

# 三维CAD建模课程作业

---

## 任务

- 1) 定义半边数据结构;
- 2) 基于半边数据结构, 实现五个基本欧拉操作;
- 3) 基于欧拉操作, 实现扫掠操作 (必须允许二维区域带多个内环), 并将基于扫掠操作构建的实体模型进行真实感图形显示。

## 结果展示

以自己名字首字母为例

实体模型:



线框模型:



多孔:



## 实现的主要功能

(1)通过课上讲解的知识点以及相关伪代码案例实现了基于B-rep的数据结构

(2)实现了6个最常用的欧拉操作,以及扫掠操作

```
mvfs():定义一个体、一个面、一个外环、一个点
mev():构造一个新点,同时构造一条连接新点与一给定点的边
mef():构造一条新边,同时构造一个新面
kemr():消去环中的一条边,构造一个内环
kfmrh():删除一个面,并将其定义为一个内环,进而在体中构造一个柄,或将两个物体合并一个物体
semv():在给定边上构造一个新点,同时将边分裂为两条边
sweep():基于B-rep及相关的欧拉操作进行扫成操作
```

(3)利用Qt的OpenGL库进行图形绘制,将基于扫掠操作的实体模型进行可视化(可以显示线框模型和实体模型)

(4)支持↑、↓、←、→转向观察等交互操作

## 使用指南

请将exe文件放置在与输入文件Input.txt文件同级别目录下

### 输入

- 以文件txt进行输入,具体格式参考input.txt文件

```
1  polygon_num          //所有环的数量
2  5
3  point_num            //每个环包含点数量及点坐标
4  4 8 6 7 4
5
6  -20.0 20.0 0.0       //第一个环的点坐标
7  -20.0 -20.0 0.0
8  50.0 -20.0 0.0
9  50.0 20.0 0.0
10
11 -10.0 10.0 0.0       //第二个环的点坐标
12 -10.0 -10.0 0.0
13 10.0 -10.0 0.0
14 10.0 -5.0 0.0
15 -5.0 -5.0 0.0
16 -5.0 5.0 0.0
17 5.0 5.0 0.0
18 10.0 10.0 0.0
19
20
21 15.0 10.0 0.0        //第三个环的点坐标
22 15.0 -10.0 0.0
23 25.0 -10.0 0.0
24 25.0 -5.0 0.0
25 20.0 -5.0 0.0
26 20.0 10.0 0.0
27
28 30.0 10.0 0.0        //第四个环的点坐标
29 30.0 -10.0 0.0
30 40.0 -10.0 0.0
31 42.0 -12.0 0.0
```

```

32 44.0 -10.0 0.0
33 40.0 -6.0 0.0
34 40.0 10.0 0.0
35
36 32.5 5.0 0.0 //第五个环的点坐标
37 32.5 -5.0 0.0
38 35.0 -5.0 0.0
39 35.0 5.0 0.0
40
41 0.0 0.0 50.0 //扫成向量

```

- 格式为：第1行底面的环数量polygon\_num（包括外环和所有内环）；
- 然后输入每个环的数据，每个环的数据格式为：第1行为1个整数m，表示该环有m个顶点，之后的m行按逆时针（俯视角）顺序每行输入1个点的坐标（浮点数），x、y、z坐标以空格分隔；
- 最后，输入扫掠操作的向量。

## 输出

### 采用多种方式的输出

#### 1.图形输出

按↑, ↓控制模型绕y轴旋转，按←, →建控制模型绕x轴旋转

鼠标右键唤醒菜单栏

#### 2.控制台txt输出

控制台输出为扫成构造的实体的所有表面及其中的环、组成环的坐标。

## 开发环境和配置

操作系统: Windows10 x64

集成开发环境: Qt OpenGL

编译器: vs2017

## 算法详细

详细参考代码

```

1 //mvfs操作，构造一个实体，得到一个面一个点，一个环
2 Solid* mvfs(Point *p)

```

```

1 //构造一个新点，同时构造一条连接新点与给定点的边
2 HalfEdge* mev(Point *oldPoint, Point *newPoint, Loop *lp);
3 //判断v1 点是否在solid中
4 //建立半边结构，添加到lp中
5 //判断是否是lp的首条边
6 //不是第一条边，则先建立he2相关连接

```

```

1 //在两给定点间构造一个新边，同时构造一个新面，以及一个外环
2 Loop* mef(Point *p1, Point *p2, Loop *lp);
3 //外环根据右手法则指向体外
4 //将v2为起点的环作为外环(原lp)，he1在外环中；v1为起点的环作为内环（newLp），he2在内环中
5 //返回的为内环

```

```

1 //消去环中的一条边，构造一个内环(innerLp)，原环为外环(lp)
2 Loop* kemr(Point *p1, Point *p2, Loop *lp);
3 //删除边，分四种情况 边在中间、末尾、顶端、单边

```

```

1 //删除一个面，将其定义为另一个面的内环，在体中构造一个柄/或将两物体合并
2 void kfmrh(Loop *OuterLp ,Loop *InnerLp);
3 //先判断环是否正确
4 //删除面，分四种情况

```

```

1 //Sweeping operation
2 void sweeping (Face *face ,Point *sweepPoint);
3 //平移扫成操作,轨迹线为一条直线，轨迹线为原点和给定的扫描点组成的直线
4 //遍历每个lp

```

## 项目结构

