Homework 4作业报告

16337341 朱志儒

1. **程序说明**

本次作业的两个题目均使用立即渲染模式编写的，题目2使用了轻量级第三方库OpenMesh。

在编写题目1的过程与前几个作业相似，但对每次鼠标点击操作的相应有所不同，每当有鼠标点击时，将鼠标点击的坐标存入名为points的数组中以供绘制图像使用。

编写初始化函数Init()时，和之前几个作业相似，首先清除颜色缓冲区，设置点的大小，接着设置为对模型视景的操作，然后将矩阵转换为单位矩阵，最后调用二维剪裁函数设置整个坐标系。

编写drawLine()函数，用于绘制线段。首先设置线段的颜色和大小，接着根据鼠标输入的点的坐标绘制相应的线段。

编写drawCurve()函数，用于绘制三次Bezier曲线，首先设置曲线的颜色和大小，接着使用公式



（P0、P1、P2、P3分别代表四个点）

求出曲线中点(x，y)的坐标值，然后将这些点绘制出来即可得到Bezier曲线。

编写Display()函数，用于绘制整个图像。首先清除颜色缓冲区，当存储的点的数目少于4时，只绘制点，当存储的点的数目等于4时，绘制线段和曲线。

编写Mouse()函数，用于响应用户的鼠标操作。每当有鼠标点击操作时，先判断当前points数组的长度，若小于4，则将坐标点存入points数组中，若等于4，则清除points数组，再将坐标点存入points数组中。

最后编写主函数，先初始化GLUT库，设置RGB像素格式窗口，接着设置窗口的位置、大小和名称，然后调用初始化函数Init()，接着设置绘制函数Display()、鼠标响应函数的回调Mouse()，最后让绘制线程循环。

编写题目2十分的困难，编写的过程也十分的艰辛，这个题目花了我大量的时间，不过收获颇丰。

在编写题目2时，首先声明多个全局变量，方便多个函数共同使用。声明鼠标当前状态mouse\_state为0，点击之前物体的位置(ox, oy)为(0, 0)，旋转角度相关的两个参数xr和yr，与放缩矩阵相关的参数scale，声明读取的文件名file1、file2、file3，当前读取的文件号current\_file，声明两个显示列表show\_face\_list、show\_wire\_list，显示状态show\_face、show\_wire、show\_flat\_lines，用于设置显示模式。

编写初始化函数initial\_GL()，和题目1相似，但是需要开启深度缓冲区、启用光照功能和设置第一个光源，再申请连续索引值赋给显示列表，接着定义第一个显示列表show\_wire\_list，再定义第二个显示列表show\_face\_list。在定义show\_wire\_list时，先关闭光照功能，再设置线的大小和颜色，然后根据读取的多边形网格数据以Wireframe模式绘制，最后开启光照功能。在定义show\_face\_list时，根据读取的多边形网格数据以Flat模式绘制。

编写读取文件函数read\_file(string file)，首先请求顶点法线，如果存在则读取文件，如果不存在顶点法线，就通过面法线计算顶点法线，最后释放面法线。

编写reshape(Glint w, Glint h)函数，用于设置显示窗口的属性，把用glOrtho创建的正交平行视景体截取图像显示到窗口。

编写显示函数display()，首先清除之前的颜色缓存和深度缓存，将矩阵置为单位矩阵，再设置旋转矩阵、位移矩阵和缩放矩阵，最后根据显示状态show\_face、show\_wire、show\_flat\_lines来执行相应的显示列表以实现Flat、Wireframe和Flat lines三个显示模式。

编写函数special(int key, int x, int y)，用于响应用户的按键操作。当用户按下F1时，显示cow.obj文件；按下F2时，显示cactus.ply文件；按下F3时，显示Armadillo.off文件。当用户按下F4时，显示模式在Flat、Wireframe和Flat lines中切换。

编写mouse\_move()函数和mouse()函数，用于响应用户的鼠标操作。编写mouse\_move()函数时，根据鼠标位置的变化情况修改物体旋转的角度。编写mouse()函数时，首先判断鼠标的操作，如果是点击操作，则修改鼠标的位置坐标。如果是滑动滚轮，则修改放缩矩阵参数scale。

最后编写主函数main()，与题目1相似，但增加glutMotionFunc()和glutSpecialFunc()用于响应用户的鼠标移动和按键操作。

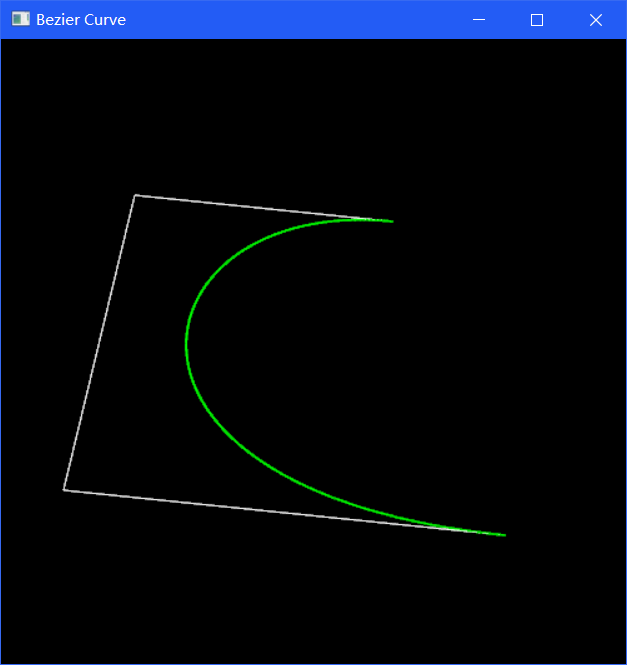
1. **运行方法**

在“用OpenGL实现交互式三次Bezier曲线的构建”文件夹中，有Bezier Curve.exe和freeglut.dll两个文件。运行Bezier Curve.exe文件，然后在窗口中随机点击4下，就会在窗口中显示三条线段和一条三次Bezier曲线。再点击4下也是同样的效果。

在“用OpenGL实现简单的多边形网格数据读取和操作”文件夹中，有Reading and Manipulation.exe可执行文件、freeglut.dll动态链接库文件和Armadillo.off、cactus.ply、cow.obj三个多边形网格文件。运行Reading and Manipulation.exe文件，读取并显示cow.obj文件，此时可以使用鼠标拖动物体使其旋转，也可操作滚轮来放大或缩小视角，按下F4即可切换显示模式。再按下F2则读取并显示cactus.ply文件，按下F3则读取并显示Armadillo.off文件，在显示这两个文件时均可进行上述操作修改显示效果。

1. **程序运行结果**

运行Bezier Curve.exe文件效果如图所示。



运行Reading and Manipulation.exe文件效果如图所示。

