import random  
  
from OpenGL.GL import \*  
from OpenGL.GLUT import \*  
from OpenGL.GLU import \*  
  
  
# <!--MID POINT LINE START-->  
  
  
def draw\_points\_lines(x, y):  
 glPointSize(2)  
 glBegin(GL\_POINTS)  
 glVertex2f(x, y)  
 glEnd()  
  
  
def find\_zones(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 7  
 elif dx<=0 and dy<=0:  
 return 4  
 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\_zone0(x1, y1, x2, y2, zone):  
 if zone == 0:  
 return x1, y1, x2, y2  
 elif zone == 1:  
 return y1, x1, y2, x2  
 elif zone == 2:  
 return y1, -x1, y2, -x2  
 elif zone == 3:  
 return -x1, y1, -x2, y2  
 elif zone == 4:  
 return -x1, -y1, -x2, -y2  
 elif zone == 5:  
 return -y1, -x1, -y2, -x2  
 elif zone == 6:  
 return -y1, x1, -y2, x2  
 elif zone == 7:  
 return x1, -y1, x2, -y2  
  
  
def convert\_original\_zone(x, y, zone):  
 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\_count(x1, y1, x2, y2):  
 dx = x2 - x1  
 dy = y2 - y1  
 zone\_no = find\_zones(dx, dy)  
 x1, y1, x2, y2 = convert\_to\_zone0(x1, y1, x2, y2, zone\_no)  
 d0 = 2\*dy - dx  
 dNE = 2\*(dy - dx)  
 dE = 2\*dy  
 x = x1  
 y = y1  
  
 while x < x2:  
 p = x  
 q = y  
 p, q = convert\_original\_zone(x, y, zone\_no)  
 draw\_points\_lines(p, q)  
 x = x+1  
  
 if(d0 < 0):  
 d0 = d0 + dE  
 else:  
 d0 = d0 + dNE  
 y = y+1  
  
# <!--MID POINT LINE END-->  
  
  
# <!--MID POINT CIRCLE START-->  
  
  
def points(x, y):  
 glPointSize(2)  
 glBegin(GL\_POINTS)  
 glVertex2f(x, y)  
 glEnd()  
  
  
def draw\_point\_circle(x, y, center\_x, center\_y, x\_values\_array, y\_values\_array):  
 x\_values\_array = x\_values\_array + [x + center\_x, y + center\_x, y + center\_x, x + center\_x, -x + center\_x, -y + center\_x, -y + center\_x, -x + center\_x]  
 y\_values\_array = y\_values\_array + [y + center\_y, x + center\_y, -x + center\_y, -y + center\_y, -y + center\_y, -x + center\_y , x + center\_y, y + center\_y]  
 for (j, k) in zip(x\_values\_array, y\_values\_array):  
 points(j, k)  
  
  
def mid\_Point\_Circle(radius, center\_x, center\_y, x\_values\_array, y\_values\_array):  
 x = 0  
 y = radius  
 d = 1 - radius  
 draw\_point\_circle(x, y, center\_x, center\_y, x\_values\_array, y\_values\_array)  
 while x < y:  
 if d > 0:  
 d += 2 \* x - 2 \* y + 5  
 x += 1  
 y -= 1  
 else:  
 d += 2 \* x + 3  
 x += 1  
 draw\_point\_circle(x, y, center\_x, center\_y, x\_values\_array, y\_values\_array)  
  
  
value\_x\_array = []  
value\_y\_array = []  
  
  
# <!--MID POINT CIRCLE END-->  
  
  
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 display\_screen():  
 glClearColor(0, 0, 0, 0)  
 glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)  
 glLoadIdentity()  
 iterate()  
 glColor3f(1, 1, 1)  
  
  
 # Sun  
 mid\_Point\_Circle(20, 400, 400, value\_x\_array, value\_y\_array)  
  
 mid\_point\_count(423, 423, 429, 429)  
 mid\_point\_count(360, 400, 370, 400)  
 mid\_point\_count(430, 400, 440, 400)  
 mid\_point\_count(400, 438, 400, 428)  
 mid\_point\_count(370, 368, 378, 376)  
 mid\_point\_count(378, 425, 369, 434)  
  
  
 # Hills  
 mid\_point\_count(0, 350, 500, 350)  
 mid\_point\_count(110, 350, 0, 450)  
 mid\_point\_count(82, 380, 173, 445)  
 mid\_point\_count(238, 380, 170, 430)  
 mid\_point\_count(210, 350, 299, 427)  
 mid\_point\_count(380, 350, 300, 360)  
 mid\_point\_count(370, 360, 500, 420)  
  
  
 # Swing (Dolna)  
 mid\_point\_count(15, 280, 175, 280) # swing stant  
 mid\_point\_count(60, 280, 60, 145) # left rope  
 mid\_point\_count(130, 280, 130, 145) # Right rope  
 mid\_point\_count(40, 145, 150, 145) # swing seat1  
 mid\_point\_count(40, 142, 150, 142) # swing seat2  
  
  
 # River line  
 mid\_point\_count(70, 0, 249.5, 209.5)  
 mid\_point\_count(280, 210, 360, 260)  
 mid\_point\_count(360, 260, 400, 350)  
  
  
 # Boat  
 mid\_point\_count(350, 150, 450, 150) # top  
 mid\_point\_count(350, 150, 330, 180) # left side  
 mid\_point\_count(450, 150, 470, 180) # Right side  
 mid\_point\_count(315, 140, 350, 190,) # boitha  
  
 mid\_Point\_Circle(7, 340, 185, value\_x\_array, value\_y\_array) # majhi  
 mid\_point\_count(340, 176, 340, 160)  
  
  
 glutSwapBuffers()  
  
  
glutInit()  
glutInitDisplayMode(GLUT\_RGBA)  
glutInitWindowSize(500, 500)  
glutInitWindowPosition(0, 0)  
wind = glutCreateWindow(b"Lab Final")  
glutDisplayFunc(display\_screen)  
  
glutMainLoop()

