HW8 Report

16340164 吕雪萌

- 代码链接

实现效果

Beizer 曲线定义

贝塞尔曲线 Bézier curve 是计算机图形学中相当重要的参数曲线。 更高维度的广泛化贝塞尔曲线就称作贝兹曲面，其中贝兹三角是一种特殊的实例。

Beizer 公式

- 代码实现：

```
//bezier curve 公式
float* Bezier(vector<glm::vec2> point) {
    //beizer curve 点个数
    int num = 0;
    //beizer curve 点坐标
    float* bezierCurve = new float[10000];
    for (float t = 0; t < 1; t += 0.001, num++) {
        // x,y 坐标 = 0
        bezierCurve[num * 2] = 0;
        bezierCurve[num * 2 + 1] = 0;
        for (int i = 0, n = point.size() - 1; i <= n; i++) {
            // 常数部分 C=n!/i!(n-i)!
            float C = jiecheng(n) / (jiecheng(i) * jiecheng(n - i));
            //X
            bezierCurve[num * 2] += C * point[i].x  * pow(1 - t, n - i) * pow(t, i);
            //Y
            bezierCurve[num * 2 + 1] += C * point[i].y  * pow(1 - t, n - i) * pow(t, i);
        }

    }

    return bezierCurve;
}
```

- 自己写阶乘函数

```
// 阶乘
long int jiecheng(int x) {
    if (x == 0) return 1;
    int result = 1;
    for (int i = 1; i <= x; i++) {
        result *= i;
    }
    return result;
}
```

动态呈现

- 实现动画循环播放，中间停顿1秒。

```
// 动态呈现效果
        if (point.size() >= 3) {
            animation(point);
            time += 0.001;
            if (time > 1) {
                Sleep(1000);
                time = 0;
            }
        }
```

- animation 为动态呈现的实现代码：递归实现

```
// 动态呈现
void animation(vector<glm::vec2> vertex) {
    float animationVertex[10000];
    int n = vertex.size();
    if (n == 1) return;

    vector<glm::vec2> next = vector<glm::vec2>();
    for (int i = 0; i < n - 1; i++) {
        float tempx = (1 - time) * vertex[i].x + time * vertex[i + 1].x;
        float tempy = (1 - time) * vertex[i].y + time * vertex[i + 1].y;
        glm::vec2 temp = glm::vec2(tempx, tempy);
         animationVertex[i * 2] = tempx;
         animationVertex[i * 2 + 1] = tempy;
        next.push_back(temp);
    }

    glGenVertexArrays(1, &VAO);
    glBindVertexArray(VAO);
    glGenBuffers(1, &VBO);
    glBindBuffer(GL_ARRAY_BUFFER, VBO);

    glBufferData(GL_ARRAY_BUFFER, 2 * sizeof(float) * next.size(),  animationVertex,
GL_STATIC_DRAW);
    glVertexAttribPointer(0, 2, GL_FLOAT, GL_FALSE, 2 * sizeof(float), (void*)0);
    glEnableVertexAttribArray(0);
```

```
    glPointSize(10.0f);
    glDrawArrays(GL_POINTS, 0, next.size());

    glPointSize(1.0f);
    glDrawArrays(GL_LINE_STRIP, 0, next.size());

    animation(next);
}
```