

Air Hockey 项目报告

黄予 2013011363 计 34

一、项目描述

在 Visual Studio 平台上实现 Air Hockey 小游戏，要求正确绘制场景，正确计算 puck 的运行轨迹，实现镜头转动与鼠标控制等功能。

该项目由本人独立完成。

二、实现功能

1. 绘制场景
2. puck 与 wall 之间的碰撞与反弹
3. puck 与 mallet 之间的碰撞与反弹
4. 通过鼠标控制玩家的 mallet
5. 电脑 AI：简单与困难
6. 镜头转动：上下左右
7. 额外完成：增加困难的电脑 AI；镜头转动增加向上与向下移动

三、效果与原理

1. 场景绘制

使用 opengl 进行实时场景绘制，效果如下图所示：



2. 碰撞

判断 puck 与 wall 是否碰撞：通过计算圆柱到矩形的距离以及圆柱的柱心在矩形所在面的投影点是否位于矩形内来判断。

计算 puck 与 wall 碰撞后的方向：以矩形面的法向量为轴，计算碰撞前方向的轴对称方向即可。

判断 puck 与 mallet 是否碰撞：通过计算两个圆柱体的柱心间的距离是否小于两圆柱半径之和来判断。

计算 puck 与 mallet 碰撞后的方向：以两柱心间的连线为法向量，计算碰撞前的轴对称方向即可。

3. 鼠标控制

利用 `gluUnProject` 函数求得鼠标所在处代表的“视线”方向，求得视线与桌面的交点，将交点的 x、y 坐标设置为玩家 mallet 的 x、y 坐标，再将坐标限制在玩家所属的桌面范围内即可。

4. AI

简单 AI: AI mallet 的位置较靠近桌面中心, 移动速度较慢, 通过计算与 puck 在 y 轴上的相对位置得出下一步的移动方向, 希望尽量接近 puck 的 y 坐标。

困难 AI: AI mallet 的位置较靠近桌面边缘, 移动速度快, 通过计算与 puck 在 y 轴上的相对位置得出下一步的移动方向, 在 puck 向人类玩家方运动时远离 puck, 在 puck 向己方运动时靠近 puck。

5. 镜头转动

通过三个参量确定相机的位置: alphaXY、alphaZ、dist, 其中 alphaXY 指相机与视点(lookat)连线在 xy 平面上的投影与 x 轴正方向的夹角, 可活动范围设为 $[-60, 60]$; alphaZ 指相机与视点连线与 xy 平面的夹角, 可活动范围设为 $[30, 70]$; dist 指相机与视点间的距离, 设为 14。

通过改变 alphaXY 实现镜头的左右转动, 改变 alphaZ 实现镜头的上下转动, 转动时 dist 不变。

四、 用户手册与文件说明

见 README.md

五、 遇到的问题与解决方案

1. 纹理贴图一片空白: 使用 `glEnable(GL_TEXTURE_2D);` 开启纹理功能
2. “YOU WIN!” 与 “YOU LOSE!” 字样随着镜头的转动而转动: 将投影矩阵设置成二维裁剪矩阵, 再绘制文字, 绘制完毕后恢复透视投影矩阵。
3. 在后面的物体遮挡住在前面的物体: 开启深度测试 `glEnable(GL_DEPTH_TEST);`
4. 绘制地面纹理时影响到其他物体的绘制: 在绘制地面的函数的头尾分别用 `glEnable(GL_TEXTURE_2D);` 与 `glDisable(GL_TEXTURE_2D);` 括起来。

六、 收获与感想

通过本次大作业, 对 opengl 的使用更加熟练, 对面向对象编程有了更深的理解。最后, 感谢老师与助教在本学期的帮助与指导!