

## 311170030 Introduction to Computer Graphics

### Project 1: Creating a 2D Scene (30%)

Due Time: 12:00pm, November 6 (Wednesday), 2024

*Late penalty: 10% per day.*

*Fail the course if you copy*

#### I. Introduction

This first programming assignment will introduce you to the OpenGL graphics programming interface. In this programming assignment, you will be creating different 2D objects to model interesting shapes. The objective of this assignment is to apply your understanding of the computer graphics theories and give you an introduction to the OpenGL programming library.

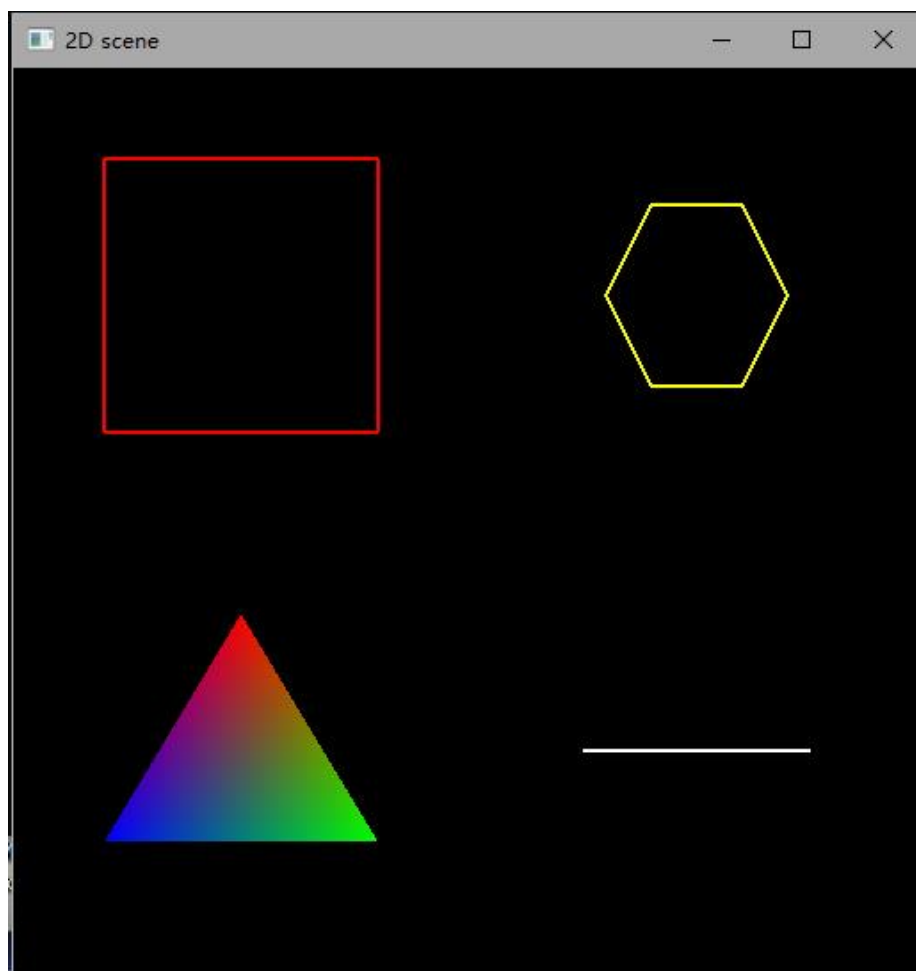


Fig. 1 The scene drawn by the demo program

Your goal is to design a 2D space, which consists of at least *six* different geometric primitives, such as the point, line, triangle, square, polygons, curves, circles and etc. And you should change the attributes of above

primitives, such as the line type or width, filling pattern, anti-aliasing, etc. Besides, you should be able to apply arbitrary transformations — translation, rotation, scaling — to them.

To enhance your project, you will implement multiple viewports within the same window. Set up at least *two* viewports, where each viewport can display different geometric primitives or unique transformations of the same objects. For example, one viewport may show a static scene, while another allows user interaction for transformations. Users should be able to switch between viewports using keyboard and/or mouse inputs, with the ability to modify attributes like color and line style in each viewport and the ability to translate, rotate, and scale the object.

Once you have implemented all the basic functionality, we expect you to construct an interesting background with your program. This will deepen your exploration of OpenGL's capabilities and create a more dynamic visual experience.

## II. Implementation Details

In this assignment package, we have provided you with a template program (i.e., *submit.c*) that includes the necessary functions you are going to use and callback functions in the GLUT interface toolkit. Use this template as the basis for your implementation. There is also a file (i.e., *readme.txt*) indicating the keyboard usage of the demo program (i.e., *demo.exe*) for the users. You may design your own function to process the keyboard events, but you should also submit a file like this to specify the keyboard (and/or mouse) events you designed in your program. Otherwise, the mark for related items will be deducted.

All programs should meet reasonable programming standards: header comment, in-line comments, good modularity, clear printout, efficiency.

### Constraints:

1. Draw at least six different geometric primitives in the 2D space;
2. Could change the attributes of above primitives by the keyboard event;
3. Create at least 6 and at most 10 keyboard or mouse events;
4. Design diverse objects transformations, such as rotation, translating, scaling;
5. Set interesting material properties to different objects.
6. Create at least two viewports to show the 2D space.

### Non-constraints

You are free to add objects, move them, deal with their material attributes, and whatever you wish to make your scene interesting.

## III. Grading Scheme

Your assignment will be graded by the following marking scheme:

- |   |     |
|---|-----|
| ● At least six different geometric primitives | 20% |
|---|-----|
-

---



---

● Change the attributes of above primitives	15%
● Six to ten keyboard events (mouse event is optional)	20%
● Object transformation animation	20%
● Viewport	15%
● Creativity (background and one another creative point designed by yourself....)	5%
● Programming style	5%

---

Total 100%

**Note: no grade will be given if the program is incomplete.**

#### IV. Guidelines to submit programming assignments

- 1) You are suggested to write your programs on Windows, since there will be enough technical support. If you developed the program in other platforms, *make sure your program can be compiled and executed on Windows as the program will only be tested on this platform.*
- 2) Modify the provided *submit.c*, and provide all your code in this file. No more additional .c or .h files are allowed. Type your full name and student ID in *submit.c*. *Missing such essential information will lead to mark deduction.*
- 3) Zip the source code file (i.e. *submit.c*), the executable file (i.e., *submit.exe*), and the readme file (i.e., *readme.txt*, please state the keyboard/mouse event and your creative point in the readme file ) in a .zip or .rar file. Name it with your own student ID (e.g. 2014333333.zip). That is, there should be exactly **three** files in your submitted package.
- 4) Submit the zip file by the URL: <https://send2me.cn/Usy4wS46/SyGEQR2W5eSR3w>
- 5) Please save the screenshot of successful submission, the TA will present the Unsubmitted List after the deadline, if there is any problem, please feel free to contact her.
- 6) In case of multiple submissions, only the latest one will be considered.
- 7) *Fail the course if you copy.*