Intro:Init_Menu:LTLoad:LTScore:Mes1:QBITS_Maze



Intro Screen

Opening screen



```

1030 DEFine PROCEDURE QBITS_Maze
1031 REPeat Maze_Ip
1032 IF gst=1:Score:ELSE Gclk=DATE:Score
1033 IF gdel=0:MGuard:END IF:IF mck=0:
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 :MazView :REMark [N]ew (Maze Check)
1045 = 76,108:MLoad::GPause:MazView :REMark [L]oad (Maze check)
1046 = 83,115:IF mck=1:MSave :REMark [S]ave
1047 = 69,101:QExit :REMark [Q]uit
1048 =192:IF gst=1:fd=1:MazView :REMark [L]eft West
1049 =200:IF gst=1:fd=2:MazView :REMark [R]ight East
1050 =208:IF gst=1:fd=3:MazView :REMark [U]p North
1051 =216:IF gst=1:fd=4:MazView :REMark [D]own 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:CUSOR#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_Ip
1072 END DEFine

```

```

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

```

```

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.


```

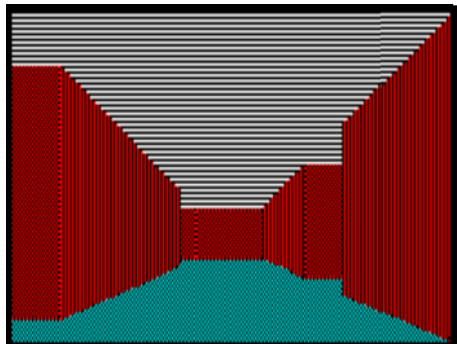
1080 DEFine PROCEDURE MazView
1081 fvn=0:INK#2,6:CURSOR#2,178,190:PRINT#2,dir$(fd):CLS#2,4
1082 FOR n=1 TO 5
1083   fv(n)=0 :REMark fv forward view
1084   IF fd=1
1085     cw=grid(px-n+1,py):IF Walls(8,cw)=1:fv(n)=1
1086     IF Walls(4,cw)=1:fv(n)=fv(n)+2 :REMark cw cell walls
1087     IF Walls(1,cw)=0:fvn=n:EXIT 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-eighties 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.

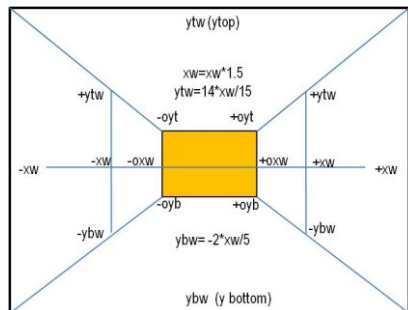
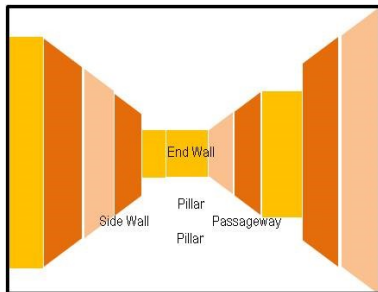
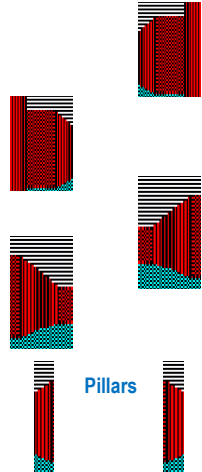


1115 REPEAT sidewalls

```

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
1144 END DEFine

```

:REMark identify fd (forward direction)

1146 DEFine PROCEDURE Wallcalc

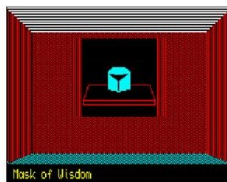
```

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

```



Coin 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#2,240,190
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

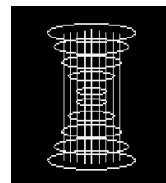
```

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

```

1175 DEFine PROCedure PortChk
1176 CURSOR#2,240,190: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#2,240,190:PRINT#2,'Defeat All Guardians':CLS#2,4:RETurn
1180     BLOCK#2,240,10,240,190,0::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#2,240,10,240,190,0
1189 END IF
1190 END DEFine

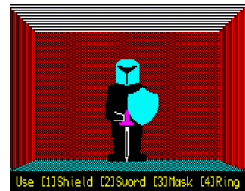
```



```

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#2,240,190:PRINT#2,'Use [1]Shield [2]Sword:CLS#2,4
1198 IF km=1 AND snum>50*50*lev:CURSOR#2,378,190:PRINT#2,'[3]Mask'
1199 IF kr=1 AND snum>500*500*lev:CURSOR#2,426,190:PRINT#2,'[4]Ring'
1200 k=CODE(INKEY$(-1))
1201 IF k=49:snun=snun-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:snun=snun-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:snun=snun-50*50*lev:gdel=120:EXIT G_lp
1210 IF k=52 AND snum>500*500*lev
1211   snun=snun-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: MGuard PROCEDURE activates after a set number of moves; [120 / RND (2 to 4) {i.e. 30 40 60 moves}]

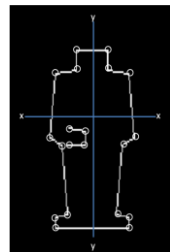
Mask - Banishes (glev) Level Guardians for 120 moves **Ring** - Reduces Guardians (glev=0 & gmax by glev)

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

```

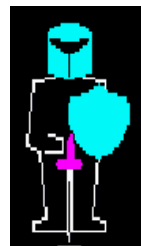


Note: The Phantom Guardians

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.

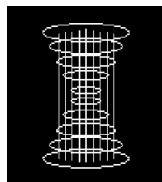
The body with head Masked by a helmet, then a Shield over the left arm and a Sword held with the blade tip down at the feet.

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 two objects.

1270 **DEFine PROCEDURE Ring(ch,x,y)**

```
1271 INK#ch,6:FILL#ch,1:CIRCLE#ch,x,y,7,.6,PI/2:FILL#ch,0
1272 INK#ch,0:FILL#ch,1:CIRCLE#ch,x,y-.8,6,.4,PI/2:FILL#ch,0
1273 INK#ch,6:FILL#ch,1:CIRCLE#ch,x,y+5,2,.5,PI/2:FILL#ch,0
1274 INK#ch,0:LINE#ch,x-2,y+5 TO x-2,y+2.8 TO x+2,y+2.8 TO x+2,y+5
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:
```



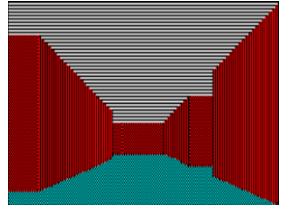
1285 REMark New Game / Level Change

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:CUSOR#2,18,190:PRINT#2,Direction Move:INK#2,7
1291 CUSOR#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
1294   IF x>1 AND grid(x-1,y)=0 : p=p+1:pm(p)=1
1295   IF x<w AND grid(x+1,y)=0 : p=p+1:pm(p)=2
1296   IF y>1 AND grid(x,y-1)=0 : p=p+1:pm(p)=3
1297   IF y<h AND grid(x,y+1)=0 : p=p+1:pm(p)=4
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

```



:REMark p - Pass / mp - 0.5 Pause Delay

:REMark West wall

:REMark East wall

:REMark North wall

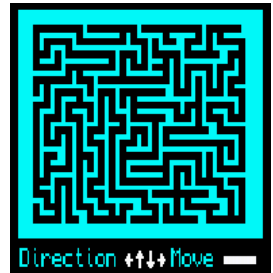
:REMark South wall

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 Kamak 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 MSeI

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

Note: (V) Toggles ON/OFF

Highlighted

1385 IF TM=1:BLOCK#3,4,3,2+Tres(7,1)*6,1+Tres(7,2)*5,241:PAUSE 30 Cells seen on the 2D Maze Layout

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



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

<Enter>

1398 PRINT#2,'Select Maze Algorhythm [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 MSeI: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:MSeI:RETURN

1405 =10:BLOCK#2,250,10,240,48,0:CLS:EXIT New_lp

1406 END SElect

1407 END REPEAT New_lp

1407 GTS=0:Gclk=DATE:sm=0:snum=2000:Score:gck=1:gst=1

1408 gdel=120/RND(3 TO 4):gmax=16:glev=2

1409 lev=1:col=5:MazLev:MazKey:w=20:h=16:mp=.5

1410 MazNew:MazHall:MazTres:mp=0:GPause:KOpen:GTS=0:Gclk=DATE:mck=1

1411 END DEFine

1413 REMark File Management

View Pause New Load Save Exit
Select: ↑ ↓ win2_MazeData_0 ↔ ← → Enter symbol

1415 DEFine PROCEDURE SelPath

```
1416 REMark TS=(DATE-Gclk+GTS):INK#2,6
1417 INK#2,6:CURSOR#2,240,48:PRINT#2,'Select: ↑ ↓ 'SD$;' ← → ← → Enter symbol
1418 BLOCK#2,12,3,412,52,6:BLOCK#2,2,4,432,50,6
1419 REPEAT Path_lp
1420 CURSOR#2,300,48:PRINT#2,drv$(dn%):CURSOR#2,384,48:PRINT#2,fnum
1421 k=CODE(INKEY$(-1))
1422 SElect ON k
1423 =192:fnum=fnum-1:IF fnum<0:fnum=9
1424 =200:fnum=fnum+1:IF fnum>9:fnum=0
1425 =208:dn%=dn%-1:IF dn%<0:dn%=dm%
1426 =216:dn%=dn%+1:IF dn%>dm%:dn%=0
1425 =10:pck=1:EXIT Path_lp
1427 =32:pck=0:EXIT Path_lp
1428 END SElect
1429 END REPEAT Path_lp
1430 END DEFine
```

1432 DEFine PROCEDURE FCheck

```
1433 CURSOR#2,240,48:PRINT#2,'Searching...':CLS#2,4
1434 PAUSE 20:DELETE drv$(dn%)&'FList'
1435 OPEN _NEW#99,drv$(dn%)&'FList':DIR#99,drv$(dn%):CLOSE#99
1436 OPEN _IN#99,drv$(dn%)&'FList'
1437 REPEAT Dir_lp
1438 IF EOF(#99):CLOSE#99:BLOCK#2,250,10,240,48,0:pck=0:EXIT Dir_lp
1439 INPUT#99,Fchk$:IF Fchk$==SD$&fnum:CLOSE#99:pck=1:EXIT Dir_lp
1440 END REPEAT Dir_lp
1441 END DEFine
```

Searching...

1443 DEFine PROCEDURE MLoad

```
1444 SelPath:IF pck=0:BLOCK#2,250,10,240,48,0:RETURN:ELSE FCheck
1445 IF pck=0 OR eck=1
1446 CURSOR#2,240,48:PRINT#2,'File Not Found...':CLS#2,4
1447 PAUSE 20:BLOCK#2,250,10,240,48,0:eck=0:RETURN
1448 END IF
1449 OPEN _IN#99,drv$(dn%)&SD$&fnum:CURSOR#2,240,48:PRINT#2,'Loading...';
1450 FOR n=1 TO 12:INPUT#99,Tres(n,3):PRINT#2,',:PAUSE 2
1451 INPUT#99,m\lev\gmax\glev\km\kr\GTS\sm\snnum:CLOSE#99
1452 CLS:MazNew:MazHall:MazTres:MazKey
1453 MSel:col=5:MazLev:Score:MazView
1454 IF km=1:Mask 4,220,20 :
1455 IF kr =1:Ring 4,250,22
1456 BLOCK#2,250,10,240,48,0:gdel=120/RND(2 TO 4):gst=1:gck=1:mck=1
1457 END DEFine
```

File Not Found...

Loading.....

1459 **DEfIne PROCEDURE MSave**

1460 **SeIPath**:IF pck=0 OR gck=0:BLOCK#2,250,10,240,48,0:RETurn :ELSE **FCheck**

1461 IF eck=1

1462 CURSOR#2,240,48:PRINT#2,'DEVICE ERROR...':CLS#2,4

DEVICE ERROR...

1463 PAUSE 20:BLOCK#2,250,10,240,48,0:eck=0:RETurn

1464 END IF

1465 IF pck=1

1466 CURSOR#2,240,48:PRINT#2,'Overwrite y/n':PAUSE

Overwrite y/n

1467 IF KEYROW(5)<>64:BLOCK#2,250,10,240,48,0:RETurn

1468 END IF

1469 DELETE drv\$(dn%)&SD\$&fnum:OPEN_NEW#99,drv\$(dn%)&SD\$&fnum

1470 CURSOR#2,240,48:PRINT#2,'Saving...';

Saving.....

1471 FOR n=1 TO 12:PRINT#99,Tres(n,3):PRINT#2,',';PAUSE 2

1472 PRINT#99,m\lev\gmax\glev\km\kr\GTS\sm\snun:CLOSE#99

1473 BLOCK#2,250,10,240,48,0

1474 **END DEfIne**

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

Exit Game (Y/N)

1476 **DEfIne PROCEDURE QExit**

1477 INK#2,6:CURSOR#2,240,48:PRINT#2,'Quit Game (Y/N)':CLS#2,4:PAUSE

1478 IF KEYROW(5)=64:LRUN dn\$:STOP:ELSE CURSOR#2,200,48:CLS#2,4

1479 **END DEfIne**

1481 **REMark Sphere of Destiny - Matching Keystones**

1483 **DEfIne PROCEDURE KeyStone**

1484 **KStone 4,340,20**:scol=6:**SDeSt:SRing**

1485 CURSOR#2,258,44:PRINT#2,'Activate the Sphere of Destiny'

1486 CURSOR#2,240,54:PRINT#2,'by Matching the Sphere and Maze Keys'

1487 INK#2,5:CURSOR#2,250,190:PRINT#2,'Use to Match and Test Keys'

1488 INK#2,7:CURSOR#2,274,190:PRINT#2,'←↑↓→←':BLOCK#2,2,4,310,192,7 ← Enter Symbol

1489 check=0:col=0:FOR kp=1 TO 4:ks=kp:**GetKey**

1490 **REPeat key_lp**

1491 IF snun<500:snun=0:**Score:Mes2:EXIT key_lp**

1492 **GetKey**:k=CODE(INKEY\$(-1))

1493 **SElect ON k**

1494 =192:kp=kp-1:IF kp<1:kp=4

1495 =200:kp=kp+1:IF kp>4:kp=1

1496 =208:ks=ks+1:IF ks>4:ks=1

1497 =216:ks=ks-1:IF ks<1:ks=4

1498 = 10:MatchKey:IF schk<5:snun=snun-500:**Score:ELSE Mes3:EXIT key_lp**

1499 =248:IF col=0:col=5:ELSE col=0

1500 **END SElect**

1501 **END REPeat key_lp**

1502 gst=0:BLOCK#2,250,30,240,40,0:**Mes1:QBITS_Maze**

1503 **END DEfIne**

1505 **DEfIne PROCEDURE MatchKey**

1506 schk=1:FOR i=1 TO 4:IF SKey(i,3)=Mkey(i):schk=schk+1

1507 **END DEfIne**

```

1509 DEFine PROCEDURE GetKey
1510 IF kp=1:Mkey(1)=ks:c=414
1511 IF kp=2:Mkey(2)=ks:c=430
1512 IF kp=3:Mkey(3)=ks:c=466
1513 IF kp=4:Mkey(4)=ks:c=484
1514 RESTORE 1516:INK#4,col:FOR i=1 TO 4:READ a:CURSOR#4,a,212:PRINT#4,SKey(i,3)
1515 RESTORE 1516:INK#4,5 :FOR i=1 TO 4:READ a:CURSOR#4,a,240:PRINT#4,Mkey(i)
1516 INK#4,6:CURSOR#4,c,240:PRINT#4,Mkey(kp)
1517 DATA 414,430,466,484
1518 END DEFine

```

```

1520 DEFine PROCEDURE Mes2
1521 BLOCK#2,256,30,232,40,0:INK#2,6:IF gst=0:RETurn
1522 CURSOR#2,240,190:PRINT#2,' Hard Luck You FAILED - Try AGAIN '
1523 CLS:CLS#3:lev=1:glev=2:MazLev:SEnd:col=0:fil=1:Guard(1):Mes1
1524 END DEFine

```

```

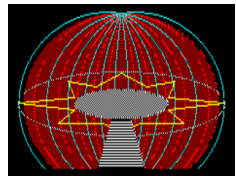
1526 DEFine PROCEDURE Mes3
1527 BLOCK#2,256,30,232,40,0:INK#2,6:INK 7:SPort
1528 CURSOR#2,240,190:PRINT#2,' The Past has Changed - Humanity Saved '
1529 PAUSE 80:CLS:CLS#3:SEnd:BLOCK#2,240,10,240,190,0:LName:Mes1
1530 END DEFine

```

```

1532 DEFine PROCEDURE SDest
1533 col=5:ss=8:x=0:y=10:CLS
1534 REPEAT SP_lp
1535 FOR i=0 TO 1.1 STEP .1
1536 ARC x,y+ss TO x,y-ss,PI*i:ARC x,y+ss TO x,y-ss,-PI*i
1537 INK col:IF col=5:col=2:ELSE col=5
1538 END FOR i
1539 BEEP 2000,8,20,-8,0,0,0:ss=ss+8:IF ss>56:EXIT SP_lp
1540 PAUSE 5:INK 2,0,3:FILL 1:CIRCLE x,y,6+ss,.9,PI/2:FILL 0
1541 END REPEAT SP_lp
1542 INK 248:CIRCLE x,y,66,.3,PI/2:y=20:INK 6
1543 LINE x,y+9 TO x-16,y TO x-34,y+4 TO x-30,y-6 TO x-66,y-10
1544 LINE TO x-30,y-14 TO x-40,y-22 TO x-12,y-22 TO x,y-28 TO x+12,y-22
1545 LINE TO x+40,y-22 TO x+30,y-14 TO x+66,y-10 TO x+30,y-6 TO x+34,y+4
1546 LINE TO x+16,y TO x,y+9:INK 0,7,1:FILL 1
1547 LINE x-6,y-20 TO x+6,y-20 TO x+15,y-50 TO x-15,y-50 TO x-6,y-20
1548 FILL 0:INK 248:FILL 1:CIRCLE x,y-10,30,.3,PI/2:FILL 0
1549 END DEFine

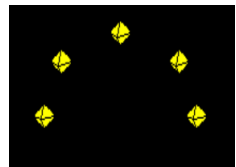
```



```

1551 DEFine PROCEDURE SRing
1552 ch=1:scol=6:FOR i=1 TO 5:KStone 1,SKey(i,1),SKey(i,2):PAUSE 5
1553 END DEFine

```



```

1555 DEFine PROCEDURE SPort
1556 INK 7:FOR i=6 TO 9:CIRCLE 0,6+i*2.5,i,.3,PI/2:CIRCLE 0,36-i*3,i,.3,PI/2
1557 FILL 1:LINE -2,46 TO 0,10 TO 2,46 TO -2,46:FILL 0
1558 FILL 1:LINE -2,42 TO 30,32 TO 2,42 TO -2,42:FILL 0
1559 FILL 1:LINE -2,42 TO -30,32 TO 2,42 TO -2,42:FILL 0
1560 FILL 1:LINE -2,42 TO 38,4 TO 2,42 TO -2,42:FILL 0
1561 FILL 1:LINE -2,42 TO -38,4 TO 2,42 TO -2,42:FILL 0
1562 END DEFine

```

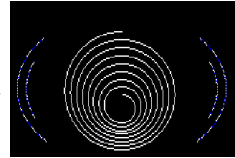


1563 DEFINE PROCEDURE SEnd

```

1565 FOR i=1 TO 24 STEP 2
1566   INK 241:CIRCLE 0,14,i*3,.7,PI/2:BEEP 2000,40,120,90,0,0,0:PAUSE 2
1567 END FOR i
1568 INK 0:FILL 1:CIRCLE,0,14,50:FILL 0:BEEP 30000,1,250,90,-8,15,15:INK 7
1569 FOR i=50 TO 15 STEP -5
1570   ARC 0,i TO 0,-i/2,PI:ARC 0,-i/2 TO 0,i,-5,PI:PAUSE i/5
1571 END FOR i
1572 BEEP 10000,4,200,190,0,0,0:PAUSE 20
1573 END DEFINE

```



QBITS Maze – Game Checks

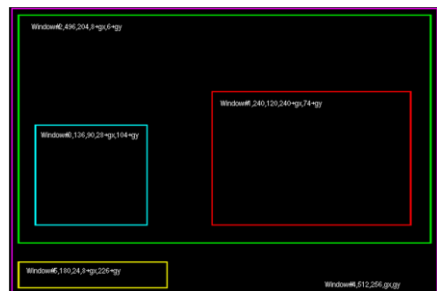
The hidden commands. These are the cheats that Gamers seek to uncover. Press [#] to toggle ON/OFF.

- 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**, F5 identifies the Sphere Key Stones Order.



1575 REMark QBITS Maze Screen Setup



1577 DEFINE PROCEDURE Init_Win

```

1578 OPEN#5,scl_:WINDOW#5,180,24,8+gx,226+gy :PAPER#5,0
1579 OPEN#4,scl_:WINDOW#4,512,256,gx,gy :PAPER#4,0
1580 OPEN#3,scl_:WINDOW#3,136,90,28+gx,106+gy :PAPER#3,5
1581 CSIZE#5,2,1:INK#5,6:BORDER#4,1,3:CLS#4:SCALE#4,240,0,0
1582 WINDOW#2,496,204,8+gx,6+gy :PAPER#2,0:BORDER#2,1,5
1583 WINDOW#1,248,120,240+gx,74+gy :PAPER#1,0:SCALE#1,100,-80,-30
1584 WINDOW#0,496,42,gx+8,gy+212 :PAPER#0,0:CSIZE#0,0,0:INK#0,7
1585 END DEFINE

```

```

1587 DEFine PROCEDURE Init_Intro
1588 CSIZE#2,2,1:OVER#2,1:CLS#2:RESTORE 1609
1589 INK#2,2:FOR i=1 TO 2:CUSOR#2,7+i, 3:PRINT#2,'QBITS Karnak'
1590 INK#2,6:FOR i=1 TO 2:CUSOR#2,9+i, 4:PRINT#2,'QBITS Karnak'
1591 CSIZE#2,0,0:OVER#2,0:INK#2,7
1592 CUSOR#2,40,34:PRINT#2,"Press 'Esc' to Skip Intro ";:cnt=0
1593 REPEAT Ans_Ip
1594 PRINT#2,':':PAUSE 5:cnt=cnt+1:IF cnt>36:EXIT Ans_Ip
1595 IF KEYROW(1)=8:Coin 4,140,22:Shield 4,180,32:Sword 4,206,6:RETurn
1596 END REPEAT Ans_Ip
1597 BLOCK#2,400,10,40,34,0:KEnter
1598 FOR a=1 TO 3:READ M$:CURSOR#2,44,24+a*10:TypeMess:PAUSE 2
1599 Score:Coin 4,140,22:Shield 4,180,32:INK#2,5
1600 FOR a=1 TO 2:READ M$:CURSOR#2,20,70+a*10:TypeMess:END FOR a:KOpen
1601 FOR a=1 TO 5:KStone 4,280+a*12,20:PAUSE 2
1602 FOR a=1 TO 2:READ M$:CURSOR#2,20,94+a*10:TypeMess
1603 CLS:MPort:Guard 1:PAUSE 20
1604 FOR a=1 TO 2:READ M$:CURSOR#2,20,114+a*10:TypeMess
1605 Sword 4,206,6:Mask 4,228,18:Ring 4,260,22:BEEP 2000,12,5,6,0,0,0:PAUSE 20
1606 FOR a=1 TO 2:READ M$:CURSOR#2,20,142+a*10:TypeMess
1607 SDest:PAUSE 20:SRing:INK#2,3:CURSOR#2,28,182
1608 PRINT#2,'Press any key to continue...':PAUSE:INK 7:SPort:PAUSE 20
1609 DATA 'The Human Race is under Threat of Extinction by a Rogue DNA Gene'
1610 DATA ' propagated by an EVENT in Prehistory. In the Tombs of Karnak'
1611 DATA ' lies a TIME PORTAL called the SPHERE of DESTINY.'
1612 DATA ' Search the Karnak Tombs for the','Keystones to unlock the PORTAL.'
1613 DATA ' The Levels are protected by','the GUARDIANS [Phantom Knights]'
1614 DATA ' To Defeat them use the Sword ' ' or find the Mask and Ring.'
1615 DATA 'Use the PORTAL to Go Back in TIME','and Prevent the EVENT happening.'
1616 END DEFine

```

```

1618 DEFine PROCEDURE TypeMess
1619 FOR b=1 TO LEN(M$):PRINT#2,M$(b):PAUSE 2
1620 END DEFine

```

View (P)ause (N)ew (L)oad (S)ave (Q)uit

```

1622 DEFine PROCEDURE Init_Menu
1623 BLOCK#2,490,40,2,28,0:BLOCK#2,220,136,2,60,0:CLS;CLS#3
1624 col=0:lev=0:MazLev:INK#2,7:Guard(4):OVER#2,1
1625 CURSOR#2,230,30:PRINT#2,'(V)iew (P)ause (N)ew (L)oad (S)ave (Q)uit '
1626 CURSOR#2,231,30:PRINT#2,' V P N L S Q '
1627 INK#2,5:FOR i=0 TO 1:CUSOR#2,8,30+i:PRINT#2,'LEVEL',,, ' GUARDIANS'
1628 OVER#2,0:gst=0:gck=0:gdel=120:pck=0
1629 INK#4,7:CIRCLE#4,140,74,9,.6,PI/2
1630 INK#2,3:FILL#2,1:LINE#2,65,8 TO 71,12 TO 76,8 TO 65,8:FILL#2,0
1631 END DEFine

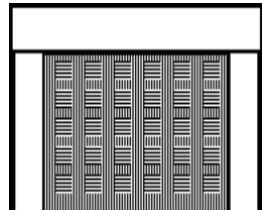
```



```

1633 DEFine PROCEDURE KEnter
1634 PAPER 0:CLS:BLOCK 188,24,30,6,0:BLOCK 184,86,32,30,0
1635 BLOCK 184,22,32,7,7:BLOCK 20,84,34,31,7:BLOCK 20,84,194,31,7
1636 BLOCK 136,82,56,32,7,0,2:BLOCK#2,248,10,240,190,0
1637 FOR a=1 TO 5:FOR b=1 TO 6:BLOCK 12,10,42+b*22,20+a*15,7,0,1
1638 END DEFine

```



```

1640 DEFine PROCEDURE KOpen
1641 FOR i=1 TO 11:BLOCK i*12,82,124-i*6,32,0:PAUSE 5
1642 END DEFine

```

```

1644 DEFine PROCEDURE TMode
1645 IF TM=1:INK#2,7:ELSE INK#2,5:END IF :CURSOR#2,230,12:PRINT#2,["#]
1646 END DEFine

```

```

1648 REMark League Table

```

```

1650 DEFine PROCEDURE LScore
1651 FOR i=1 TO 7
1652   PAUSE 5:BLOCK 200,12*i,28,68-i*6,0
1653   BLOCK 200,2,28,68-i*6,2:BLOCK 200,2,28,68+i*6,2
1654 END FOR i

```

```

1655 INK 6:OVER 1:CSIZE 2,1:FOR i=0 TO 1:CURSOR 52+i,32:PRINT 'League Table'
1656 INK 5:OVER 0:CSIZE 0,0:CURSOR 32,54:PRINT 'Points/Moves   Time   Gamer'
1657 FOR a=1 TO 3
1658   CURSOR 38,56+a*12:PRINT FILL$(" ",5-LEN(Grad(a,1)))&Grad(a,1)
1659   CURSOR 74,56+a*12:PRINT FILL$(" ",4-LEN(Grad(a,2)))&Grad(a,2)
1660   CURSOR 162,56+a*12:PRINT name$(a) :HST$=DATES$(Grad(a,3))
1661   CURSOR 106,56+a*12:PRINT HST$(13 TO 20)
1662 END FOR a
1663 END DEFine

```

```

1665 DEFine PROCEDURE LName
1666 GTS=GTS+DATE-Gclk:Gmr=0:LScore:REMark Gclk$=DATES$(GTS)
1667 FOR i=1 TO 3:IF Grad(i,1)<snum:Gmr=i:EXIT i
1668 IF Gmr=0:RETURN
1669 Grad(Gmr,1)=snum:Grad(Gmr,2)=sm:Grad(Gmr,3)=GTS
1670 ch=6:OPEN#ch,con_:WINDOW#ch,60,10,402+gx,128+gy+Gmr*12
1671 PAPER#ch,0:CLS#ch:INK#ch,6:INPUT#ch,name$(Gmr):CLOSE#ch:LTSave
1672 END DEFine

```

```

1674 DEFine PROCEDURE LSave
675 DELETE dev$&'QBMAzeLT':OPEN_NEW#99,dev$&'QBMAzeLT'
1676 FOR a=1 TO 3:PRINT#99,name$(a)\Grad(a,1)\Grad(a,2)\Grad(a,3)
1677 CLOSE#99
1678 END DEFine

```

```

1680 DEFine PROCEDURE LLoad
1681 OPEN_IN#99,dev$&'QBMAzeLT'
1682 FOR a=1 TO 3:INPUT#99,name$(a)\Grad(a,1)\Grad(a,2)\Grad(a,3)
1683 CLOSE#99
1684 END DEFine

```

```

1686 DEFine PROCEDURE LTDefault
1687 name$(1)='QBITS   ':Grad(1,1)=1730:Grad(1,2)=756:Grad(1,3)=1072
1688 name$(2)='         ':Grad(2,1)= 0:Grad(2,2)= 0:Grad(2,3)=0
1689 name$(3)='         ':Grad(3,1)= 0:Grad(3,2)= 0:Grad(3,3)=0:LSave
1690 END DEFine

```

```

1692 DEFine PROCEDURE LTRreset
1693 name$(1)='         ':Grad(1,1)=0:Grad(1,2)=0:Grad(1,3)=0
1694 name$(2)='         ':Grad(2,1)=0:Grad(2,2)=0:Grad(2,3)=0
1695 name$(3)='         ':Grad(3,1)=0:Grad(3,2)=0:Grad(3,3)=0:LSave
1696 END DEFine

```

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

Note: From command line
 LRUN LTDefault or LTRreset