

The purpose of this programming assignment is to generate a canvas that plots points at specific locations to create a Sierpinski triangle. You will be using the Point class from earlier, and tweaking the CoordinateSystem class from earlier. This can be achieved by the following pseudo-code:

```
v1, v2, v3 ← three vertices of an equilateral triangle
plot v1, v2, and v3
p1, p2 ← two randomly selected vertices from v1, v2, and v3
m ← the midpoint of p1 and p2
plot m (black color)
repeat
    v ← a randomly selected vertex from v1, v2, and v3
    m' ← the midpoint of m and v
    plot m'
    m ← m'
until midpoints plotted >= 1500 or tired
```

Specification

The program consists of two parts:

- Make sure the size of the canvas is 600 * 520
 - A Point class that defines a blueprint for a 2D point.
 - A ChaosGame class that manages a Tkinter Canvas and plots Point objects in random colors onto the canvas.
-

Point Class

Your Point class will likely remain unchanged from last programming assignment.

ChaosGame Class

Rename CoordinateSystem class to ChaosGame.

- Set the window title to “The Chaos Game”.
- Set the number of points to plot to 50,000 by default.
- Plotted points should be individual instances of your Point class.
- Vertices should be plotted in a color of your choice and have a radius greater than 0 (i.e., they must be larger than the generated midpoints that make up the fractal).
- Midpoints making up the fractal should be formed from the last plotted midpoint and a randomly chosen vertex (i.e., through the midpt() function in your Point class).
- Plot the first midpoint using any two vertices of the equilateral triangle.
- Midpoints making up the fractal should be plotted in a color that is different than the vertices so that the vertices are clearly visible. The midpoints must all be of the same color.
- You are free to set the canvas background to a color of your choice (however, ensure that the vertices and fractal are clearly visible).
- You are free to define other functions in your ChaosGame class as appropriate.

Hint: First, plot the three vertices. Experiment with the color and radius of the vertices. Once plotted, then move on to plot the midpoints.

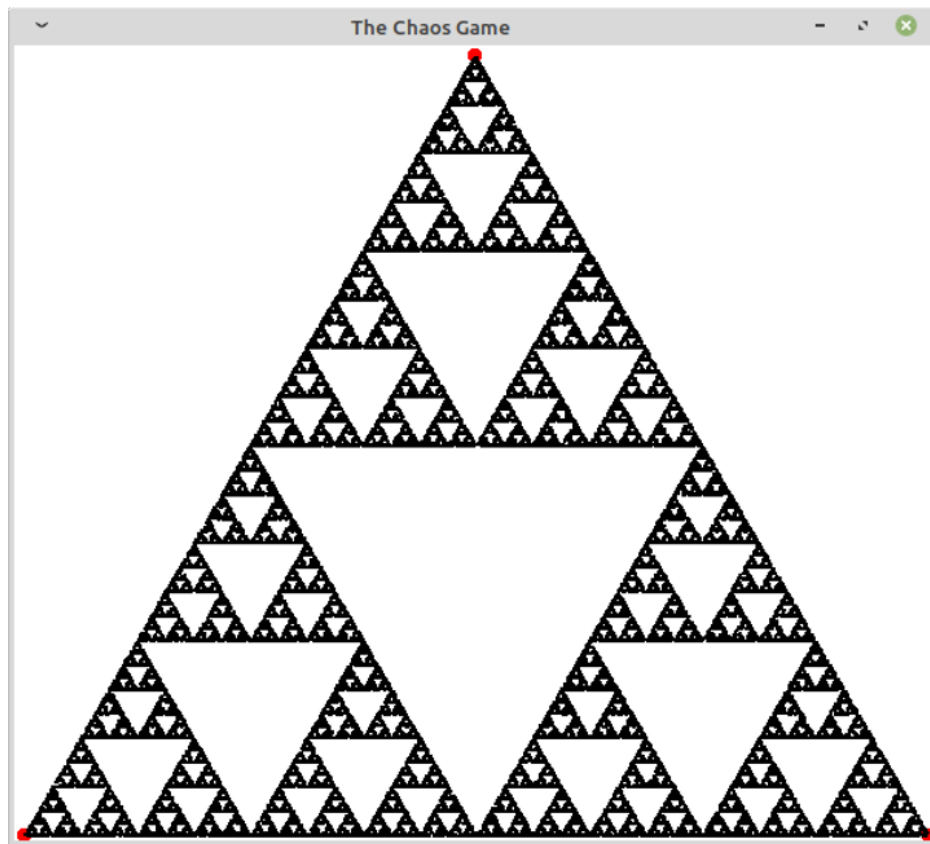


Figure 1: Sample Output

Deliverable

- Submit the Point.py and ChaosGame.py. Both classes can also be present in the same file.

Rubric

Item	Points
Good coding style	2
Appropriate Comments & Header	3
Window size & Title	3
Point class	5
50,000 Instances plotted	4
Vertices (color & size)	5
Midpoints (color & size)	5
Midpoints correctly calculated	4
First midpoint from two random vertices	3
Output is correct	6
Total	40

終わる