**Final Project – Zen Designer**

519021910475 王廷宇

1. 概述

Zen Designer 是一款古风建模游戏，允许玩家设计自己的模型并布置场景。

1. 关于Qt+OpenGL的说明
2. 关于程序编写

该项目基于Qt项目框架构建，OpenGL作用于其中主要子窗口的渲染。具体而言，OpenGL相关代码主要在ObjectViewWidget和SceneViewWidget类中。

Qt库为OpenGL中的VAO、VBO等进行了简单的对象化封装，为了更好的代码结构性，本项目大部分代码运用了这种封装。但对OpenGL代码的编写流程，如VAO、VBO的绑定与解绑，基本不产生影响。对于个别由于Qt封装而简化较多的技术实现，下文中会详细说明。

1. 关于项目运行

根据以往的经验，Qt打包出的项目可能会缺少某些动态链接库。此次打包的项目中已经包含了已知的可能缺少的库，若仍不能正常运行，可联系本人解决。

1. 程序使用说明

进入游戏后，默认为建模模式，可以通过Switch Mode按钮在建模模式和场景模式之间切换。

1. 建模模式（Object Mode）
   1. 控制点编辑

添加点：点击“Points”右侧加号，再点击屏幕空白处，在点击位置添加点。

点的移动：可直接拖拽或通过属性面板修改。

点的修改、删除：选中该点或左侧列表中对应标签，弹出属性面板，即可修改或删除点。

切换到纹理编辑：“Edit Textures”。

* 1. 纹理编辑

设置材质纹理属性：“Material”。

设置光照：“Light”。

切换到控制点编辑：“Edit Points”。

1. 场景模式

退出视角移动：Esc。

恢复视角移动：点击场景窗口即可。

添加物体：点击“Objects”右侧加号，弹出文件选择窗口，选择对应模型文件即可。

物体的修改、删除：选中左侧列表对应标签，弹出属性面板，即可修改或删除物体。

设置环境及特效：“Environment”。

1. 技术实现
2. 摄像机

使用Qt的事件响应函数，处理用户的键盘和鼠标输入，转化成玩家位置的移动和俯仰角、巡航角的变化， 并计算对应的观察矩阵。相关公式如下：

1. 光照

采用Blinn-Phong光照模型。相关公式如下：

1. 材质贴图

采用漫反射贴图+镜面反射贴图+法线贴图（部分材质可能有部分贴图缺失，此时相关参数由玩家设置）。其中漫反射贴图和镜面反射贴图取代材质参数中的对应属性，而法线贴图使顶点法向量偏移，突出材质的凹凸感。具体实现上，通过向片段着色器传入从切线空间变换到世界空间的变换矩阵（TBN矩阵），从而将法线贴图中的数据转化为正确的顶点法向量。

1. 模板测试

利用模板缓冲实现了场景中选中物体有高亮边框显示。首先渲染该物体并写入模板缓冲，之后禁用模板测试并渲染场景中的其他物体，最后启用模板测试渲染一个单一颜色的并且稍微放大的该物体，丢弃掉已写入模板缓冲的部分像素，从而得到高亮的边框。

1. 天空盒

本项目中的天空盒是利用Unity制作简单场景，然后渲染到图片从而制作而成。在OpenGL中使用立方体贴图（Cube Map）进行渲染，并且丢弃观察矩阵中的平移，从而达到天空盒的效果。

1. 抗锯齿

本项目中采用Qt封装中带有的一个额外帧缓存进行MSAA（多重采样抗锯齿），该帧缓存用于简化各种常用的的后处理。普通OpenGL项目中的MSAA应创建一个带有多重采样纹理附件的帧缓存，再通过图像位块传送（Blit）渲染到屏幕上。

1. 阴影映射

*\*该功能仍不完善*

先从光源方向渲染一张深度贴图，在渲染物体时使用该深度贴图检查是否在阴影中，以得到阴影效果。并且采用了阴影偏移技术和简单的PCF，但实现效果仍不理想，故设为默认关闭阴影，可在属性面板中打开。

1. 粒子系统

制作了简单的2D粒子系统，所有粒子在玩家右上方生成，按照一定的速度与加速度移动，并且带有一定的旋转。