

# 计算机图形学作业 1 报告

515110910011 李晗东 1737078184@qq.com

## 一、 题目描述

作业 1: 基于 OpenGL 编程实现

写一个 Loader 读入三维模型 (.max、.3ds 或.obj 格式) 并绘制, 导入 FKR 模型 (20 分)

分析并模拟 FKR 行动策略, 建立 FKR 运动计算模型 (20 分)

模拟 FKR 飞行前进、搜索、随机运动、躲避的运动, 仿真度尽量高 (40 分)

作业报告 (20 分)

## 二、 基本设定

1. 这次作业采用的无人机模型为 obj 格式
2. 假设当目标在 FKR 感知范围内时, FKR 自带的摄像头可以给出目标的坐标参数, 当在感知范围内有物体靠近时, FKR 自带的传感器也能 给出一个较短时间内物体的两次坐标。
3. 假定给定空间中没有障碍物。

## 三、 实现简述

### 1. Loader 的实现

概述:

obj 文件是以纯文本的形式存储了模型的顶点、法线和纹理坐标和材质使用信息。obj 文件的每行格式如下: 前缀 参数 1 参数 2 参数 3 ...其中, 前缀标识了这一行所存储的信息类型, 参数则是具体的数据。因此 Loader 的实现就是对 obj 文件的每一行进行处理, 根据每一行的前缀和数据将数据保存到不同的集合中。

相关代码说明:

本部分的代码在 ObjLoader.h 与 ObjLoader.cpp 中, loader 的实现大部分代码参考网上教程。

### 2. 类的设计

这次作业中设计的类及其简述如下表

类名称	简述	代码文件
BaseFloor	参考平面, 用于模拟地板	BaseFloor.h
Camera	用户视角的 camera, 便于视角转换控制	Camera.h
Drone	无人机的类, 其中包含了无人机的基本参数与各种飞行模式的实现	Drone.h Drone.cpp
MarkSphere	用于标记起始点与终点的小球	MarkSphere.h

其中大部分飞行策略都在 Drone.h 与 Drone.cpp 中实现。

### 3. FKR 的运动计算模型概述

在一定的时间间隔下, 将 FKR 的飞行划分为三部分:

加速度改变——>速度改变——>位置改变

其中加速度改变包含了三种情况:

- a. 在一定大小范围内, 生成方向完全随机的加速度
- b. 在一定大小范围内, 指定一个向量, 生成一个与该向量夹角小于  $90^\circ$  的加速度

c. 在一定大小范围内，给定两个点，以这两个点作为对角顶点构成一个封闭空间，生成一个在物体靠近封闭空间边缘时能够使其减速的并回到空间中间的加速度速度改变包含两种情况：

- a. 旧速度加上变化量后如果未超过范围限定，则更新速度
  - b. 旧速度加上变化量后如果超过了范围限定，则取范围限定的速度
- 位移改变仅仅是将旧位移加上新的偏移量。

相关代码说明：

加速度的改变在 Drone.cpp 的 `changeAcc()`, `changeAccWithTargetDirect()`, `changeAccWithLimitSpace()` 中实现。

速度的改变在 Drone.h 的 `changeSpeed()` 中实现。

位置的改变在 Drone.h 的 `changePos()` 中实现。

#### 4. FKR 的飞向目的地的策略

概述：

FKR 得到目标坐标时，会计算从目标位置到当前位置的向量，并以此为参数调用指定方向向量的加速度生成函数，再根据当前加速度改变速度，再根据速度改变位移。

相关代码说明：

本部分的代码在 Drone.cpp 的 `flyToPos2()` 函数中实现。函数首先计算当前位置与目标位置之间的方向向量，并随机生成与此方向向量夹角超过  $30^\circ$  但小于  $90^\circ$  的加速度（为了使得随机运动更明显），再根据当前加速度改变速度，并根据速度改变位移。

#### 5. FKR 的快速接近目的策略

概述：

该运动模式模拟 FKR 攻击目标时的飞行表现：快速直线飞向目标。

相关代码说明：

本部分的代码在 Drone.cpp 的 `flyToPos()` 函数中实现。函数首先计算当前位置与目标之间的方向向量，并计算出在最大运动速度的情况下与该方向向量平行的速度向量，直接改变速度，并根据速度改变位移。在达到目标位置之后停下。

#### 6. FKR 的搜索策略

在一定坐标位置范围内，FKR 会采取与悬停类似的策略，在给定空间范围内随机生成加速度，以达成随机飞行搜寻目标的效果。

相关代码说明：

本部分的代码在 Drone.cpp 的 `searchInArea()` 函数中实现。函数在给定的空间范围内调用限制空间范围的加速度生成函数，再根据当前加速度改变速度，根据速度改变位移。

#### 7. FKR 的随机运动策略

FKR 悬停时，会随机在 XYZ 上产生一定的位移，但总体不会远离悬停点一定的距离。

相关代码说明：

本部分的代码在 Drone.cpp 的 `hoverAtPos()` 函数中实现。函数以悬停点为参数，首先在以悬停点为中心构建一个封闭立方空间，再以该空间的两个对角点作为参数调用包含空间限制的加速度变化函数，再根据当前加速度改变速度，根据速度改变位移。

其中加速度的生成的逻辑是检测当前某个方向的位置是否超过了限制，如果超过了则

#### 8. FKR 的躲避策略

概述：

当有物体以超过某个速度快速接近时，传感器会给出最近两次物体的坐标，FKR 会计算出物体运动的方向向量，并在与该向量垂直的面上随机选择一个点（有一定的范围限制），以该点为目标进行飞行。

相关代码说明：

本部分的代码在 Drone.cpp 的 escapeFromPos() 函数中实现。函数以两个点 a, b 的坐标为参数（假定采集 b 的时间晚于 a），首先计算出物体运动的方向向量 x，再随机生成一个向量 y，与方向向量进行叉乘，得到垂直于方向向量的一个向量 z，将 z 进行单位化后乘以一定范围内的随机整数得到新的变量 v，最后将 v 与坐标 b 相加，即得到闪避目标的坐标。然后调用 flyToPos2，使得无人机飞向此处完成闪避。（备注：为了显示效果当前实现将在闪避后目标处停止）

### 四、 创新设计

1. FKR 靠近目标时，不采用最简单的直线路径而是采用随机的曲线路径，这样可以减少路径被预测而被击落的概率，同时如果目标在人群中可以达到隐藏目标的目的。同时针对要迅速靠近目标并发动击杀的场合则采用了直线路径，更符合需求。
2. 要使得 FKR 产生随机位移时，不采用每一帧随机生成位移偏移量的设计而是采用变化三个方向加速度的方法，同时只有在一定的位移间隔后才会再次变化加速度。这是因为加速度的变化可以使速度的变化连续，因而也更合理，设置较长的时间间隔可以使得随机位移更加明显。

### 五、 测试方法

使用 visual studio 打开项目并编译运行后，采用键盘按键（小写英文输入法）控制，按键与对应功能如下表：

按键	功能
a	退出程序
s	强制重置 Drone 的状态到初始位置
d	使得 Drone 以直线快速接近目标
f	使得 Drone 以曲线接近目标
g	使得 Drone 在当前位置悬停
h	使得 Drone 回避一次来自(0,0,0),(1,1,1)的攻击
j	使得 Drone 在(0,10,0),(100,40,100)的封闭空间内搜索

### 六、 其他说明

1. 这次的作业的编写环境如下：  
Windows 10 x64  
Visual studio 2015（编译选项为 release x86）  
其中 OpenGL 库采用 vs 提供的 NuGet 工具获得（使用方法：工具-NuGet 包管理器-管理解决方案的 NuGet 程序包，搜索 openGL）；visual studio 的编译选项不可以选择 debug 模式，因为此模式会很卡顿，完全无法正常渲染。
2. 这次作业采用的 obj 模型从 free3Dmodel 获得，obj 原文件已经附在项目 lab1/data 文件夹下。

3. 项目的 Github 链接如下: <https://github.com/NutterLee/CG/tree/master/lab1> , 如编译遇到报错或其他运行问题希望可以邮件联系我。
4. 参考资料:
  1. <https://www.cnblogs.com/Anzhongliu/p/6092048.html>
  2. <https://cloud.tencent.com/developer/article/1141806>
  3. [https://blog.csdn.net/xie\\_zi/article/details/1911891](https://blog.csdn.net/xie_zi/article/details/1911891)