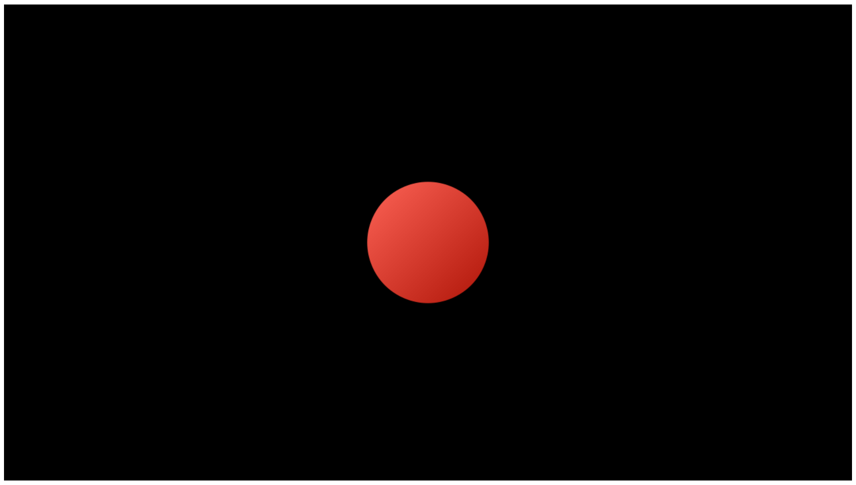
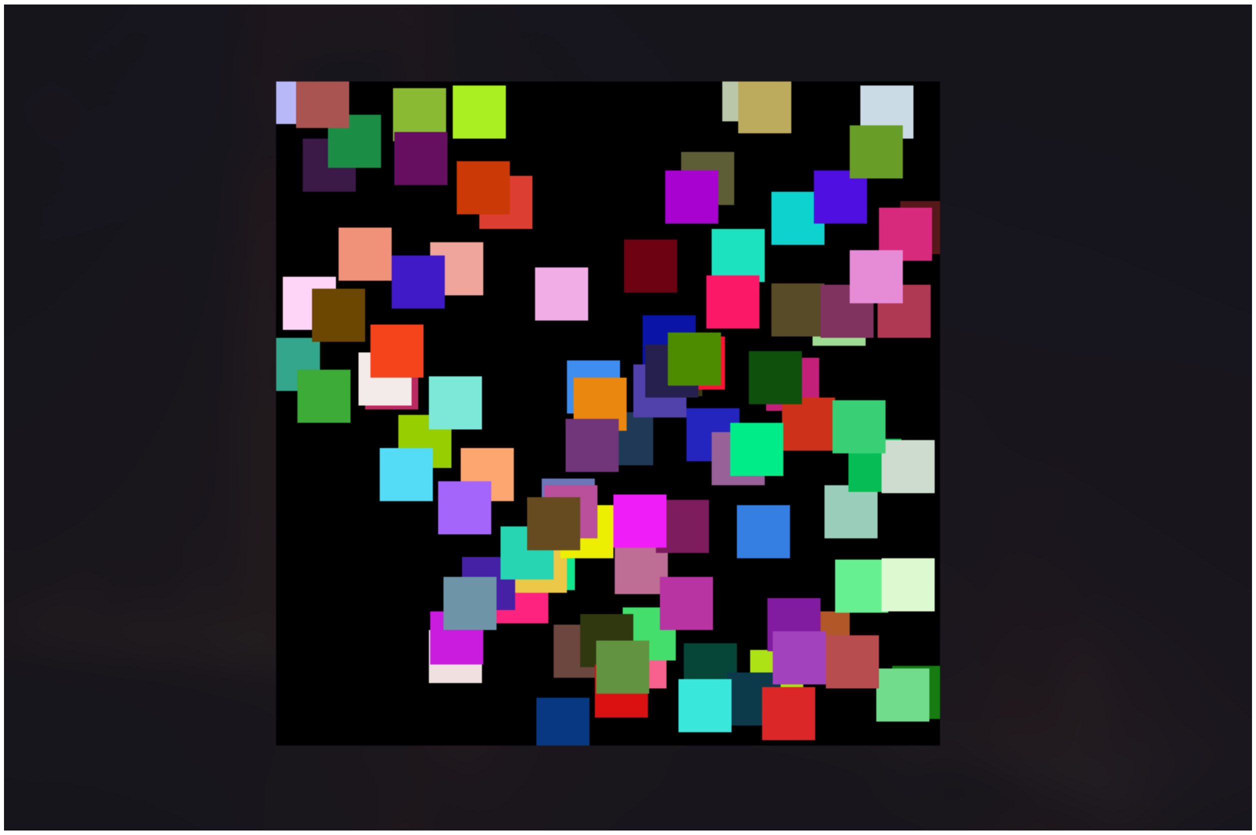


CSE423: Computer Graphics
Lab 01

Submitted by:
Asim Baidya
20301239

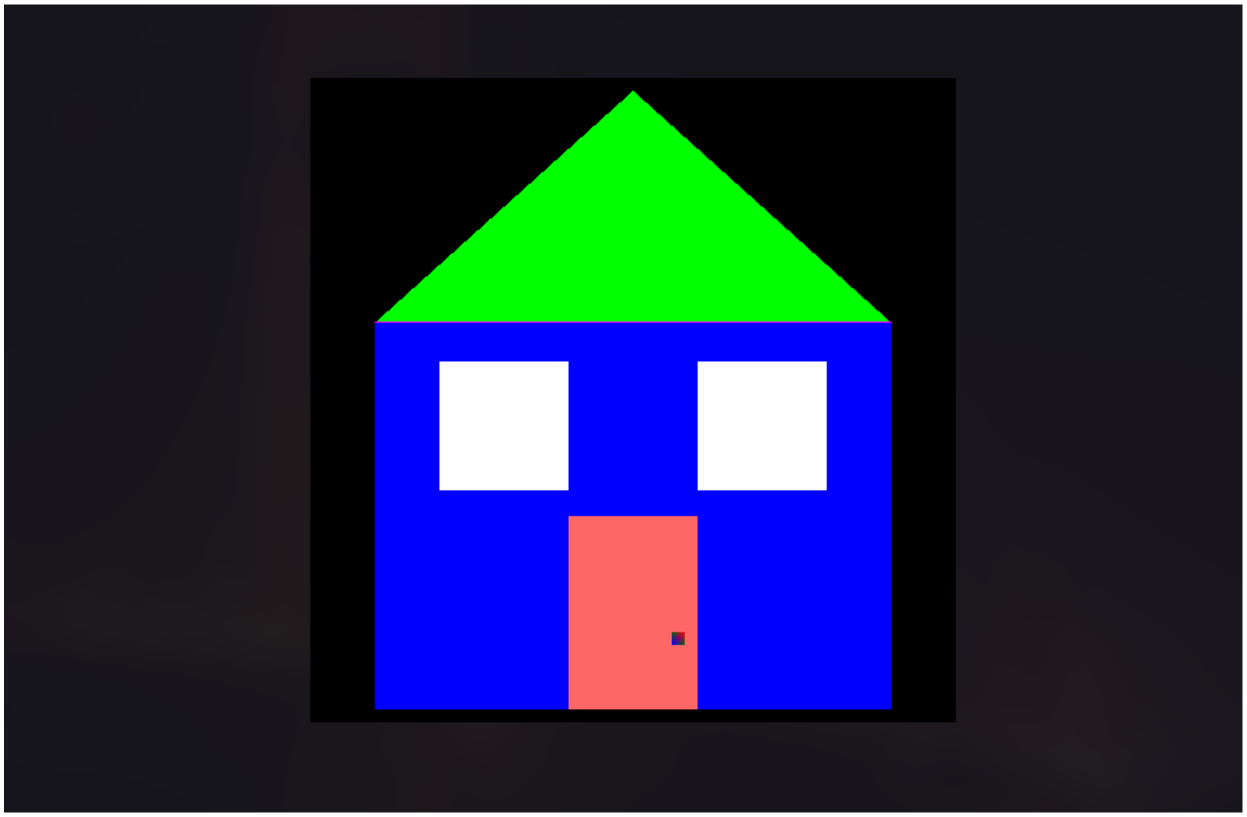
Task 01

```
# status = DONE

from random import randint, random, seed

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

POINT_COUNT = 50
HEIGHT = 500
WIDTH = 500


def random_point():
    global POINT_COUNT
    x, y = randint(0, WIDTH), randint(0, HEIGHT)
    r, g, b = random(), random(), random()
    glBegin(GL_POINTS)
    glColor3f(r, g, b)  # konokichur color set (RGB)
    glVertex2f(x, y)  # jekhane show korbe pixel
    glEnd()


def task(point_size=40):
    glPointSize(point_size)

    for _ in range(POINT_COUNT):
        random_point()


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 showScreen():
    ## error due to type issue.
    # glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glClear(GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    iterate()
    # glColor3f(1.0, 1.0, 0.0)  # konokichur color set (RGB)
    # call the draw methods here
    task()
    glutSwapBuffers()


def main():
    global HEIGHT, WIDTH
    glutInit()
    glutInitDisplayMode(GLUT_RGBA)
    glutInitWindowSize(WIDTH, HEIGHT)  # window size
    glutInitWindowPosition(0, 0)

    # (important for xmonad to ingnore tiling)
    wind = glutCreateWindow(b"opengl")

    glutDisplayFunc(showScreen)

    glutMainLoop()


if __name__ == "__main__":
    seed(42)
    main()
```



Task 02

```
# status = DONE

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

POINT_COUNT = 5000
HEIGHT = 500
WIDTH = 500


def draw_body(start_x, start_y, width, height):
    bottom_left = (start_x, start_y)
    bottom_right = (start_x + width, start_y)
    top_left = (start_x, start_y + height)
    top_right = (start_x + width, start_y + height)

    glBegin(GL_TRIANGLES)

    glColor3f(0, 0, 1)
    glVertex2f(*bottom_left)
    glVertex2f(*bottom_right)
    glVertex2f(*top_right)

    glVertex2f(*top_right)
    glVertex2f(*top_left)
    glVertex2f(*bottom_left)
    glEnd()

    return top_right, top_left


def top_cone(top_right, top_left, center):
    glBegin(GL_TRIANGLES)
    glColor3f(0, 1, 0)
    glVertex2f(*top_right)
    glVertex2f(*top_left)
    glVertex2f(*center)
    glEnd()


def draw_line(start, end):
    glBegin(GL_LINES)
    glColor3f(1, 0, 1)
    glVertex2f(*start)
    glVertex2f(*end)
    glEnd()


def draw_window(start_x, start_y, width, height):
    bottom_left = (start_x, start_y)
    bottom_right = (start_x + width, start_y)
    top_left = (start_x, start_y + height)
    top_right = (start_x + width, start_y + height)

    glBegin(GL_TRIANGLES)
    glColor3f(1, 1, 1)
    glVertex2f(*bottom_left)
    glVertex2f(*bottom_right)
    glVertex2f(*top_right)

    glVertex2f(*top_right)
    glVertex2f(*top_left)
    glVertex2f(*bottom_left)
    glEnd()


def draw_door(start_x, start_y, width, height):
    bottom_left = (start_x, start_y)
    bottom_right = (start_x + width, start_y)
    top_left = (start_x, start_y + height)
    top_right = (start_x + width, start_y + height)

    glBegin(GL_TRIANGLES)
    glColor3f(1, 0.4, 0.4)
    glVertex2f(*bottom_left)
    glVertex2f(*bottom_right)
    glVertex2f(*top_right)
    glVertex2f(*top_right)
    glVertex2f(*top_left)
    glVertex2f(*bottom_left)
    glEnd()


def draw_door_nob(start_x, start_y, width, height):
    bottom_left = (start_x, start_y)
    bottom_right = (start_x + width, start_y)
    top_left = (start_x, start_y + height)
    top_right = (start_x + width, start_y + height)

    glBegin(GL_TRIANGLES)

    glColor3f(0, 0, 1)
    glVertex2f(*bottom_left)
    glColor3f(0, 0.4, 0)
    glVertex2f(*bottom_right)
    glColor3f(1, 0, 0)
    glVertex2f(*top_right)

    glColor3f(1, 0, 0)
    glVertex2f(*top_right)
    glColor3f(0, 0.4, 0)
    glVertex2f(*top_left)
    glColor3f(0, 0, 1)
    glVertex2f(*bottom_left)
    glEnd()


def task(point_size=3):
    glPointSize(point_size)

    width, height = 400, 300
    start_x, start_y = 50, 10

    top_right, top_left = draw_body(start_x, start_y, width, height)

    center = (WIDTH / 2, start_y + height + 180)
    top_cone(top_right, top_left, center)

    draw_line(top_right, top_left)

    # both window
    draw_window(start_x + 50, start_y + 170, 100, 100)
    draw_window(start_x + 250, start_y + 170, 100, 100)

    draw_door(start_x + 150, start_y, 100, 150)

    draw_door_nob(start_x + 150 + 80, start_y + 50, 10, 10)


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 showScreen():
    ## error due to type issue.
    # glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glClear(GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    iterate()
    # glColor3f(1.0, 1.0, 0.0)  # konokichur color set (RGB)
    # call the draw methods here
    task()
    glutSwapBuffers()


def main():
    global HEIGHT, WIDTH
    glutInit()
    glutInitDisplayMode(GLUT_RGBA)
    glutInitWindowSize(WIDTH, HEIGHT)  # window size
    glutInitWindowPosition(0, 0)
    # (important for xmonad to ingnore tiling)
    wind = glutCreateWindow(b"opengl")

    glutDisplayFunc(showScreen)

    glutMainLoop()


if __name__ == "__main__":
    main()
```



Task 03

```
# status = DONE

from random import random, seed

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

HEIGHT = 500
WIDTH = 500


def generate_digits_coordinates(start_x, start_y, width, height):
    mid_y = height / 2

    a = (start_x, start_y)
    b = (start_x + width, start_y)
    c = (start_x, start_y + mid_y)
    d = (start_x + width, start_y + mid_y)
    e = (start_x, start_y + height)
    f = (start_x + width, start_y + height)
    pa = (start_x + int(width * 0.3), start_y)
    pc = (start_x + int(width * 0.2), start_y + mid_y + int(mid_y * 0.5))

    return [a, b, c, d, e, f, pa, pc]


def generate_digits_line(digits):
    a, b, c, d, e, f, pa, pc = digits
    digit_lines = {
        "0": [(a, e), (e, f), (f, b), (b, a)],
        "1": [(pc, f), (f, b)],
        "2": [(e, f), (f, d), (d, c), (c, a), (a, b)],
        "3": [(e, f), (f, d), (d, c), (d, b), (b, a)],
        "4": [(e, c), (c, d), (f, b)],
        "5": [(f, e), (e, c), (c, d), (d, b), (b, a)],
        "6": [(f, e), (e, a), (a, b), (b, d), (d, c)],
        "7": [(e, f), (f, pa)],
        "8": [(a, e), (e, f), (f, b), (b, a), (c, d)],
        "9": [(c, d), (c, e), (e, f), (f, b), (b, a)],
    }

    return digit_lines


def draw_line(start, end):
    glBegin(GL_LINES)
    glVertex2f(*start)
    glVertex2f(*end)
    glEnd()


def task(point_size=3):
    glPointSize(point_size)

    my_id = "20301239"

    start_x, start_y = 10, 200
    gap = 10
    width = 50
    height = 100
    for d in my_id:
        r, g, b = random(), random(), random()
        glColor3f(r, g, b)  # konokichur color set (RGB)

        digits = generate_digits_coordinates(start_x, start_y, width,
height)
        line_maps = generate_digits_line(digits)
        lines = line_maps[d]
        for line in lines:
            draw_line(*line)

        start_x += width + gap


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 showScreen():
    ## error due to type issue.
    # glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glClear(GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    iterate()
    # glColor3f(1.0, 1.0, 0.0)  # konokichur color set (RGB)
    # call the draw methods here
    task()
    glutSwapBuffers()


def main():
    global HEIGHT, WIDTH
    glutInit()
    glutInitDisplayMode(GLUT_RGBA)
    glutInitWindowSize(WIDTH, HEIGHT)  # window size
    glutInitWindowPosition(0, 0)

    # (important for xmonad to ingnore tiling)
    wind = glutCreateWindow(b"opengl")

    glutDisplayFunc(showScreen)

    glutMainLoop()


if __name__ == "__main__":
    seed(42)
    main()
```

20301239