计算机图形学（2022秋）

——曲线和曲面实验

一、实验目的

本次实验需要实现spline曲线的计算，掌握通过简单的曲线旋转生成曲面的建模方法，实验目的是为了理解spline以及局部和全局坐标系统转换。

二、运行环境

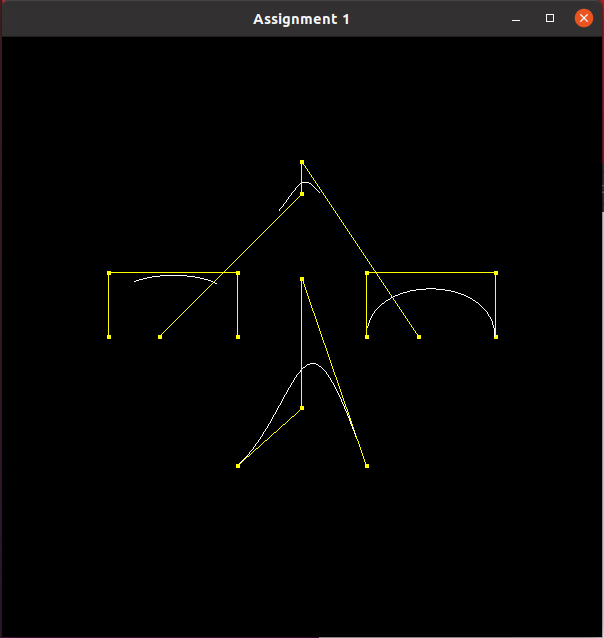
推荐ubuntu18.04以上操作系统，需要安装opengl。安装命令可参考

apt-get install freeglut3-dev

Assignment(linux)文件夹为linux平台下项目代码。此外，Assignment（vs）文件夹为win平台下的项目代码，经测试win10下visual studio 2019可运行。在Visual studio右键项目->属性->配置属性->调试，可修改程序运行前的命令行参数。

实验需要用到vecmath库用来处理3d向量和矩阵的运算（头文件已包含在Assignment目录下）。用Vector3f类来表示三维向量（有成员函数x（）、y（）、z（）返回x、y、z坐标值，normalize（）将向量单位化等），CurvePoint表示曲线上的点，Curve类表示曲线。关于vecmath更多可参考（https://github.com/ydm/mit-vecmath）。

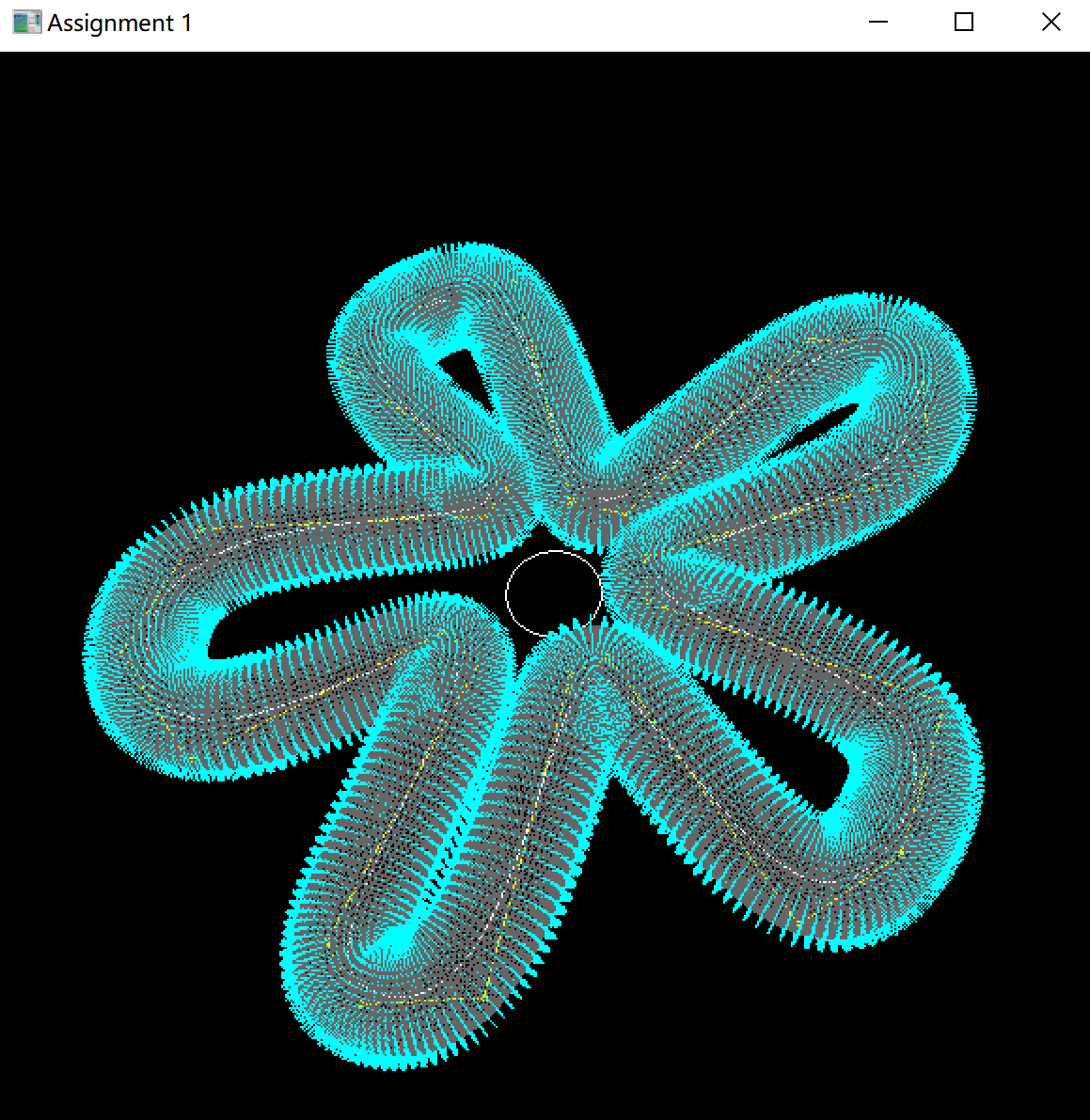
三、编译和运行示例代码

将Assignment(linux)文件夹拷贝至unbuntu（虚拟机）上，进入该文件夹，示例文件的名字叫a1soln，可能需要用chmod命令改可执行权限才能运行。打开示例程序并载入swp场景文件（输入 ./a1soln swp/core.swp ）可以看到曲线部分的展示。示例程序将core.swp读取至ifstream对象。在函数parseFile（parse.h中）已经将所需要的曲线control point数据读取到你写代码所需要用到的数据结构里面。

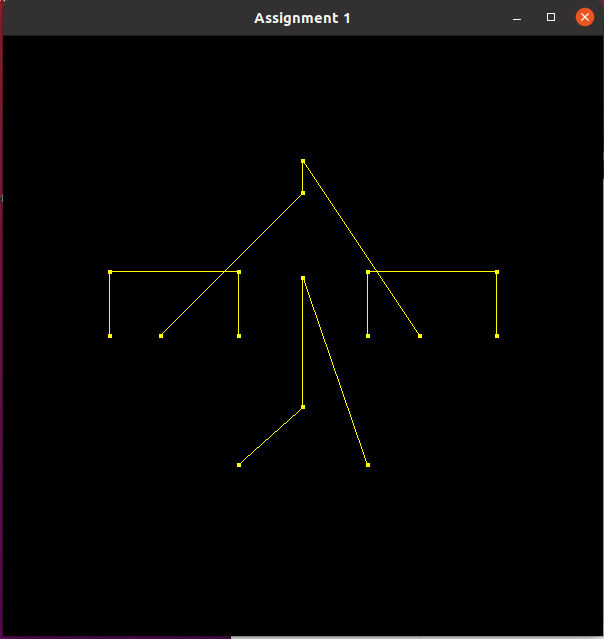
Core.swp包含4个曲线，control points以黄色显示出来，曲线本身用白色显示，按c键可以切换显示曲线的local coordinate frames。按p键可以切换是否显示control point.按esc退出程序。

Weird.swp显示一个生成的圆柱。按s键可以切换渲染模式，默认是smooth shading，可以切换成线框显示。

Flircle.swp是一个生成的飞行圈。按s键可以切换渲染模式，目前显示的是另外一种渲染模式。



使用make命令生成a1程序，目前这个程序不具备上述部分功能，需要你在cpp文件中的TODO部分注释下面的区域完善代码。输入 ./a1 swp/core.swp 目前应该能看到control point但看不到曲线本身。



四、实验要求

3.1 曲线

在Curve.cpp中增加代码使得程序可以根据control points生成和显示cubic Bezier以及B-spline曲线、正确计算沿着曲线的局部坐标线框（曲线本身用白色显示，N,B,T向量分别用红色、绿色、蓝色显示，N代表法向量，T代表速度向量）。

3.2 曲面

在surf.cpp中的makeSurfRev和makeGenCyl函数中增加代码使得程序能够生成和显示旋转曲面（由一条位于xy平面的曲线绕y轴旋转）和圆柱（generalized cylinder）。法向量也要计算正确。

五、算法提示

参考assn1.pdf的4.2和5.1、5.2。

//c被赋值为a与b的叉乘

Vector3f a，b，c；

a.x() = 1.0; a.y() = 0.0; a.z() = 0.0;

b.x() = 0.0; b.y() = 1.0; b.z() = 0.0;

c = Vector3f::dot(a, b);

//c被赋值为a与b的叉乘

c = Vector3f::cross(a, b);

//c单位化

c.normalize（）；

Curve curve;

CurvePoint point;

//CurvePoint类有VTNB四个Vector3f成员变量，用来存储本身的坐标、速度向量、法向量等

point.V=a;

point.T=b;

point.N=c;

point.B=c;

//向curve中加入point

curve.push\_back(point);

Surface surface;

//向surface中加入顶点

surface.VV.push\_back(a);

//向surface中加入对应顶点的法向量

surface.VN.push\_back(b);