BRAC
UNIVERSITY

Inspiring Excellence

CSE423 : Computer Graphics

Project Title : Eating The Fish Game

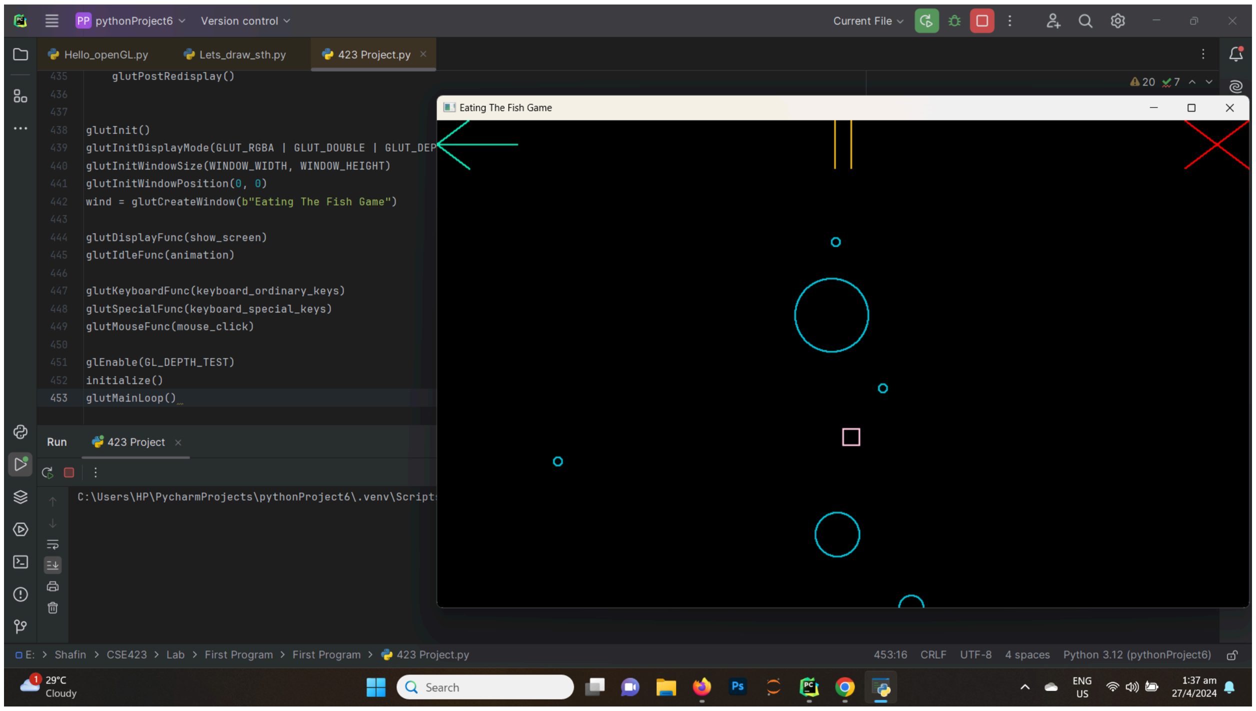
Group No : 04, CSE423, Lab Section : 2, Spring 2024	
ID	**Name**
21141058	Mohammad Omar Raihan
21201446	Nafisa Rahman
21201193	Smita Biswas

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

WINDOW_WIDTH = 1000
WINDOW_HEIGHT = 600

currentTime = 0
previousTime = 0
elapsedTime = 0
score = 0
lvl2 = 5
lvl3 = 10
status = 'playing'


def initialize():
	global currentTime
	glViewport(0, 0, WINDOW_WIDTH, WINDOW_HEIGHT)
	glMatrixMode(GL_PROJECTION)
	glLoadIdentity()
	glOrtho(0.0, WINDOW_WIDTH, 0.0, WINDOW_HEIGHT, 0.0, 1.0)
	glMatrixMode(GL_MODELVIEW)
	glLoadIdentity()
	currentTime = glutGet(GLUT_ELAPSED_TIME)


# midpoint line
def drawPixel(x, y):
	glPointSize(2)
	glBegin(GL_POINTS)
	glVertex2f(x, y)
	glEnd()
	glFlush()


def draw8way(x, y, slope):
        if slope == 0:
        drawPixel(x, y)
        elif slope == 1:
        drawPixel(y, x)
        elif slope == 2:
        drawPixel(-y, x)
        elif slope == 3:
        drawPixel(-x, y)
        elif slope == 4:
        drawPixel(-x, -y)
        elif slope == 5:
        drawPixel(-y, -x)
        elif slope == 6:
        drawPixel(y, -x)
        elif slope == 7:
        drawPixel(x, -y)


def MidpointLine(x0, y0, x1, y1, slope):
        dx = x1 - x0
        dy = y1 - y0
        delE = 2 * dy
        delNE = 2 * (dy - dx)
        d = 2 * dy - dx
        x = x0
        y = y0
        while x < x1:
        draw8way(x, y, slope)
        if d < 0:
        d += delE
        x += 1
        else:
        d += delNE
        x += 1
        y += 1


def drawLine(x0, y0, x1, y1):
        dx = x1 - x0
        dy = y1 - y0
        if abs(dx) >= abs(dy):  # zone 0, 3, 4, and 7
        if dx >= 0:
        if dy >= 0:
                MidpointLine(x0, y0, x1, y1, 0)
        else:
                MidpointLine(x0, y0, -x1, -y1, 7)

        else:
        if dy >= 0:
                MidpointLine(-x0, y0, -x1, y1, 3)
        else:
                MidpointLine(-x0, -y0, -x1, -y1, 4)
        else:  # zone 1, 2, 5, and 6
        if dx >= 0:
        if dy >= 0:
                MidpointLine(y0, x0, y1, x1, 1)
        else:
                MidpointLine(-y0, x0, -y1, x1, 6)

        else:
        if dy >= 0:
                MidpointLine(y0, -x0, y1, -x1, 2)
        else:
                MidpointLine(-y0, -x0, -y1, -x1, 5)


# midpoint circle
def circ_points(x, y, cx, cy):
        glVertex2f(x + cx, y + cy)
        glVertex2f(y + cx, x + cy)

        glVertex2f(y + cx, -x + cy)
        glVertex2f(x + cx, -y + cy)

        glVertex2f(-x + cx, -y + cy)
        glVertex2f(-y + cx, -x + cy)

        glVertex2f(-y + cx, x + cy)
        glVertex2f(-x + cx, y + cy)
```

```
def mid_circle(cx, cy, radius):
        d = 1 - radius
        x = 0
        y = radius

        while x < y:
        if d < 0:
        d = d + 2 * x + 3
        else:
        d = d + 2 * x - 2 * y + 5
        y = y - 1
        x = x + 1
        circ_points(x, y, cx, cy)


# top screen buttons
def back():
        # button width=100 height = 60
        glColor3f(0.0465, 0.930, 0.724)
        drawLine(0, 570, 100, 570)
        drawLine(0, 570, 40, 600)
        drawLine(40, 540, 0, 570)


def cross():
        # button width=80 height = 60
        glColor3f(1, 0, 0)
        drawLine(1000, 600, 920, 540)
        drawLine(1000, 540, 920, 600)


def pause():
        # button width=20 height = 60
        glColor3f(0.980, 0.765, 0.0588)
        drawLine(510, 600, 510, 540)
        drawLine(490, 600, 490, 540)


def play():
        # button width=60 height = 60
        glColor3f(0.980, 0.765, 0.0588)
        drawLine(540, 570, 480, 600)
        drawLine(540, 570, 480, 540)
        drawLine(480, 600, 480, 540)


coll = False
# draw characters
self_x = 500  # bottom left coordinate of the box
self_y = 200
self_w = 20  # 20, 40, 65
self_h = 20


def draw_self():
        global coll, score, self_w, self_h, self_y, self_x, lvl2, lvl3
        glColor3f(1.0, 0.8, 0.9)

        if score > lvl3:  # for box size level 3, w & h becomes 65
        self_h = 65
        self_w = 65
        elif score > lvl2:  # for box size level 2, w & h becomes 40
        self_h = 40
        self_w = 40

        drawLine(self_x, self_y, self_x + self_w, self_y)
        drawLine(self_x + self_w, self_y, self_x + self_w, self_y + self_h)
        drawLine(self_x + self_w, self_y + self_h, self_x, self_y + self_h)
        drawLine(self_x, self_y + self_h, self_x, self_y)


cre_r = []  # radius
cre_cx = []  # center x
cre_cy = []  # center y
cre_start = []  # start value of each creature (needed to determine direction in animation function)
cre_dist = 500  # min dist between two characters on the screen on the same y-line


def draw_creatures():
        global cre_r, cre_cx, cre_cy, cre_start, cre_dist, score, lvl2, lvl3

        if score > lvl3:
        cre_dist = 200
        choice = random.choice(['t', 't', 'f'])
        elif score > lvl2:
        cre_dist = 300
        choice = random.choice(['t', 't', 't', 'f'])
        else:
        choice = random.choice(
        ['t', 't', 't', 't', 'f'])  # choice condition added to randomize/delay when creatures are
appearing

        if choice == 'f':
        cre_radlist = [5, 15, 27, 45]
        cre_cyrandom = random.randrange(0, 495, 90)
        cre_cxrandom = random.choice([-50, 1050])
        if len(cre_r) != 0:
        if cre_cyrandom in cre_cy:  # condition to check if a creature already exists in that y-line
of the screen
        idx = max(i for i, elem in enumerate(cre_cy) if
        elem == cre_cyrandom)  # to get the index/position of the most recent creature on
the y-line of the screen
        if (cre_cx[idx] >= cre_dist and cre_start[idx] == -50) or (cre_cx[idx] <= cre_dist
and cre_start[
        idx] == 1050):  # condition to check if the creature that already exists, is more
than 500 pixels away so that another one can be created
        cre_r.append(random.choice(cre_radlist))
        cre_cx.append(cre_cxrandom)
        cre_cy.append(cre_cyrandom)
        cre_start.append(cre_cxrandom)
        else:
        pass
        else:  # if creature doesnt exist already, a new one is generated on that y-line
        cre_r.append(random.choice(cre_radlist))
        cre_cx.append(cre_cxrandom)
        cre_cy.append(cre_cyrandom)
        cre_start.append(cre_cxrandom)
        else:  # at the very beginning when there are 0 creatures on screen, new ones are
generated
        cre_r.append(random.choice(cre_radlist))
        cre_cx.append(cre_cxrandom)
        cre_cy.append(cre_cyrandom)
        cre_start.append(cre_cxrandom)

        glColor3f(0, 0.8, 0.9)
        glPointSize(2)
        glBegin(GL_POINTS)

        if len(cre_r) != 0:
        for i in range(len(cre_r)):
        mid_circle(cre_cx[i], cre_cy[i], cre_r[i])  # this function draws the creatures

        glEnd()


def check_collision():
        global cre_r, cre_cx, cre_cy, cre_start, self_x, self_y, self_w, self_h, coll, score, lvl2, lvl3
        right_self = self_x + self_w  # 4 sides of the box
        left_self = self_x
        top_self = self_y + self_h
        bottom_self = self_y

        cre_remove = []

        # circ
        for i in range(len(cre_r)):
        cre_left = cre_cx[i] - cre_r[i]
        cre_right = cre_cx[i] + cre_r[i]
        cre_top = cre_cy[i] + cre_r[i]
        cre_bottom = cre_cy[i] - cre_r[i]
        if cre_right >= left_self and cre_left <= right_self and cre_top >= bottom_self and
cre_bottom <= top_self:
        coll = True
        cre_area = 3.1416 * (cre_r[i]) ** 2
        area_self = self_w * self_h

        if area_self >= cre_area:
                # box size increment and removing the circle from the list/screen
                cre_remove.append(i)
                score += 1
                print("Current score:", score)
                if score == lvl2 + 1:
                print("You have levelled up to level 2!")
                if score == lvl3 + 1:
                print("You have levelled up to level 3!")
        else:
                # the circle will eat the box/ window or session closes
                print("Goodbye :(")
                print("Final Score:", score)
                glutDestroyWindow(wind)

        for i in range(len(cre_remove)):
        # removing eaten circles from screen
        cre_r.pop(cre_remove[i])
        cre_cx.pop(cre_remove[i])
        cre_cy.pop(cre_remove[i])
        cre_start.pop(cre_remove[i])


def show_screen():
        global previousTime, currentTime, elapsedTime
        previousTime = currentTime
        currentTime = glutGet(GLUT_ELAPSED_TIME)
        elapsedTime = currentTime - previousTime
        # this function should contain the logic for drawing objects
        # DO NOT do game logic here (e.g. object movement, collision detection, sink detection
etc.)
        # clear the screen
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        # draw stuffs here
        draw_creatures()
        draw_self()
        back()
        cross()
        if status == "playing":
        pause()
        else:
```

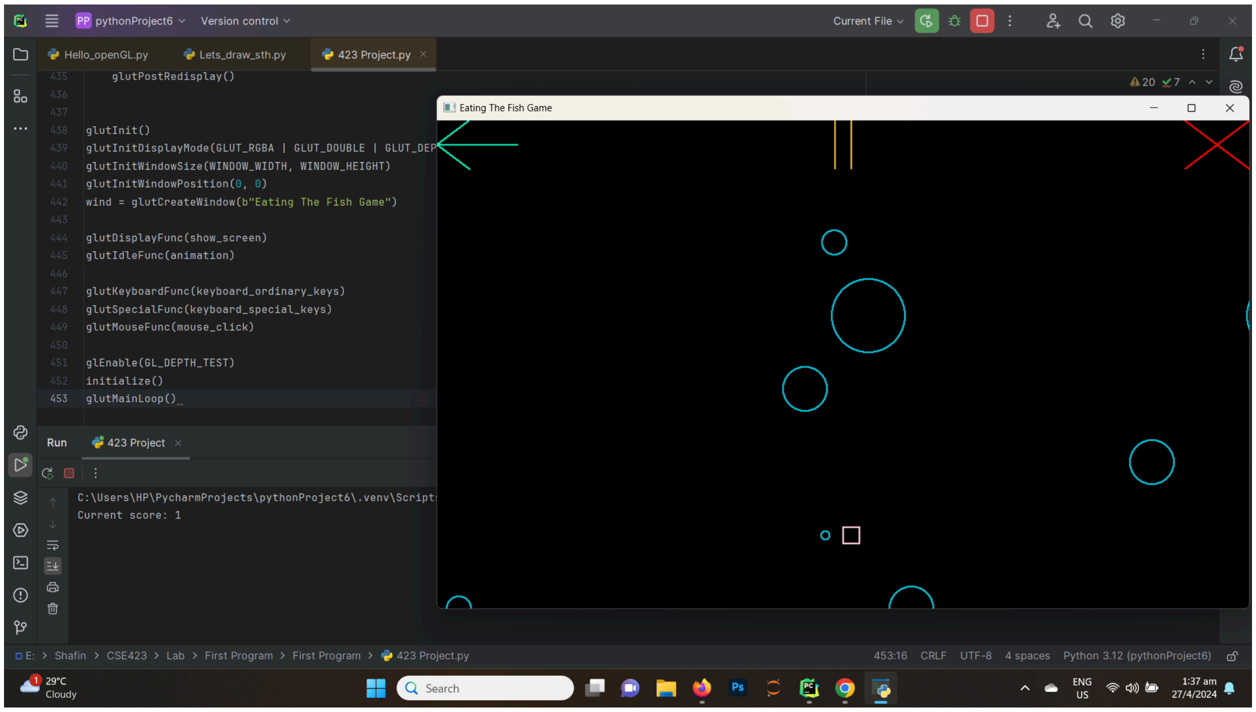
```
        play()
        animation()
        # do not forget to call glutSwapBuffers() at the end of the function
        glutSwapBuffers()


def keyboard_ordinary_keys(key, _, __):
        # check against alphanumeric keys here (e.g. A..Z, 0..9, spacebar, punctuations)
        # must cast characters to binary when comparing (e.g. key == b'q')
        glutPostRedisplay()


box_speed = 20


def keyboard_special_keys(key, _, __):  # function to control movement of self box
        global self_x, self_y, self_w, self_h

        if key == GLUT_KEY_UP:
        if self_y + self_h < 510:
        self_y += box_speed
        elif key == GLUT_KEY_DOWN:
        if self_y > 0:
        self_y -= box_speed
        elif key == GLUT_KEY_LEFT:
        if self_x > 0:
        self_x -= box_speed
        elif key == GLUT_KEY_RIGHT:
        if self_x + self_w < 990:
        self_x += box_speed

        glutPostRedisplay()


prev_cre_speed = 0
prev_box_speed = 0


def mouse_click(button, state, x, y):
        global status, cre_speed, box_speed, prev_cre_speed, prev_box_speed, score, self_x,
self_y, self_w, self_h, cre_r, cre_cx, cre_cy, cre_start
        mx, my = x, WINDOW_HEIGHT - y

        # do your click detection here using button, state, mx, my
        if mx > 480 and mx < 540 and my < 600 and my > 540:
        # play()
        if button == GLUT_LEFT_BUTTON:
        if (state == GLUT_DOWN):
            if status == "playing":
            status = "paused"
            prev_cre_speed = cre_speed
            prev_box_speed = box_speed
            cre_speed = 0
            box_speed = 0
            print("Game paused!")
            else:
            box_speed = prev_box_speed
            cre_speed = prev_cre_speed
            status = "playing"

        if mx > 0 and mx < 100 and my < 600 and my > 540:
        if button == GLUT_LEFT_BUTTON:
        if state == GLUT_DOWN:
            # reset all variables
            score = 0
            self_x = 500
            self_y = 200
            self_w = 20
            self_h = 20
            cre_r = []  # radius
            cre_cx = []  # center x
            cre_cy = []  # center y
            cre_start = []  # start value of each creature (needed to determine direction in
animation function)
            box_speed = 20
            cre_speed = 30
            prev_cre_speed = 0
            prev_box_speed = 0
            status = "playing"
```

```
			print('Starting Over')

		glutPostRedisplay()

		if mx > 920 and mx < 1000 and my < 600 and my > 540:
		# cross()
		if button == GLUT_LEFT_BUTTON:
		if state == GLUT_DOWN:
			print("Goodbye :(")
			print("Final Score: ", score)
			glutDestroyWindow(wind)

		glutPostRedisplay()


cre_speed = 30


def animation():
		global cre_cx, elapsedTime, box_speed, cre_start, cre_speed, score, lvl2, lvl3
		# print(cre_cx)   #print this to check how list of creatures updates overtime

		if cre_cx != []:
		for i in range(len(cre_cx)):
		if cre_cx[0] < -50 or cre_cx[0] > 1050:  # condition to remove creatures depending on
pixel position
			cre_cx.pop(0)
			cre_cy.pop(0)
			cre_r.pop(0)
			cre_start.pop(0)

		if score > lvl3 and status == "playing":
		cre_speed = 40
		elif score > lvl2 and status == "playing":
		cre_speed = 35

		for i in range(len(cre_cx)):
		if cre_cx[i] >= -50 and cre_start[
                    i] == 1050:  # object moves leftward if it started at right end of the screen and
vice versa
                    cre_cx[i] -= cre_speed * elapsedTime / 1e3
          elif cre_cx[i] <= 1050 and cre_start[i] == -50:
                    cre_cx[i] += cre_speed * elapsedTime / 1e3

          check_collision()

          # don't forget to call glutPostRedisplay()
          # otherwise your animation will be stuck
          glutPostRedisplay()


glutInit()
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH)
glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Eating The Fish Game")

glutDisplayFunc(show_screen)
glutIdleFunc(animation)

glutKeyboardFunc(keyboard_ordinary_keys)
glutSpecialFunc(keyboard_special_keys)
glutMouseFunc(mouse_click)

glEnable(GL_DEPTH_TEST)
initialize()
glutMainLoop()s
```



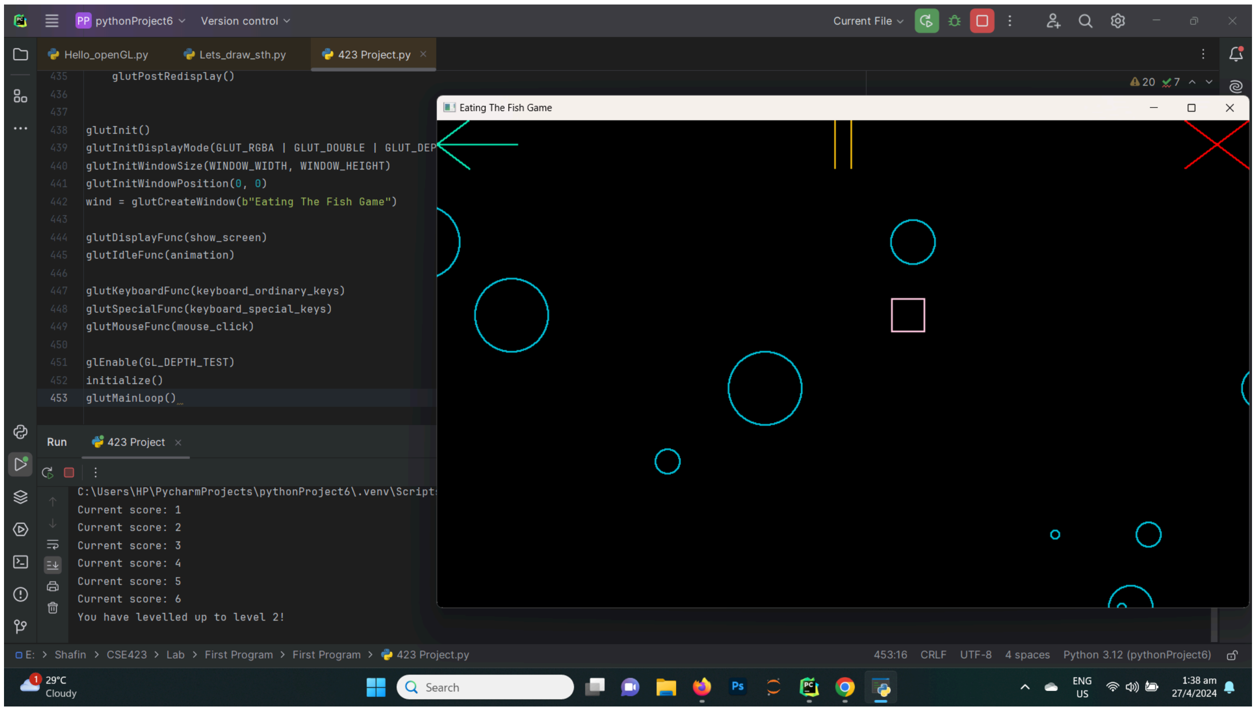
```
    glutPostRedisplay()


glutInit()
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEP
glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Eating The Fish Game")

glutDisplayFunc(show_screen)
glutIdleFunc(animation)

glutKeyboardFunc(keyboard_ordinary_keys)
glutSpecialFunc(keyboard_special_keys)
glutMouseFunc(mouse_click)

glEnable(GL_DEPTH_TEST)
initialize()
glutMainLoop()
```

```
C:\Users\HP\PycharmProjects\pythonProject6\.venv\Script
```



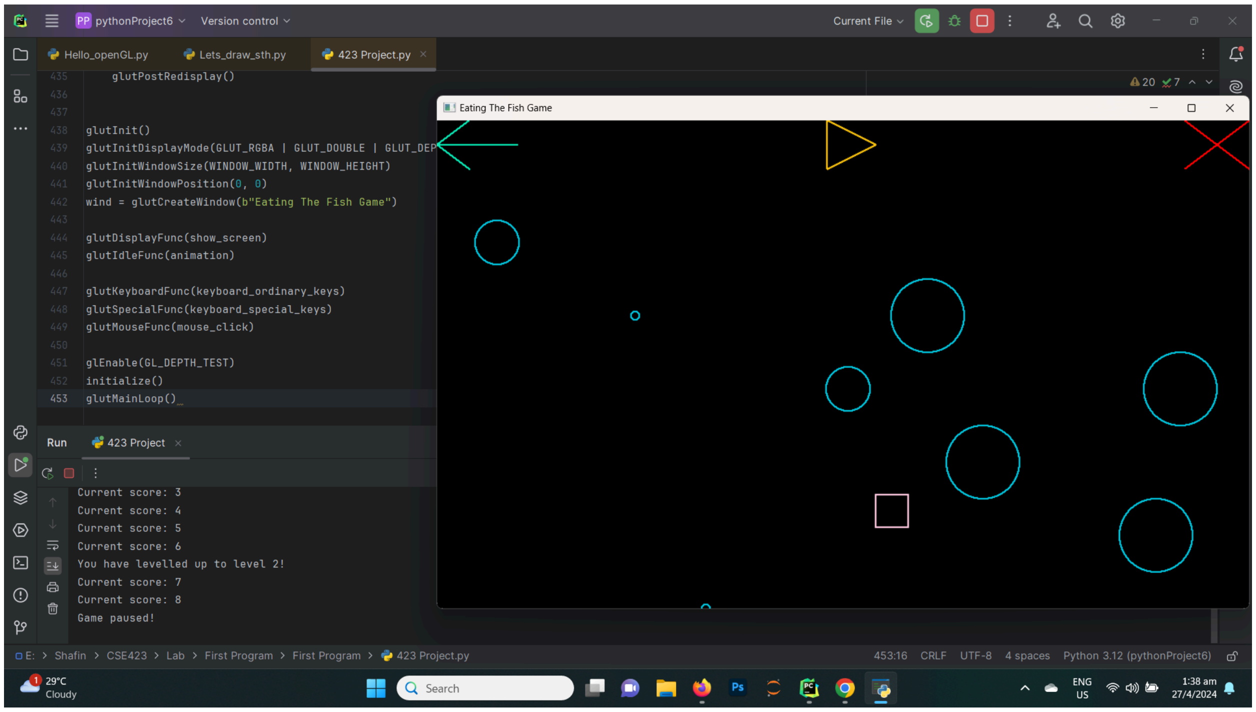
```
    glutPostRedisplay()


glutInit()
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEP
glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Eating The Fish Game")

glutDisplayFunc(show_screen)
glutIdleFunc(animation)

glutKeyboardFunc(keyboard_ordinary_keys)
glutSpecialFunc(keyboard_special_keys)
glutMouseFunc(mouse_click)

glEnable(GL_DEPTH_TEST)
initialize()
glutMainLoop()
```

```
C:\Users\HP\PycharmProjects\pythonProject6\.venv\Script
Current score: 1
```



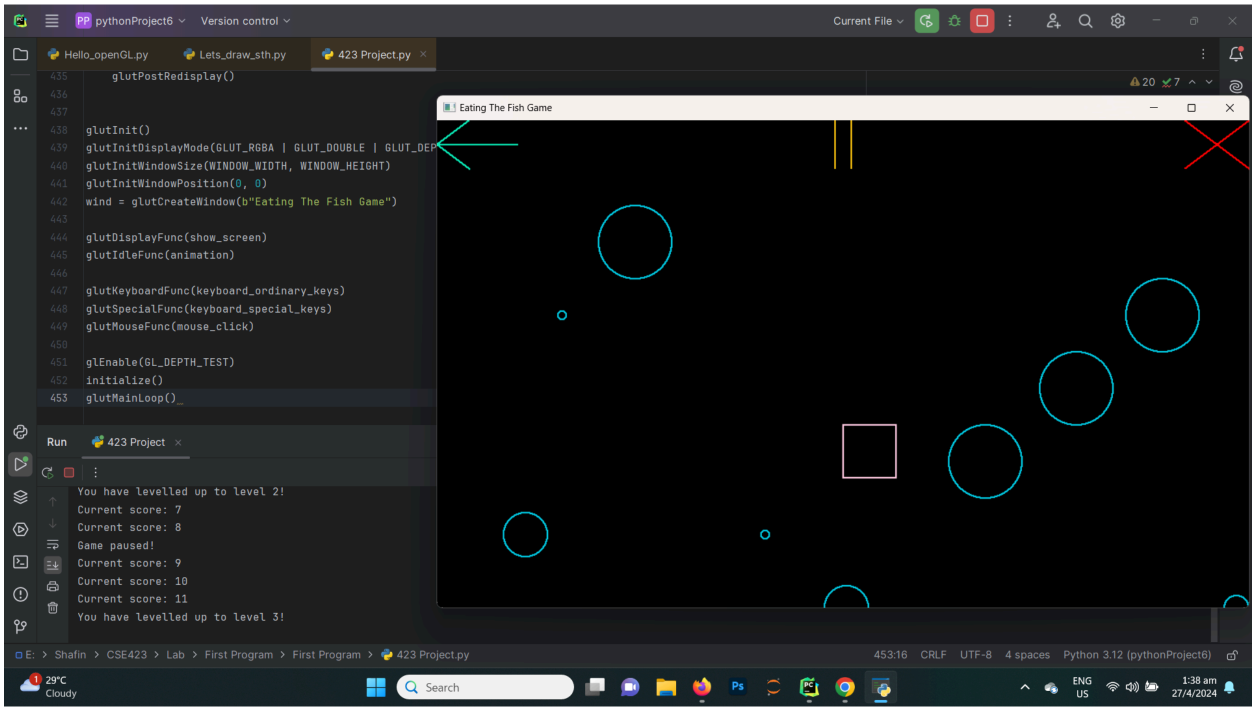
```
    glutPostRedisplay()


glutInit()
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEP
glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Eating The Fish Game")

glutDisplayFunc(show_screen)
glutIdleFunc(animation)

glutKeyboardFunc(keyboard_ordinary_keys)
glutSpecialFunc(keyboard_special_keys)
glutMouseFunc(mouse_click)

glEnable(GL_DEPTH_TEST)
initialize()
glutMainLoop()
```

```
C:\Users\HP\PycharmProjects\pythonProject6\.venv\Script
Current score: 1
Current score: 2
Current score: 3
Current score: 4
Current score: 5
Current score: 6
You have levelled up to level 2!
```



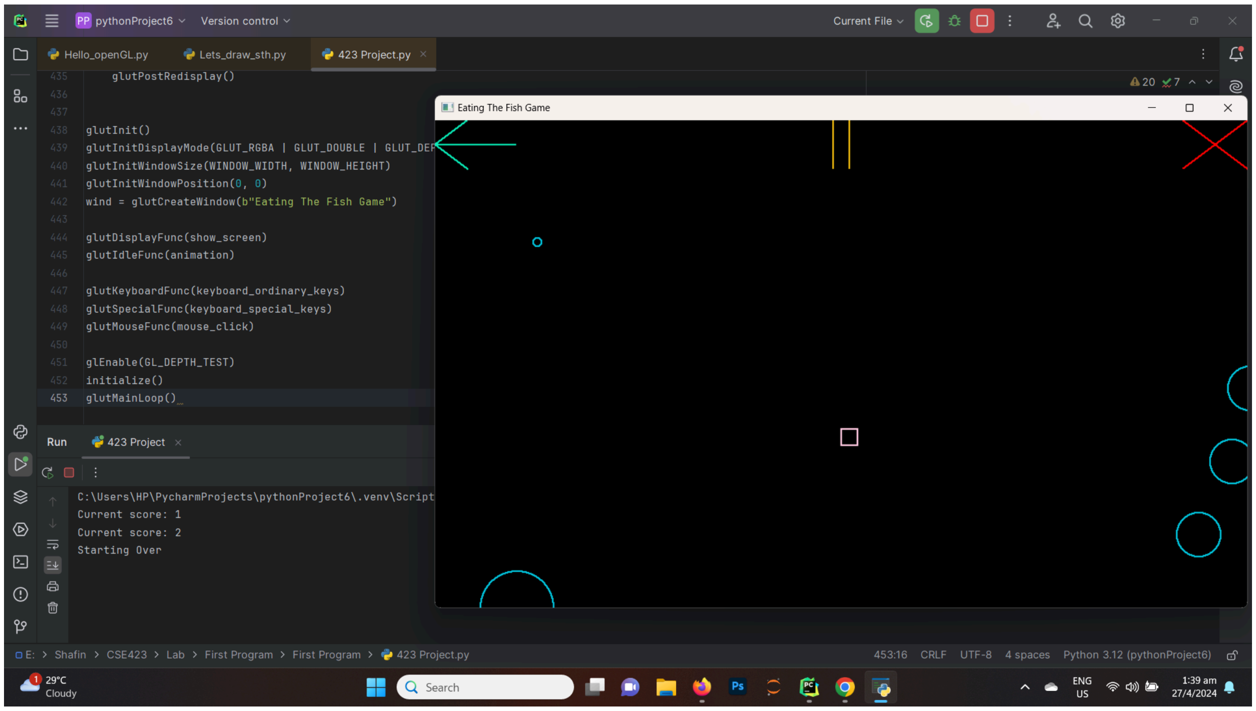
```
    glutPostRedisplay()


glutInit()
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEP
glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Eating The Fish Game")

glutDisplayFunc(show_screen)
glutIdleFunc(animation)

glutKeyboardFunc(keyboard_ordinary_keys)
glutSpecialFunc(keyboard_special_keys)
glutMouseFunc(mouse_click)

glEnable(GL_DEPTH_TEST)
initialize()
glutMainLoop()
```

```
Current score: 3
Current score: 4
Current score: 5
Current score: 6
You have levelled up to level 2!
Current score: 7
Current score: 8
Game paused!
```



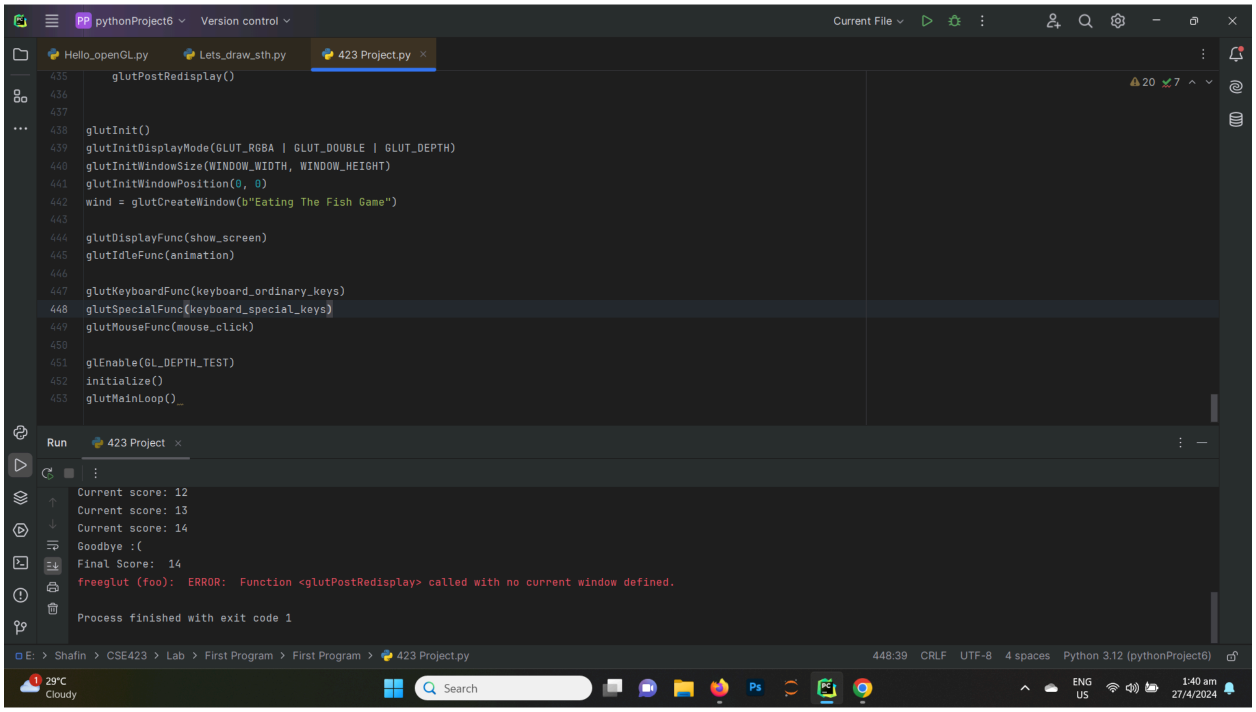
```
    glutPostRedisplay()


glutInit()
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEP
glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Eating The Fish Game")

glutDisplayFunc(show_screen)
glutIdleFunc(animation)

glutKeyboardFunc(keyboard_ordinary_keys)
glutSpecialFunc(keyboard_special_keys)
glutMouseFunc(mouse_click)

glEnable(GL_DEPTH_TEST)
initialize()
glutMainLoop()
```

```
You have levelled up to level 2!
Current score: 7
Current score: 8
Game paused!
Current score: 9
Current score: 10
Current score: 11
You have levelled up to level 3!
```



```
    glutPostRedisplay()


glutInit()
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEP
glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Eating The Fish Game")

glutDisplayFunc(show_screen)
glutIdleFunc(animation)

glutKeyboardFunc(keyboard_ordinary_keys)
glutSpecialFunc(keyboard_special_keys)
glutMouseFunc(mouse_click)

glEnable(GL_DEPTH_TEST)
initialize()
glutMainLoop()
```

```
C:\Users\HP\PycharmProjects\pythonProject6\.venv\Script
Current score: 1
Current score: 2
Starting Over
```



```
    glutPostRedisplay()


glutInit()
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH)
glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Eating The Fish Game")

glutDisplayFunc(show_screen)
glutIdleFunc(animation)

glutKeyboardFunc(keyboard_ordinary_keys)
glutSpecialFunc(keyboard_special_keys)
glutMouseFunc(mouse_click)

glEnable(GL_DEPTH_TEST)
initialize()
glutMainLoop()
```

```
Current score: 12
Current score: 13
Current score: 14
Goodbye :(
Final Score:  14
freeglut (foo):  ERROR:  Function <glutPostRedisplay> called with no current window defined.

Process finished with exit code 1
```