



浙江大學
ZHEJIANG UNIVERSITY

计算机图形学

dream car

3220103422 刘韬

3220103422@zju.edu.cn

2024 年 11 月 29 日

目录

1	Bézier 曲面	3
2	Polygonal Mesh	4
3	L-System	6
4	最终效果	7
5	总结	8

本次作业使用三种建模技术来对车的形状进行建模，分别是 Bézier 曲面，Polygonal Mesh 和 L-System。下面分别进行阐述

1 Bézier 曲面

Bézier 曲面是由 Bézier 曲线推广而来的，它是由一个或多个 Bézier 曲线组成的。由二维数组描述其控制点，每个控制点是一个三维坐标。Bézier 曲面的方程如下：

$$\mathbf{B}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{P}_{ij} B_i^n(u) B_j^m(v) \quad (1)$$

在 opengl 中，我们按如下方式实现了 Bézier 曲面：

```
1 void draw_Bézier(){
2     GLfloat ctrlpoints[4][4][3] = {
3         {{-3, -1.5, 0.0}, {-0.5, -1.5, 0.0}, {0.5, -1.5, 0.0}, {3, -1.5, 0.0}},
4         {{-3, -0.5, -2.0}, {-0.5, -0.5, -2.0}, {0.5, -0.5, -2.0}, {3, -0.5, -2.0}},
5         {{-3, 0.5, -2.0}, {-0.5, 0.5, -2.0}, {0.5, 0.5, -2.0}, {3, 0.5, -2.0}},
6         {{-3, 1.5, 0.0}, {-0.5, 1.5, 0.0}, {0.5, 1.5, 0.0}, {3, 1.5, 0.0}}
7     };
8     glEnable(GL_AUTO_NORMAL);
9     glEnable(GL_NORMALIZE);
10    // 2D Bézier surface
11    glMap2f(GL_MAP2_VERTEX_3, 0.0, 1.0, 3, 4, 0.0, 1.0, 12, 4, &ctrlpoints[0][0][0]);
12    glEnable(GL_MAP2_VERTEX_3);
13    glMapGrid2f(50, 0.0, 1.0, 50, 0.0, 1.0);
14    glEvalMesh2(GL_FILL, 0, 50, 0, 50);
15
16 }
```

首先我们定义了一个 4x4 的控制点数组，然后使用 `glMap2f` 函数将其映射到 3D 空间中，然后使用 `glMapGrid2f` 函数将其映射到网格上，最后使用 `glEvalMesh2` 函数绘制 Bézier 曲面。

效果如下图所示：

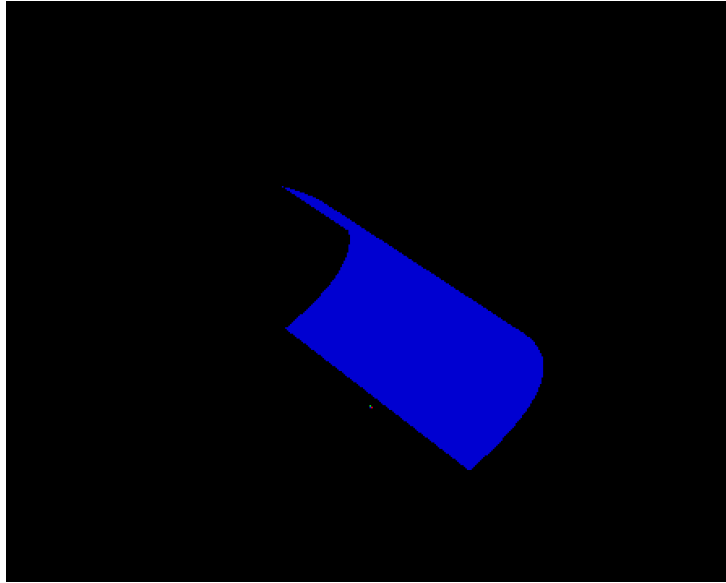


图 1: Bézier 曲面

2 Polygonal Mesh

这里我们采用导入模型的方式来实现多边形网格。根据要求，我们首先从网络上找了车的模型，然后实现函数来导入模型。导入模型的代码如下：

```
1 void loadOBJ(const char* filename) {
2     std::vector<GLfloat> vertices;
3     std::vector<GLuint> indices;
4     std::ifstream file(filename);
5     if (!file.is_open()) {
6         std::cerr << "Failed to open file: " << filename << std::endl;
7         return;
8     }
9
10    vertices.reserve(10000);
11    indices.reserve(10000);
12
13    std::string line;
14    while (std::getline(file, line)) {
15        if (line.substr(0, 2) == "v ") {
16            std::istringstream s(line.substr(2));
17            GLfloat x, y, z;
18            s >> x; s >> y; s >> z;
19            vertices.push_back(x);
20            vertices.push_back(y);
21            vertices.push_back(z);
```

```

22     } else if (line.substr(0, 2) == "f ") {
23         std::istringstream s(line.substr(2));
24         std::string a, b, c;
25         s >> a; s >> b; s >> c;
26         indices.push_back(std::stoi(a.substr(0, a.find('/'))) - 1);
27         indices.push_back(std::stoi(b.substr(0, b.find('/'))) - 1);
28         indices.push_back(std::stoi(c.substr(0, c.find('/'))) - 1);
29     }
30 }
31 file.close();
32
33 glColor3f(0.8, 0.8, 0.0);
34 glBegin(GL_TRIANGLES);
35 for (size_t i = 0; i < indices.size(); i+=3) {
36     glVertex3f(vertices[indices[i] * 3], vertices[indices[i] * 3 + 1],
37         ↪ vertices[indices[i] * 3 + 2]);
38     glVertex3f(vertices[indices[i+1] * 3], vertices[indices[i+1] * 3 + 1],
39         ↪ vertices[indices[i+1] * 3 + 2]);
40     glVertex3f(vertices[indices[i+2] * 3], vertices[indices[i+2] * 3 + 1],
41         ↪ vertices[indices[i+2] * 3 + 2]);
42 }
43 glEnd();
44 }

```

首先我们解析.obj 文件，我们主要关注 v 和 f 两个部分，其中 v 表示顶点，f 表示面。f 是由点进行描述的，部分是有四边形描述的，对于这种我们要将其拆分为两个三角形，这个函数的实现我使用了 python。然后我们将其绘制出来。由于电脑性能缘故，我只能绘制相对简单的图案，效果如下：



图 2: Polygonal Mesh

3 L-System

最后我们尝试了 L-System 来绘制车的形状。L-System 是一种递归的字符串替换系统，我们可以通过 L-System 来绘制各种各样的图案。这里我们使用 L-System 来绘制车的装饰形状。我们首先定义了一个 L-System，然后使用递归来绘制图案。代码如下：

```
1 void drawBranch(int depth, float length){
2     if (depth == 0) {
3         glutSolidCube(length);
4         return;
5     }
6
7     // 绘制当前段
8     glutSolidCube(length);
9
10    // 生成三个子块向外扩散
11    for (int k = 0; k < 6; ++k) {
12        glPushMatrix();
13        glRotatef(k * 60.0, 0.0, 0.0, 1.0); // Rotate to create 3 sub-branches
14        glTranslatef(length, 0.0, 0.0); // Translate to position the sub-branch
15        drawBranch(depth - 1, length * 0.5); // 递归绘制子块
16        glPopMatrix();
17    }
18
19    // 在 Y 轴方向上延伸
20    glPushMatrix();
21    glTranslatef(0.0, length, 0.0); // Translate along Y axis
22    drawBranch(depth - 1, length * 0.5); // 递归绘制子块
23    glPopMatrix();
24 }
```

绘制时做了更多位移和旋转操作，使得图案在画布中的位置更加合理。效果如下：

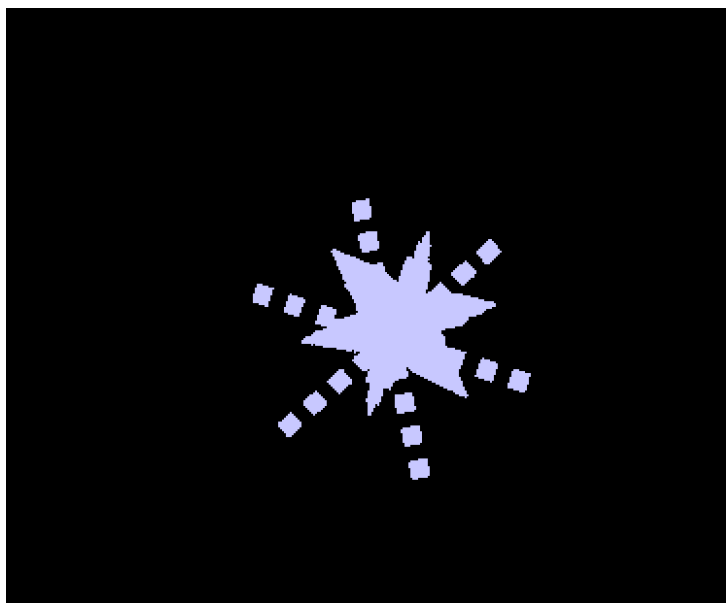


图 3: L-System

4 最终效果

本次实验还要求我们把建模得到的车子放入之前的太阳系中，为此，我对之前的模块进行了封装，在本次调用了这些模块，将车子放入太阳系中。效果如下：

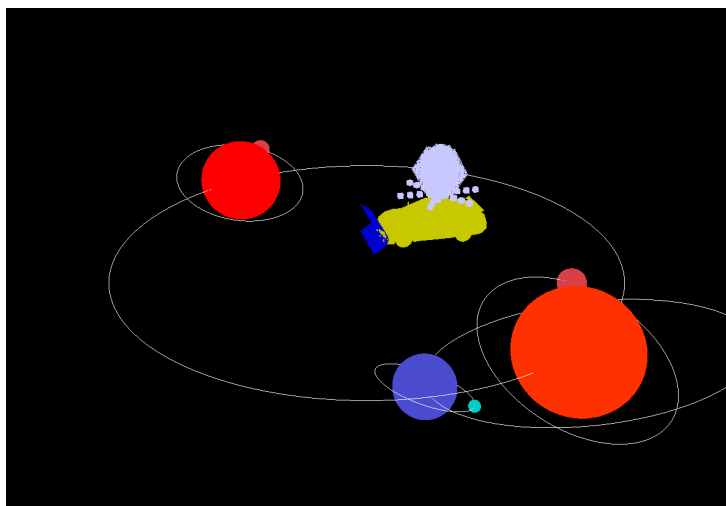


图 4: 最终效果

5 总结

本次作业的最大收获是学会了如何导入已经建模好的模型，以及对模块的封装做了实践，方便之后更好的调用，为大作业做好基础。