研究生课程考试成绩单

（试卷封面）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 院 系 | 计算机科学与工程学院 | 专业 | | 软件工程 | | |
| 学生姓名 | 李娜 | 学号 | | 171674 | | |
| 课程名称 | 计算机图形学 | | | | | |
| 授课时间 | 2017年9月至2018年1月 | | 周学时 | 18 | 学分 | 3 |
| 简  要  评  语 |  | | | | | |
| 考核论题 | 3D场景中的庭院 | | | | | |
| 总评成绩  （含平时成绩） |  | | | | | |
| 备注 |  | | | | | |

任课教师签名：

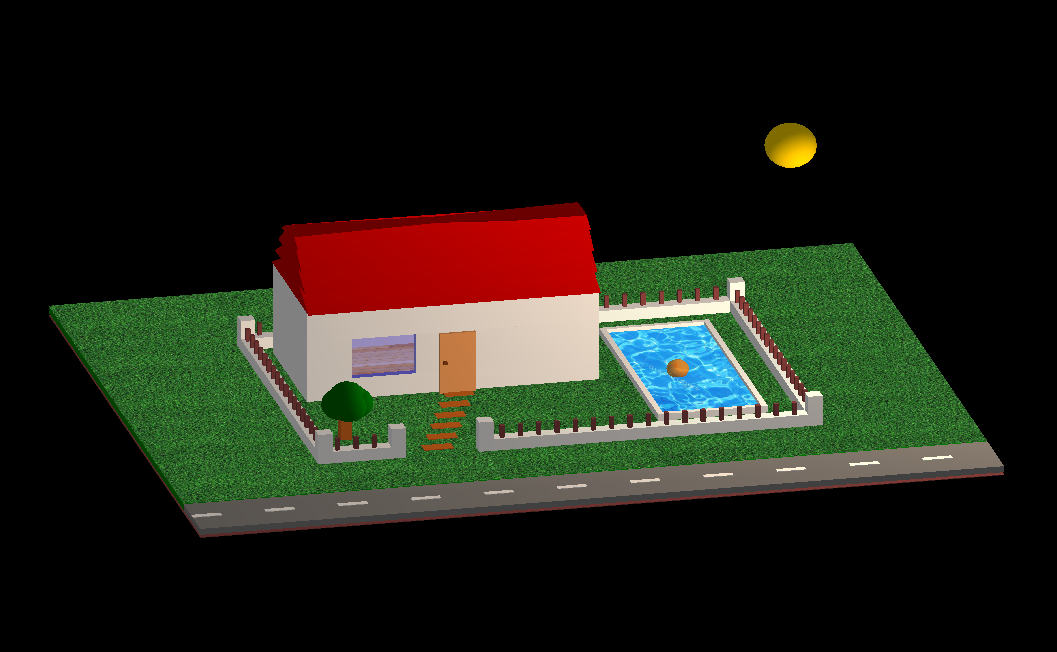
日期：

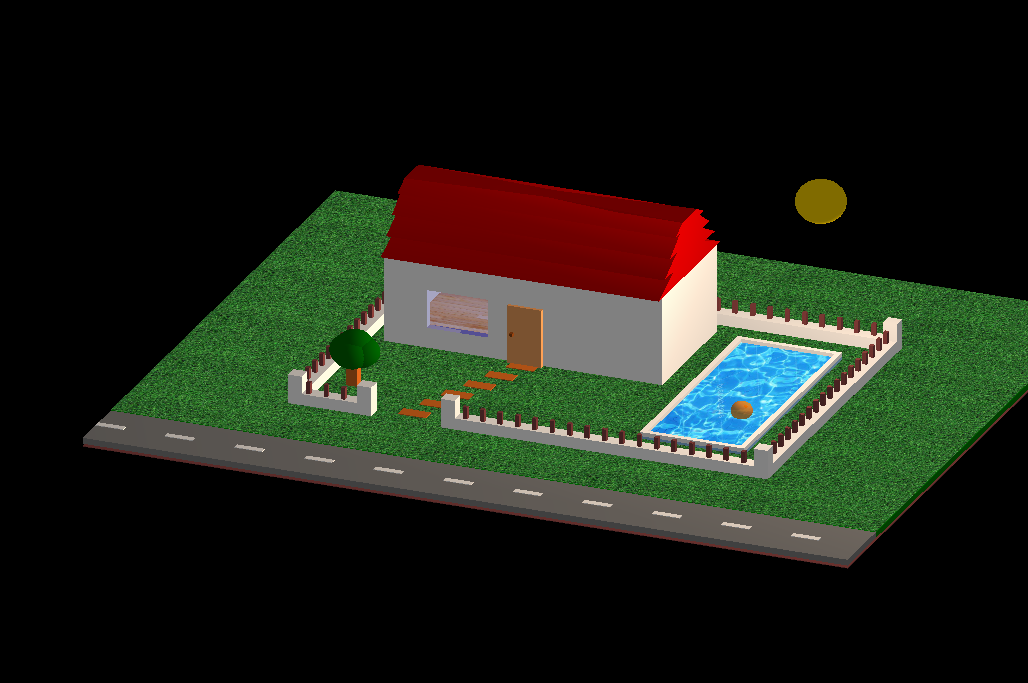
注：1. 以论文或大作业为考核方式的课程必须填此表，综合考试可不填。“简要评语”栏缺填无效。

2. 任课教师填写后与试卷一起送院系研究生秘书处。

3. 学位课总评成绩以百分制计分。

计算机图形学实验报告

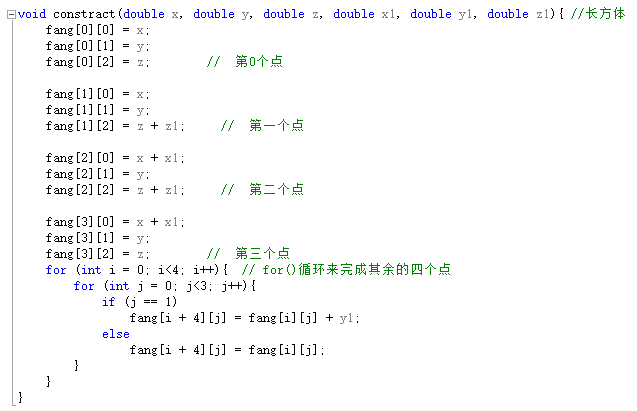
1. **使用手册**
2. 该程序在Visual Studio + OpenGL的平台上模拟了一个三维立体场景中的庭院，场景中包括草地、树木、房屋、泳池、小球、栅栏、小路、马路和太阳等对象。
3. 场景观测视角旋转：键盘↑/↓/←/→分别对应控制视角上移、下移、左移和右移。
4. 场景缩放：键盘F7/F8分别对应控制场景放大和缩小。
5. 小球运动：泳池中小球会随着水波自动沉浮，并且可以通过键盘F1/F2/F3/F4分别控制小球向前、向后、向左、向右游动。

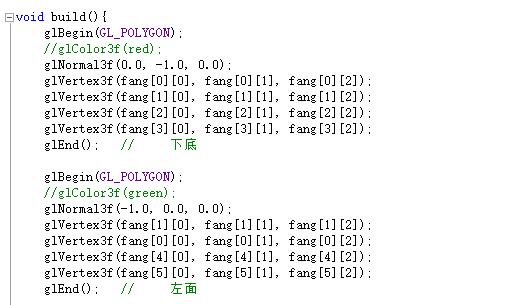


1. **设计思路**
2. 建模。对三维立体场景以及场景中的各对象进行模型建立，包括光照和面绘制。设置了环境光和平行光两种光照类型，面绘制则采用了两种绘制方式，单层绘制负责圆形的绘制，两层层级绘制则用于其他图形的绘制。
3. 纹理绘制。实现场景中的草地、泳池和房内地板的纹理映射。
4. 交互控制。实现键盘对场景的各类交互控制。
5. 动态纹理绘制。实现泳池的水的动态水波和小球的动态沉浮，考虑使用基于网格绘制的方法，通过物理模拟（各种波动方程）来计算网格点位置，达到动态效果。
6. **实现**
7. 建模
8. 首先，简单地调用OpenGL的初始化函数，实现应用程序的初始化工作。之后，建立窗口的位置，定义窗口的宽度和高度，指定窗口的显示模式。在init函数中对光照和视角进行了初始化，注册了绘图函数display负责三维场景中所有对象的绘制。

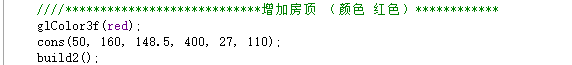


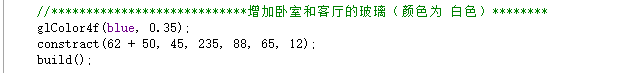
1. 面绘制中两层层级绘制方式，上层设置了constract函数和cons函数分别根据输入的坐标参数对长方体和梯形的顶点数组进行赋值，再由下层build2函数和build函数分别根据顶点坐标进行面绘制。

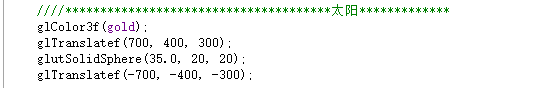




1. 场景中各对象的实现只需调用上层constract函数和cons函数或glutSolidSphere函数即可。



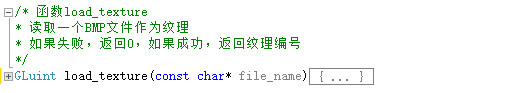


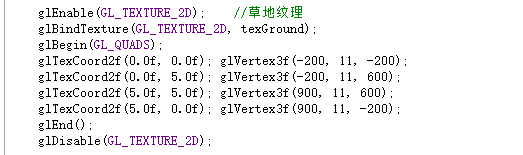


1. 纹理绘制

首先对纹理对象进行了定义，在纹理绑定函数中读取纹理绘制所需的bmp图片。之后进行启动纹理、绑定纹理、关闭纹理三步骤。

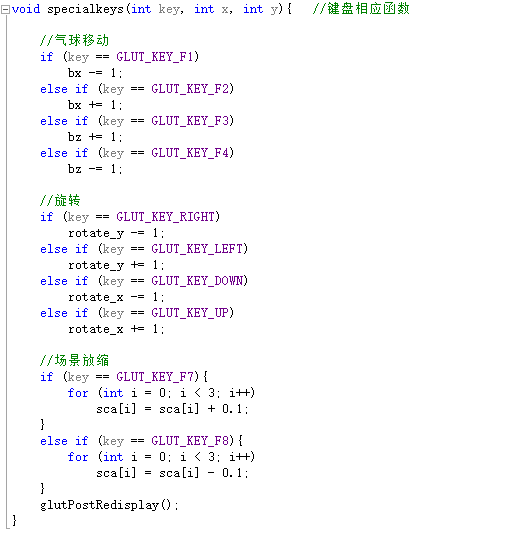






1. 交互控制

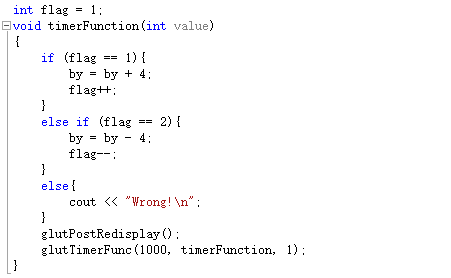
键盘的交互控制通过specialkeys函数实现，函数参数为键盘按下的键key。F1/F2/F3/F4控制泳池中小球的移动，该移动通过改变小球坐标的偏移量bx、bz实现。方向键上下左右控制三维场景的观测视角，该旋转通过改变场景轴旋转变量rotate\_x、rotate\_y实现。F7/F8控制三维场景的放大和缩小，该放缩通过修改放缩数组sca[3]实现。



1. 动态纹理绘制。

这一部分在设计中是计划绘制出泳池中水涌动带动小球沉浮的场景。了解了真实的水面模拟在绘制方面有四种实现方式，分别是：贴图置换、基于网格绘制，通过物理模拟（各种波动方程）来计算网格点位置，达到动态效果、动态的凹凸纹理映射、使用粒子系统模拟水面；渲染则包括反射和折射效果的实现、高光（亮斑）的添加、半透明等方面。在这次实验中曾尝试基于网格绘制实现动态水面模拟，很遗憾这部分代码最终没有调试成功，因而最终只能通过定时器的方式实现小球的自动沉浮。





1. **小结**

一学期的图形学课程学习让我对图形学的概念有了基础的了解，学习到了很多图形绘制技术，也对OpenGL的相关操作做了简单的练习，理解了GL是如何绘制并显示图形的，尤其是对3D图形绘制。以往,使用游戏引擎时,往往使用的是其封装好的OpenGL,无法窥视其内部实现原理。

我觉得图形学作为研讨课，presentation的研讨拓宽我在计算机图形学的知识面，之前一直认为图像大多是和计算机硬件无关的，而这门课则让我建立起了图像和硬件处理息息相关的基本认识。在实验的部分也让我收获非常多，对于自己的一些小想法付诸了实践，达成了一定的效果。实验中从“实验思路的设计”到“运用课堂知识来实现思路”的过程中存在着很多从“遇到问题”到“解决问题”的过程，这个过程中对自学能力和资料的查找能力是有很大的帮助的。譬如在实验过程中动态效果的实现在使用基于网格绘制受阻时，退而求其次使用了定时器方式和贴图方式实现了简易动画效果。

在这里也有一些小建议，在完成这个实验的过程中，其实花了很多时间在查阅书籍和网上资料解决一些问题，这些问题其实是简单入门级、甚至老师上课有提及的。我想边学边实践可能会使我们获得更好的体验和更深的印象，在课堂上与老师同步完成一些基础入门功能的实现后，在课后才能在理解背后原理、熟悉OpenGL语言的基础上实现一些更加深入高级的图像效果，因此建议老师可以考虑在机房开设课程。