

具体工作

- 学习了活化边表的扫描线算法
- 读取文件顶点数据
- 构造边表，处理奇点
- 构造活化边表
- 在活化边表中手动进行深度检测并两两配对填充颜色
- 删除活化边表中的边
- 更新活化边表

详见代码注释

技术方案

- fstream 读取文件顶点数据

```
#include<fstream>
    fstream file;
    file.open("rec.txt");
```

- 构造 Point 类存储顶点数据

```
class Point
{
public:
    int x;
    int y;
    int z;
    int r;
    int g;
    int b;
    Point(int x, int y, int z, int r, int g, int b)
    {
        this->x = x;
        this->y = y;
        this->z = z;
        this->r = r;
        this->g = g;
        this->b = b;
    }
};
```

- 以链表形式存储边表和活化边表

```
//以 (3) 为例
Edge *pET>windowHeight];
Edge *AET;
for (int i = 0; i < maxY; i++)
{
    pET[i] = new Edge();
    pET[i]->next = nullptr;
}
AET = new Edge();
AET->next = nullptr;
```

- 作业 (3) 数据通过先写入.txt文件后导入

实现展示





