from OpenGL.GL import \*

from OpenGL.GLUT import \*

from OpenGL.GLU import \*

import random

import math

def draw\_pixels(x, y):

glPointSize(2)

glBegin(GL\_POINTS)

glVertex2f(x,y)

glEnd()

def iterate():

glViewport(0, 0, 500, 500)

glMatrixMode(GL\_PROJECTION)

glLoadIdentity()

glOrtho(0.0, 500, 0.0, 500, 0.0, 1.0)

glMatrixMode (GL\_MODELVIEW)

glLoadIdentity()

def Circle\_pixes(x, y, x0, y0):

draw\_pixels(x + x0, y + y0) # zone 1

draw\_pixels(y + y0, x + x0) # zone 0

draw\_pixels(-x + x0, y + y0) # zone 2

draw\_pixels(-y + y0, x + x0) # zone 3

draw\_pixels(-y + y0, -x + x0) # zone 4

draw\_pixels(-x + x0, -y + y0) # zone 5

draw\_pixels(x + x0, -y + y0) # zone 6

draw\_pixels(y + y0, -x + x0) # zone 7

def Midpoint\_circle(radius, x0, y0):

d = 1 - radius # initial d value

x = 0

y = radius

Circle\_pixes(x, y, x0, y0)

while (x < y):

if (d < 0): # choose East

d = d + 2\*x + 3

x += 1

else: # choose South East

d = d + 2\*x - 2\*y + 5

x += 1

y -= 1

Circle\_pixes(x, y, x0, y0)

def showScreen():

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)

glLoadIdentity()

iterate()

glColor3f(0.0, 1.0, 1.0)

x, y, radius = 250, 250, 220 # center & radius for outer circle

Midpoint\_circle(radius, x, y) # outer circle

glColor3f(0.0, 1.0, 0.0)

Midpoint\_circle(radius / 2, x + radius / 2, y) # right

Midpoint\_circle(radius / 2, x - radius / 2, y) # left

Midpoint\_circle(radius / 2, x, y + radius / 2) # top

Midpoint\_circle(radius / 2, x, y - radius / 2) # bottom

n = 8

m = 360 / (4\*(n//4))

for i in range((n//4) - 1):

glColor3f(random.randint(0,1), random.randint(0,1), random.randint(0,1))

Midpoint\_circle(radius / 2, x + math.sin(math.radians(m)) \* radius / 2, y + math.sin(math.radians(m)) \* radius / 2) # top right diagonal

Midpoint\_circle(radius / 2, x - math.sin(math.radians(m)) \* radius / 2, y + math.sin(math.radians(m)) \* radius / 2) # top left diagonal

Midpoint\_circle(radius / 2, x - math.sin(math.radians(m)) \* radius / 2, y - math.sin(math.radians(m)) \* radius / 2) # bottom left diagonal

Midpoint\_circle(radius / 2, x + math.sin(math.radians(m)) \* radius / 2, y - math.sin(math.radians(m)) \* radius / 2) # bottom right diagonal

m += m

glutSwapBuffers()

glutInit()

glutInitDisplayMode(GLUT\_RGBA)

glutInitWindowSize(500, 500) #window size

glutInitWindowPosition(0, 0)

wind = glutCreateWindow(b"OpenGL Coding Practice") #window name

glutDisplayFunc(showScreen)

glutMainLoop()