

“魁地奇桌球基础建模及简单动画”设计报告

沈斯杰 5130379036

2015. 10. 26

一、项目简介

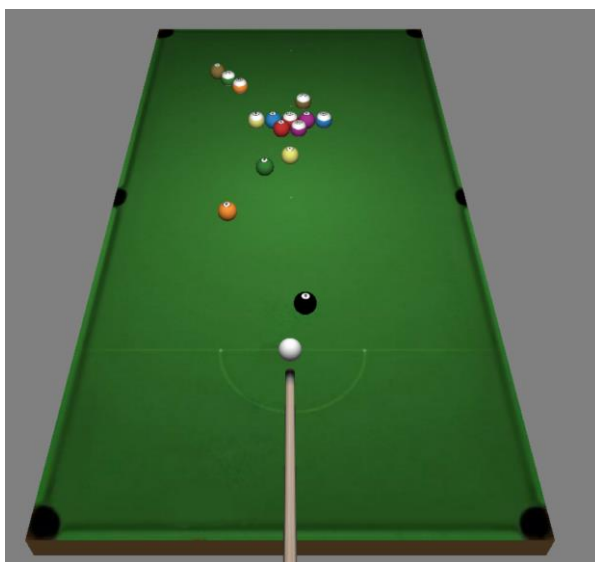
(一) 项目要求

此次项目的具体要求如下：

- (1) 设计搭建桌球模型（15 分）；
- (2) 桌面放置 6 个静止的“鬼球”，初始状态是静止（后续作业中鬼球在受到其他球碰撞时会滚动并由于摩擦力停止）（10 分）；
- (3) 桌面还有 6 个任意缓慢移动的“游走球”，沿任意路径缓慢滚动，（后续作业中游走球会在缓慢滚动时避让其他球，只有在被撞击速度加快时会撞到其他的球）；（20 分）；
- (4) 1 个“金色飞贼”小球，桌面上空随机飞行，但会间歇性落到平面上休眠，在游戏中击中该球会压倒性地取得高分；（20 分）；
- (5) 1 个白色母球，受球杆打击会沿球杆运动方向以某种速度滚动，并撞击其他球；（15 分）；
- (6) 设计游戏模式和积分规则；（10 分）；
- (7) 详细的设计报告。（10 分）

以上要求基本全部实现。

(二) 游戏截图



(三) 游戏基础设置

1. 游戏的按键设置如下：

按键	功能
A/D	转动杆的方向
Shift	推杆
空格	为杆蓄力
J/L	左右移动球台
U/O	上下移动球台
I/M	前后移动球台
K	复原最初位置和视角
↑ / ↓ / ← / →	转移视角

2. 球台上有 15 个球，其中球号为 5/6/7/13/14/15 为移动的“游走球”，黑 8 为“飞贼”。初始状态下，“鬼球”会在桌面平面内随机移动，“飞贼”会在桌球及其上空随机移动，经过一段时间后停在桌面上休息，如此循环。

(四) 目前已经实现的与尚未实现的工作

1. 除了以上所提到的部分，目前已经实现的工作还有以下部分：

- (1) 场景的基础建模
- (2) 球的基本运动
- (3) 球杆的蓄力与推杆
- (4) 纹理与光照的设置
- (5) 游戏规则的设计
- (6) 键盘控制，后面有望实现鼠标控制

2. 尚未实现的工作：

- (1) 物理运动模型，包括球的旋转、球的碰撞(球与球之间、球与边间之间)，目前仅仅是以停止替代
- (3) 球随机路线的优化
- (4) 游戏模式(双人交换、游戏积分)的实现
- (5) 场景的进一步优化(美化背景)

二、游戏规则

此游戏预期达到以下的效果。

(一) 游戏流程

类似于美式桌球。本游戏为双人交换进行，除下面罗列的特殊情况外，每人一次推杆的机会。开局后，玩家第一次进洞的球色作为该玩家本局游戏的球色(全色、半色)。当某玩家进自己的球后，获得一次再次推杆的机会。

(二)胜败条件

1. 本游戏一局的限定时间为 15 分钟（或许可以自己设置），在此过程中，如果一方将自己的球全部打进，且最后把黑球打进，首先达到此目标的玩家获胜。
2. 在自己本色球未打完的情况下，直接打入黑球，判该玩家输。
3. 若超过 15 分钟后，胜负仍未定，那么将按照积分定输赢，积分高的获胜。积分获得见下表：

事件	积分
打进非本色球（除黑 8）	-5
未打中本色球*	-10
白球进洞*	-10
打进本色球（“鬼球”）*	+5
打进本色球（“游走球”）*	+10
打中黑 8（非打进）	+15

4. 上表中，处罚事件中标记为*的给对手多一次推杆机会，奖励事件中标记为*的自己多一次推杆机会。
5. 最后打黑 8 指袋（可能实现？）

三、代码架构

目前的代码文件中，有以下文件：

头文件	.cpp 文件	职能
background.h	background.cpp	绘制（底层）背景
draw.h	draw.cpp	绘制模型(包括光照等)
	Texture.cp	绘制纹理
entity.h	entity.cpp	球和球杆的实体类
	billard.cpp	游戏逻辑
	main.cpp	初始化和入口函数

(一) 底层背景

这部分代码主要参考了 NeHe 教程中的代码，框架也是该教程中提供的，这部分代码与上层做到了解耦，基本上不动。它提供三个重要的接口，CreateGLWindow 初始化窗口，LoopGLWindow 每次刷新帧，KillGLWindow 作为程序结束后对窗口的析构。

其他底层需要用的函数，用户在上层写好后通过句柄进行传入。为此，特地写了一个结构体 Handlers 传入所需要的函数，包括初始化、绘制和贴纹理。

(二) 绘制

这部分代码作为使用图形库最多的代码是将后台逻辑进行显示。这部分代码提供 drawGLScene 和 loadGLTexture，作为句柄传入底层中。

loadGLTexture 中使用 glut 进行贴纹理，比较常规。其中图片选用了网上的一些素材。

`drawGLScene` 中需要设置光照、渲染不同模型、并与游戏逻辑进行交互。这部分代码中，对于视角的转移、以及物体运动是坐标轴的变化应用有很多。尤其是坐标轴的变化，为了达到旋转和移动的效果，在实际实现中着实头疼。对于按键的处理也在这部分中，当然，后续可能实现鼠标控制。

(三) 主文件

在主文件中，主要分为初始化函数 `initGL` 和程序的入口函数 `WinMain`。

初始化函数中进行了这个程序的一些设置，包括打开纹理、打开光照、二次图形的建模等等。

入口函数中，制作句柄传入底层进行渲染即可。