

# (Sierpinski's Gasket) x 3

John Hanna

## T<sup>3</sup> - Teachers Teaching with Technology

4/25/97

Sierpinski's Gasket, that notorious 'triangle within triangle' figure that permeates many discussions of fractal geometry, occurs in many unique situations. The following TI-82 programs illustrate three of the ways in which the shape can be generated.

Program SIERAUTO uses 'cellular automata' (like Jon Conway's 'Game of Life', only in one dimension). A cell (pixel) is plotted at the center top of the screen. Each row of pixels represents a generation. In each generation (row), the prior generation is checked: If there is a living cell with 0 or 1 neighbors (to the left or right) then the cell will continue to live in this row. If an empty cell has 1 or 2 neighbors then it will be 'born' in this row.

```
PROGRAM:SIERAUTO
:FnOff
:PlotsOff
:AxesOff
:ClrDraw
:47→L
:0→Y
:Pxl-On(Y,47)
:While Y<46
:Y+1→Y
:L-1→L
:For(I,L,47,1)
:If pxl-Test(Y-1,I)
:Then
:If (not (pxl-Test(Y-1,I-1) and (pxl-
Test(Y-1,I+1)))
:Then
:Pxl-On(Y,I)
:Pxl-On(Y,94-I)
:End
:Else
:If pxl-Test(Y-1,I-1) or pxl-Test(Y-1,I+1)
:Then
:Pxl-On(Y,I)
:Pxl-On(Y,94-I)
:End
:End
:End
:End
:End
```

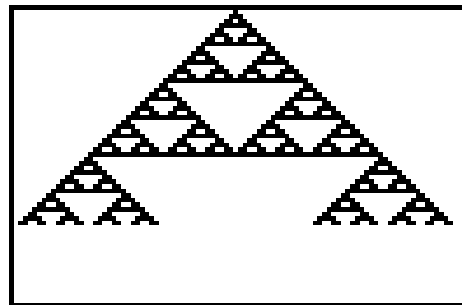


Figure 1: SIERAUTO

Program SIERRAND demonstrates the concept of ‘order out of chaos’. The viewing window is  $[0,1] \times [0,1]$ . A point  $P(A,B)$  is chosen at random. This point is plotted. A vertex of the triangle  $[(0,0), (1,0), (.5,1)]$  is selected at random and the midpoint of the segment joining  $P$  to this vertex is chosen as the ‘new’ point  $P$  (using a new *super-elegant algorithm!*). Press any key to end the program. At first, the points appear to be random, but after awhile, the Sierpinski Gasket begins to take shape! Some points in the figure may be ‘out of place’, due to the random initial point. For an interesting computer project, choose any regular polygon rather than the triangle and use the number of the selected vertex to plot the points in color.

```
PROGRAM:SIERRAND
:FnOff
:PlotsOff
:AxesOff
:ClrDraw
:0→Xmin
:1→Xmax
:0→Ymin
:1→Ymax
:rand→A
:rand→B
:While getKey=0
:Pt-On(A,B)
:int (3*rand)→R
:(A+(R≠0)/((R=0)+R))/2→A ;cool!
:(B+(R=2))/2→B ;cool, too!
:End
```

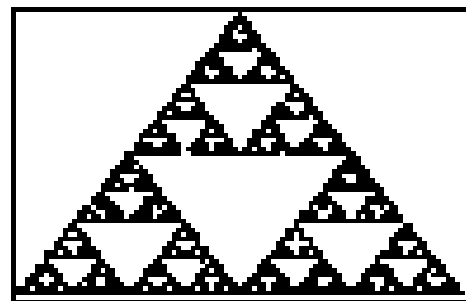


Figure 2: SIERRAND

Program SIERPAS generates a Sierpinski Gasket from Pascal’s Triangle: Look at each number in Pascal’s Triangle: if it is ODD, then plot a point, otherwise, leave it off. The figure gets distorted near the bottom of the screen due to overflow in the calculation of  $(N \text{ nCr } K)$ . For an interesting computer project, try plotting (Pascal’s Triangle) mod  $N$  on a color computer with the ‘mod value’ determining the color of the pixel.

```
PROGRAM:SIERPAS
:FnOff
:PlotsOff
:AxesOff
:ClrDraw
:47→J
:For(N,0,62,1)
:For(K,0,N,1)
:If fPart ((N nCr K)/2)=.5
:Then
:Px1-On(N,J+K)
:End
:End
:J-(N/2=int (N/2)→J
:End
```

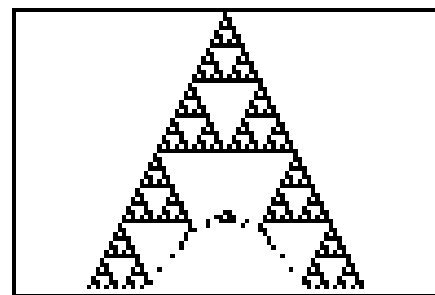


Figure 3: SIERPAS