COSC6372 HW1 – Frame Buffer

Due: February 16 at 23:59, 2023

During the assignments, you will build a small but useful graphics library. We'll call it the Gz library, and all classes, functions, data types, etc, will be named accordingly. Certain standards will apply to make the code interfaces consistent. Consistency will be established using provided include header files, which contain definitions about classes and their interface functions.
The material for each assignment will be provided in a single zip file. After extracting this zip file, you may find:

- All provided source code in folder **src**
- For those using Microsoft Visual C++, you should create your project file (sln) to include all the source code.
- The **makefile** for those use GNU C++ (g++)
- A main program **src/main.cpp** which uses the Gz library
- Several ready classes and functions
- Several incomplete classes and functions with declaration only

Your task is to implement incomplete functions of the Gz library so that the main application can be compiled and run correctly. You are free to add extra classes, functions, and variables to complete the task; but you are not allowed to remove any pre-defined function of the library, even some might not be used currently. In the source code, you can also find some comments and instructions that might help you to complete the task.

After finishing, you need to submit your source code with project file (MSVC) or makefile (GNU C++) and your results. All files must be packed together in a single zip file and submitted to GitHub.

In this assignment, you will build the frame buffer with the depth buffer (z-buffer) for the Gz library. On top of that, the provided main application will read several rectangles from file and use the z-buffer to draw the output image. For each rectangle, the coordinate of 2 corners, color, and z-coordinate (depth) will be provided. The pixel with larger z value will be on top. If the two pixels have the same z value, the pixel drawn later will be on top.

All the data you need for this assignment is put in the zip file hw1.zip. Check the file **handout.pdf** for an overview of the assignment. There are several other files there that may be useful:

File	Description	Type
Gz.h	Contains the major declaration and implementation of the Gz library. These files will be updated in every assignment.	Provided
Gz.cpp		
GzImage.h	Supports manage bitmap images. In this assignment, you use it to write your rendering result to BMP files.	
GzImage.cpp		
GzFrameBuffer.h	Given the deceleration of the GzFrameBuffer class, you need to give the implementation in cpp file.	Incomplete
GzFrameBuffer.cpp		
rects.txt	An input text file contains the information about rectangles. You can figure out the format by reading the source code in file main.cpp.	Data files

Requirements:

1. Do the assignment independently.
2. You need to write a detailed report (50% points of the assignment, pdf format), you should state the assignment problem, explain the algorithm or method you use, explain details of implementation, discuss your results, etc.
3. Save your results as images (with and without depth buffer).
4. Upload every necessary file to GitHub
5. In your GitHub readme file, put your name and student ID there, and the coding environment and compiling method (command).
6. You can only use the library we provide.
7. You will lose points if violate any requirement above