from OpenGL.GL import \*

from OpenGL.GLUT import \*

from OpenGL.GLU import \*

def draw\_points(x, y):

glPointSize(5) #pixel size. by default 1 thake

glBegin(GL\_POINTS)

glVertex2f(x,y) #jekhane show korbe pixel

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 find\_zone(dx, dy):

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

if dx >= 0 and dy >= 0:

return 0

elif dx <= 0 and dy >= 0:

return 3

elif dx <= 0 and dy <= 0:

return 4

elif dx >= 0 and dy <= 0:

return 7

else:

if dx >= 0 and dy >= 0:

return 1

elif dx <= 0 and dy >= 0:

return 2

elif dx <= 0 and dy <= 0:

return 5

elif dx >= 0 and dy <= 0:

return 6

def convert\_to\_zone\_zero(zone, x, y):

if zone == 0:

return x, y

if zone == 1:

return y, x

if zone == 2:

return y, -x

if zone == 3:

return -x, y

if zone == 4:

return -x, -y

if zone == 5:

return -y, -x

if zone == 6:

return -y, x

if zone == 7:

return x, -y

def convert\_to\_original\_zone(zone, x, y):

if zone == 0:

return x, y

if zone == 1:

return y, x

if zone == 2:

return -y, x

if zone == 3:

return -x, y

if zone == 4:

return -x, -y

if zone == 5:

return -y, -x

if zone == 6:

return y, -x

if zone == 7:

return x, -y

def mid\_point\_line(x1, y1, x2, y2, zone):

dx = x2 - x1

dy = y2 - y1

d\_init = (2\*dy) - dx

del\_E = 2\*dy

del\_NE = 2 \* (dy-dx)

x = x1

y = y1

while x < x2:

x\_org, y\_org = convert\_to\_original\_zone(zone, x, y)

draw\_points(x\_org, y\_org)

if d\_init < 0:

x += 1

d\_init += del\_E

else:

x += 1

y += 1

d\_init += del\_NE

def drawLine(x1, y1, x2, y2):

dx = x2 - x1

dy = y2 - y1

zone = find\_zone(dx, dy)

new\_x1, new\_y1 = convert\_to\_zone\_zero(zone, x1, y1)

new\_x2, new\_y2 = convert\_to\_zone\_zero(zone, x2, y2)

mid\_point\_line(new\_x1, new\_y1, new\_x2, new\_y2, zone)

def number(x,y,a):

if a == 0:

drawLine(x, y, x+100, y)

drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

#drawLine(x, y+100, x+100, y+100)

drawLine(x, y+100, x, y+200)

drawLine(x, y + 200, x+100, y + 200)

drawLine(x+100, y + 100, x+100, y + 200)

elif a == 1:

#drawLine(x, y, x+100, y)

#drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

#drawLine(x, y+100, x+100, y+100)

#drawLine(x, y+100, x, y+200)

#drawLine(x, y + 200, x+100, y + 200)

drawLine(x+100, y + 100, x+100, y + 200)

elif a == 2:

drawLine(x, y, x+100, y)

drawLine(x, y+100, x, y)

#drawLine(x+100, y, x+100, y+100)

drawLine(x, y+100, x+100, y+100)

#drawLine(x, y+100, x, y+200)

drawLine(x, y + 200, x+100, y + 200)

drawLine(x+100, y + 100, x+100, y + 200)

elif a == 3:

drawLine(x, y, x+100, y)

#drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

drawLine(x, y+100, x+100, y+100)

#drawLine(x, y+100, x, y+200)

drawLine(x, y + 200, x+100, y + 200)

drawLine(x+100, y + 100, x+100, y + 200)

elif a == 4:

#drawLine(x, y, x+100, y)

#drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

drawLine(x, y+100, x+100, y+100)

drawLine(x, y+100, x, y+200)

#drawLine(x, y + 200, x+100, y + 200)

drawLine(x+100, y + 100, x+100, y + 200)

elif a == 5:

drawLine(x, y, x+100, y)

#drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

drawLine(x, y+100, x+100, y+100)

drawLine(x, y+100, x, y+200)

drawLine(x, y + 200, x+100, y + 200)

#drawLine(x+100, y + 100, x+100, y + 200)

elif a == 6:

drawLine(x, y, x+100, y)

drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

drawLine(x, y+100, x+100, y+100)

drawLine(x, y+100, x, y+200)

drawLine(x, y + 200, x+100, y + 200)

#drawLine(x+100, y + 100, x+100, y + 200)

elif a == 7:

#drawLine(x, y, x+100, y)

#drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

#drawLine(x, y+100, x+100, y+100)

#drawLine(x, y+100, x, y+200)

drawLine(x, y + 200, x+100, y + 200)

drawLine(x+100, y + 100, x+100, y + 200)

elif a == 8:

drawLine(x, y, x+100, y)

drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

drawLine(x, y+100, x+100, y+100)

drawLine(x, y+100, x, y+200)

drawLine(x, y + 200, x+100, y + 200)

drawLine(x+100, y + 100, x+100, y + 200)

elif a == 9:

drawLine(x, y, x+100, y)

#drawLine(x, y+100, x, y)

drawLine(x+100, y, x+100, y+100)

drawLine(x, y+100, x+100, y+100)

drawLine(x, y+100, x, y+200)

drawLine(x, y + 200, x+100, y + 200)

drawLine(x+100, y + 100, x+100, y + 200)

def showScreen():

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

glLoadIdentity()

iterate()

glColor3f(1, 0.3, 0.8) #konokichur color set (RGB)

#call the draw methods here

#draw\_points(250, 250)

#number(100,150,9)

id = '21201785'

x = 100

j = -2

for i in range(2):

number(x,150,int(id[j]))

x += 150

j += 1

glutSwapBuffers()

glutInit()

glutInitDisplayMode(GLUT\_RGBA)

glutInitWindowSize(500, 500) #window size

glutInitWindowPosition(0, 0)

wind = glutCreateWindow(b"A2: Drawing last 2 digit of Student ID ") #window name

glutDisplayFunc(showScreen)

glutMainLoop()

