Name: Nowshin Sumaiya

ID: 21301276

Sec:06

Course:CSE423 (lab03)

```
import random
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

total_lives = 3
no_of_life_lost = False
missed = False #one
misses = 0  #more than one
damaged = 0

def draw_points(x,y) :
   glPointSize(5)
   glBegin( GL_POINTS )
   glVertex2f(x,y )
   glEnd()

def midpoint_circle_draw(rad, qx, qy) :
    x = 0
    y = rad
    d = 1 -rad
    draw_circle(x, y, qx, qy )
    while y > x :
        if d < 0 :
            d = d+ 2 * x + 3
        else :
            d =d+ 2 * (x - y) +5
            y = y-1
        x =x+ 1
        draw_circle(x, y, qx, qy)
#---------------------------------------------
def draw_circle(x, y, qx, qy) :
    glPointSize(4)
    glBegin(GL_POINTS)

    glVertex2f(x + qx, y + qy)
    glVertex2f(-x + qx, y + qy)
    glVertex2f(x + qx, -y + qy)
    glVertex2f(-x + qx, -y + qy)
    glVertex2f(y + qx, x + qy)
    glVertex2f(-y + qx, x + qy)
    glVertex2f(y + qx, -x + qy)
    glVertex2f(-y + qx, -x + qy)
    glEnd()
#-----------------------------------------------
def zone_convert(x, y, zoneN): #zoneN to zone 0
    if zoneN == 0:
        return x, y
    elif zoneN == 1:
        return y, x
    elif zoneN == 2:
        return -y, x
    elif zoneN == 3:
        return -x, y
    elif zoneN == 4:
        return -x, -y
    elif zoneN == 5:
        return -y, -x
    elif zoneN == 6:
        return y, -x
    elif zoneN == 7:
        return x, -y

def find_the_zone(x, y):
    if x >= 0:
        if y >= 0:
            if x >= y:
                return 0
            else:
                return 1
        else:
            if x >= -y:
                return 7
            else:
                return 6
    else:
        if y >= 0:
            if -x >= y:
                return 3
            else:
                return 2
        else:
            if x <= y:
                return 4
            else:
                return 5
#--------------------------------
def midpoint_algo(x1, y1, x2, y2, zoneN):
    dx = abs(x2 - x1)
    dy = abs(y2 - y1)
    sx = 1 if x1 < x2 else -1
    sy = 1 if y1 < y2 else -1
    temp = dx - dy
    while True:
        xo, yo = zone_convert(x1, y1, zoneN)
        draw_points(xo, yo)
        if x1 == x2 and y1 == y2:
            break

        e2 = 2 * temp
        if e2 > -dy:
            temp -= dy
            x1 += sx
        if e2 < dx:
            temp += dx
            y1 += sy

def draw_lines(x1, y1, x2, y2):
   dxo = x2 - x1
   dyo = y2 - y1
   if dyo == 0:
       if dxo > 0:
           zoneN = 7
       else:
           zoneN = 2

   if dxo == 0:
       if dyo > 0:
           zoneN = 1
       else:
           zoneN = 5
   else:
       zoneN = find_the_zone(dxo, dyo)
   x1i, y1i = zone_convert(x1, y1, zoneN)
   x2i, y2i = zone_convert(x2, y2, zoneN)
   midpoint_algo(x1i, y1i, x2i, y2i, zoneN)
#----------------------------------------------------------------------
def iteration():
   glViewport(0, 0, 500, 710)
   glMatrixMode(GL_PROJECTION)
   glLoadIdentity()
   glOrtho(0.0, 500.0, 0.0, 720.0, 0.0, 1.0)
   glMatrixMode(GL_MODELVIEW)
   glLoadIdentity()
#-----------------------------------------
def control(cx, cy) :
    global game, scores, dimension_y, spd_r, dimension_x, color,
r_x,connect, dimension_x, bullet, misses, damaged,total_lives
    if 650 <= cy <= 700 :
        if 450 <= cx <= 500 :
            print('bye. Come back soon') #if we press cross button the game
will be over
            print('final score:', scores)
            glutLeaveMainLoop()
        if 230 <= cx <= 270:
            print('Pause') # if we want to pause the game
            if game == '':
                game = 'pause'
            elif game == 'pause':
                game = ''
        if 5 <= cx <= 50:
            print('restart') # if we want to restart the game
            total_lives = 3
            game = ''
            spd_r = 1
            damaged = 0
            scores = 0
            dimension_y = 640
            r_x = 250
            dimension_x = random.randint(10, 480)
            misses = 0
            color= [random.randint(5, 10 ) / 10 for _ in range(3)]
            connect = []
            for z in range(5):
                connect.append([random.randint(15, 26), random.randint(10,
480), 615])
            bullet = []
#----------------------------------------------
r_x = 250
def keyboardListener(key, x, y):
    global bullet, r_x
    if game != 'pause' and game != 'over':
        if key == b' ':
            bullet.append([7, r_x, 25])
            glutPostRedisplay()
        if key == b'd':
            if r_x <= 475:
                r_x += 10
        if key == b'a':
            if r_x >= 25:
                r_x -= 10

def mouseListener(button, state, x, y) :
    if button == GLUT_LEFT_BUTTON and state == GLUT_DOWN:
        control(x, 720 - y )
        glutPostRedisplay()

def specialKeyListener(key, x, y):
    global r_x
    if game != 'pause' and game != 'over':
        if key == GLUT_KEY_RIGHT:
            if r_x <= 475:
                r_x += 10
        if key == GLUT_KEY_LEFT:
            if r_x >= 25:
                r_x -= 10
    glutPostRedisplay()
#-------------------
dimension_y= 640
spd_r = 1
```

```
scores = 0
def animate():
    global spd_r, r_x, game, connect, bullet, total_lives
    for g in range(len(connect)):
        dia_x = connect[g][1]
        dia_y = connect[g][2]
        if game == '':
            if dia_y > 45:
                if spd_r % 7 == 0:
                    dia_y -= 1
            else:
                if game != 'over':
                    dia_x = 800
                    connect.pop(g)
                    connect.append([random.randint(15, 26), random.randint(10,
480), 615])
                spd_r = 1
                dia_x = random.randint(10, 480)

            spd_r += 1
        connect[g] = [connect[g][0], dia_x, dia_y]
        if game == 'pause':
            spd_r = spd_r
    for h in range(len(bullet)):
        if game != 'pause':
            bullet[h][2] += 1

    glutPostRedisplay()
#-------------------------------------
def scoree():
    global scores, game,color,bullet, connect, no_of_life_lost , missed,
damaged

    for p in connect:
        if (p[2] - p[0]) <= 45:
            if (p[1] - p[0]) <= (r_x - 20) <= (p[1] + p[0]) or (p[1] - p[0]) <=
(r_x + 20) <= (p[1] + p[0]):
                damaged =damaged+ 1
            else:
                no_of_life_lost  = True
                connect.remove(p)
                connect.append([random.randint(15, 26), random.randint(10,
480), 615])

        for q in bullet:
            if q[2] > 700:
                missed = True
                bullet.remove(q)
                print('missed')

            if (p[1] - p[0]) < (q[1] - q[0]) < (p[1] + p[0]) or (p[1] - p[0]) <
(q[1] + q[0]) < (p[1] + p[0]):
                if (p[2] - p[0]) <= (q[2] + q [0]):
                    scores =scores + 1
                    print('Your Score:', scores)
                    bullet.remove(q)
                    connect.remove(p)
                    connect.append([random.randint(15, 26),
random.randint(10, 480), 615])
dimension_x = random.randint(10, 480)
```

```
connect = []
color = [random.randint(5, 10) / 10 for _ in range(3)]
for f in range(5):
  connect.append([random.randint(15, 26), random.randint(10, 480),
615])
#------------------------------
bullet= []
def showScreen():
  global dimension_x, color, r_x, total_lives, no_of_life_lost , game,
misses, missed,damaged,connect
  glClearColor(0, 0, 0, 1.0)
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
  glLoadIdentity()
  iteration()
  if game == 'pause':  #to draw pause button after pressed
    draw_lines(240, 700, 240, 647)
    draw_lines(240, 647, 275, 675)
    draw_lines(240, 700, 275, 675)

  glColor3f(1,1,0) #to draw pause button before pressed
  if game == '' or game == 'over':
    draw_lines(240, 702, 240, 657)
    draw_lines(260, 702, 260, 657)

  glColor3f(1, 0, 0) #to draw cross button
  draw_lines(437, 658, 490, 708)
  draw_lines(437, 708, 490, 658)

  glColor3f(0, 1, 1)  #to draw restart button
  draw_lines(7, 677, 60, 677)
```

```
    draw_lines(7, 677, 40, 700)
    draw_lines(7, 677, 40, 657)
#--------------------
    if game != 'over':
        glColor3f(1, 1, 1)
    else:
        glColor3f(1, 0, 0)
    midpoint_circle_draw(20, r_x, 25)
    glColor3f(0.5, 0, 0.5)
    for s in bullet:
        midpoint_circle_draw(s[0], s[1], s[2])
    glColor3f(color[0], color[1], color[2])
    for t in connect:
        rad = t[0]
        x_axis = t[1]
        y_axis = t[2]
        midpoint_circle_draw(rad, x_axis, y_axis)

    scoree()
    if missed == True:
        misses += 1
        print('Missed:', misses)
        missed = False
        if misses == 3:
            game = 'over'
            print("Missed more than 2 times. game over")
            print('final score:', scores)

    if no_of_life_lost == True:
        total_lives -= 1
        print('Lives:', total_lives)
        no_of_life_lost = False
        if total_lives == 0:
            game = 'over'
            print("the planet is destroyed.game over")
            print('final score:', scores)

    if damaged == 1:
        print("you are fully damaged.game over")
        print('final score:',scores)
        game = 'over'

    if game != 'over':
        glutSwapBuffers()
    if game == 'over':
        connect = []
        damaged = 0
        glutSwapBuffers()
#------------------------
game = ""
glutInit()
glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE |
GLUT_RGB)
glutInitWindowSize(500,710)
glutInitWindowPosition(700, 0)
wind = glutCreateWindow(b"Assignment03")
glutDisplayFunc(showScreen)
glutIdleFunc(animate)
glutSpecialFunc(specialKeyListener)
glutKeyboardFunc(keyboardListener)
```

```
glutMouseFunc(mouseListener)
if game != 'pause':
   glutMainLoop()
```

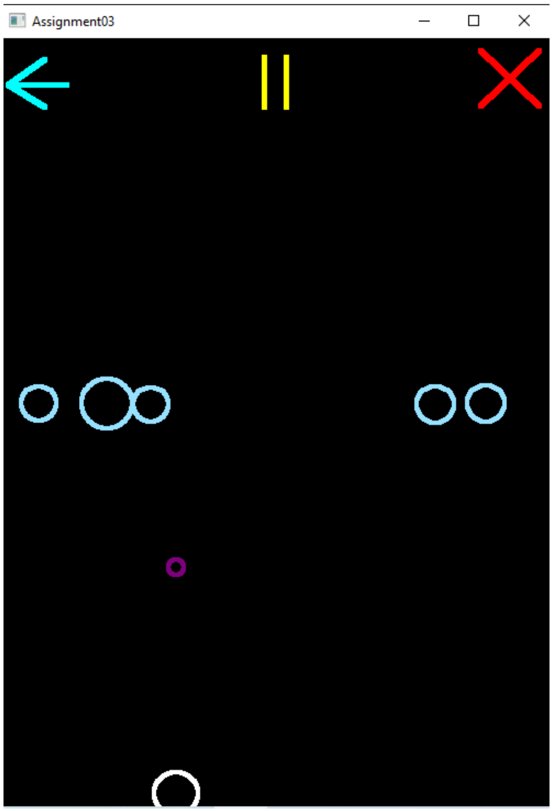

FIG: Starting the game

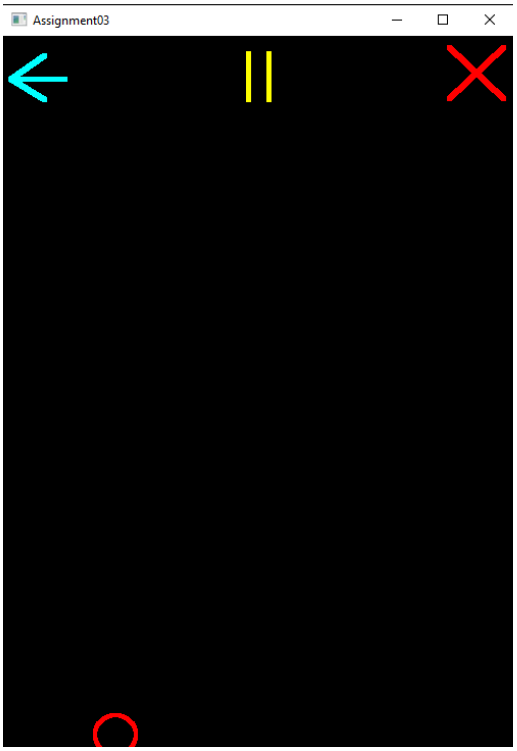

FIG: Game over

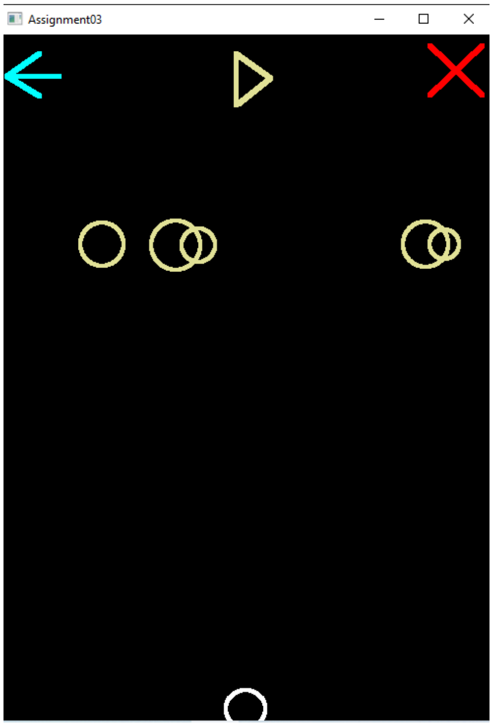

FIG: game is paused

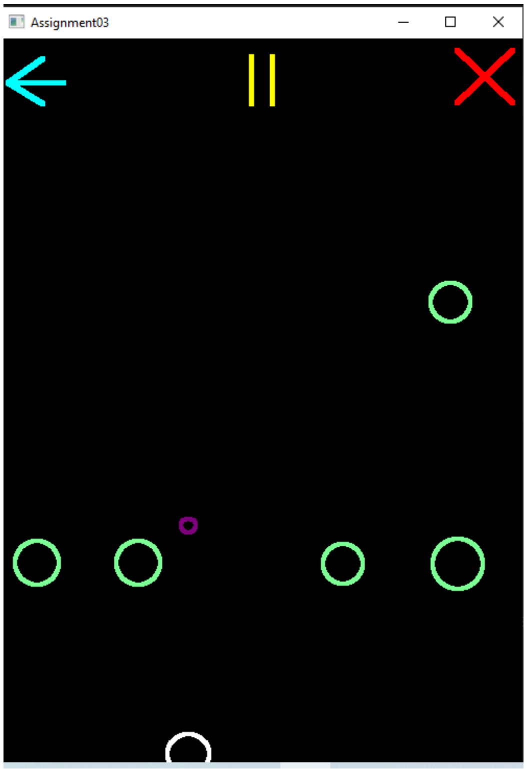

FIG: misfire