

图形学第二次大作业报告

515110910011 李晗东

一、 题目概述

在作业 1 基础上添加场景建模、视觉效果及动画

优化 FKR 行动策略，自动决策，并根据决策实现加速度、速度和位置的变换
(20 分)

实现目标对象、Anti-FKR 的模型导入和运动决策，实现场景动画 (35 分) ;

构建故事发生的场景 3D 模型，并设置光照 (25 分)

作业报告 (20 分)

二、 基本设定

1. 当目标在 FKR 的感知范围内是，FKR 可以发现目标并得到目标的参数
2. 假定给定空间中，从 FKR 到小人的运动空间内没有障碍物
3. 假定人只在广场的平台上走动
4. 假定广场上装有 antiFKR，它是一种远距离电磁脉冲炮，即发射的并不是实体而是干扰信号，能够有效的使得某个方向的一定范围内的电子设备瘫痪
5. FKR 的运动缺陷分析：
 - a. FKR 的目标很明确，是人的头骨，因此可以着重保护人的头骨来进行防范。
 - b. FKR 可以躲避低速的实体攻击，但这意味着对于无实体的攻击（如强电磁信号）FKR 没有躲避能力。（本次实现采用这种方式）
 - c. FKR 通过人脸识别+红外追踪来锁定目标，可以通过在空气中释放干扰烟雾和干扰热源来使得 FKR 失去目标。

三、 实现简述

1. 用到的库：
 - 1) Assimp v3.3.1
 - 2) SOIL2
 - 3) glm
 - 4) GLFW
 - 5) GLEW

2. 类的设计

这次作业中设计的类及其简述如下：

类名称	简述	代码文件
Mesh	用于处理 Assimp 加载的模型数据，此部分代码是参考网上的教程完成的	Mesh.h
Drone	无人机的基本实现，包含了无人机的基本参数和运动模式的实现	Drone.h Drone.cpp
Camera	用于处理鼠标对视角的改变	Camera.h
Human	人物的基本实现，包含了人的基本参数和运动模型的实现	Human.h Human.cpp
Model	使用 Assimp 将库导入，实现了绘制的功能	Model.h

Mortar	antiFKR 的基本实现，包含了 antiFKR 的基本参数和运动模型的实现	Mortar.h Mortar.cpp
Shader	读取 shader 文件，并对其编译，此部分的代码在实现时参考了网上的教程	Shader.h

3. FKR 运动模型

与作业 1 中相同，FKR 的飞行划分为三部分：

加速度改变—>速度改变—>位置改变

位置的改变通过绘制的时候体现出来，加速度和速度的信息则保存在对象中用于下一次对位置的修改。

1) FKR 的加速度生成策略：

每次调用加速度生成函数时，在三个坐标上都随机生成一个加速度，这些加速度的大小在受最大加速度的限制，三个相互垂直的加速度合成了这次运动的加速度。

在此基础上，可以通过添加其他参数实现如下三个加速度生成函数：

函数名	描述	使用场景
GenerateAcc	最基本的加速度生成函数，生成大小方向完全随机的加速度	在无穷大空间巡逻和搜索目标时采用
GenerateAccWithTarget	指定一个方向向量，生成一个与该向量夹角小于 90° 的大小随机的加速度	当有物体快速接近，FKR 需要躲避时
GenerateAccWithLimit	给定两点，以他们为对角线构成一个立方体空间，当物体靠近空间边缘时将生成将其拉回空间内部的加速度	在给定空间中悬停或巡逻时

2) FKR 的速度改变策略

在不同的场景下加速度改变后，会根据对应的加速度更新速度，注意如果更新后的值超过了速度的上限，将只采取上限的速度。

3) FKR 的飞行策略

在演示的场景中，FKR 将在场景上空指定区域内一直盘旋搜索目标，如果发现了目标就飞向该目标。

实现方法：采用上表中限定空间位置的加速度，使得 FKR 在一定空间中随机运动，当 FKR 的距离与目标的距离小于一定值时，切换 FKR 的模式，将 FKR 设定为以最大速度飞向目标的模式。

4) FKR 的坠落

在演示的场景中，FKR 若被 antiFKR 的电磁脉冲炮击中，会失去飞行能力原地下落。下落的实现就是通过将 Y 方向的加速度恒定设置为指向地面的某个值来实现。

4. 人物运动模型

这次作业中实现了人物的运动模型如下：

人物会在固定的坐标点之间走动，因为人走路的时候会走走停停，人物也不是连续的在几个坐标点中走动，而是有概率会停下来。同时在改变走路的目标方向时会身体前方朝向目标方向，与人真实的运动情况较为符合。

实现方法：

人物每次运动的时候都会做一次随机判断，只有在判断成立的情况下才会继续运动，否则就停止不动。当人物的目的地坐标更改以后，人体也会随之转向。

5. antiFKR 的运动模型

antiFKR 被设置为固定点可旋转的电磁炮台，可以对某个方向发射电磁脉冲炮（无实体，特效暂未实现），来达到使得该方向一定距离内的电子设备瞬间瘫痪的目的。因为电磁波的传播速度接近光速，且无实体，所以几乎没有被防范的可能。同时由于可能会有很多其他正常的无人机在空中飞行，antiFKR 只有在某个无人机靠近需要保护的目标太近的时候才会进行攻击。

实现方法：

在演示的场景中，antiFKR 会随机旋转炮口，但发现有无人机距离目标太近的时候会锁定目标（将炮口对准目标）并发动攻击。如果攻击成功目标将瞬间瘫痪并下落。

6. 光照设置

实现方法：在 main 中将光源的位置传入 shader，shader 根据光源的位置和预设的参数分别计算出环境光（环境光强度已预设好）、漫反射光和镜面反射光，将他们叠加后与物体的颜色混合，输出对应的颜色。

7. 场景布置

这次作业采用了网上下载的场景，由于免费版没有对应的贴图与纹理所以采用了简单的上色与其他模型区分。

场景设置在小镇上的广场上，广场上安装有 antiFKR，目标这在广场上反复走动。

四、 创新设计

1. FKR 运动时都是通过加速度来调整运动状态，更符合物理逻辑，运动也更流畅。
2. 采用 shader 进行着色和光照渲染，可以更好的控制渲染效果。

五、 测试方法

使用 Visual Studio 2015 打开 Lab.sln 文件，选择 release X86 并点击本地 Windows 调试器，即可运行代码。

注：直接点击 Release 文件夹下的 Lab.exe 无法正常渲染出模型。

六、 参考资料

1. Modern OpenGL 3.0+ Tutorials
<https://www.youtube.com/playlist?list=PLRtjMdoYXLf6zUMDJVRZYV-6g6n62vet8>
2. LearnOpenGL
<https://learnopengl-cn.github.io/>

七、 其他说明

1. 项目的 github 链接如下: <https://github.com/NutterLee/CG/tree/master/Lab>, 如编译错误或者遇到其他问题希望邮件联系我。
2. 项目的编写环境如下:
Windows 10 X64
Visual studio 2015 (选择 release X86 编译)
3. 笔记中的证明在第 22 页 pdf 中, 与下图证明一致:

由插值公式:

$$C(t) = P_0 \frac{n!}{0!(n-0)!} (1-t)^n + P_1 \frac{n!}{1!(n-1)!} t(1-t)^{n-1} + P_2 \frac{n!}{2!(n-2)!} t^2(1-t)^{n-2} + \dots$$

将t的多项式的常系数简化, 记做k1, k2, k3...

$$C'(t) = -k_1 * P_0 * n(1-t)^{n-1} + k_2 * P_1 * (1-t)^{n-1} - k_2 * P_1 * t(1-t)^{n-2} + \dots$$

$$+ k_{n-1} * P_{n-1} * (n-1)t^{n-2} * (1-t) - k_{n-1} * P_{n-1} * t^{n-1} + k_n * P_n * n * t^{n-1}$$

当t=0, 即起点时:

$$C'(t) = -k_1 P_0 n + k_2 P_1$$

而由之前简化, 有k1*n=k2, 所以切矢量的方向C'(t)=n(P1-P0)与第一条边重合,

同理, t=1时, C'(t)=n(Pn-Pn-1), 所以切矢量方向与最后一条边重合。