

Project report for the CG 100433 course

第三组

组长：梁伦豪；

组员：袁蜀煜、黄一脚、金颖鑫、陆毅帆、张欢、郑博超。

1. Project Title

3d showcase

2. Team member

第三组；

组长：梁伦豪；

组员：袁蜀煜、黄一脚、金颖鑫、陆毅帆、张欢、郑博超。

3. Project Title

桃花源

4. Abstract

确定主题为 3d showcase 后，经过讨论，我们决定实现桃花源第一段描述的场景，包括房屋，树木，溪流等，希望能够通过人机交互从各个角度查看场景，并实现水流、落花等效果。整个项目主要基于 OpenGL 开发，也涉及 3D 模型的导入。

5. Motivation

这就是人脑中一闪而过的微光，我们在 showcase 的选择中讨论了宇宙，沙滩，深海等等发现不仅有重复的冲突可能也有实现上的技术困难。最后我们选择了初中课文中的桃花源记，不仅画面感强烈而且很有意义。

6. Goal of the project

目标是再现桃花源记原文中第一段故事情节。从第一人称的视角进行再现，将小溪，小船，两岸的树和落花、最后的发光的山洞和通过山洞后的世外桃源。首先完成树、小船等模型的建立，再完成世外桃源的设计，最后实现。

7. Scope of the project

- a. 未能加入花瓣飘落的效果；
- b. 导入的模型仅有树，船等几种，未能添加更多更精细的模型；
- c. 场景限于一个包含木屋，溪流，树木的天空盒内，没有实现整个桃花源的景观。
- d. 未能实现光影效果。

8. Involved CG techniques

C 语言编译器：用于编译编写代码；

OpenGL 库：用于模型和场景在计算机中生成；

AUTODESK MAYA：建立 3D 模型并导出为图元参数。

9. Project contents

- a. Obj 格式模型的导入和生成;
- b. Mtl 格式的模型纹理贴图文件的导入和生成;
- c. 平面贴图的实现;
- d. 天空盒的建立;
- e. 水面效果的建立和生成;
- f. 材质光影;
- g. 背景音乐

10. Implementation

- a. Obj 格式模型的导入和生成:

Obj 格式文件定义的模型相关参数如下

v : 顶点的坐标

vn : 顶点法向量

f : 面, 指的是面的三个顶点

将以上参数导入程序中的三个数组再读取绘制这三个数组中定义的图形可以实现 obj 格式模型的导入和绘制。

- b. Mtl 格式的模型纹理贴图文件的导入和生成:

材质库中包含材质的漫射(diffuse), 环境(ambient), 光泽(specular)的 RGB(红绿蓝)的定义值, 以及反射(specularity), 折射(refraction), 透明度(transparency) 等其它特征。 "usemtl"指定了材质之后, 以后的面都是使用这一材质, 直到遇到下一个"usemtl"来指定新的材质。

mtl 格式文件定义的贴图相关参数如下

Ka r g b

defines the ambient color of the material to be (r,g,b). The default is (0.2,0.2,0.2);

Kd r g b

defines the diffuse color of the material to be (r,g,b). The default is (0.8,0.8,0.8);

Ks r g b

defines the specular color of the material to be (r,g,b). This color shows up in highlights. The default is (1.0,1.0,1.0);

d alpha

defines the transparency of the material to be alpha. The default is 1.0 (not transparent at all) Some formats use **Tr** instead of **d**;

Tr alpha

defines the transparency of the material to be alpha. The default is 1.0 (not transparent at all). Some formats use **d** instead of **Tr**;

Ns s

defines the shininess of the material to be s . The default is 0.0;

illum n

denotes the illumination model used by the material. **illum = 1** indicates a flat material with no specular highlights, so the value of **Ks** is not used. **illum = 2** denotes the presence of specular highlights, and so a specification for **Ks** is required.

map_Ka filename

names a file containing a texture map, which should just be an ASCII dump of RGB values;

将以上参数导入程序中的数组再读取贴图可以实现 mtl 格式模型材质的导入和绘制。

c. 平面贴图的实现;

使用 opengl 中 Bitmap 和 Texture 相关函数将 bmp、jpg、png 格式的材质图形文件信息读取出来, 并且设定好贴图 id 编号, 通过 glBindTexture() 函数将贴图编号绑定到当前绘制的图形上, 实现可以对于绘制平面进行贴图的功能。

d. 天空盒的建立;

如图所示, 天空盒的思想就是绘制一个大的立方体, 然后将观察者放在立方体的中心, 当相机移动时, 这个立方体也跟着相机一起移动, 这样相机就永远不会运动到场景的边缘。这与我们真实世界中的情况一样的, 我们可以看见远处的天空接触到了地平线, 但是不论我们怎么朝着那个方向移动, 都不可能到达那个地方。



e. 水面效果的建立和生成:

1.水波的运动位移 (z) 满足偏微分方程:

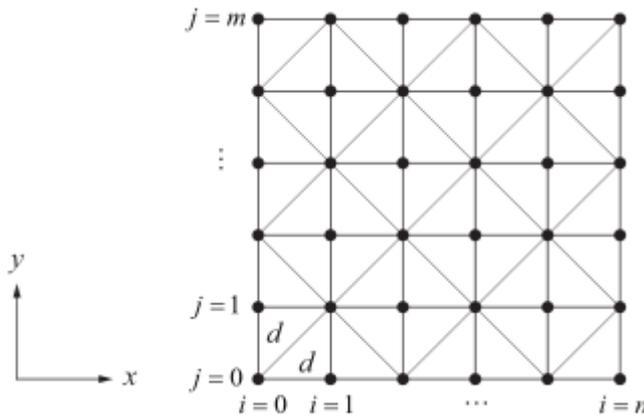
$$\frac{\partial^2 z}{\partial t^2} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - \mu \frac{\partial z}{\partial t},$$

其中,c 是波速, u 是描述阻力大小的系数,x,y,z 是空间坐标, t 是时间。

2.这个方程求解很麻烦, 我们用近似的方法得到运动位移的公式:

$$z(i, j, k+1) = \frac{4-8c^2t^2/d^2}{\mu t+2} z(i, j, k) + \frac{\mu t-2}{\mu t+2} z(i, j, k-1) + \frac{2c^2t^2/d^2}{\mu t+2} [z(i+1, j, k) + z(i-1, j, k) + z(i, j+1, k) + z(i, j-1, k)],$$

在这里, 我们用三角形网格来表达水面, 如下图, i,j 代表点位置的索引, k 是时间。



3.为了保证迭代方程收敛, t和 c 要满足:

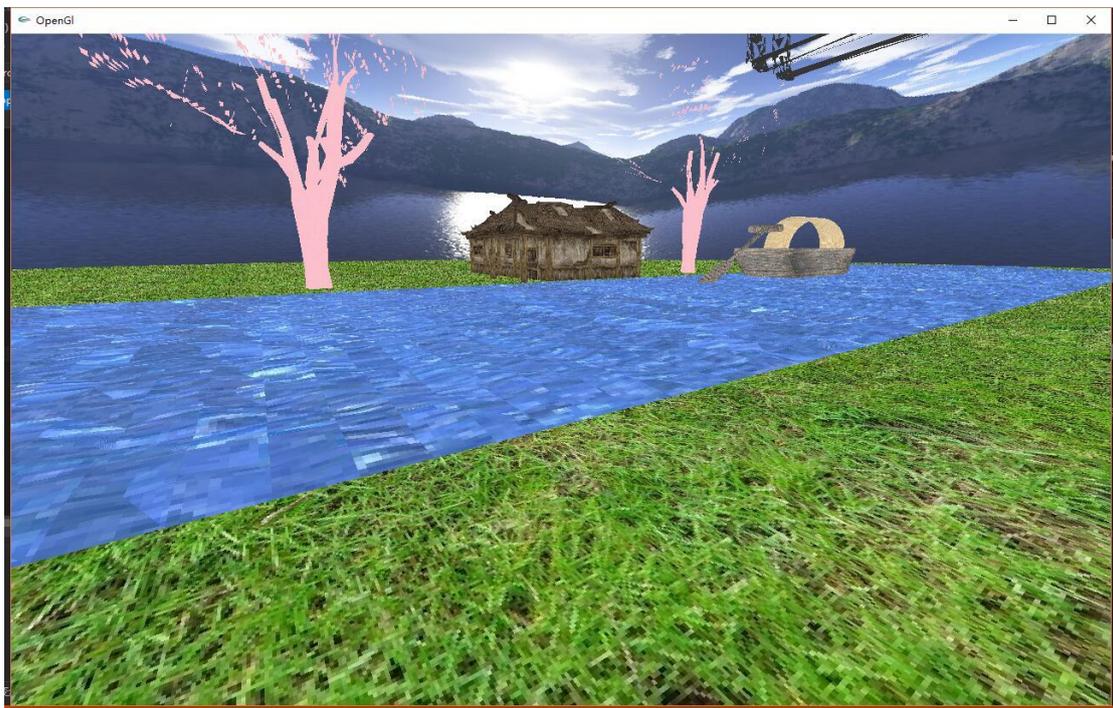
$$0 < c < \frac{d}{2t} \sqrt{\mu t + 2}.$$

$$0 < t < \frac{\mu + \sqrt{\mu^2 + 32c^2/d^2}}{8c^2/d^2}.$$

绘制时我们需要输入的信息包括：

- 1.每个顶点的坐标（包括高度信息）
2. 每个顶点的法向量（在这里省略了）
- 3.顶点的下标索引（描述一个三角形由哪几个顶点组成，只存索引，不存具体的顶点坐标）
- 4.纹理坐标（描述纹理图片是如何映射到三角形上的）

通过绘制满足上述数学关系的三角形变换，加上水面的材质贴图可以模拟水面波动效果



f. 材质光影；

先使用一下函数定义光源的位置、漫射、环境、光泽等信息

```
glLightfv(GL_LIGHT0, GL_POSITION, light_position);  
glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);  
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);  
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);  
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, light_direction);  
glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 10.0f);  
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 120.0f);
```

再使用一下函数定义材质相对于光源的漫射、环境、光泽等相关信息可以实现贴图材质的阴影效果

```
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient_color);
```

```
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);  
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);  
glMaterialfv(GL_FRONT, GL_SHININESS, high_shininess);  
glMaterialfv(GL_FRONT, GL_EMISSION, mat_emission);
```

g. 背景音乐

使用 `playsoundapi.h` 头文件库中的 `playsound()` 函数可实现背景音乐的播放

运行效果：



11. Results

- 场景在一个天空盒内；
- 场景包括木屋，小船，树木，溪流等；
- 通过鼠标拖动可以自由旋转视角；
- 实现了模型纹理贴图 and 地面贴图；
- 实现了溪流蜿蜒，水面波动的效果；
- 实现了小船随水漂流的效果；

12. Roles in group

代码编写：梁伦豪 郑博超；

模型：陆懿帆 张欢；

资料收集：黄一卿；

测试：金颖鑫 袁蜀煜。

13. References

ave Shreiner, Graham Sellers, John Kessenich, Bill Licea-Kane, 2013, OpenGL Programming Guide:the Official Guide to Learning OpenGL, Version 4.3, Eighth Edition, Addison-Wesley Professional. 《华章程序员书库:OpenGL 编程指南（原书第 8 版）》，机械工业出版社；第 1 版（2014 年 10 月 1 日）。