图形学作业 1: 曲线和曲面 Curves & Surfaces

一、工程目录

```
project
|   README.pdf
|   Work_one.sln（解决方案文件）
└──.vs
|   └──Work_one
|       └──v15
|           |.suo（项目配置相关文件）
|
└──data（数据文件夹）
|   |   spline01_bezier.txt
|   |   spline02_bspline.txt
|   |   ...
|
└──include（.h文件夹）
|   |
|   └──GL
|       |   glut.h
|
└──lib（库文件夹）
|   |   glut32.lib
|
└──release（项目可执行文件夹）
|   |   curve_editor.exe
|   |   glut32.dll
|   |   raytracer.exe
|   |   ...
|
└──temp（编译中间文件目录）
|   |   ...
|
└──Work_one（源代码文件夹）
|   |   arg_parser.h
|   |   Curve.cpp
|   |   ...(源代码文件)
```

二、编译环境

Windows 10 + Visual Studio 2017

三、功能实现

基础部分: *全部实现*

1. 工程建立与代码运行成功。
2. 绘制 Bezier 曲线和 B 样条曲线。(四个控制点)
3. 实现 Bezier 曲线和 B 样条曲线的转换（四个控制点)，保存转换后的文件。
4. 绘制多于四个控制点的 Bezier 曲线和 B 样条曲线。
5. 编辑两种给出的曲线（增，移动，删除控制点）并保存编辑后的曲线文件。(Bezier 不支持删除控制点)。

6. 实现旋转面（包括增删改旋转曲线的控制点）并保存至 obj 文件。
7. 实现 Bezier 张量积曲面并保存至 obj 文件。
8. 茶壶数据可以成功画出。

附加分:

1. 实现了几何法画 Bezier 曲线算法 de Casteljau's algorithms。
2. 实现了多控制点情况下 B 样条曲线向 Bezier 曲线转化并保存为数据文件，没有实现多控制点 Bezier 曲线向 B 样条曲线转化。

四、可执行文件使用

为了做到和给出既定的使用命令一致，方便使用该工程已将可执行文件编译为 release 文件夹下的 curve_editor.exe 可执行文件，同时配合已经给出的 raytracer.exe 使用。因此，测试结果时将这两个可执行文件放在保存数据文件的文件夹下，可以**直接**使用文档中给定的所有命令，如:

```
curve_editor -input spline01_bezier.txt -gui -curve_tessellation 30
```

五、问题解决过程

1. 关于类的设计部分，我就是设计为基类 Spline 基本所有函数为纯虚函数，交由子类来实现。
2. Curve 类的 Paint 问题。画 Bezier 曲线或者 B 样条曲线按照给定的公式:

$$Q(t) = \mathbf{GBT(t)} \quad = \text{Geometry } \mathbf{G} \cdot \text{Spline Basis } \mathbf{B} \cdot \text{Power Basis } \mathbf{T(t)}$$

其中样条矩阵是固定不变的，给定拟合的直线段数 tess 的话 T 矩阵也是可以确定的，因此我在 Spline 类中封装了下面两个函数，通过给定的控制点矩阵和分辨率可以直接画出相应的曲线。

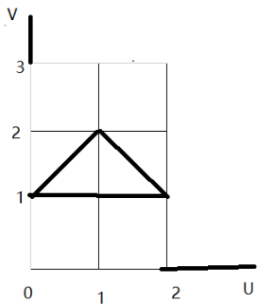
```
void drawBezier(Matrix m,int tessellation)
void drawBSpline(Matrix m, int tessellation)
```

在 Curve::Paint 中绘制控制点和控制点的连线。子类 BezierCurve 和 BSplineCurve 中 Paint()实现即可先调用父类 Curve::Paint()绘制控制点，然后分别通过循环给出控制点矩阵并调用以上两个对应的函数即可绘制出曲线。

3. 两种曲线的转换问题。使用公式直接求即可，会用到 Matrix 类的求逆矩阵函数 Inverse，还是比较简单的，求解点的过程和绘制曲线时差不多。OutputBezier 和 outputBSpline 函数按照给出的数据文件示例的格式输出即可。这里需要注意的是数据文件第一行 nums_spline 已经被框架中代码输出至文件，自己需要做的就只是输出曲线类型和分行输出一系列控制点。
4. 两种曲线在多控制点下互相转换的问题。对于多控制点 B 样条曲线，控制点数>4，每连续的四个控制点都可以通过公式直接转换为同形状的 Bezier 曲线的控制点坐标，但是 Bezier 曲线不行。

   三次 B 样条处处是光滑的，但是多个三次 Bezier 曲线连接在一起，连接点处就不一定是光滑的，因此多条 Bezier 曲线连接在一起是不一定能转化为单条 B 样条曲线的。一种可能的解决方式是，这多条 Bezier 曲线转化为对应的多条 B 样条曲线，但是针对这个项目不可行。因为正如前面所说，打开数据文件后 nums_spline 是框架读入并固定的，转化后 output 文件是框架自己做好的，这就意味着，我们从数据文件中读入 nums_spline 条曲线，写入文件就还是 nums_spline 条文件。那么，如果 Bezier 曲线数据文件中给出了一条由多条 Bezier 曲线连接而成的曲线，nums_spline 是 1，转换为 B 样条曲线输出的话也只能是 1 条，因此框架代码不允许我们分割。故我没有实现。
5. 处理控制点多于 4 个的曲线绘制。这个就是滑动窗口的设计，Bezier 曲线每次窗口滑动 3 个控制点，B 样条每次滑动 1 个控制点。
6. 实现曲线编辑的功能。框架中已经给出了事件调用什么对应的函数，并且在做出更改后及时重绘。Add,move,delete 函数直接添加、修改和删除数组中存的对应的点即可。B 样条对于这三种操作是没有问题的，但是 Bezier 曲线要求点的数量必须符和它的规律，因此针对 Bezier 曲线，我添加点时采取的方式是在同一个地方添加三个相同的点，保证曲线形状基本不变，delete 我禁用了。另外就是要打印以下过程，看看框架代码到底 pick 和 pickEdge 了对应的什么位置的点和边，这个对于插入控制点很重要。添加点的空间不足时，空间扩大二倍处理。
7. 旋转面问题。我们需要做的就是 return 一个 TriangleMesh 指针即可，这个网络会被自动 output 为一个 obj 文件，供 raytracer 加载对应的场景文件显示。关于 TriangleMesh 和 TriangleNet 还是很有意思的。TriangleNet 更像是一个 Grid，即网格，它的两个参数就是这个网格的横向及纵向的网格数。如下图是一个 U=2,V=3 的

Net，只需要指定 (1,1), (2,1), (1,2)点的具体坐标，这三个点形成的三角形就自动构建好了。



使用 TriangleNet 的方便之处在于，指定坐标上的点之间的约束已经形成，给出具体位置的点坐标后，在约束下构建的三角形也是固定的。因此我们要给出旋转曲面的网格点及三角形，只需要在指定分辨率下的既定网格内的指定坐标位置添加具体的点的坐标即可。换句话说，用指定分辨率的 TriangleNet 可以“包住”一个曲面。因此主要工作就是计算旋转面上点的坐标。

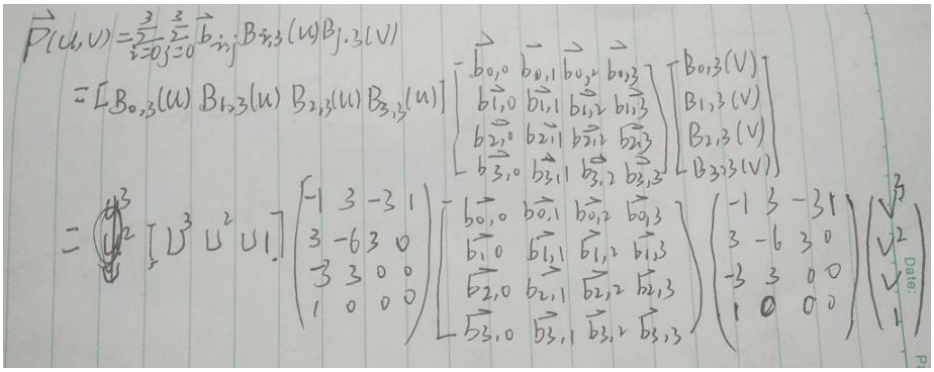
Bezier 曲线和 B 样条曲线上的点的坐标可以直接获得，绕 Y 轴旋转 theta 角度后的点的坐标也可以通过旋转矩阵变换求得，因此旋转面上所有的点的坐标都是可以求得的。构成旋转曲面的 TriangleNet 的 U，V 参数即分辨率，其中横向的 U 是由 revolution_tessellation 决定的，纵向是由拟合该曲线的直线条数决定的，即等同于由 curve_tessellation 决定的。因此，将对应坐标的点填入该 Net 即可。

8. Bezier 张量积曲面绘制。题目要求绘制的是双三次 Bezier 张量积曲面，张量积曲面的定义是：

$$\vec{P}(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}\vec{b}_{i,j}B_{i,m}(u)B_{j,n}(v)$$

$$=[B_{0,m}(u)\ B_{1,m}(u)\cdots B_{m,m}(u)]\begin{bmatrix}\vec{b}_{0,0} & \vec{b}_{0,1} & \cdots & \vec{b}_{0,n}\\ \vec{b}_{1,0} & \vec{b}_{1,1} & \cdots & \vec{b}_{1,n}\\ \vdots & \vdots & \vdots & \vdots\\ \vec{b}_{m,0} & \vec{b}_{m,1} & \cdots & \vec{b}_{m,n}\end{bmatrix}\begin{bmatrix}B_{0,n}(v)\\ B_{1,n}(v)\\ \vdots\\ B_{n,n}(v)\end{bmatrix}$$

那么针对三次 Bezier 张量积曲面的式子为：



其中 b(I, j)是对应的向量。已经给出了 8 条 Bezier 曲线的 16 个控制点，则可以根据给出的分辨率构建 TriangleNet，通过上述矩阵运算出对应（u, v）的点的坐标，填入 TriangleNet 对应的（I, j）即可。其中（u, v）取值可通过循环在 [0,1]区间循环给出。

9. De Casteljau 算法的实现。这个可以用分治算法实现，也可以循环迭代实现。我才用了后者，实现的函数为 Spline 类的 drawBezierWithDeCasteljau (Matrix m, int tessellation)函数，可以用于替代 drawBezier (Matrix m, int tessellation)函数。

六、实习总结

这次实习难度还行，本身实习内容让我对 Bezier 曲线和 B 样条曲线掌握的更加扎实，对旋转曲面、张量积曲面也有了更深层的理解。框架代码中矩阵运算的计算过程实现的非常精妙，TriangleNet 的思想也着实惊艳，我上课时候问过老师有没有好的计算旋转面上点和三角形的算法，这个 TriangleNet 就是用来解决这个问题的，给出分辨率就可以绘制出网格。框架代码中有很多工程构思很值得我学习。

姓名：AnDJ