



中山大學
SUN YAT-SEN UNIVERSITY

《计算机图形学作业》 实验报告

(作业四)

学 院 名 称 : 数据科学与计算机学院

学 生 姓 名 : 姚雪辉

学 号 : 15355119

专业 (班级) : 16 软件工程四 (7) 班

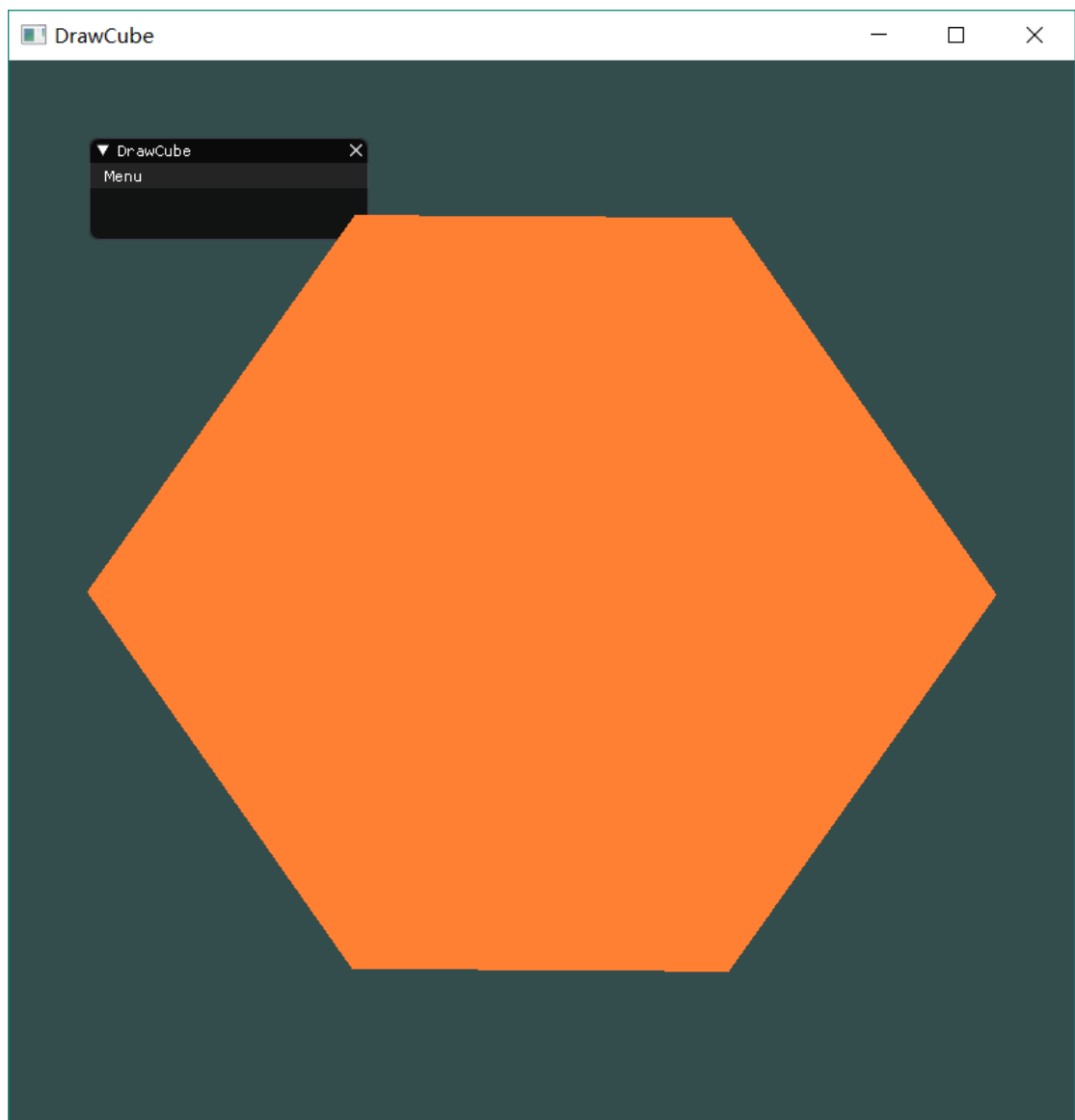
时 间 : 2019 年 4 月 7 日

Basic

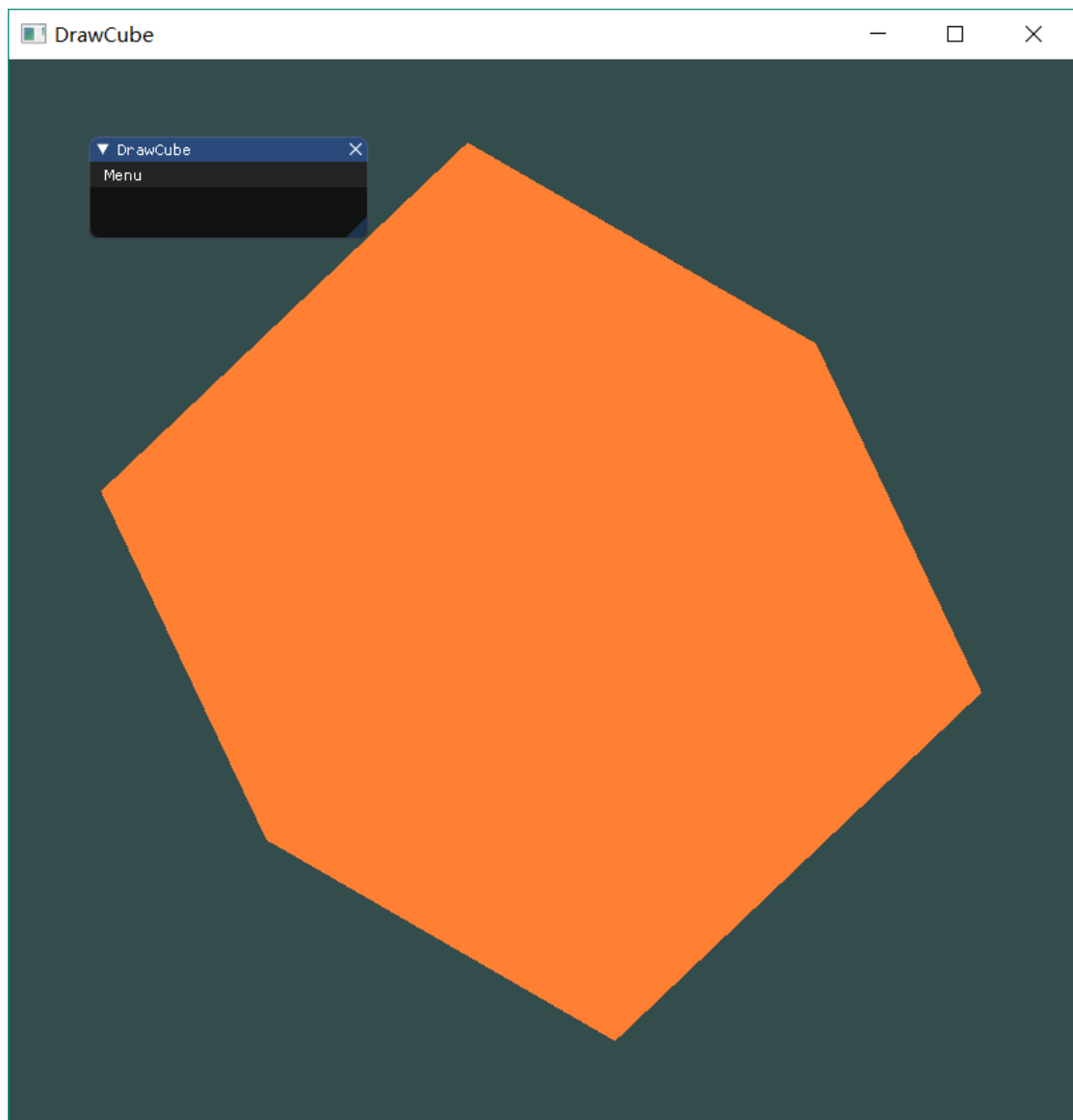
1. 画一个立方体(cube): 边长为 4, 中心位置为(0, 0, 0)。分别启动和关闭深度测试

`glEnable(GL_DEPTH_TEST)`、`glDisable(GL_DEPTH_TEST)`，查看区别，并分析原因。

未开启深度检测



开启深度检测



似乎看不出来开不开深度检测的区别，在 `glEnable(GL_DEPTH_TEST)` 的同时，需要 `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)`，可能是因为我所有的面都用了一个颜色，导致结果不明显，实际上不开启深度检测，渲染出来的会有覆盖的效果，开启了深度检测之后，会有遮盖效果，比如正方体只会显示三个面，而遮住后面的三个面。第二版增加了摄像头：GLSL 改写成如下代码

```
const char *vertexShaderSource = "#version 330 core\n"  
    "layout (location = 0) in vec3 aPos;\n"  
    "uniform mat4 transform;\n"  
    "uniform mat4 view;\n"  
    "uniform mat4 projection;\n"  
    "void main()\n"
```

```

    "{\n"
    "    gl_Position = projection * view * transform * vec4(aPos, 1.0);\n"
    "}\0";

```

VIEW 部分

```

glm::mat4 view = glm::mat4(1.0f);
view = glm::lookAt(glm::vec3(6.0f, 4.0f, -10.0f), glm::vec3(0.0f, 0.0f, 0.0f), glm::vec3(0.0f, 1.0f, 0.0f));
unsigned int viewLoc = glGetUniformLocation(shaderProgram, "view");
glUniformMatrix4fv(viewLoc, 1, GL_FALSE, glm::value_ptr(view));

```

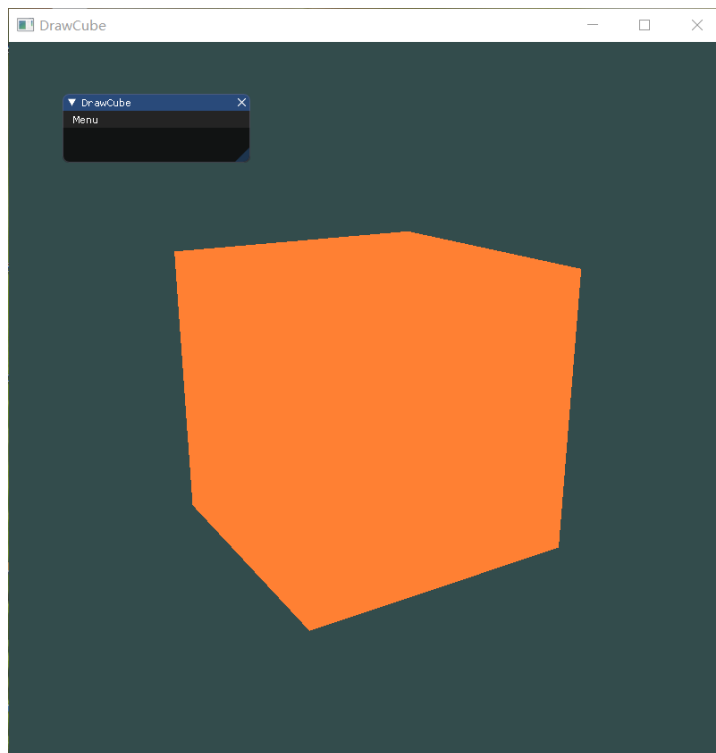
PROJECTION 部分

```

glm::mat4 projection = glm::mat4(1.0f);
projection = glm::perspective(glm::radians(45.0f), (float)SCR_WIDTH / (float)SCR_HEIGHT, 0.1f, 100.0f);
unsigned int projectionLoc = glGetUniformLocation(shaderProgram, "projection");
glUniformMatrix4fv(projectionLoc, 1, GL_FALSE, glm::value_ptr(projection));

```

正常情况下显示如下：



2. 平移(Translation): 使画好的 cube 沿着水平或垂直方向来回移动。

```

transX = sin((float)glfwGetTime())/2;
transform = glm::translate(transform, glm::vec3(transX, 0.0f, 0.0f));

```

tranX 是根据时间变化的三角函数，所以会在屏幕的左右来回移动。

3. 旋转(Rotation): 使画好的 cube 沿着 XoZ 平面的 x=z 轴持续旋转。

```
transform = glm::rotate(transform, (float)glfwGetTime(), glm::vec3(1.0f, 0.0, 1.0f));
```

旋转的角度和时间挂钩，旋转的轴是 x=z

4. 放缩(Scaling): 使画好的 cube 持续放大缩小。

```
scaleFactor = sin((float)glfwGetTime());  
transform = glm::scale(transform, glm::vec3(scaleFactor, scaleFactor, scaleFactor));
```

旋转因子根据时间变化的三角函数，会缩小到无再恢复成原状

5. 在 GUI 里添加菜单栏，可以选择各种变换。

```
ImGui::Begin("DrawCube",&isOpen, ImGuiWindowFlags_MenuBar);  
    if (ImGui::BeginMenuBar()) {  
        if (ImGui::BeginMenu("Menu")) {  
            if (ImGui::MenuItem("Translate")) {  
                isTranslate = true;  
                isRotate = false;  
                isScale = false;  
            }  
            if (ImGui::MenuItem("Rotate")) {  
                isTranslate = false;  
                isRotate = true;  
                isScale = false;  
            }  
            if (ImGui::MenuItem("Scale")) {  
                isTranslate = false;  
                isRotate = false;  
                isScale = true;  
            }  
            if (ImGui::MenuItem("Bonus")) {  
                isTranslate = true;  
                isRotate = true;  
                isScale = true;  
            }  
        }  
        ImGui::EndMenu();  
    }
```

```
    }  
    ImGui::EndMenuBar();  
}  
  
ImGui::End();
```

通过三个参数控制，再通过菜单选择

6. 结合 Shader 谈谈对渲染管线的理解

Shader 是图形可编程方案的程序片段。主要分为顶点着色器和片段着色器。渲染管线是由一个三维场景出发，生成一张二维图像，在这个过程中，需要很多数据，而 shader 就是处理这些数据，使得数据传入下一个阶段直到显示在屏幕上。

Bonus

1. 将以上三种变换相结合，打开你们的脑洞，实现有创意的动画。比如：地球绕太阳转等

```
if (ImGui::MenuItem("Bonus")) {  
    isTranslate = true;  
    isRotate = true;  
    isScale = true;  
}
```

介于之前的深度检测不太明显，所以要做地球绕太阳的动画也不太明显，所以直接组合了三个动画。