```
1 import tkinter
 2 import math
 3
 4 class Window:
 5
       def __init__(self, xdim, ydim):
           self.root = tkinter.Tk()
 6
 7
           self.xdim = xdim
 8
           self.ydim = ydim
 9
           self.canvas = tkinter.Canvas(self.root, bg="white", height=ydim, width=xdim)
           self.canvas.pack()
10
11
       def drawPixel(self, x, y, colour):
12
           self.canvas.create_rectangle((x, y)*2, outline=colour)
13
14
15
       def sketchFunc(self, palette, iterations):
16
           for y in range(self.ydim):
17
               for x in range(self.xdim):
                   colour = palette(x,y, iterations)
18
19
                   self.drawPixel(x, y, colour)
20
21 def update(win, palette, iterations):
       win.canvas.delete("all")
22
       win.sketchFunc(palette, iterations)
23
24
25 resolutionx = 700
26 resolutiony = 400
27
28 iterations = [[0 for x in range(resolutionx)] for y in range(resolutiony)]
29 maxIterations = 100
30
31 def palette(x,y,iterations):
32
       i = iterations[y][x]
       if i == 0:
33
           return "white smoke"
34
       elif i < 100:
35
36
           return "grey"+str(100-i)
37
       else:
           return "black"
38
39
40 def xoff(x):
       #return x/200 - 2.5
41
42
       return x/2500 - 0.25
43
44 def yoff(y):
45
       #return x/200 - 1
46
       return y/2500 - 1
47
48 for y in range(resolutiony):
49
       for x in range(resolutionx):
50
           iteration = 0
51
           x0 = xoff(x)
52
           y0 = yoff(y)
53
           xn, yn = 0, 0
54
           while math.pow(x0,2)+math.pow(y0,2)<math.pow(2.2,2) and iteration <
   maxIterations:
55
               #print(iteration)
56
               try:
                   xtemp = math.pow(xn,2)-math.pow(yn,2) + x0
57
58
                   yn = 2*xn*yn + y0
```

```
59
                                                                                                 xn = xtemp
                                                                                                 iteration += 1
60
61
                                                                            except:
62
                                                                                                  break
                                                       #if iteration > 0:
63
                                                                                print(str(x)+","+str(y))
64
65
                                                       iterations[y][x] = iteration
                                   if y%10==0:
66
67
                                                       print(y)
68
69 win = Window(resolutionx, resolutiony)
70 \#update(win, [[lambda x, y: iterations[int(y)][int(x)]==0, "white smoke"], [lambda x, y: iterations[int(y)][int(x)]=0, "white smoke"], [lambda x, y: iter
              y: iterations[int(y)][int(x)] in range(1,100), "grey"+str(100-iterations[int(y)]
              [int(x)])],["black"]])
71 update(win, palette, iterations)
72 #print(iterations[0:10])
73 win.root.mainloop()
```

```
1 import tkinter
 2 import math
 3
 4 class Window:
 5
       def __init__(self, xdim, ydim):
           self.root = tkinter.Tk()
 6
 7
           self.xdim = xdim
 8
           self.ydim = ydim
 9
           self.canvas = tkinter.Canvas(self.root, bg="white", height=ydim, width=xdim)
           self.canvas.pack()
10
11
       def drawPixel(self, x, y, colour):
12
           self.canvas.create_rectangle((x, y)*2, outline=colour)
13
14
       def sketchFunc(self, palette, iterations):
15
16
           for y in range(self.ydim):
               for x in range(self.xdim):
17
                   self.drawPixel(x, y, palette(x, y, iterations))
18
19
20 resolutionx = 1400
21 resolutiony = 800
23 iterations = [[0 for x in range(resolutionx)] for y in range(resolutiony)]
24 global maxIterations
25 maxIterations = 500
26
27 def update(win, palette, iterations):
28
       win.canvas.delete("all")
       win.sketchFunc(palette, iterations)
29
30
31 def palette(x,y,iterations):
32
       global maxIterations
       colours = ["navy","medium blue","royal blue","cornflower blue","light sky
33
   blue", "LightSkyBlue2", "ghost white", "lemon chiffon", "light goldenrod yellow", "light
   goldenrod","black"]
34
       i = iterations[y][x]
       if i == 0:
35
           return colours[0]
36
       base = math.pow(maxIterations,1/(len(colours)-1))
37
38
       return colours[math.floor(math.log(i,base))]
39
40 def xoff(x):
41
       return x/500 - 2.5
42
       #return x/5000 - 0.4
43
44 def yoff(y):
45
       return y/500 - 1
       #return y/5000 - 0.7
46
47
48 for y in range(resolutiony):
49
       for x in range(resolutionx):
           iteration = 0
50
           x0 = xoff(x)
51
52
           y0 = yoff(y)
53
           xn, yn = 0, 0
           while math.pow(x0,2)+math.pow(y0,2)<math.pow(2.2,2) and iteration <
54
   maxIterations:
55
               try:
56
                   xtemp = math.pow(xn,2)-math.pow(yn,2) + x0
```

```
57
                   yn = 2*xn*yn + y0
58
                   xn = xtemp
                   iteration += 1
59
60
               except:
61
                   break
           iterations[y][x] = iteration
62
63
       if y%100==0:
64
           print(y)
65
66 win = Window(resolutionx, resolutiony)
67 update(win, palette, iterations)
68 win.root.mainloop()
69
```