

北京林业大学 2023--2024 学年第 2 学期实验任务书

课程名称: 三维动画原理与制作 开课学院: 信息学院

考试班级: 数媒 21-1、2 命题人: 杨 猛

实验环境: Unity 3D 2019 或以上 实验学时: 2

实验题目(范围): 实验 5 Free-Form Deformation

请详细说明该设计的方案、内容、要求、进度等

严禁剽窃、抄袭等作弊行为!

实验目的:

1. 理解并实现论文《Free-Form Deformation of Solid Geometric Models》中描述的 FFD 算法。
2. 了解 NURBS 曲线的实现方法。
3. 探索变形算法原理。

实验内容:

1. 了解 Cmake 项目的构建方法, 主要包括:
 - (1) 下载并安装 Cmake。
 - (2) 使用 Cmake 命令, 构建项目代码。
 - (3) 运行项目了解算法实现原理。
2. 依照“**实验方法**”中步骤进行对算法进行了解和学习, 并给提供源码的核心代码添加注释。

实验环境要求:

1. 实验环境: 建议使用 Clion 2021 或 Visual Studio 2019 或更高版本。

实验方法: (注: 以下演示步骤均在 Visual Studio 2019 环境中完成)

本次实验提供源代码, 需要将源代码构建成功(可选)后, 针对算法核心部份填写注释。

1. 项目构建(可选)

注意, **源代码构建**根据你的项目构建经验以及网络, 机器, 系统环境可能会占据大

量的时间，所以完全是可选的，你也可以直接参照可执行文件（.exe）来进行源代码的阅读。

首先，打开 Visual Studio 2019，选择继续但是不需要代码

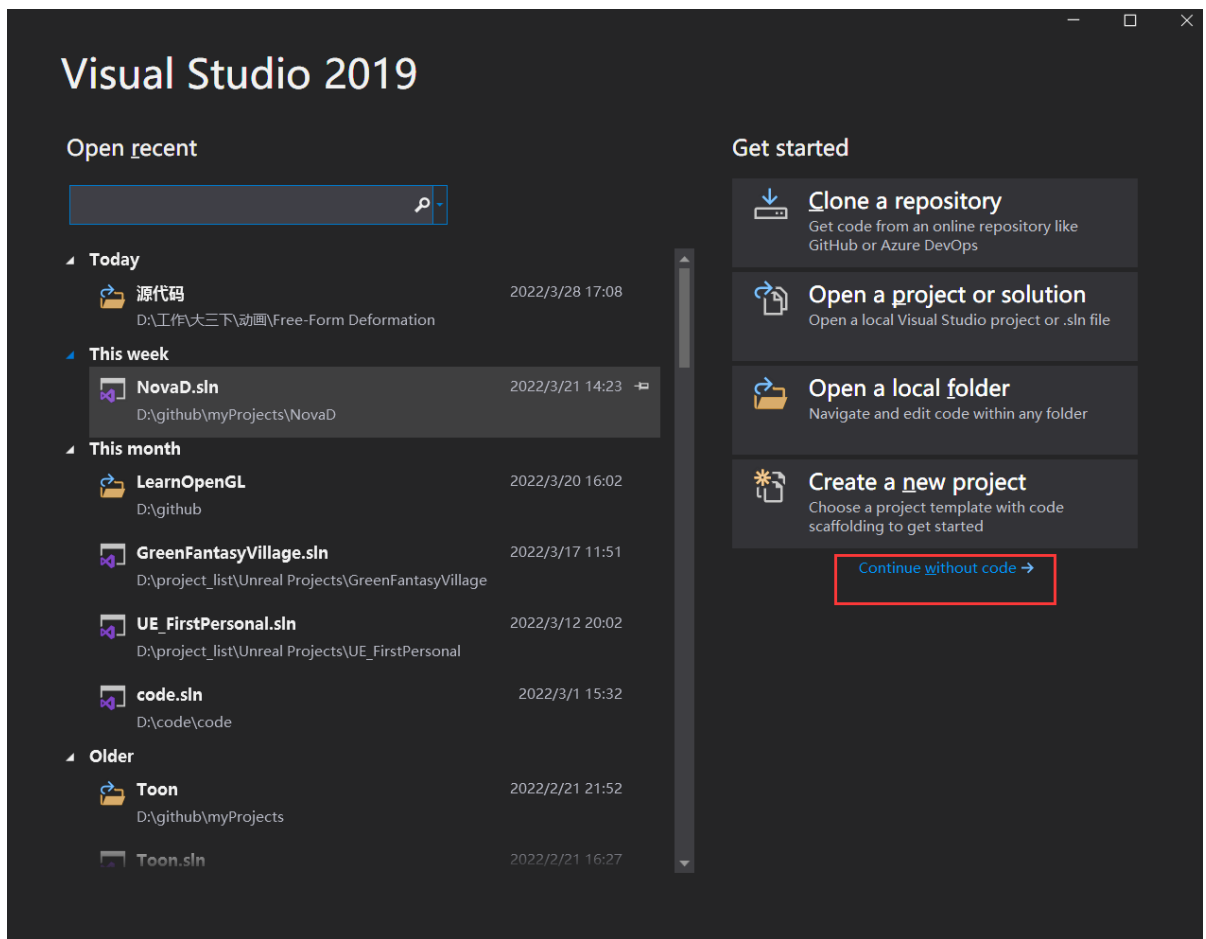


图 1 打开 Visual Studio

选择文件->打开->Cmake（如图 2 所示）。

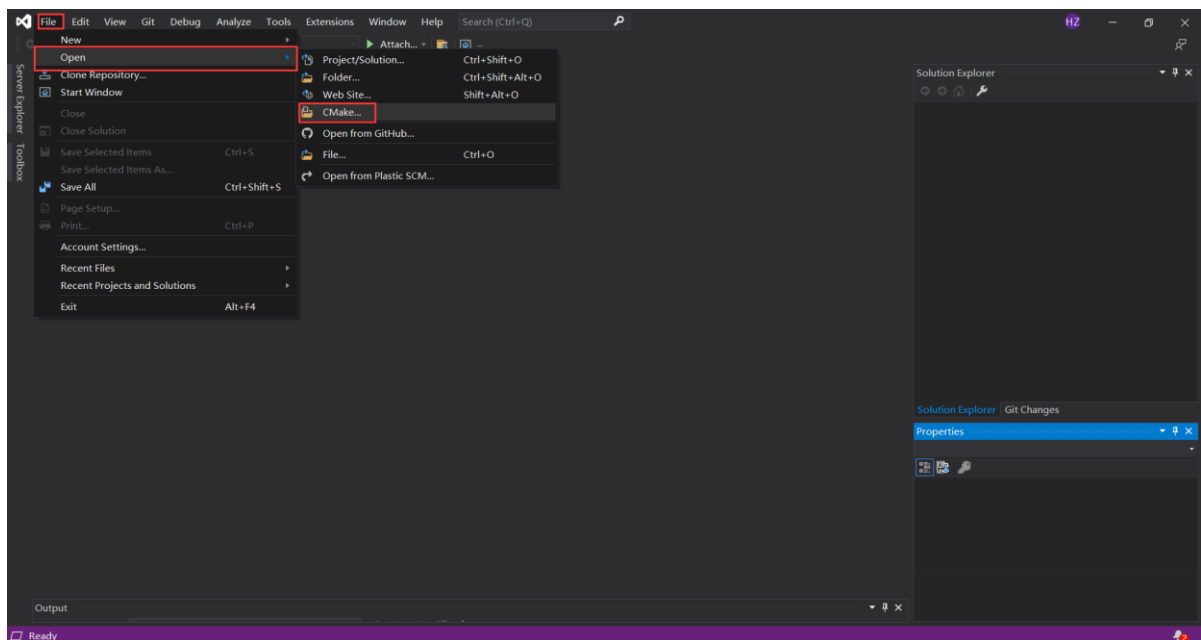


图 2 打开 Cmake 文件夹

选择提供源代码所在文件夹的 CMakeLists.txt 文件，点击打开

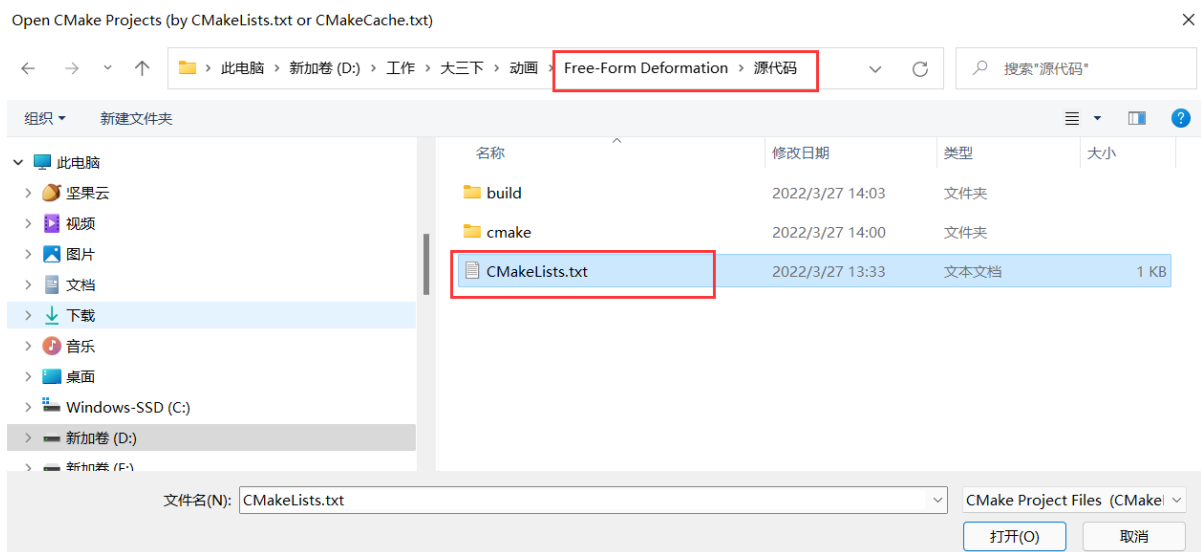


图 3 导入 Cmake 项目

观察 Visual Studio 的输出窗口，此时 Visual Studio 正在进行 Cmake 项目的配置，需要等待一段时间，直到看到 Cmake Generation finished。

注意，这一步可能会因为网络原因等待时间过久或者配置失败。

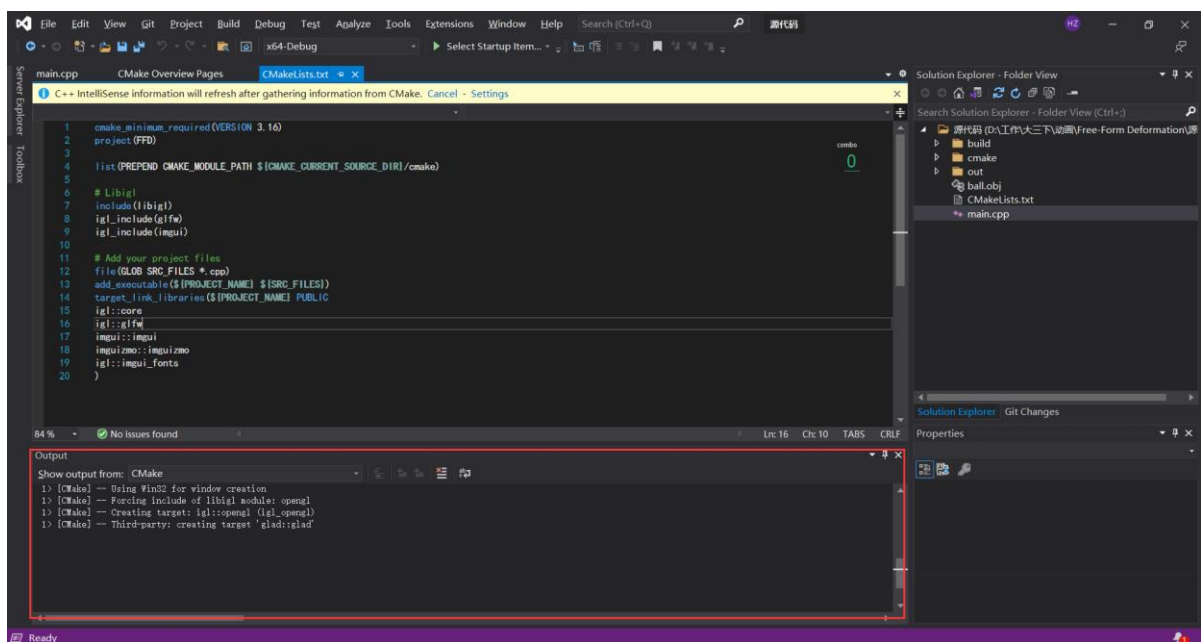


图 4 配置 Cmake 项目

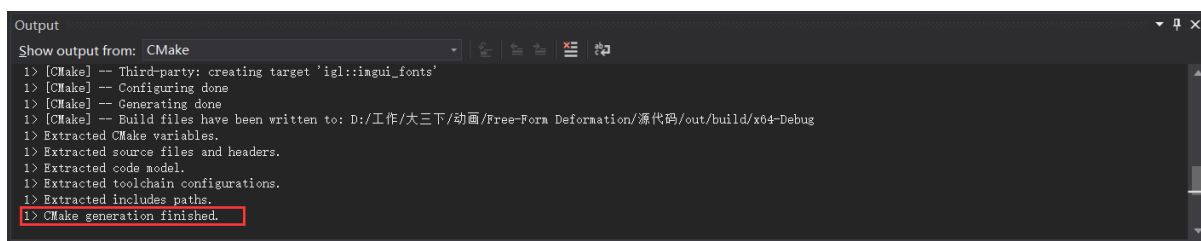


图 5 配置完成

点击 Visual Studio 上方工具栏中的三角按钮右侧的下拉键，选择 FFD.exe

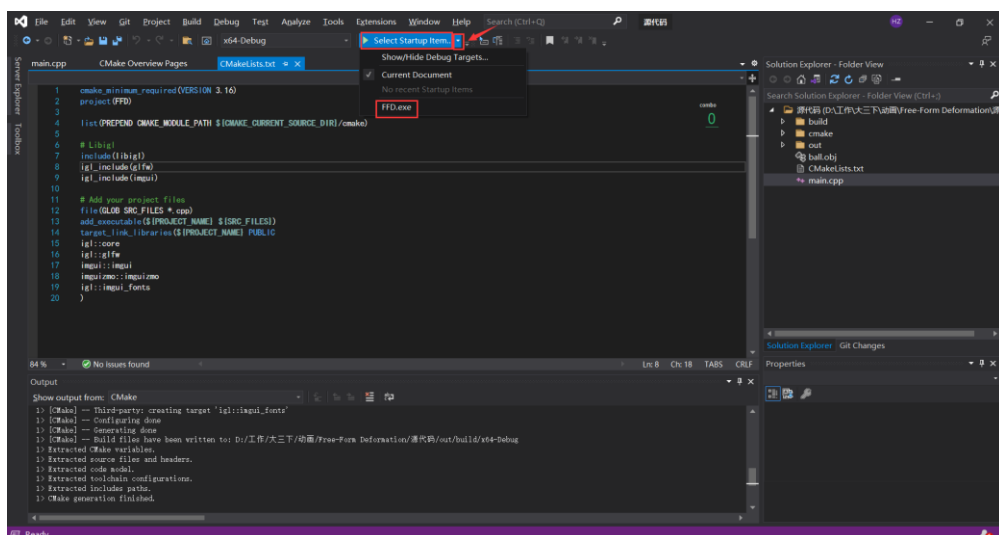


图 6 选择项目

此时直接点击运行可能会出错，这是因为缺少资源的问题，打开源代码的文件夹，将 ball.obj 拷贝到 out/build/x64-Debug(Release)/文件夹下

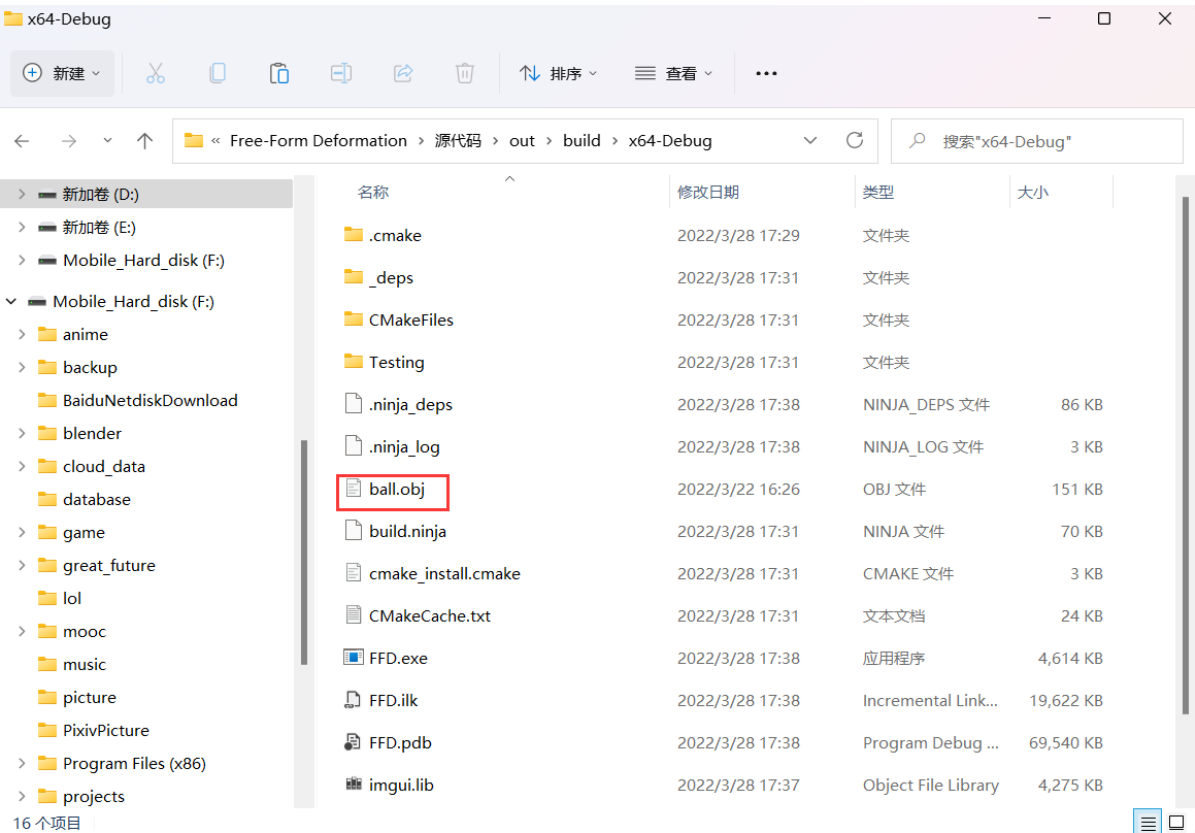


图 7 配置资源

此时点击运行能够启动该程序，如下图所示

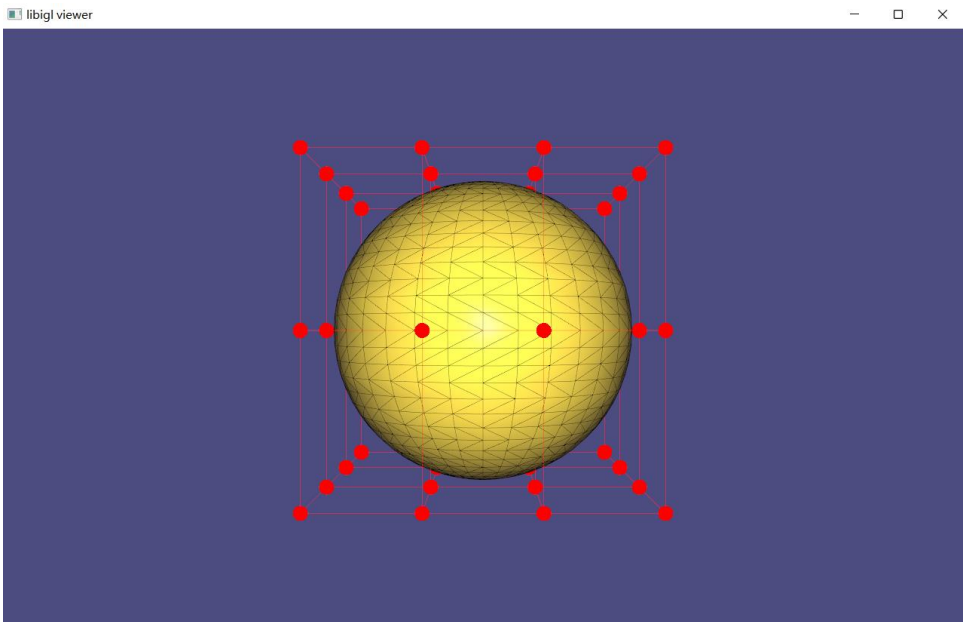


图 8 正确运行项目

2. 算法概述

2.1 坐标处理

对于模型任意顶点 $X = (x, y, z)$ ，都可以表示为以模型中某一点的原点的**相对坐标** (s, t, u) ：

$$X = X_0 + sS + tT + uU \quad (1)$$

为了便于理解，我们取 $X_0 = (x_{\min}, y_{\min}, z_{\min})$ ， S, T, U 分别为 X 轴, Y 轴, Z 轴三个方向，大小分别为 $x_{\max} - x_{\min}, y_{\max} - y_{\min}, z_{\max} - z_{\min}$ 。可以得出 $0 \leq s \leq 1, 0 \leq t \leq 1, 0 \leq u \leq 1$ 。

2.2 控制点定义

通过相对坐标，计算控制点的**绝对坐标** P_{ijk} 。

$$P_{ijk} = X_0 + \frac{i}{l}S + \frac{j}{m}T + \frac{k}{n}U \quad (2)$$

下图中，正方体线框上的**白色晶格**， $l = 1, m = 2, n = 3$ 。

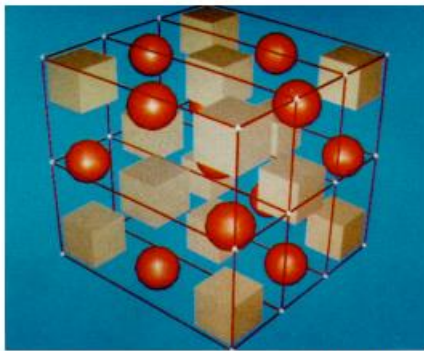


Fig.4 Undisplaced Control Points

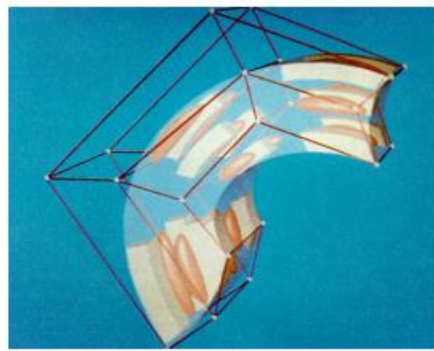


Fig. 5 Control Points in Deformed Position

图 9 参考图片

2.3 变形算法

$$X_{fd} = \sum_{i=0}^l C_l^i (1-s)^{l-i} s^i \left(\sum_{j=0}^m C_m^j (1-t)^{m-j} t^j \left(\sum_{k=0}^n C_n^k (1-u)^{n-k} u^k P_{ijk} \right) \right) \quad (3)$$

C_l^i 是组合数， $C_n^r = \frac{A_n^r}{r!} = \frac{n!}{r!(n-r)!}$ 。阶乘： $n! = 1 \times 2 \times 3 \times \cdots \times (n-1) \times n$ 。

注意 P_{ijk} 是绝对坐标，因此变形后的模型各顶点坐标 X_{fd} 为绝对坐标，不再需要转换。

3. 示例程序说明

红点为控制点，鼠标左键单击后出现 Gizmo 即可拖动。按下 R 键初始化模型。

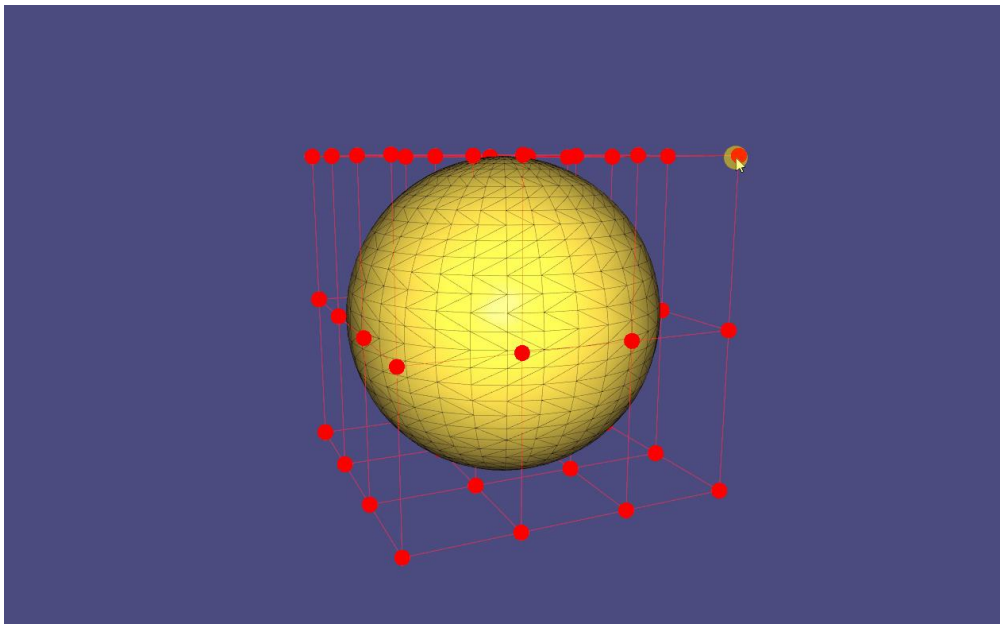


图 10 示例程序

4. 实例程序源代码简述

你不应该花大量时间阅读完整源代码！这对你完成报告没有帮助。只需参阅下方 1、2 点了解算法编写的过程即可。

1. 163 行~192 行 为初始化相对坐标 (s, t, u) ，并计算控制点的绝对坐标
2. 121 行~134 行 为计算变形后的模型各顶点坐标 X_{ffd} 。
3. 算法部分共 44 行，其余部分均为用户界面交互代码。
4. 如果你想修改源代码并编译或使用示例程序提供的框架，请仔细完成项目构建部份。

5. 再探变形算法

现在你可能还一头雾水，不明白为什么一个公式就能得到形变的效果。请冷静一下并恢复耐心，在这一部分会引导你详细了解公式。

$$X_{ffd} = \sum_{i=0}^l C_l^i (1-s)^{l-i} s^i \left(\sum_{j=0}^m C_m^j (1-t)^{m-j} t^j \left(\sum_{k=0}^n C_n^k (1-u)^{n-k} u^k P_{ijk} \right) \right) \quad (4)$$

(s, t, u) 为模型各顶点的相对坐标， P_{ijk} 为控制点的绝对坐标

6. 报告要求

1. 思考为什么这个公式可以将控制点与模型的各个顶点平滑的联系起来？
2. 观看 GAMES102 课程视频片段（P4 三次样条函数 从 1:09:54 开始至结

束)

1. $\binom{n}{r} == C_n^r$
2. <https://www.bilibili.com/video/BV1NA411E7Yr?p=4&t=4194.0>
3. 比对 Bernstein 基函数与 FFD 算法的联系与区别。
3. 观看 **P5 Bezier 曲线 B 样条曲线**
 1. <https://www.bilibili.com/video/BV1NA411E7Yr?p=5>
 2. 思考如下问题
 1. Bezier 曲线的公式?
 2. Bezier 曲线的性质?
 3. B 样条曲线的公式?
 4. B 样条曲线的性质?
 5. 如何将示例程序用 B 样条曲线改写?
 4. 结合以上问题与你的思考完成报告。
 5. 如果你还有兴趣, 可以继续了解 NURBS 曲线, 将思考写入报告。

结论分析:

列举实验中遇到的问题、解决的方法, 总结实验的收获和体会以及尚存在的问题。

实验要求:

1. 确认机器已经安装 Visual Studio2019 (或更高版本); 参见“**实验环境要求**”。
2. 根据“**实验方法**”中步骤依次学习 FFD 算法, 并达到**熟练掌握**程度。
3. 给实验中添加/修改的代码逐句添加注释, 可参考“参考书目”中教材或网络上资料。
4. 提交材料: **本次**实验(实验 5)需要填写实验报告并且提交添加注释后的源代码。实验报告内容包括实验步骤、结果图、注释后的代码以及“**结论分析**”中所述内容等。实验报告必须是在 **Microsoft Office Word** 中进行格式排版后的报告。
5. 命名规则: 学号_姓名, 注意中间为“下划线”, **未按规则命名者按未交作业处理**。
6. **本课程所有实验的报告要求集成在一个文档中, 并在课程结束后规定时间内提交即可, 即课程结束前不必分开提交“实验 1”、“实验 2”等实验材料。**

7. 提交地址: ftp://211.71.149.149/yang_meng/homework/相应目录下。

参考书目:

- 计算机动画算法与编程基础, 雍俊海 著, 2008.7 , 清华大学出版社。

教研室主任意见:

签字: _____ 年 月 日

学院负责人意见:

签字: _____ 年 月 日

注: 此表一式两份, 一份于考前交到考试中心, 一份随学生课程设计材料上交学院。