

7/15/98

ANIMATED FRACTALS

Mathematics and Computer Education, Vol.33 (1999), pp.236-243.

Thomas J. Osler
Mathematics Department
Rowan University
Glassboro, NJ 08028

osler@rowan.edu

1. Introduction

In this paper we show how to animate fractals on the computer screen. Fractal images are by nature very complex. In the classroom, it can be very instructive to watch the fractal image evolve before our eyes as we slowly change some important parameter, or zoom in and out of the image. All that is required is a version of BASIC or some other programming language that permits the user to store and retrieve a graphics image as a disk file. In this paper QUICK BASIC is used for the programs that are given. We will assume that the reader is familiar with the elements of fractal geometry and has already studied their creation as given in the excellent reference of Devaney [1].

Most fractals involve a parameter. For example, the famous Julia sets, (see [1], page 83) iterate the function

$$f(z, c) = z^2 + c$$

in which the parameter is c and the variable of iteration is z . In the illustrations we see twelve consecutive images of the computer screen showing Julia sets evolving as the parameter c changes. In the first image the value of c is -1.4333. In the second the value has changed slightly to -1.3667. In further images it continues to increase until it reaches $c = 0.3667$ in the final

frame. It is easy to imagine that these are selected frames from an animated computer image of the changing Julia sets.

In the next sections we will show how to create this movie version of the Julia sets with simple BASIC programs. These programs are easily modified to show other Julia sets or any other fractal having a parameter of some sort that can vary from frame to frame.

2. Creating the Fractal Movie

The first program we write will create the frames of our movie as individual files on the computer disk. For example the program MAKEMOVI.BAS can create files named JULIA1.SCN, JULIA2.SCN, JULIA3.SCN, ..., JULIA30.SCN. These 30 computer files are the 30 frames of our movie. In each frame the parameter c will change a small amount, causing the Julia set itself to change slightly. In our second program MOVIE.BAS (described in the next section) we will read these 30 files back onto our computer screen in rapid succession, thereby creating our fractal movie.

We now describe how the program MAKEMOVI.BAS works.

Lines 100 through 118 allow the user to enter the required variables for the movie. Picture the computer screen as a segment of the complex plane. The coordinates of the upper left hand corner are (XMIN, YMAX), and those of the lower right hand corner are (XMAX, YMIN). The parameter c is a complex number. In this program its real and imaginary parts are the variables CX and CY. The first complex value of c used is entered into the program in line 115 as CX0 and CY0. The final value is entered in the next line as CX1. We will assume in this movie that the imaginary part of c given by CY0 does not change, (We used CY0 = 0 for the illustration.) In the next two lines we enter the number of movie frames and the name we select for the disk

files that contain these frames. Keep the file name short (5 or less characters) since the program will add the frame number as well as the extension ".SCN" to each file name.

In lines 120 and 130 we identify the desired graphics mode and dimensions. We have selected low resolution mode (SCREEN 1) so as to speed up the creation of the frames. On our Pentium 90 PC we can create 30 frames in about 10 minutes.

In lines 140 and 150 we calculate the distance between screen pixels. Later in lines 170 and 180 these are used as the STEPs for the real and imaginary parts of z given by X and Y .

Lines 155 to 270 form the major loop in the program. Each time this loop is executed, a new frame is created. The variable for the loop is CX , the real part of the parameter c . A glance at the STEP size for CX shows that it is the range of the variable c , $(CX1 - CX0)$ divided by the NUMBER of desired computer frames.

Lines 170 to 240 create the image of a particular Julia set on the screen. The details of how this works can be found in Devaney [1]. Naturally, these lines would be changed to create movies of other types of fractals.

Lines 241 to 246 add text information to each frame. These are a nice professional touch, but are not necessary and can be omitted. This is the information highlighted in black in the illustrations.

Lines 250 to 260 are all that is needed to copy the screen image of our Julia set into a file on the disk. Lines 250 and 260 are QUICK BASICs way of writing the screen image to a disk file whose name is stored in the variable $F\$\$. In line 254 we add the number of the frame and the extension ".SCN" to the filename.

This completes our discussion of how the frames of our fractal movie are created by the program MAKEMOVI.BAS .

```
'PROGRAM MAKEMOVI.BAS
' Requiring QUICK BASIC (May not run under QBASIC)

'User enters fractal info and filename:
100 INPUT "ENTER XMIN, XMAX "; XMIN, XMAX
110 INPUT "ENTER YMIN, YMAX "; YMIN, YMAX
115 INPUT "ENTER initial CX,CY "; CX0, CY0
116 INPUT "ENTER final CX "; CX1
117 INPUT "ENTER NUMBER OF FRAMES "; NUMBER
118 INPUT "ENTER FILENAME "; F1$: N = 0

'Set screen mode and window size:
120 SCREEN 1: XPIX = 320: YPIX = 200
130 WINDOW (XMIN, YMAX)-(XMAX, YMIN)

'Find step size:
140 STX = (XMAX - XMIN) / XPIX
150 STY = (YMAX - YMIN) / YPIX

'Create fractal image frames:
155 FOR CX = CX0 TO CX1 STEP (CX1 - CX0) / NUMBER
156 CY = CY0
160 CLS
170 FOR Y = YMIN TO YMAX STEP STY
180  FOR X = XMIN TO XMAX STEP STX
182   U = X: V = Y
185   FOR I = 1 TO 25
       AS = INKEY$: IF AS <> "" THEN STOP
187   UU = U * U - V * V + CX
190   VV = 2 * U * V + CY
200   U = UU: V = VV
210   R = U * U + V * V: IF R > 4 THEN PSET (X, Y): GOTO 230
220   NEXT I
230  NEXT X
240 NEXT Y

'Add text to fractal image:
241 LOCATE 1, 1: PRINT "JULIA SET": PRINT "Z = Z^2 + C";
242 LOCATE 23, 1
244 PRINT "("; XMIN; ", "; XMAX; ") - ("; YMIN; ", "; YMAX; ") CX =", CX; " CY =", CY;
```

```
246 LOCATE 24, 1: PRINT USING "CX = ##.##### "; CX;
```

‘Memory location for our fractal image: (Required by QUICK BASIC)

```
250 DEF SEG = &HB800
```

‘Create and number our image’s filename:

```
254 N = N + 1: N$ = STR$(N): L = LEN(N$) - 1: F$ = F1$ + RIGHT$(N$, L) + ".SCN"
```

‘Store this fractal frame on the disk:

```
260 BSAVE F$, 0, &H4000
```

```
270 NEXT CX
```

3. Showing the Fractal Movie

After running the previous program, we now have say, 30 new files on the disk named JULIA1.SCN, JULIA2.SCN, ..., JULIA30.SCN. We now wish to read these files back into memory and flash them on the screen in rapid succession. For this purpose we will use our second BASIC program called MOVIE.BAS

The first line sets the screen mode to low resolution. This must match the resolution used in the program that created the fractals.

Lines 110 and 210 work together. They are QUICK BASICs way of reading the disk file into memory and flashing the image on the screen. The name of the disk file is stored in the variable F\$.

In lines 112 and 114 the user inputs the file name and the number of frames created by the previous program. Note that you must remember how many frames you created. If you forget, you can look at the disk directory and check the filenames.

In line 116 you enter a number DEL which works with line 145 to create a delay loop so that the frames do not run too rapidly. How rapidly the movie runs will depend on the speed of the disk drive and the CPU. Once the movie starts running, lines 2000 to 2020 allow you to alter the

speed at which it runs. By touching the "S" key line 2000 will increase the value of DEL, to slow down the delay between frames. By pressing the "F" key line 2010 will allow the user to decrease the value of DEL and make the movie run faster. Line 2020 will make the program quit by pressing the "Q" key.

Since the movie has only 30 frames in this example, it will be finished in a few seconds. Therefore once frames 1, 2, 3, ... 30 are shown, the program runs them in reverse. That is frames 30, 29, 28, ..., 1 are then flashed on the screen. Then the whole thing is repeated again in an endless loop. (You can quit the program by pressing "Q".) Lines 120 to 150 run the frames in the forward direction, while lines 200 to 230 run the frames in reverse. line 240 starts the show repeating again.

In subroutine 1000 we change the number in the filename so that the proper frame is read.

In subroutine 2000 the program checks to see if the user wants to run the movie <S>lower, <F>aster, or <Q>uit.

```
'PROGRAM MOVIE.BAS
'Running in QUICK BASIC (May not run in some versions of QBASIC

'Set screen mode and memory location for file frames
100 SCREEN 1
110 DEF SEG = &HB800

'User enters file information:
112 INPUT "FILENAME 5 CHAR MAX "; F1$
114 INPUT "ENTER NUMBER OF FRAMES "; NF
116 INPUT "ENTER DELAY (1 TO 10000) "; DEL

'Loop to flash fractal frames on the screen:
120 FOR N = 1 TO NF
130 GOSUB 1000
140 BLOAD F$, 0
145 FOR C = 1 TO DEL: NEXT C
150 NEXT N
```

```

'Loop to flash fractal frames in reverse:
200 FOR N = NF TO 1 STEP -1
207 GOSUB 1000
210 BLOAD F$, 0
220 FOR C = 1 TO DEL: NEXT C
230 NEXT N

```

```

'Repeat the movie:
240 GOTO 120

```

```

'Get filename with number:
1000 N$ = STR$(N): L = LEN(N$) - 1: F$ = F1$ + RIGHT$(N$, L) + ".SCN"

```

```

'Look for user pressing hot keys:
' <S>lower, <F>aster, and <Q>uit
1003 A$ = INKEY$: IF A$ <> "" THEN GOSUB 2000
1010 RETURN

```

```

'Decode hot keys:
2000 IF A$ = "S" THEN DEL = 1.5 * DEL: IF DEL > 100000 THEN DEL = 100000
2010 IF A$ = "F" THEN DEL = DEL / 1.5: IF DEL < 10 THEN DEL = 10
2020 IF A$ = "Q" THEN STOP
2030 A$ = ""
2500 RETURN

```

4. Running the Programs.

We will now show the parameters that were entered into our computer to generate the movie of the Julia sets shown in the illustrations.

The following shows the prompts that appear on the screen when you run the program MAKEMOVI.BAS. To the right we show the values that we entered.

ENTER XMIN, XMAX ?	-2,2
ENTER YMIN, YMAX ?	-2,2
ENTER initial CX, CY ?	-1.5, 0
ENTER final CX ?	0.4

ENTER NUMBER OF FRAMES ? 30

ENTER FILENAME JULIA

You will now see each frame generated on the screen and saved on your disk. You can now relax while the computer works to manufacture the movie. This may take from 10 minutes to 30 minutes or more.

Now that you have made the movie, you want to show it. Run the program MOVIE.BAS. Here are the prompts you will see as you run the program and the values you can enter.

FILENAME 5 CHAR MAX ? JULIA

ENTER NUMBER OF FRAMES ? 30

ENTER DELAY ? 1000

You now see your movie running forwards and backwards again and again. You can touch the following three keys (all in CAPS) to alter the show:

S makes the show run slower

F makes the show run faster

Q makes the show quit

Finally we note that our movies could be used to describe mathematical topics other than fractals. For example, we could show the graphs of the partial sums of a Taylor's series

$$f(x) = \lim f_n(x)$$

where

$$f_n(x) = \sum_{k=0}^n a_k x^k$$

as n varies through $0, 1, 2, \dots$. In this way the reader could use these movie techniques to advantage illustrating mathematical ideas.

5. Reference

[1] Devaney, "Chaos, Fractals and Dynamics", Addison-Wesley, 1990