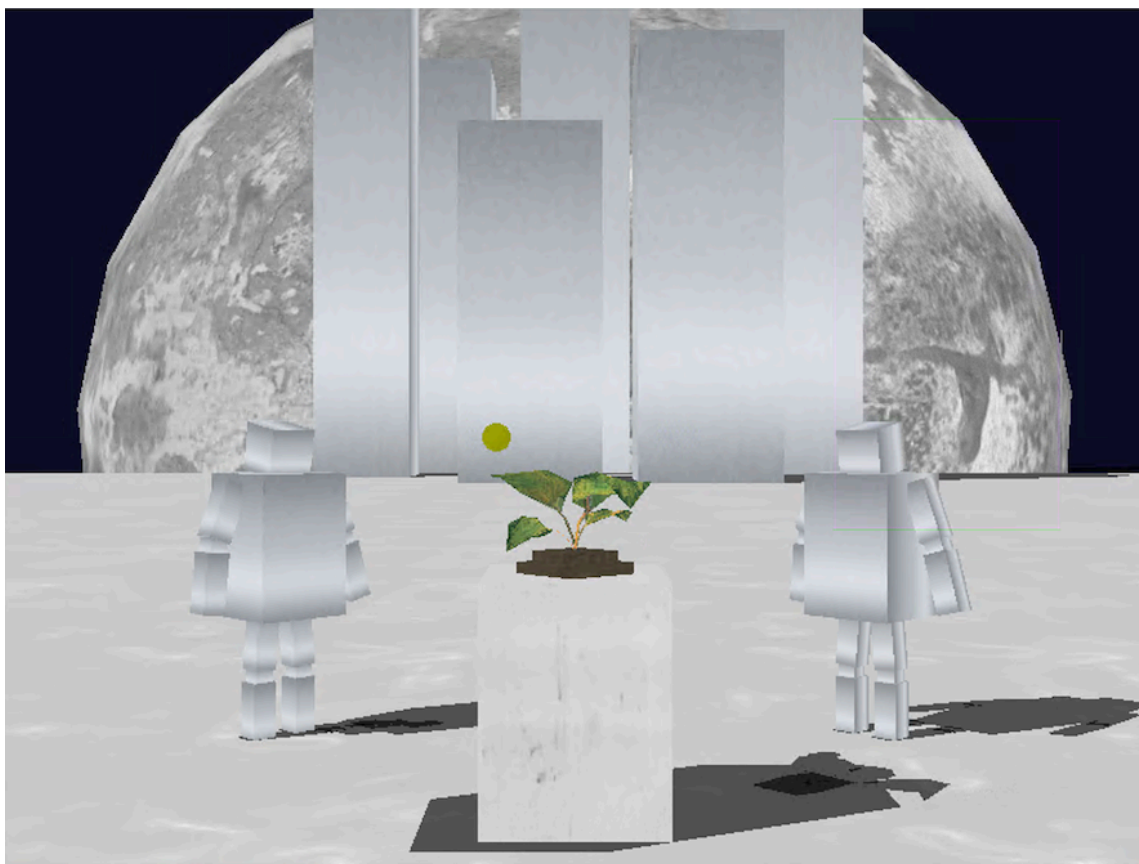


1122 電腦圖學期末專題報告

本文連結 (<https://hackmd.io/AwfGrpO5TayYzYuTotDX7Q>).

Demo (1., 6.)



- 影片 (https://drive.google.com/file/d/1idmQiLKadt_YjIYDv7_qL9EVUqSV22Yk/view?usp=share_link).

🔗 簡介

這個世界大概就是在很遠很遠的未來
不只是人類而是所有的生物都已消失
僅存那顆小草
以前不以為意
現在卻是生命存在過的最後的證據

Texture (2.)

```
// index
#define GROUND_TEXTURE 0    // 白土
#define TORUS_TEXTURE 1    // 白泥
#define SPHERE_TEXTURE 2   // 黑土
#define ROBOT_TEXTURE 3    // 金屬
#define BUILDING_TEXTURE 3 // 建築
#define PLANET_TEXTURE 4   // 行星

#define NUM_TEXTURES 5
GLuint textureObjects[NUM_TEXTURES];

const char *szTextureFiles[] = {"D:\\code\\graph\\final\\snow_02_rough_1k.tga", \
                                "D:\\code\\graph\\final\\polystyrene_disp_1k.tga", \
                                "D:\\code\\graph\\final\\brown_mud_02_diff_1k.tga", \
                                "D:\\code\\graph\\final\\metallic-textured-background.tga", \
                                "D:\\code\\graph\\final\\cracked_concrete_wall_diff_1k.tga"};
```

共五種，用於土壤、底座、機器人、建築以及後方大行星

Load object (3.)

沒錯，是草



雖然跟未來世界好像沒有什麼關係

不過這在背景設定下是很重要的！！（？

使用之前寫的ObjLoader再進行修改
變成可以貼圖的形式

```
// load obj file
ObjLoader* grassObj = new ObjLoader("D:\\code\\graph\\Lab13\\final_sampleCode\\testOBJ.obj", \
                                     "C:\\Users\\selab\\Downloads\\ImageToStl.com_nettle_plant_1k\\nettle_plant_dry_diff_1k_2.png");
```

```
cv::Mat grassImg;
std::string texturePath;
ObjLoader(string filename, string texturePath) {
    ifstream file(filename);
```

並且加入自轉的設定

```
// 自轉的草
glRotatef(yRot, 0.0f, 1.0f, 0.0f);
glPushMatrix();
    glDisable(GL_CULL_FACE);
    grassObj->init();
    glScalef(5.0f, 5.0f, 5.0f);
    grassObj->rotate();
    grassObj->translate();
    grassObj->draw(nShadow);
    glEnable(GL_CULL_FACE);
glPopMatrix();
```

動畫 (4.)

參考demo章節的影片

附上程式碼截圖

```
glTranslatef(0.0f, 0.1f, -2.5f);
glPushMatrix();
    glPushMatrix();
        // 旋轉小太陽
        glRotatef(-yRot * 2.0f, 0.0f, 1.0f, 0.0f);
        glTranslatef(0.1f, 0.0f, -0.1f);
        gltDrawSphere(0.02f, 21, 11, nShadow);
    glPopMatrix();

    if (nShadow == 0)
    {
        // Torus alone will be specular
        glMaterialfv(GL_FRONT, GL_SPECULAR, fNoLight);
    }

    // 自轉的草
    glRotatef(yRot, 0.0f, 1.0f, 0.0f);
    glPushMatrix();
        glDisable(GL_CULL_FACE);
        grassObj->init();
        glScalef(5.0f, 5.0f, 5.0f);
        grassObj->rotate();
        grassObj->translate();
        grassObj->draw(nShadow);
        glEnable(GL_CULL_FACE);
    glPopMatrix();
    // 土壤
    glPushMatrix();
        glBindTexture(GL_TEXTURE_2D, textureObjects[SPHERE_TEXTURE]);
        glTranslated(0.0f, -0.17f, 0.0f);
        drawTorso(0.04f, 0.02f, 0.04f, nShadow);
        glTranslated(0.0f, -0.005f, 0.0f);
        drawTorso(0.055f, 0.01f, 0.055f, nShadow);
    glPopMatrix();
glPopMatrix();
```

```
// 行星
glPushMatrix();
    // 旋轉
    glBindTexture(GL_TEXTURE_2D, textureObjects[PLANET_TEXTURE]);
    glTranslatef(0.0f, 0.0f, -60.0f);
    glRotatef(-yRot * 0.2f, 0.0f, 0.0f, 1.0f);
    drawPlanet(22.0f, 21, 11, nShadow);
glPopMatrix();
```

Shadow (5.)

```
void drawTorso(GLfloat x, GLfloat y, GLfloat z, int shadowMode)
{
    if(shadowMode ==1){
        glColor4d(0.0, 0.0, 0.0, 0.6);
    }
    else{
        glColor3d(1.0, 1.0, 1.0);
    }
    glBegin(GL_QUADS);
    // Front Face
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.0f, 0.0f);
    glVertex3f(-x, -y, z);
    glTexCoord2f(1.0f, 0.0f);
    glVertex3f(x, -y, z);
    glTexCoord2f(1.0f, 1.0f);
    glVertex3f(x, y, z);
    glTexCoord2f(0.0f, 1.0f);
    glVertex3f(-x, y, z);
}
```

ObjLoader部分

```
glNormal3f(normal.x, normal.y, normal.z);
glBindTexture(GL_TEXTURE_2D, textures[0]);
glTexCoord2f( (vt[firstTextureIndex])[0], (vt[firstTextureIndex])[1]);
glVertex3f(a.x, a.y, a.z);
glTexCoord2f( (vt[secondTextureIndex])[0], (vt[secondTextureIndex])[1]);
glVertex3f(b.x, b.y, b.z);
glTexCoord2f( (vt[thirdTextureIndex])[0], (vt[thirdTextureIndex])[1]);
glVertex3f(c.x, c.y, c.z);
```

Code (6.)

- [Github](https://github.com/Octopus1031/112-computer-graphics-final-project) (https://github.com/Octopus1031/112-computer-graphics-final-project).
- [Google Drive](https://drive.google.com/file/d/1lwF8rNK8Z0WYu_UgBTQ2lrIFJUPBE4_r/view?usp=drive_link) (https://drive.google.com/file/d/1lwF8rNK8Z0WYu_UgBTQ2lrIFJUPBE4_r/view?usp=drive_link).
- 隨作業壓縮檔附上

How to use

setup

- 將 **sphereworld.cpp** 以及 **objLoader.h** 匯入專案
- 將專案環境設定完成
- 更改texture路徑(**sphereworld.cpp** line 618 & 3240)

key control

N/A

Difficulty

- **貼圖以及obj檔解析**
在尋找網路上的obj資源時才了解obj的格式不僅限於上課所給的只有v和f的形式，也會搭配所使用的材質而有vt, vn等參數，在了解其意義時花了點時間閱讀相關的文章，經過嘗試修改過去程式碼後成功貼上了。
- **尋找合適的素材**
到處找就對了
- **發想主題**
想了很久，有想做的也不見得有可以免費使用的素材可以利用，最後在時間與資源與技術的妥協下，至少還是有做出一點想表達的東西...？