**fractals.py**

```
class SerpinskyCarpet(Scene):
    def construct(self):
        sideLen = 8/3
        s1 = Square(sideLen, fill_opacity = 1).center()
        g1 = VGroup(s1)
        self.add(g1)
        t = 0.1
        n = 4
        g2 = g1.copy()
        for i in range(n):
            g3 = VGroup()
            for mob in g2:
                #s = sideLen/3**(i+1) gives a sodoku like fractal
                s = sideLen/3**(i)
                h1 = mob.copy()
                h2 = mob.copy()
                h3 = mob.copy()
                h4 = mob.copy()
                h5 = mob.copy()
                h6 = mob.copy()
                h7 = mob.copy()
                h8 = mob.copy()

                if (i==0 or i==1):
                    self.play(h1.animate.scale(1/3).shift(UP*s+LEFT*s), run_time = t)
                    self.play(h2.animate.scale(1/3).shift(UP*s), run_time = t)
                    self.play(h3.animate.scale(1/3).shift(UP*s+RIGHT*s), run_time = t)
                    self.play(h4.animate.scale(1/3).shift(LEFT*s), run_time = t)
                    self.play(h5.animate.scale(1/3).shift(RIGHT*s), run_time = t)
                    self.play(h6.animate.scale(1/3).shift(DOWN*s+LEFT*s), run_time = t)
                    self.play(h7.animate.scale(1/3).shift(DOWN*s), run_time = t)
                    self.play(h8.animate.scale(1/3).shift(DOWN*s+RIGHT*s), run_time = t)
                else:
                    h1.scale(1/3).shift(UP*s+LEFT*s)
                    h2.scale(1/3).shift(UP*s)
                    h3.scale(1/3).shift(UP*s+RIGHT*s)
                    h4.scale(1/3).shift(LEFT*s)
                    h5.scale(1/3).shift(RIGHT*s)
                    h6.scale(1/3).shift(DOWN*s+LEFT*s)
                    h7.scale(1/3).shift(DOWN*s)
                    h8.scale(1/3).shift(DOWN*s+RIGHT*s)

                g3.add(h1, h2, h3, h4, h5, h6, h7, h8)

            g1.add(g3)
            self.add(g3)
            self.wait()
            g2 = g3
```

```
            t = t/10
        pass
```