from OpenGL.GL import \*

from OpenGL.GLUT import \*

from OpenGL.GLU import \*

def find\_zone(x1, y1, x2, y2):

zone = 0

dy = y2 - y1

dx = x2 - x1

if (abs(dx) > abs(dy)):

if (dx >= 0 and dy >= 0):

zone = 0

elif (dx <= 0 and dy >= 0):

zone = 3

elif (dx <= 0 and dy <= 0):

zone = 4

elif (dx >= 0 and dy <= 0):

zone = 7

else:

if (dx >= 0 and dy >= 0):

zone = 1

elif (dx <= 0 and dy >= 0):

zone = 2

elif (dx <= 0 and dy <= 0):

zone = 5

elif (dx >= 0 and dy <= 0):

zone = 6

return zone

def converted\_zone\_0(zone, X, Y):

conv\_dict = {1:[X,Y],

2:[Y,-X],

3:[-X,Y],

4:[-X,-Y],

5:[-Y,-X],

6:[-Y,X],

7:[X,-Y],

}

if zone<1 or zone>7:

return [X,Y]

else: return conv\_dict[zone]

def original\_zone(zone, X, Y):

conv\_dict = {1:[Y,X],

2:[-Y,X],

3:[-X,Y],

4:[-X,-Y],

5:[-Y,-X],

6:[Y,-X],

7:[X,-Y],

}

if zone<1 or zone>7:

return [X,Y]

else: return conv\_dict[zone]

def midpoint\_line(x1, y1, x2, y2):

zone = find\_zone(x1, y1, x2, y2)

converted\_zone1 = converted\_zone\_0(zone, x1, y1)

converted\_zone2 = converted\_zone\_0(zone, x2, y2)

x1 = converted\_zone1[0]

y1 = converted\_zone1[1]

x2 = converted\_zone2[0]

y2 = converted\_zone2[1]

dx = x2 - x1

dy = y2 - y1

d = 2 \* dy - dx

nE = 2 \* (dy - dx)

e = 2 \* dy

x = x1

y = y1

while (x <= x2):

x+=1

if (d <=0 ):

d = d + e

else:

d = d + nE

y+=1

OriginalZone = original\_zone(zone, x, y)

glVertex2f(OriginalZone[0], OriginalZone[1])

def draw():

glBegin(GL\_POINTS)

# 0

midpoint\_line(190, 230, 160, 230) #Upper Line

midpoint\_line(190, 230, 190, 170) #Right Line

midpoint\_line(190, 170, 160, 170) #Lower Line

midpoint\_line(160, 230, 160, 170) #Left Line

#1

midpoint\_line(230, 230, 230, 170) #Right Line

glEnd()

glFlush()

def myInit():

glClearColor(255, 255, 0.0, 0.0)

glPointSize(1.0)

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

glMatrixMode(GL\_MODELVIEW)

def iterate():

glViewport(50, 100, 500, 500)

glMatrixMode(GL\_PROJECTION)

glLoadIdentity()

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

glMatrixMode(GL\_MODELVIEW)

glLoadIdentity()

def showScreen():

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

glLoadIdentity()

iterate()

glColor3f(1.0, 0.0, 0.0)

# call the draw methods here

draw()

glutSwapBuffers()

glutInit()

glutInitDisplayMode(GLUT\_RGBA)

glutInitWindowSize(500, 500)

glutInitWindowPosition(0, 0)

wind = glutCreateWindow(b"Lab 02 Ques 01: Building 01 for ID: 19301101")

glutDisplayFunc(showScreen)

glutIdleFunc(showScreen)

glutMainLoop()