Computer Graphics Lab 2

OpenGL Basic

1. Code this Python and answer:

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

pygame.init()

screen_width = 1000
screen_height = 800

screen = pygame.display.set_mode((screen_width, screen_height), DOUBLEBUF | OPENGL)
pygame.display.set_caption("OpenGL in Python")

def init_ortho():
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(0, 640, 0, 480)

done = False
init_ortho()

while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    glPointSize(5)

    glBegin(GL_POINTS)
    glVertex2i(100, 50)
    glVertex2i(600, 450)
    glEnd()

    pygame.display.flip()
    pygame.time.wait(100)

pygame.quit()
```

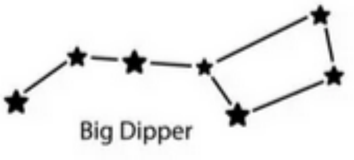
- Which dot is coordinate(100,50) and (600, 450) ?

- Explain the difference between the variables screen_width and screen_height, and the function gluOrtho2D. Specifically, discuss how each affects the rendering of graphics in an OpenGL context.

TA Checking:

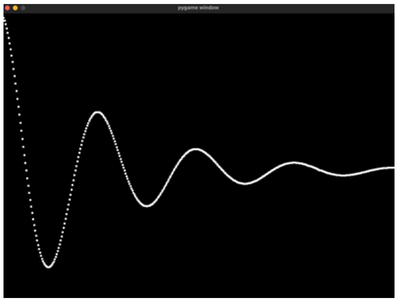
2. Make Python and OpenGL to display as below images (only dot, no line):

TA Check:



3. Plot graph from this function: $f(x) = e^{-x}\cos\left(2\Pi x\right)$

Expected result:



Hint code:

```
import math
import numpy as np
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

Your code here

def plot_graph():
    glBegin(GL_POINTS)
    for px in np.arange(0, 620, 0.01):
        py = math.exp(-px) * math.cos(2 * math.pi * px)
        glVertex2f(px, py)
    glEnd()

your code here
```

TA Check:

4. Create this Python code and observ the result:

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *
```

```
pygame.init()
screen_width = 1000
screen_height = 800
screen = pygame.display.set_mode((screen_width, screen_height), DOUBLEBUF | OPENGL)
pygame.display.set_caption("Mouse Click Coordinates")

def init_ortho():
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(0, 640, 0, 480)

def screen_to_opengl(x, y):
    """Convert screen coordinates to OpenGL coordinates"""
    return (x * 640 // screen_width,
            480 - (y * 480 // screen_height))  # Flip Y coordinate

done = False
init_ortho()

while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
        elif event.type == pygame.MOUSEBUTTONDOWN:
            if event.button == 1:  # Left mouse button
                # Get mouse position and convert to OpenGL coordinates
                mouse_x, mouse_y = pygame.mouse.get_pos()
                opengl_x, opengl_y = screen_to_opengl(mouse_x, mouse_y)
                print(f"Screen coordinates: ({mouse_x}, {mouse_y})")
                print(f"OpenGL coordinates: ({opengl_x}, {opengl_y})")
                print("---")

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

    pygame.display.flip()
    pygame.time.wait(10)

pygame.quit()
```

TA Check:

5. Base on code in 4. Create the code to plot dots at the mouse cursor position when the left mouse button is clicked.

TA Check: