

计算机图形学实验

实验7 Bézier Curve贝塞尔曲线

姓 名：马鑫\_\_\_\_\_\_\_\_\_\_\_

学 号：\_37220232203780\_

学 院：\_信息学院\_\_\_\_\_\_

专 业：\_数字媒体技术\_\_

年 级：\_2023级\_\_\_\_\_\_\_

2025年 6 月 18日

目录

[Task1：使用OpenGL画点和画线功能，实现贝塞尔曲线生成算法。 5](#_Toc24117)

[1. 完成。自行设置3个控制点，利用de Casteljau生成贝塞尔曲线。要求生成曲线形成过程的动画。 5](#_Toc31416)

[2. 完成。在上述基础上，分别增加控制点数为4、5、6、7、8，并生成相应的曲线形成动画。 9](#_Toc12030)

[3. 自行完善功能，例如提供控制点的选取或移动功能； 11](#_Toc24176)

[Task2: 使用OpenGL画点、画线和画面功能，实现贝塞尔曲面生成算法。 15](#_Toc28221)

[1. 完成。自行设置3\*3个控制点，利用de Casteljau生成贝塞尔曲面。要求生成曲面形成过程的动画。 15](#_Toc7065)

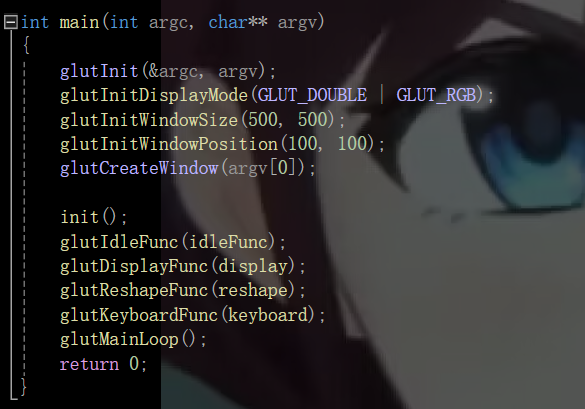
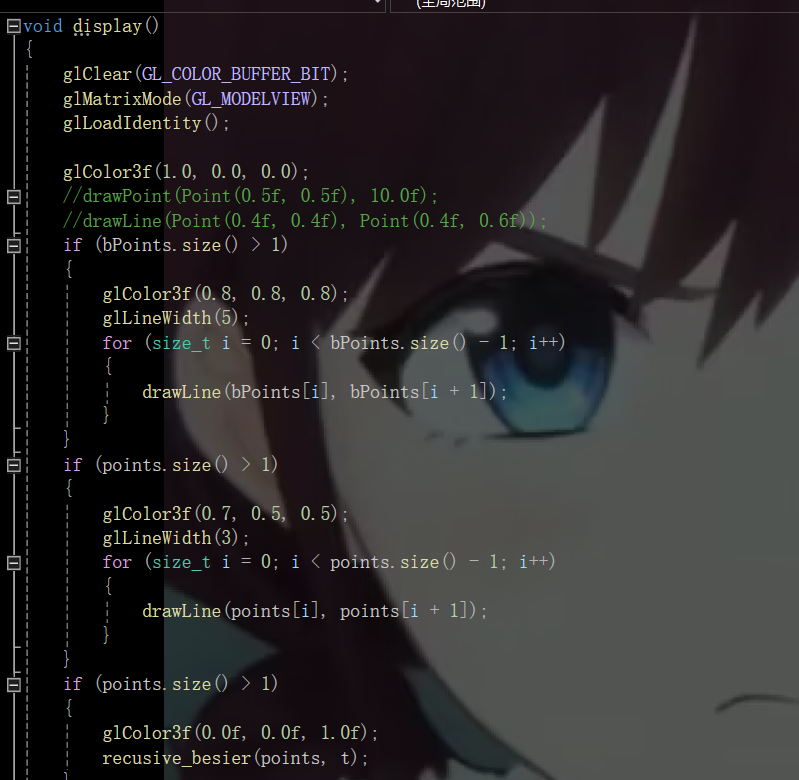
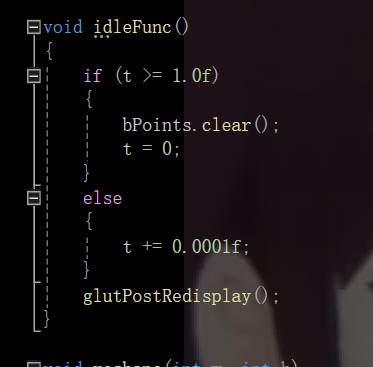
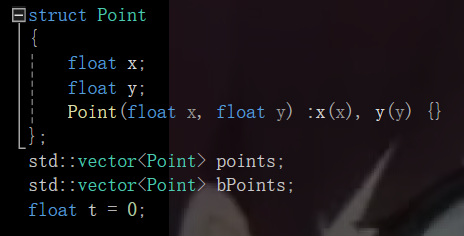
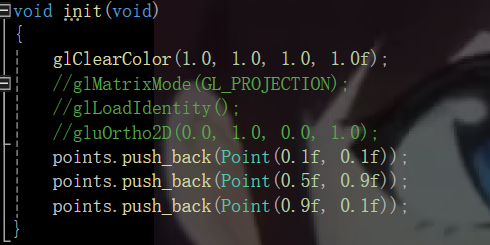
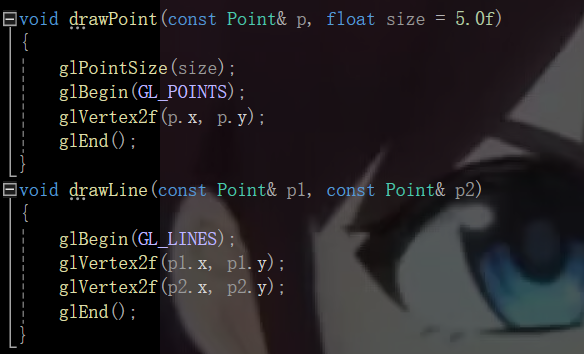
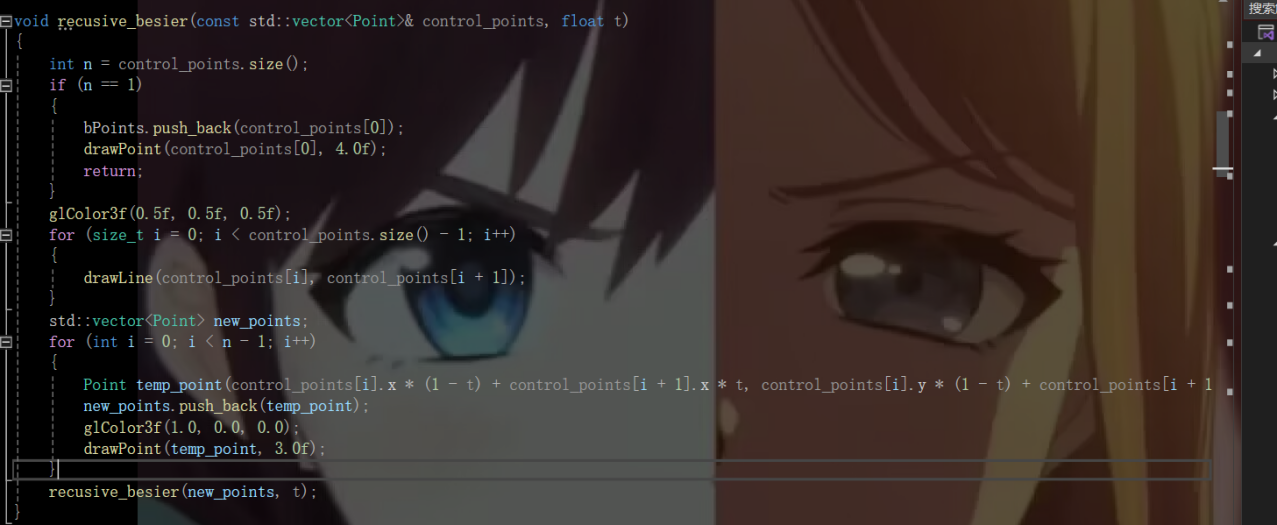
[2. 完成。在上述基础上，分别增加控制点数为4\*4、5\*5并生成相应的曲面形成动画。 20](#_Toc6058)

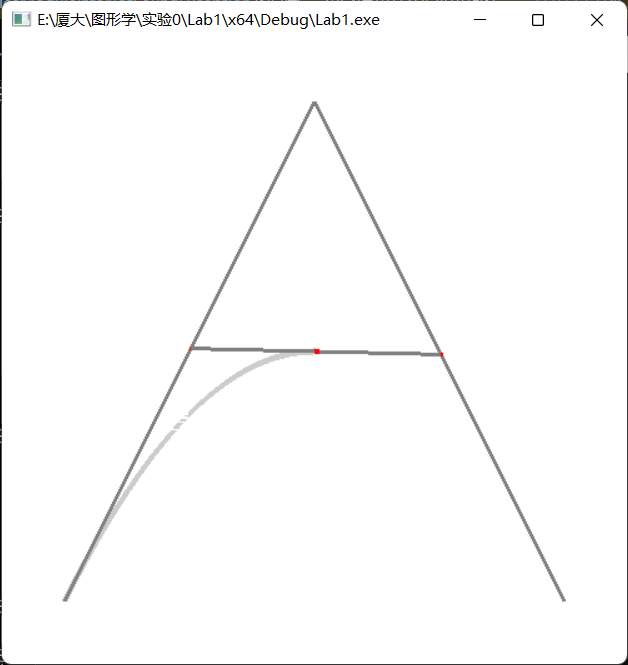
[3. 完成。自行完善功能，例如提供控制点的选取或移动功能；改善生成图形美观程度。 23](#_Toc30894)

# Task1：使用OpenGL画点和画线功能，实现贝塞尔曲线生成算法。

## 完成。自行设置3个控制点，利用de Casteljau生成贝塞尔曲线。要求生成曲线形成过程的动画。

操作流程：预设好三个点，使用de Casteljau算法生成贝塞尔曲线，根据所给的点进行线性插值，得到新的点后再重复这一过程，知道最后剩下一个点即为贝塞尔曲线上的点。使用递归函数实现这一过程，使用idle回调函数递增参数t，同时存储每个t所对应的最后剩下的那个点，在绘制时绘制出存储的每个点，从而实现曲线形成过程的动画。

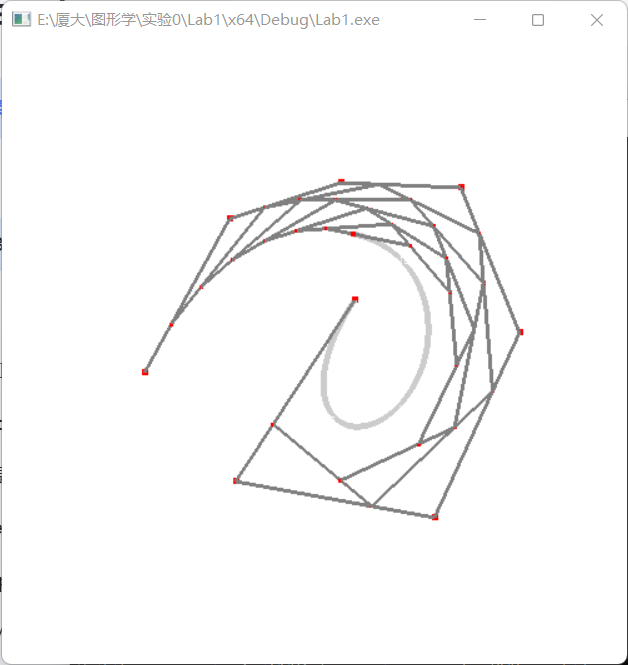
关键代码截图：

运行结果截图： 

## 完成。在上述基础上，分别增加控制点数为4、5、6、7、8，并生成相应的曲线形成动画。

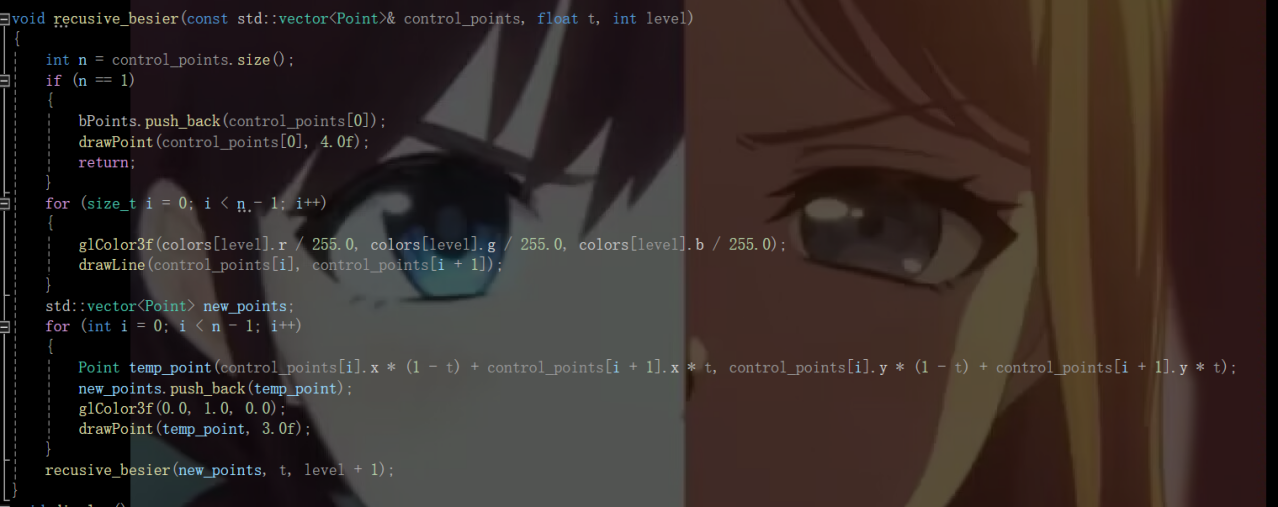
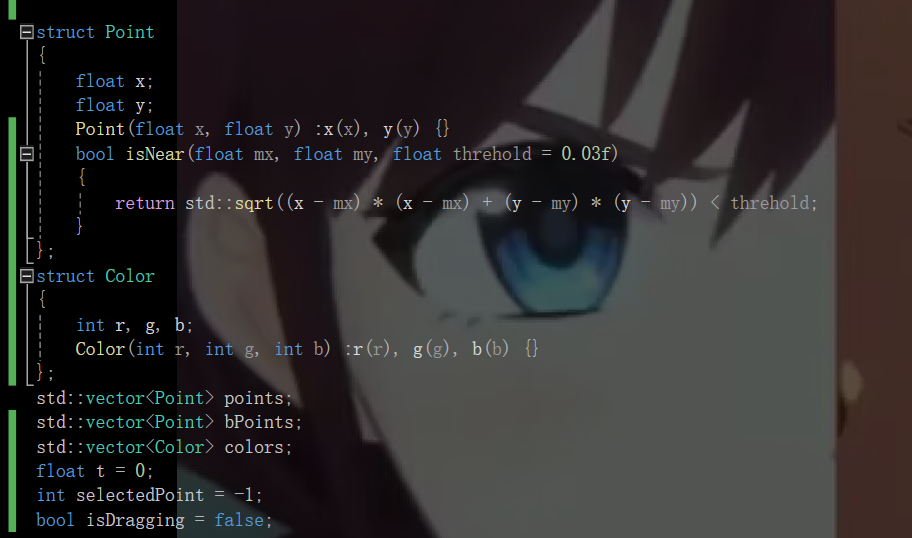
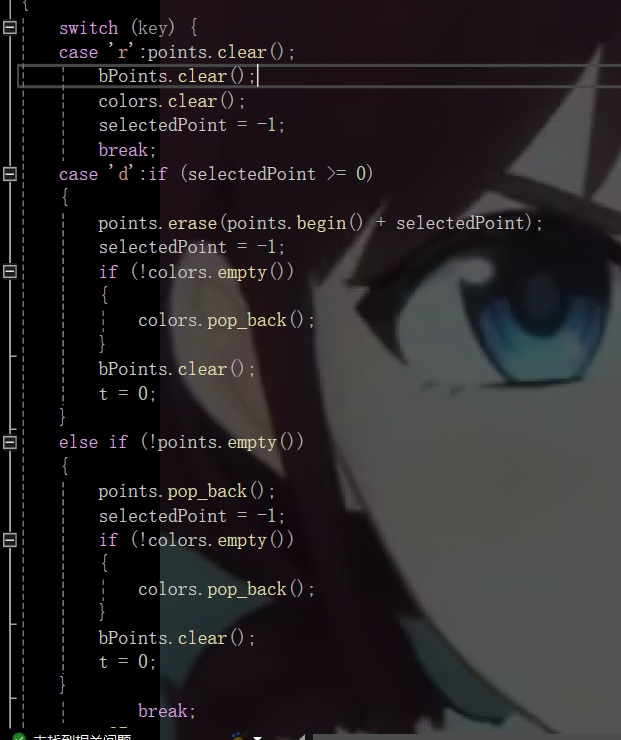
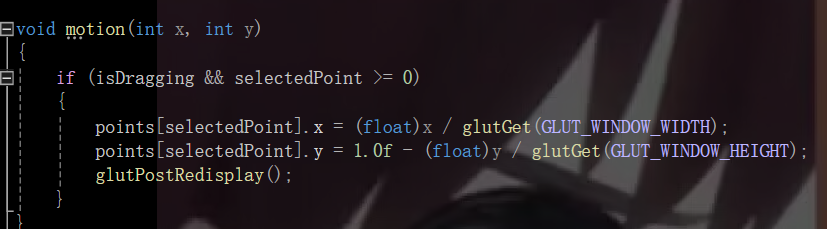
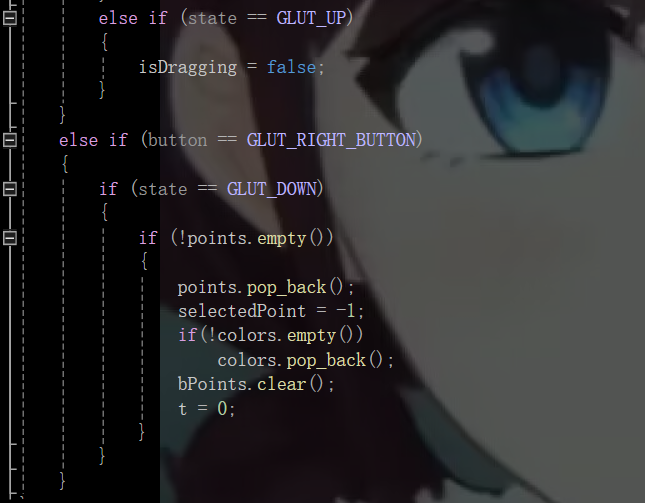
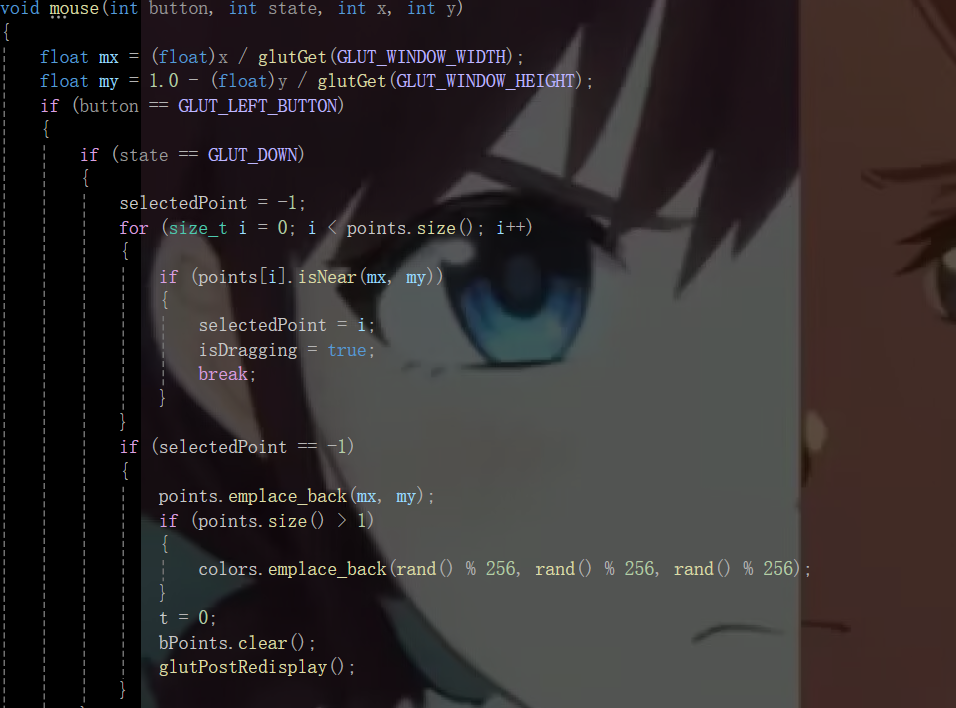
操作流程：在上一份代码的基础上，取消对点的预设，增加mouse回调函数，将鼠标左键点击的位置存入points中，增加控制点。

关键代码截图：

运行结果截图：

## 自行完善功能，例如提供控制点的选取或移动功能；

操作流程：增加控制点选取和移动功能，当鼠标点击位置接近已有控制点时，将不会创建新的控制点，而是修改相关变量，通过motion回调函数改变该控制点的位置。增加对控制点的颜色控制，每一层递归的线使用一种随机颜色。增加删除控制点和清空的功能。

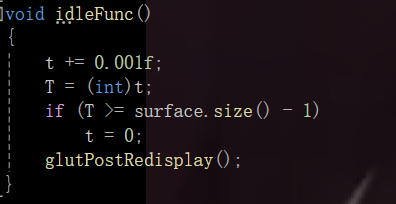
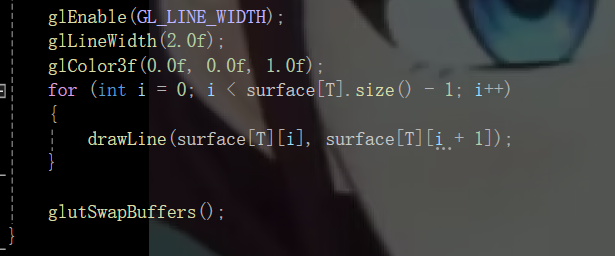
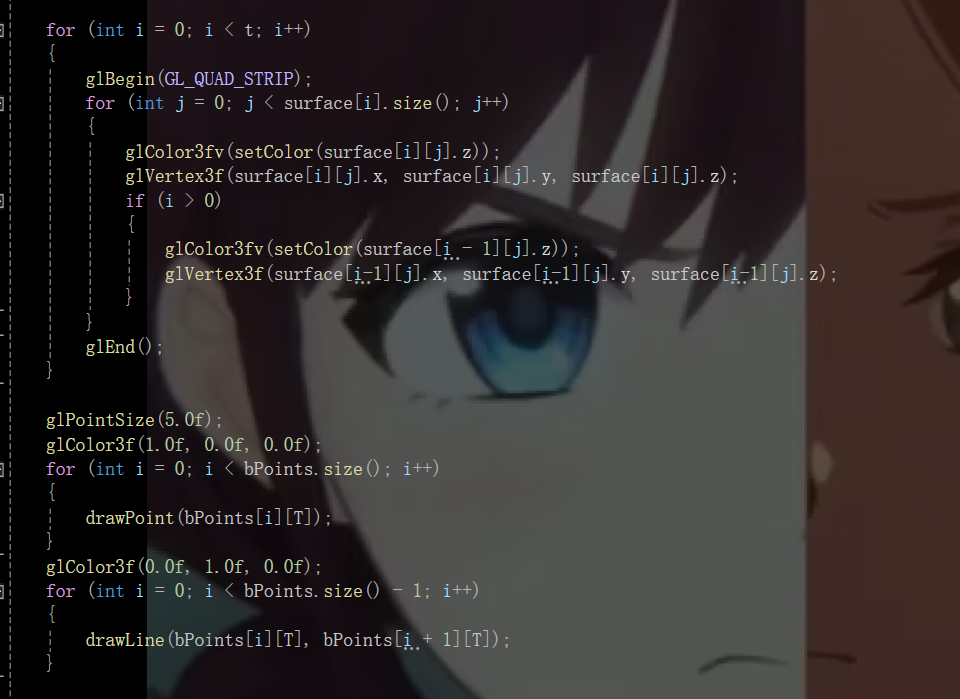
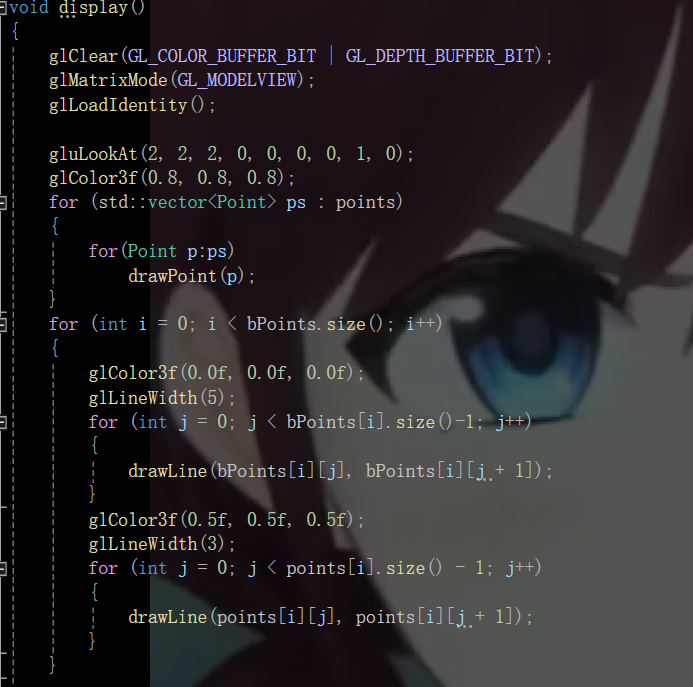
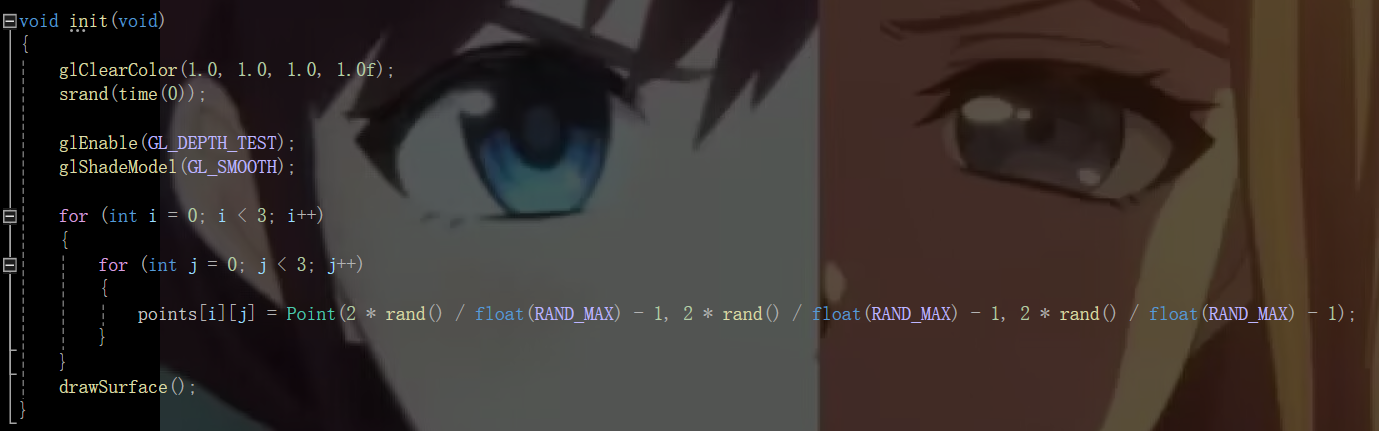
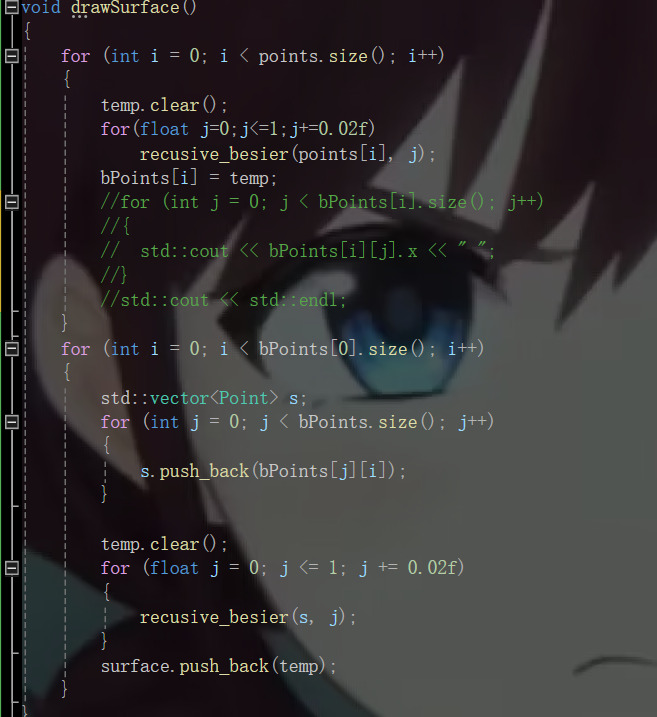
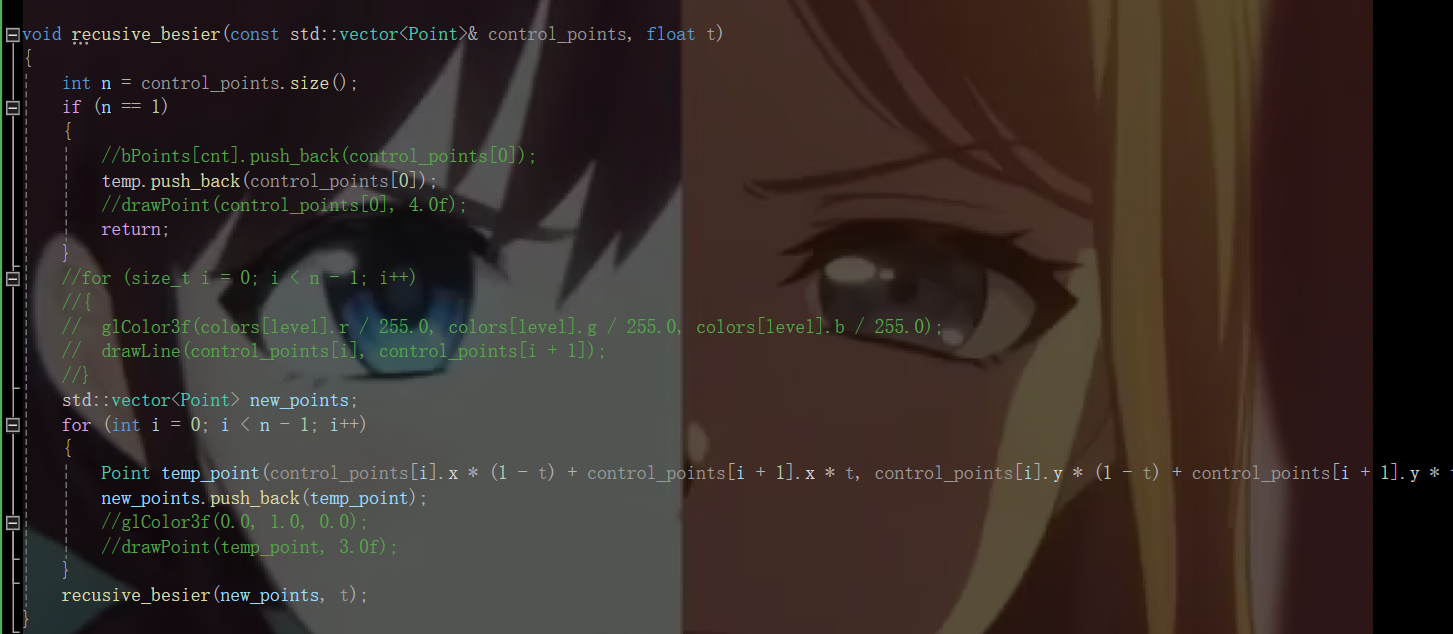
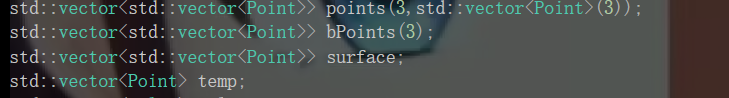
关键代码截图：

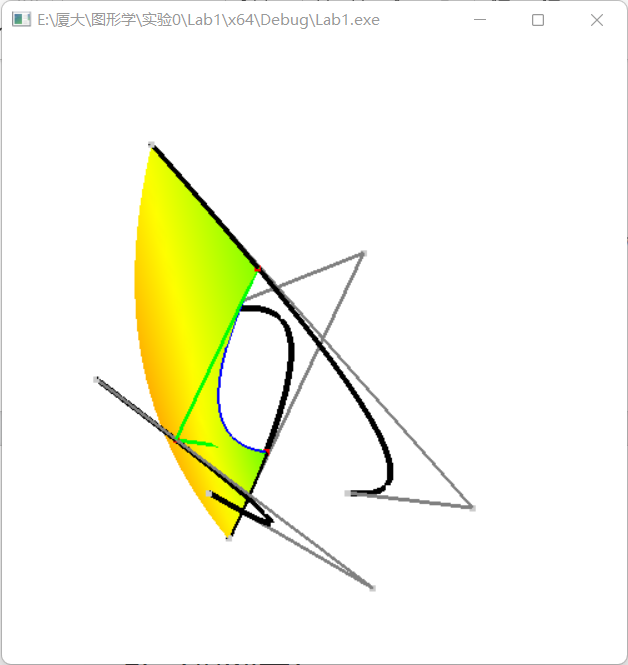
运行结果截图： 

# Task2: 使用OpenGL画点、画线和画面功能，实现贝塞尔曲面生成算法。

## 完成。自行设置3\*3个控制点，利用de Casteljau生成贝塞尔曲面。要求生成曲面形成过程的动画。

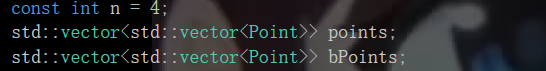
操作流程：在贝塞尔曲线的基础上进行更改，将点改为3维，增加z轴，将存储点的vector动态数组都改为二维动态数组，新增存储贝塞尔曲面的点的数组。将生成的贝塞尔曲线存储起来，再对贝塞尔曲线上的点进行遍历，使用贝塞尔曲线上的点生成新的贝塞尔曲线，该条曲线就是贝塞尔曲面的其中一条曲线，由于空间变为了三维，需要开启深度测试，使用透视摄像机等。为实现动画效果，使用t作为计时器，T作为当前节点，使用随机数预先定义3\*3个控制点，开始运行时即计算好贝塞尔曲线和曲面的点并存储起来，根据T的大小变化改变每一帧画面中遍历的点的数量。

关键代码截图：

运行结果截图： 

## 完成。在上述基础上，分别增加控制点数为4\*4、5\*5并生成相应的曲面形成动画。

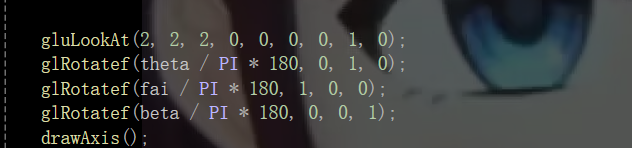
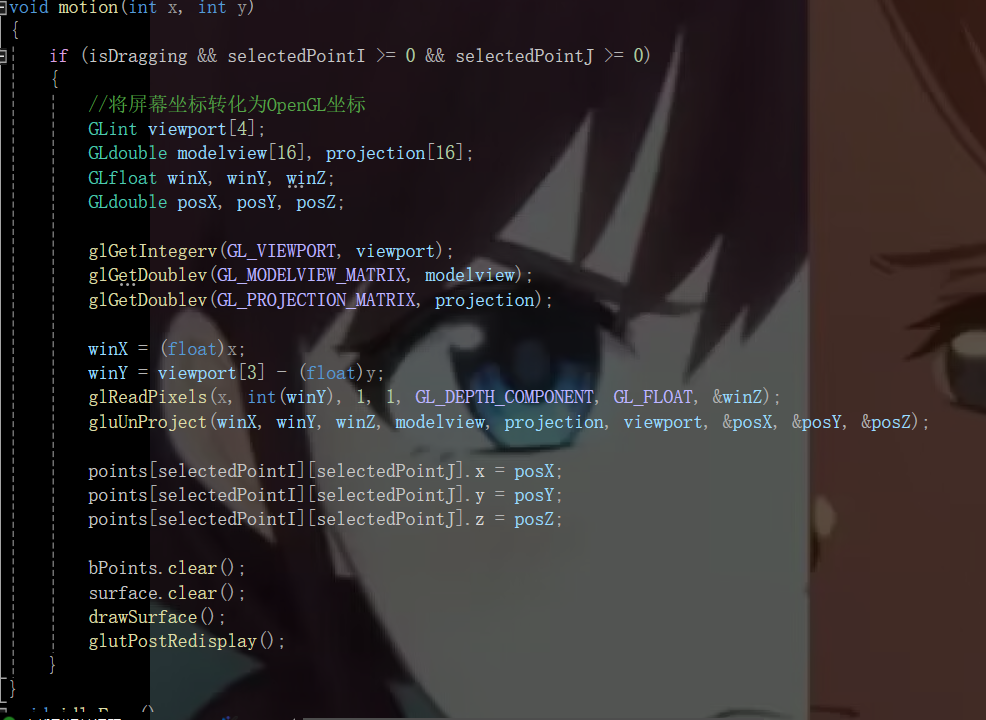
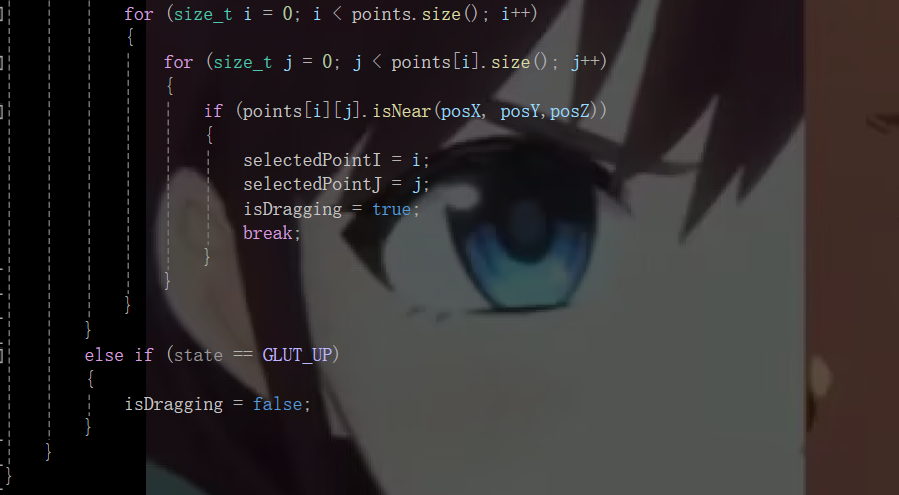
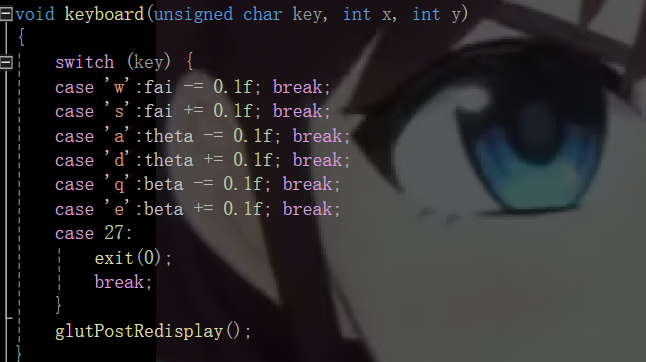
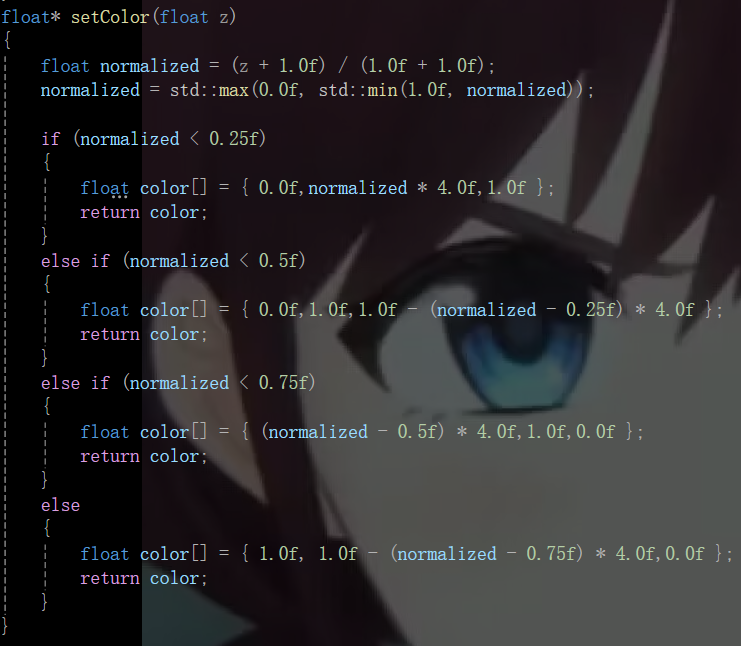
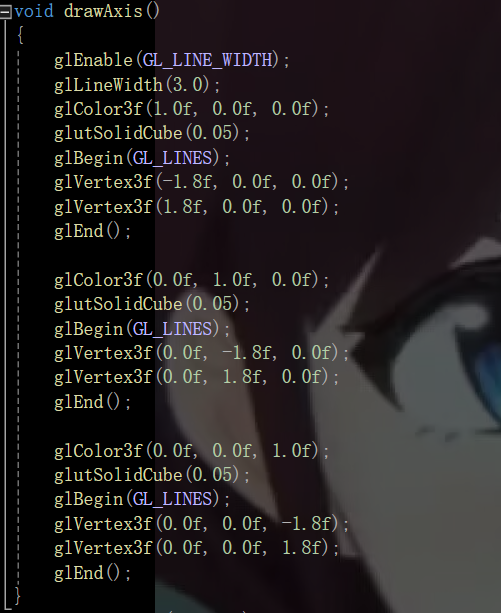
操作流程：取消提前定义动态数组的大小，定义常量n，在初始化时，随机为points生成相应随机值。

关键代码截图：

## 运行结果截图：

## 完成。自行完善功能，例如提供控制点的选取或移动功能；改善生成图形美观程度。

操作流程：添加坐标轴，在场景中画出3个固定的固体作为正交基。添加颜色，根据曲面中点的z轴变化实现颜色蓝->青-> 绿->黄->红的渐变。添加旋转，通过键盘上的wsadqe键，改变三个角度值，分别实现绕三个轴的旋转。添加控制点的选取和移动，实现mouse函数和motion函数，区别二维曲线点的选取，需要将屏幕坐标转化为OpenGL坐标。

关键代码截图：

运行结果截图： 