**Lab-03 (OpenGL Circle Spawn)**

**Codes:**

Save the codes according to their file name as shown above each code. Then just run the `task.py` file to play the scene.

**CircleModule.py**

#!/usr/bin/env python3

from OpenGL.GL import \*

from OpenGL.GLUT import \*

from OpenGL.GLU import \*

class Circle:

def \_\_init\_\_(self, x=0, y=0, radius=10, weight=2, color=(1, 1, 1)):

self.x = x

self.y = y

self.radius = radius

self.color = color

self.weight = weight

self.eightWaySymmetry = {

0: lambda x, y: (y, x),

1: lambda x, y: (x, y),

2: lambda x, y: (-x, y),

3: lambda x, y: (-y, x),

4: lambda x, y: (-y, -x),

5: lambda x, y: (-x, -y),

6: lambda x, y: (x, -y),

7: lambda x, y: (y, -x),

}

def draw(self):

self.midpointCircleAlgo()

def circlePoint(self, x, y):

for i in range(0, 8):

a, b = self.eightWaySymmetry[i](x, y)

self.drawPoint(a+self.x, b+self.y)

def drawPoint(self, x, y):

glPointSize(self.weight)

glBegin(GL\_POINTS)

glColor3f(self.color[0], self.color[1], self.color[2])

glVertex2f(x, y)

glEnd()

def midpointCircleAlgo(self):

x = 0

y = self.radius

d = 1 - self.radius

self.circlePoint(x, y)

while (x < y):

if (d < 0): # E

d += 2\*x + 3

x += 1

else:

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

x += 1

y -= 1

self.circlePoint(x, y)

**task.py**

#!/usr/bin/env python3

#####################################################

######## CSE423\_Lab03 #####################

######## Abir Ahammed Bhuiyan ##################

######## 20101197 ################

######## section: 01 ##############

##########################################

from OpenGL.GL import \*

from OpenGL.GLUT import \*

from OpenGL.GLU import \*

from CircleModule import Circle

class CircleScene:

def \_\_init\_\_(self):

self.width = self.height = 600

self.speed = 0.1

self.freeze = False

self.background = 0

self.circles = []

def drawCircles(self):

for circle in self.circles:

circle.draw()

def checkCollision(self):

if not self.freeze:

for idx, circle in enumerate(self.circles):

if circle.x-circle.radius < 0 or circle.x+circle.radius > self.width or circle.y-circle.radius < 0 or circle.y+circle.radius > self.height:

self.circles.pop(idx)

def iterate(self):

glViewport(0, 0, self.width, self.height)

glMatrixMode(GL\_PROJECTION)

glLoadIdentity()

glOrtho(0.0, self.width, 0.0, self.height, 0.0, 1.0)

glMatrixMode(GL\_MODELVIEW)

glLoadIdentity()

def showScreen(self):

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

glLoadIdentity()

self.iterate()

glClearColor(self.background, self.background, self.background, 1.0)

self.checkCollision()

self.drawCircles()

glutSwapBuffers()

def animate(self):

if not self.freeze:

for circle in self.circles:

circle.radius += self.speed

glutPostRedisplay()

def keyboardListener(self, key, x, y):

if key == b' ':

self.freeze = False if self.freeze else True

glutPostRedisplay()

def mouseListener(self, button, state, x, y):

if not self.freeze:

if button == GLUT\_RIGHT\_BUTTON and state == GLUT\_DOWN:

self.circles.append(Circle(x=x, y=self.height-y, radius=1, weight=2, color=(1, 0, 0)))

glutPostRedisplay()

def specialKeyListener(self, key, x, y):

if not self.freeze:

if key == GLUT\_KEY\_RIGHT:

if self.speed >= 0.0001:

self.speed -= 0.01

if key == GLUT\_KEY\_LEFT:

if self.speed <= 1:

self.speed += 0.01

def run(self):

glutInit()

glutInitDisplayMode(GLUT\_RGBA)

glutInitWindowSize(self.width, self.height)

glutInitWindowPosition(0, 0)

glutCreateWindow("OpenGL Circle Spawn")

glutDisplayFunc(self.showScreen)

glutIdleFunc(self.animate)

glutKeyboardFunc(self.keyboardListener)

glutMouseFunc(self.mouseListener)

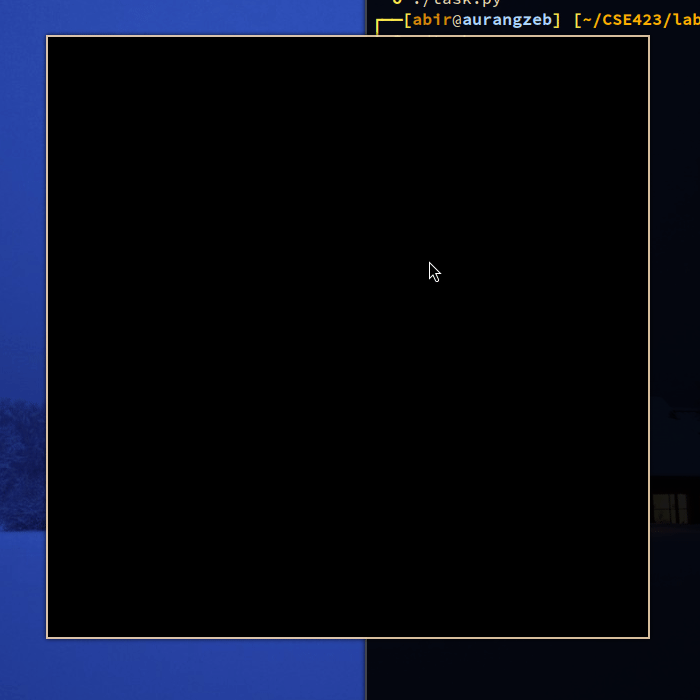
glutSpecialFunc(self.specialKeyListener)

glutMainLoop()

if \_\_name\_\_ == "\_\_main\_\_":

CircleScene().run()

**The GIF**

****

If above gif is not visible then please visit this link <https://0x0.st/HwEj.gif> for the gif.