MAPS TOOLKIT

HISTORY

The Just Words! mapping page dates from 2008 when Hugh Rooms wrote a series of articles for QL Today on GPS and mapping. This included a detailed description of the Mercator Projection and a program to convert latitude and longitude in QL x,y coordinates (*QL Today v12 i3 p35*). The databases that the program used were extremely large. The UK database contained 97,000 data pairs and was 943Kb in length. It would have taken 2 or 3 complete issues of QL Today to print it out. The solution was to set up the Just Words! mapping page so that readers could down load databases to try out Hugh's progams.

In the following issue of QL Today (*QL Today v12 i4 p9*) I wrote an article describing the problems caused by the size of the databases. In it I issued two programming challenges:

"First would anyone like to try writing a routine that would automatically adjust the scaling for the optimal reproduction of a map, given any maximum and minimum latitude and longitude?

Secondly who would like to try incorporating Hugh's routines in his own program? There are two big problems to be overcome. One is the size of the data base and the other the speed of writing. However you could investigate the possibility of reducing the size of the data base by taking say only every fifth or tenth data pairs and testing to see how this would affect the quality of the map."

No one responded to this challenge and now Just Words! has done the job for you.

MAP DISPLAY PROGRAM

The maps toolkit contains a number of programs to help you develop your own programs. They are simple and short and you are welcome to incorporate them in your own programs. The first one is a map display program.

```
100 REMark WINDOW 512,256,0,0 : REMark Standard resolution screen
110 WINDOW 700,500,0,0 : REMark High resolution screen
120 INK 0 : PAPER 7
130 CLS
140 :
150 mapdata$="dos7_maps_QLGB_dat" : REMark Amend for drive and data base to
be used.
160 OPEN_IN #4,mapdata$
170 :
180 REMark Input first coordinates
190 INPUT #4,x$
200 x=x$
210 y=x$(" "INSTR(x$) TO)
220 CLOSE #4
230 :
240 REMark Scan database to discover maximum and minimum x and y
coordinates
250 xmin =x : xmax=x : ymin=y : ymax=y
260 AT 1,1 : PRINT "PLEASE WAIT - ANALYSING DATABASE"
270 OPEN_IN #4,mapdata$
280 REPeat loop
```

```
290  IF EOF(#4) : EXIT loop
300  INPUT #4,x$
310  x=x$ : AT 3,1 : PRINT x
320  IF x<xmin : xmin = x
330  IF x>xmax : xmax = x
340  y=x$(" "INSTR(x$) TO) : AT 4,1 : PRINT y
350  IF y<ymin : ymin = y
360  IF y>ymax : ymax = y
370 END REPeat loop
380 CLOSE #4
390 :
400 CLS
410 AT 1,1 : PRINT "minimum x: ";xmin
420 AT 2,1 : PRINT "maximum x: ";xmax
430 AT 3,1 : PRINT "minimum y: ";ymin
440 AT 4,1 : PRINT "maximum y: ";ymax
450 :
460 REMark calculate optimum scaling
470 :
480 IF (ymax-ymin) > (xmax - xmin) : height= INT(ymax - ymin) + 5 : ELSE :
height =INT (1.37*(xmax - xmin)) + 5
490 :
500 xscale = INT((xmin + xmax - 1.37*height)/2)
510 :
520 yscale = INT((ymin +ymax - height)/2)
530 :
540 SCALE height, xscale, yscale
550 AT 6,1 : PRINT "SCALE ";height;",";xscale;",";yscale
560 :
570 REMark Now draw map
580 OPEN_IN #4,mapdata$
590 REPeat loop2
600  IF EOF(#4) : EXIT loop2
610  INPUT #4,x$
620  x=x$
630  y=x$(" "INSTR(x$) TO)
640  POINT x,y
650 END REPeat loop2
660 CLOSE #4
670 :
680 AT 8,1 : PRINT "COMPLETE"
690 PAUSE
```

The databases contain the mapping information in QL x,y coordinate in the form [xcordinate space y coordinate]. The first two coordinates of the UK database are:

1.906 59.625
1.908 59.621

The data is input as a string and converted to numbers in lines 200 and 210. The map is not drawn straight away, but the database is first analysed (lines 240 to 380). This process can take some time and to assure you the computer has not crashed the current co-ordinates are printed at the top left of the screen (lines 310 and 340). The purpose of the analysis is primarily to calculate the optimum scaling for the map. This is calculated in lines 450 to 560.

Once the analysis is complete the maximum and minimum x and y coordinates are displayed. This is useful information if you want to draw a box around your map. The recommended scaling is then displayed.

The map is drawn in lines 570 to 660. Please note that to draw the map you have to use the POINT command.

REDUCING THE SIZE OF A DATABASE

Most of the databases are too large to incorporate in a SuperBasic program. In practice it is possible to reduce the size of a database without too much loss of quality. In the downloads section of the Just Words! website there is a program for displaying the location of the first part of a UK postcode. The database of that program is one twentieth of the size of the full UK database. If, however, you want to print just a small part of a database then you may need to keep the full data for that area.

```
150 mapdata$="dos7_maps_QLGB_dat" : REMark Amend for drive and data base to
be used.
160 OPEN_IN #4,mapdata$
170 :
180 newmapdata$="ram1_newdata_dat"
190 OPEN_NEW #5,newmapdata$
200 :
210 cut_factor=20 : REMark Adjust for quality required
220 count=0
230 REPeat loop2
240  IF EOF(#4) : EXIT loop2
250 count=count +1
260 AT 1,1 : PRINT count
270  INPUT #4,data$
280 IF count=cut_factor : PRINT #5,data$ :count=0 : END IF
290 END REPeat loop2
300 CLOSE #4
310 CLOSE #5
320 :
```

Lines 150 and 160 open the existing database and lines 180 and 190 the new database.

Line 210 is the cut factor. A cut_factor of 20 will print every 20th coordinate pair to the new database. You can adjust this to the quality you want.

In practice you will have to experiment to find the optimum size of the new database. The size affects not only the quality, but also the speed of printing.

EXTRACTING A SECTION OF A DATABASE

If you do not want a map of a whole country it is possible to extract a small area.

```
5 REMark Insert boundaries of map
10 max_longitude=-1.02
20 min_longitude=-1.61
30 max_latitude=50.77
40 min_latitude=50.68
50 :
60 REMark convert to QL coordinates
70 xmaximum=.75*max_longitude
80 xminimum=.75*min_longitude
90 ymaximum=(180/PI)*LN(TAN((max_latitude)*PI/360+PI/4))
100 yminimum=(180/PI)*LN(TAN((min_latitude*PI)/360+PI/4))
110 :
```

```
120 OPEN_IN #4,dos7_maps_QLGB_dat : REMark Input file
130 OPEN_NEW #5,ram1_newdata_dat : REMark Output file
135 :
140 count=0
150 REPeat loop
160 IF EOF(#4) : EXIT loop
170 count=count +1
180 AT 1,1 : PRINT count
190 INPUT #4,data$
200 x=data$
210 y= data$(' ' INSTR(data$) TO)
220 IF x>xminimum AND x<xmaximum
230 IF y>yminimum AND y<ymaximum THEN PRINT #5, data$
240 END IF
250 END REPeat loop
260 CLOSE #4
270 CLOSE #5
```

You first of all have to enter the maximum and minimum latitude and longitude of the are of the map you wish to extract (lines 10 to 40). These are then converted to QL coordinates in lines 60 to 100. For an explanation of these lines you should refer to the article by Hugh Rooms published in QL Today (*QL Today v12 i3 p35*). If you do not know the maximum and minimum latitude and longitude you could try running the database through the map display program and printing the image onto graph paper. The program gives the maximum and minimum x and y coordinates and from this you can estimate them for the area you wish to extract. You can then enter these in lines 70 to 100.

Lines 120 and 130 open the source database and the new database respectively, and lines 140 to 270 extract the relevant area. The counter at the top left hand corner of the screen is to assure you the program has not crashed.

Do not forget to run the new database through the map display program to get the scaling and other statistics.

CONVERTING A DATABASE INTO SUPERBASIC DATA LINES

If you are writing a SuperBasic program it is often inconvenient to have to load a database. Instead data lines are better. You will need to use a little common sense to determine when it is practical to convert a database into SuperBasic data lines.

The UK database has some 97,000 coordinate pairs, and you would need 9,700 SuperBasic lines at 10 coordinate pairs per line. The postcodes program uses a cut down database of 4,850 coordinate pairs and thus need 485 lines at 10 coordinate pairs per line.

```
10 OPEN_IN #4,dos7_eastanglia_dat : REMark input file
20 OPEN_NEW #5, ram1_newdata_dat : REMark output file
30 count=0  : REMark number of data coordinates per line
40 line_number=10000 : REMark start lioe number
50 :
60 PRINT #5,line_number;" DATA '";
70 REPeat loop
80  IF EOF(#4) : PRINT#5, "END'" : EXIT loop
90  count=count + 1 : AT 1,1 : PRINT count
100  INPUT #4,data$
110  PRINT #5, data$;"'";
120  IF count = 10
```

```
130   line_number=line_number + 10 : count=0
140   PRINT #5
150   PRINT #5,line_number;" DATA '";
160  ELSE
170   PRINT #5, ",'";
180  END IF
190 END REPeat loop
200 CLOSE #4
210 CLOSE #5
```

Lines 10 and 20 open the source and output files. Line 30 allows you to adjust the number of data coordinate pairs per line and line 40 the starting line of the data lines. Line 80 prints "END" as the final string when the complete database has been converted to data lines.

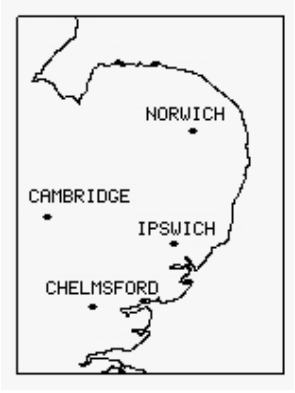
Below is a sample of the data lines produced by the program.

```
10000 DATA '.261 63.058','.263 63.053','.264 63.047','.265 63.042','.266
63.035','.267 63.029','.267 63.023','.267 63.017','.267 63.01','.267
63.004'
10010 DATA '.268 62.998','.268 62.991','.267 62.984','.267 62.977','.266
62.969','.265 62.962','.264 62.955','.262 62.949','.26 62.944','.258
62.938'
```

TWO EXAMPLES

Included in the maps toolkit are two examples of its use, one a map of East Anglia and the other of the Orkney and Shetland Islands.

East Anglia

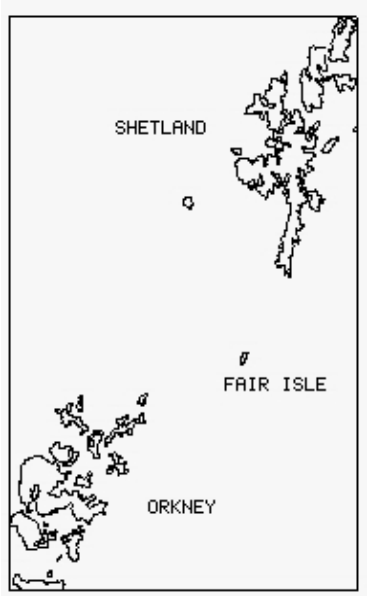


This example is of most interest to QL-ers who only have a standard resolution screen.

In this case the latitude range is from 51.40 to 53.20 and the longitude from -0.08 to 1.78. These figures were inserted into the cut database program to extract the map from the UK database.

The cut down database was then run through the make data lines program.

Orkney and Shetland



This example is of interest to QL-ers with high resolution screens.

In this case the latitude was from 58.61 to 60.88 and the longitude from -3.51 to -0.70. The values were inserted into the cut database program to extract the map.

The smaller database was then entered into the make data lines program.

PINPOINTING A LOCATION

If you want to mark a location on a map you need to know the latitude and longitude and then covert these into QL x and y coordinates. The formulae for doing this are:

*xcoordinate = .75*longitude*

*ycoordinate=(180/PI)*LN(TAN((latitude*PI)/360+PI/4))*

The detailed mathematics from which these formulae are derived as described in Hugh Room's article (*QL Today v12 i3 p35*).

ADDING TEXT

To add text to a map you should use the CURSOR keyword that integrates graphics with text. The syntax of the command is:

CURSOR [#channel,][graphics position,][pixel position]

CURSOR xcoordinate,ycoordinate,0,0 assumes the graphic is in channel 1 and places the cursor exactly on the location. In practice you will then need to adjust the pixel position to a suitable place for the text. No hard and fast rules can be given for this as it is dependent on your graphic, but you can see examples in the East Anglia and Orkney and Shetland programs.

Hopefully this manual has set you on your way to incorporate maps in your own programs. Should you find shortcomings in the manual or wich further help please contact Just Words!

Happy mapping!