

PA2 Report

计 81 包涵 2018011289

2020 年 4 月 6 日

1 OpenGL 与光线投射的不同

PA1 中的光线投射算法由视点发出光线，与物体求交，再根据光源设置计算交点颜色。而 PA2 使用 OpenGL，操作大都变为对于 OpenGL 状态机参数的设置，不再需要手工求交或使用 phong 模型计算。具体如下：

- main.cpp::drawScene()

```
void drawScene() {  
    Vector3f backGround = sceneParser->getBackgroundColor();  
    // 调用该函数设定背景颜色，而不是在发现光线和物体没有交点后  
    // 再设置对应像素点的颜色为背景颜色  
    glClearColor(backGround.x(), backGround.y(), backGround.z(), 1.0);  
    // Clear the rendering window  
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
    // 该函数中，使用 gluLookAt，直接设置视点  
    sceneParser->getCamera()->setupGLMatrix();  
    for(int i = 0; i < sceneParser->getNumLights(); i++)  
        // 调用 glEnable 和 glLightfv 设置光源参数  
        sceneParser->getLight(i)->turnOn(i);  
    // 依次调用物体的绘制函数进行绘制  
    sceneParser->getGroup()->drawGL();  
    // Dump the image to the screen.  
    glutSwapBuffers();  
}
```

```

        // Save if not in interactive mode.
        if (!savePicturePath.empty()) {
            screenCapture();
            exit(0);
        }
    }
}

```

- Group::drawGL()

```

void drawGL() override {
    // 依次调用每个 object 的 drawGL 函数完成绘制
    // 而不是依次求交
    for (auto obj: objects) obj->drawGL();
}

```

- Material::Use()

```

void Use() {
    // 调用 OpenGL 的相关函数，直接设置所绘制物体的质感参数
    glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, Vector4f(diffuseColor, 1.0f));
    glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, Vector4f(specularColor, 1.0f));
    glMaterialfv(GL_FRONT_AND_BACK, GL_SHININESS, Vector2f(shininess * 4.0, 1.0f));
}

```

- Transform::drawGL()

```

void drawGL() override {
    // 设置物体质感
    Object3D::drawGL();
    glMatrixMode(GL_MODELVIEW); glPushMatrix();
    // 进行变换
    glMultMatrixf(transform.inverse());
    // 绘制物体
    o->drawGL();
    glPopMatrix();
}

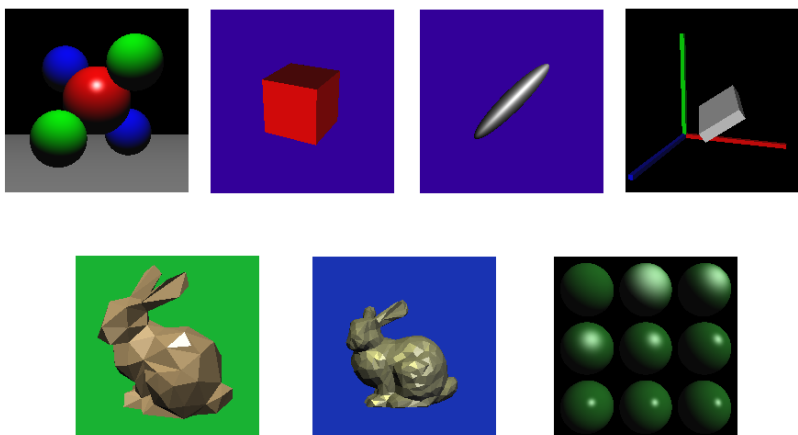
```

- Triangle::drawGL()

```
void drawGL() override {
    Object3D::drawGL();
    // OpenGL 绘制三角形的方法
    glBegin(GL_TRIANGLES);
    glNormal3fv(normal);
    glVertex3fv(vertices[0]); glVertex3fv(vertices[1]); glVertex3fv(vertices[2]);
    glEnd();
}
```

2 编译运行与结果

运行 run_all.sh 即可编译运行，结果如下



3 其他

1. OpenGL 环境配置没有遇到任何问题。
2. 没有与同学讨论，借鉴了自己 PA1 的代码。