Shiyu Fan G42039555

**Lab 2: Scan-convert & Z-Buffer**

# Main missions

Implement the z-buffer algorithm. Each polygon should be constant colored differently. Just choose a set of random colors. You need not worry about illumination models or shading yet.

# Development environment

Microsoft VS 2017(Visual C++)

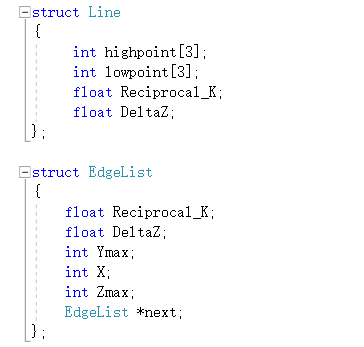
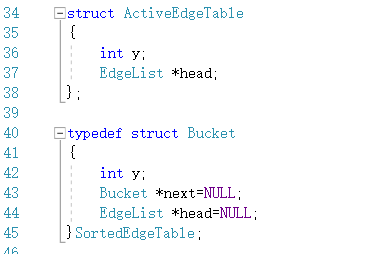
OpenGL (GLUT 3.7)

# Implementation steps

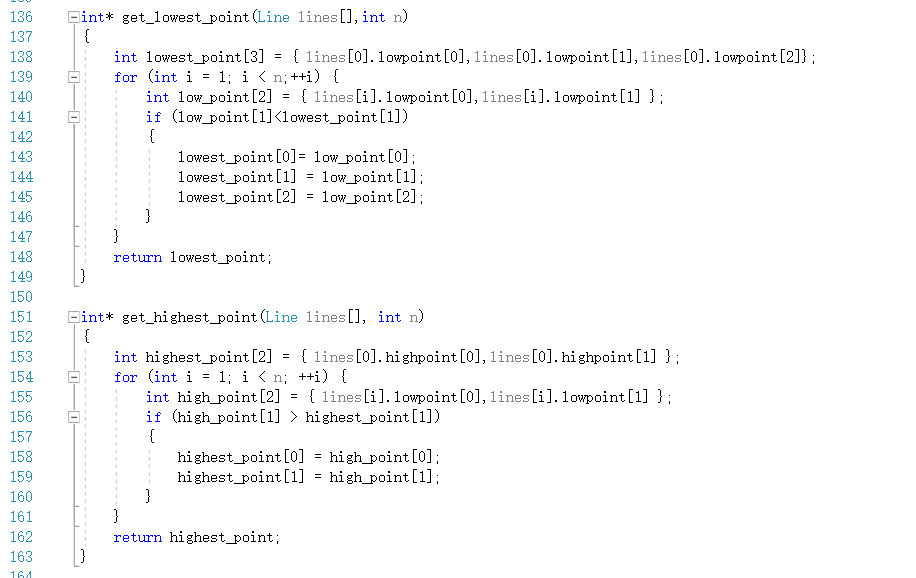
1. Build an edge table for a polygon. Sort all the lines in the polygon and find out the highest and lowest point. Store the edges and some vertices information in the initial edge table. These edge table should link to the bucket which represent the scanline number.
2. For each scanline from y=1 to y=max, calculate all the intersection point information ， and build up a new active edge table.
3. For each polygon，by using the edge table, get all the RGB color number and Z depth and store them in the Z-buffer.
4. Deal with all the front polygon by the same way. Always keep the nearest pixel color data in the Z-buffer.

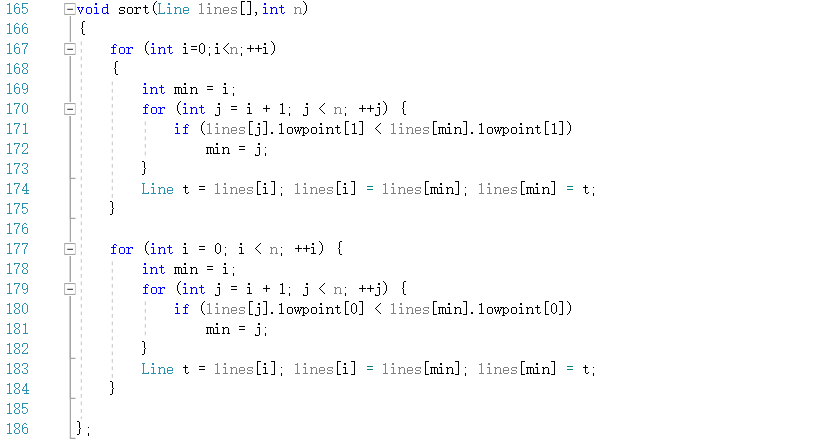
# Code

## Structures of Edge Table

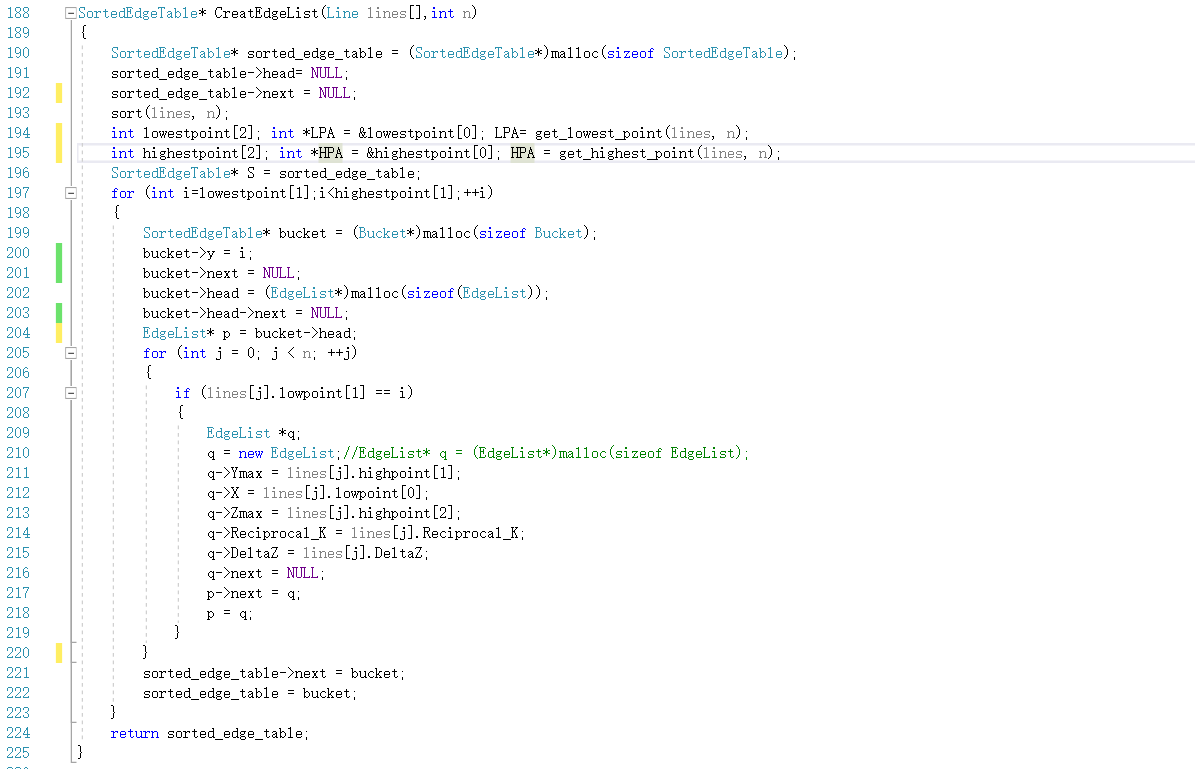
 

## Sort all the edges of the polygon

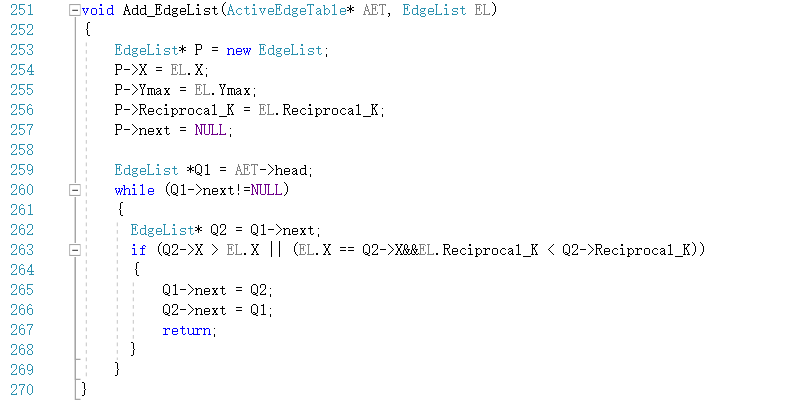




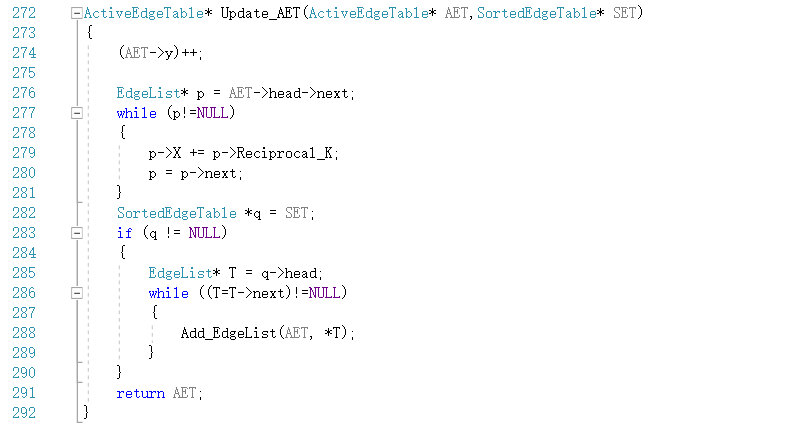
## Create Edge Table



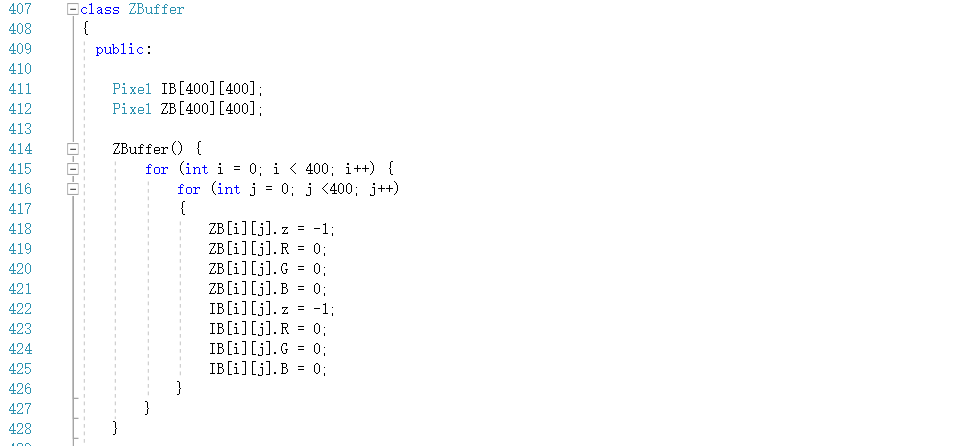
## Add Edge List

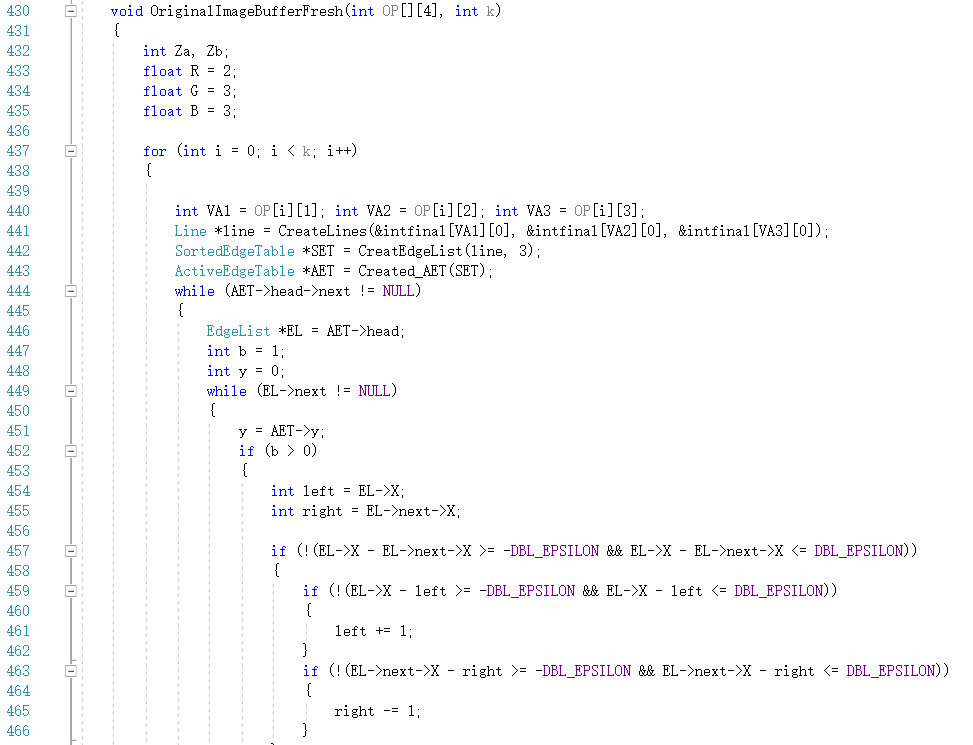


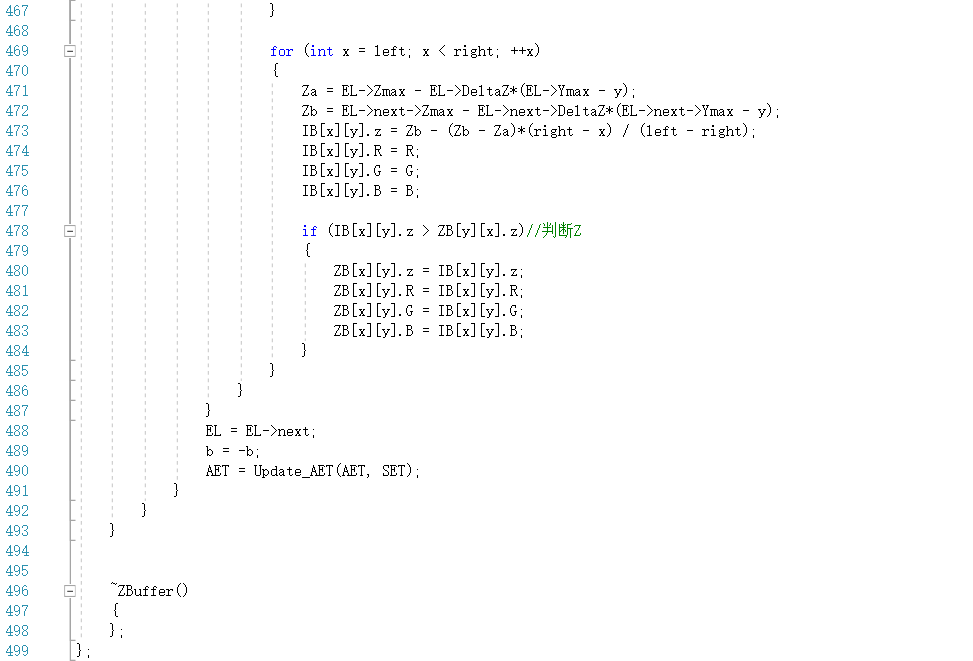
## Update the edge table



## Z-Buffer Class







# Future works

1. Since I didn’t learn any knowledge about data structure. It’s not easy for me to build up a complicate bucket table. There are still some bugs about the table’s pointer.
2. Some data and pointers are not recycled or deleted.
3. Simplify the code so that in the future lab this project can be easily modified.