CIRCLES

Jeffrey S. McArthur

Every Atari graphics programmer needs to draw circles. This tutorial will show you how to draw a circle – and draw one fast – without jumping through hoops. There are several drawing utilities here, from an elementary BASIC routine which takes 60 seconds to a machine language version that finishes in a fraction of a second. Even if you're not interested in the methodology, you can still use these subroutines in your graphics and games.

Program 1 draws circles, but takes more than a minute to draw a circle, no matter how big or small it is.

Reflections

A circle is symmetrical, so why don't we take advantage of its symmetry? If we know the value of one point, we can reflect it across the X-axis or across the Y-axis. That is, if we know (X,Y) is a point on the circle, then so is (X,-Y). The same is true for (-X,Y) and (-X,-Y). So we have to do only a quarter of the work. Circles are also symmetrical along the X=Y line. If we know (X,Y) is on the circle, then so is (Y,X). Now we have to find only an eighth of the points. Program 2 uses that method.

Unfortunately, even doing only one-eighth of the work, we still need more than ten seconds to draw the circle. Perhaps there is a better way. Instead of using sines and cosines, use the equation:

X*X +Y*Y=R*R

That isn't very useful, but we can rearrange the equation and get:

Y=SQRT (R*R - X*X)

So all we have to do is find Y for X= -R to R. However, since the square root function returns only the positive square root, we also have to plot the negative square root. Program 3 is an example of how to do that. This method is faster than using sines or cosines, but it still takes more than 16 seconds. So using Program 4, we reflect it, like we did in Program 2.

Now we have a method that takes only five seconds on a large circle and is a lot faster on the smaller ones. If you take a close look at how Program 4 draws the circle, you see it draws lines of different lengths. This method works fine on a screen, but on a plotter the circle has flat spots.

A Faster Circle

The screen is made up of an array of points. Each point is addressed by two coordinates (X,Y). However, X and Y are *always* integers. In Atari BASIC you can PLOT 0.5,0.5, but the points are rounded to integers. So if you are at one point on the circle and are trying to figure where the next point is, you can go in eight directions.

If you divide the circle into quarters, then only three of those directions are valid. If you divide the circle into eight parts, you can go in only two directions. For example, if you are on the circle at (R,0), the next point is either (R-1,0) or (R-1,1). This method is called a *potential function*. Since the screen cannot plot points except with integers, there is a small error that is not always equal to zero.

We want to keep the error as small as possible. We also reflect it eight ways as before. That takes only three seconds, and we never have to draw any long lines. Program 5 uses this method.

Notice also that you can achieve the entire result using only addition and subtraction. Such programs can be easily converted to machine language since we don't have to multiply or divide. Program 7 is a machine language program to draw a circle. Program 6 calls the machine language and takes less than two-tenths of a second to draw a circle.

The machine language is called by a USR function. The parameters that are passed to it are, in order: the address of the code, the X coordinate of the center of the circle, the Y coordinate of the center of the circle, the radius, and the mode of drawing. The mode of drawing means

0: turn point off
1: turn point on
2: invert point

The program can be converted to any 6502 machine. The only things that need to be changed are where the variables are stored and how to plot the points.

The only problem with the machine language program is that it does no checking to see if the circle goes off screen. And no clipping is done. Therefore, if your circle goes off screen, you will write over other memory.

Program 1: Sines And Cosines

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #1
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 61 SECONDS
200 DEG
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
300 FOR ALPHA=0 TO 360
310 X1=INT(R*COS(ALPHA)+0.5)
320 Y1=INT(R*SIN(ALPHA)+0.5)
330 PLOT A+X1,B+Y1
340 NEXT ALPHA
```

Program 2: Sines And Cosines Reflected

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #2
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 11 SECONDS
200 DEG
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 PLOT A+R,B
300 FOR ALPHA=0 TO 45
310 X1=INT(R*COS(ALPHA)+0.5)
320 Y1=INT(R*SIN(ALPHA)+0.5)
330 PLOT A+X1,B+Y1
340 PLOT A-X1,B+Y1
350 PLOT A+X1,B-Y1
360 PLOT A-X1,B-Y1
370 PLOT A+Y1,B+X1
380 PLOT A-Y1,B+X1
390 PLOT A+Y1,B-X1
400 PLOT A-Y1,B-X1
410 NEXT ALPHA
```

Program 3: Square Root

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #3
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 17 SECONDS
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 X0=-R:Y0=0
300 FOR X1=-R TO R
310 Y1=INT(0.5+SQR(R*R-X1*X1))
330 PLOT A+X0,B+Y0:DRAWTO A+X1,B+Y1
335 PLOT A+X0,B-Y0:DRAWTO A+X1,B-Y1
336 X0=X1:Y0=Y1
340 NEXT X1
```

Program 4: Square Root Reflected

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #4
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 5 SECONDS
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 X0=-R:Y0=0
280 X1=-R
290 Y1=INT(0.5+SQR(R*R-X1*X1))
300 PLOT A+X0,B+Y0:DRAWTO A+X1,B+Y1
310 PLOT A-X0,B+Y0:DRAWTO A-X1,B+Y1
320 PLOT A+X0,B-Y0:DRAWTO A+X1,B-Y1
330 PLOT A-X0,B-Y0:DRAWTO A-X1,B-Y1
340 PLOT A+Y0,B+X0:DRAWTO A+Y1,B+X1
350 PLOT A-Y0,B+X0:DRAWTO A-Y1,B+X1
360 PLOT A+Y0,B-X0:DRAWTO A+Y1,B-X1
370 PLOT A-Y0,B-X0:DRAWTO A-Y1,B-X1
380 X0=X1:Y0=Y1
390 IF -X1>=Y1 THEN X1=X1+1:GOTO 290
```

Program 5: Potential

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #5
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 3 SECONDS
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 PHI=0
280 Y1=0
290 X1=R
300 PHIY=PHI+Y1+Y1+1
310 PHIXY=PHIY-X1-X1+1
400 PLOT A+X1,B+Y1
410 PLOT A-X1,B+Y1
420 PLOT A+X1,B-Y1
430 PLOT A-X1,B-Y1
440 PLOT A+Y1,B+X1
450 PLOT A-Y1,B+X1
460 PLOT A+Y1,B-X1
470 PLOT A-Y1,B-X1
500 PHI=PHIY
510 Y1=Y1+1
520 IF ABS(PHIXY)<ABS(PHIY) THEN PHI
    =PHIXY:X1=X1-1
530 IF X1>=Y1 THEN 300
```

Program 6: BASIC Call To Machine Language

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #6
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 0.1833 SECONDS
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 P=7*16*16*16
300 I=USR(P,A,B,R,1)
```

Program 7:
Machine Language Circle Drawing Subroutine

```
10 REM 28000- IS SUBROUTINE
20 GOSUB 28000
30 END
28000 FOR I=0 TO 758:READ A:POKE 286
      72+I,A:NEXT I
28010 RETURN
28672 DATA 104,104,141,5,6,104
28678 DATA 141,4,6,104,141,7
28684 DATA 6,104,141,6,6,104
28690 DATA 141,9,6,141,12,6
28696 DATA 104,141,8,6,141,11
28702 DATA 6,104,104,141,10,6
28708 DATA 201,3,144,1,96,169
28714 DATA 0,141,13,6,141,14
28720 DATA 6,141,15,6,141,16
28726 DATA 6,24,173,4,6,109
28732 DATA 11,6,141,25,6,173
28738 DATA 5,6,109,12,6,141
28744 DATA 26,6,24,173,4,6
28750 DATA 109,13,6,141,29,6
28756 DATA 173,5,6,109,14,6
28762 DATA 141,30,6,56,173,4
28768 DATA 6,237,11,6,141,27
28774 DATA 6,173,5,6,237,12
28780 DATA 6,141,28,6,56,173
28786 DATA 4,6,237,13,6,141
28792 DATA 31,6,173,5,6,141
28798 DATA 14,6,141,32,6,24
28804 DATA 173,6,6,109,11,6
28810 DATA 141,33,6,173,7,6
28816 DATA 109,12,6,141,34,6
28822 DATA 24,173,6,6,109,13
28828 DATA 6,141,37,6,173,7
28834 DATA 6,109,14,6,141,38
28840 DATA 6,56,173,6,6,237
28846 DATA 11,6,141,35,6,173
28852 DATA 7,6,237,12,6,141
28858 DATA 36,6,56,173,6,6
28864 DATA 237,13,6,141,39,6
28870 DATA 173,7,6,237,14,6
28876 DATA 141,40,6,173,25,6
28882 DATA 141,0,6,173,26,6
28888 DATA 141,1,6,173,37,6
28894 DATA 141,2,6,173,38,6
28900 DATA 141,3,6,32,106,114
28906 DATA 173,27,6,141,0,6
28912 DATA 173,28,6,141,1,6
28918 DATA 32,106,114,173,25,6
28924 DATA 141,0,6,173,26,6
28930 DATA 141,1,6,173,39,6
28936 DATA 141,2,6,173,40,6
28942 DATA 141,3,6,32,106,114
28948 DATA 173,27,6,141,0,6
28954 DATA 173,28,6,141,1,6
28960 DATA 32,106,114,173,29,6
28966 DATA 141,0,6,173,30,6
28972 DATA 141,1,6,173,33,6
28978 DATA 141,2,6,173,34,6
28984 DATA 141,3,6,32,106,114
28990 DATA 173,31,6,141,0,6
28996 DATA 173,32,6,141,1,6
29002 DATA 32,106,114,173,29,6
29008 DATA 141,0,6,173,30,6
29014 DATA 141,1,6,173,35,6
29020 DATA 141,2,6,173,36,6
29026 DATA 141,3,6,32,106,114
29032 DATA 173,31,6,141,0,6
29038 DATA 173,32,6,141,1,6
29044 DATA 32,106,114,173,14,6
29050 DATA 205,12,6,240,3,144
29056 DATA 10,96,173,13,6,205
29062 DATA 11,6,144,1,96,173
29068 DATA 11,6,133,4,173,12
29074 DATA 6,133,5,173,13,6
29080 DATA 133,205,173,14,6,133
29086 DATA 206,6,4,38,5,6
29092 DATA 205,38,206,56,165,205
29098 DATA 109,15,6,141,17,6
29104 DATA 165,206,109,16,6,141
29110 DATA 18,6,24,173,17,6
29116 DATA 229,4,141,19,6,173
29122 DATA 18,6,229,5,141,20
29128 DATA 6,173,18,6,16,27
29134 DATA 73,255,141,22,6,173
29140 DATA 17,6,73,255,24,105
29146 DATA 1,141,21,6,173,22
29152 DATA 6,105,0,141,22,6
29158 DATA 24,144,9,141,22,6
29164 DATA 173,17,6,141,21,6
29170 DATA 173,20,6,16,27,73
29176 DATA 255,141,24,6,173,19
29182 DATA 6,73,255,24,105,1
29188 DATA 141,23,6,173,24,6
29194 DATA 105,0,141,24,6,24
29200 DATA 144,9,141,24,6,173
29206 DATA 19,6,141,23,6,173
29212 DATA 17,6,141,15,6,173
29218 DATA 18,6,141,16,6,24
29224 DATA 173,13,6,105,1,141
29230 DATA 13,6,173,14,6,105
29236 DATA 0,141,14,6,173,22
29242 DATA 6,205,24,6,144,39
29248 DATA 208,8,173,21,6,205
29254 DATA 23,6,144,29,173,19
29260 DATA 6,141,15,6,173,20
29266 DATA 6,141,16,6,56,173
29272 DATA 11,6,233,1,141,11
29278 DATA 6,173,12,6,233,0
29284 DATA 141,12,6,76,55,112
29290 DATA 173,2,6,133,205,169
29296 DATA 0,133,206,6,205,38
29302 DATA 206,6,205,38,206,6
29308 DATA 205,38,206,165,205,133
29314 DATA 4,165,206,133,5,6
29320 DATA 205,38,206,6,205,38
29326 DATA 206,24,165,205,101,4
29332 DATA 133,205,165,206,101,5
29338 DATA 133,206,173,0,6,133
29344 DATA 4,173,1,6,133,5
29350 DATA 70,5,102,4,70,5
29356 DATA 102,4,70,5,102,4
29362 DATA 24,165,205,101,4,133
29368 DATA 205,165,206,101,5,133
29374 DATA 206,24,165,205,101,88
29380 DATA 133,205,165,206,101,89
29386 DATA 133,206,173,0,6,41
29392 DATA 7,170,160,0,173,10
29398 DATA 6,208,10,189,41,6
29404 DATA 73,255,49,205,145,205
29410 DATA 96,201,1,208,8,189
29416 DATA 41,6,17,205,145,205
29422 DATA 96,189,41,6,81,205
29428 DATA 145,205,96,0,0,0
```

©