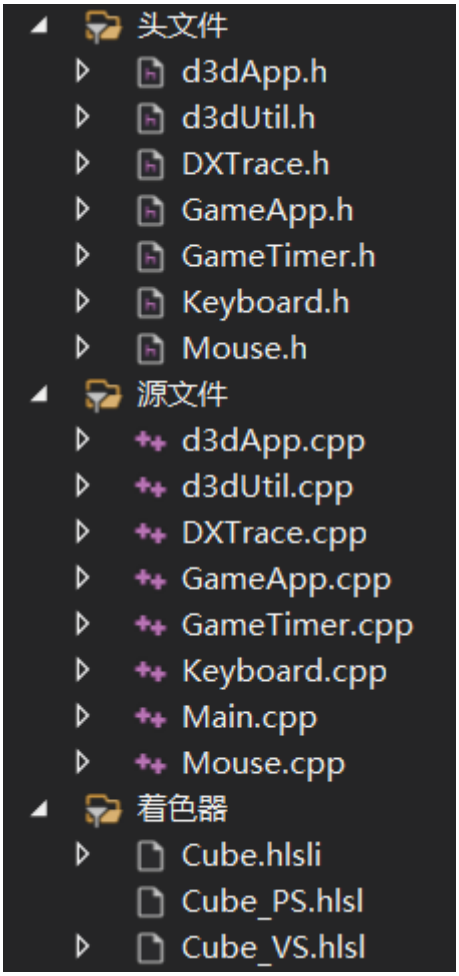
准备工作

新建项目,并配置所需的头文件和源文件



顶点缓冲区

1,输入棱锥顶点的坐标数据

2,输入底面五角星的五个角点的坐标数据

3,输入五角星的顶点之间的五个点的坐标数据

```
VertexPosColor vertices[] =
{
    { XMFLOAT3(0.0f, 1.0f, 0.0f), XMFLOAT4(1.0f, 0.0f, 0.0f, 0.0f) },    //棱锥顶点

    { XMFLOAT3(0.0f, 0.0f, 1.8f), XMFLOAT4(0.0f, 1.0f, 0.0f, 0.0f) },    //五角星的5个角点(顺时针方向,从上方开始)
    { XMFLOAT3(1.6f, 0.0f, 0.4f), XMFLOAT4(0.0f, 1.0f, 0.0f, 0.0f) },
    { XMFLOAT3(1.0f, 0.0f, -2.0f), XMFLOAT4(0.0f, 1.0f, 0.0f, 0.0f) },
    { XMFLOAT3(-1.0f, 0.0f, -2.0f), XMFLOAT4(0.0f, 1.0f, 0.0f, 0.0f) },
    { XMFLOAT3(-1.6f, 0.0f, 0.4f), XMFLOAT4(0.0f, 1.0f, 0.0f, 0.0f) },

    { XMFLOAT3(0.33f, 0.0f, 0.4f), XMFLOAT4(0.0f, 0.0f, 1.0f, 0.0f) },       //角点之间的顶点(顺时针方向,从右上开始)
    { XMFLOAT3(0.59f, 0.0f, -0.53f), XMFLOAT4(0.0f, 0.0f, 1.0f, 0.0f) },
    { XMFLOAT3(0.0f, 0.0f, -1.08f), XMFLOAT4(0.0f, 0.0f, 1.0f, 0.0f) },
    { XMFLOAT3(-0.59f, 0.0f, -0.53f), XMFLOAT4(0.0f, 0.0f, 1.0f, 0.0f) },
    { XMFLOAT3(-0.33f, 0.0f, 0.4f), XMFLOAT4(0.0f, 0.0f, 1.0f, 0.0f) },

};
```

索引缓冲区

根据顺时针的顺序输入顶点对应的索引

先构造出底面五角星,再构造出侧面.

```
DWORD indices[] = {
//底面
    5, 3, 6,
    2, 10, 4,
    1, 10, 6,
//侧面
    4, 0, 8,        //逆时针方向绘制各个侧面
    8, 0, 3,
    3, 0, 7,
    7, 0, 2,
    2, 0, 6,
    6, 0, 1,
    1, 0, 10,
    10, 0, 5,
    5, 0, 9,
    9, 0, 4
};
```

旋转(逐帧更新数据)

在UpdataScene函数中:

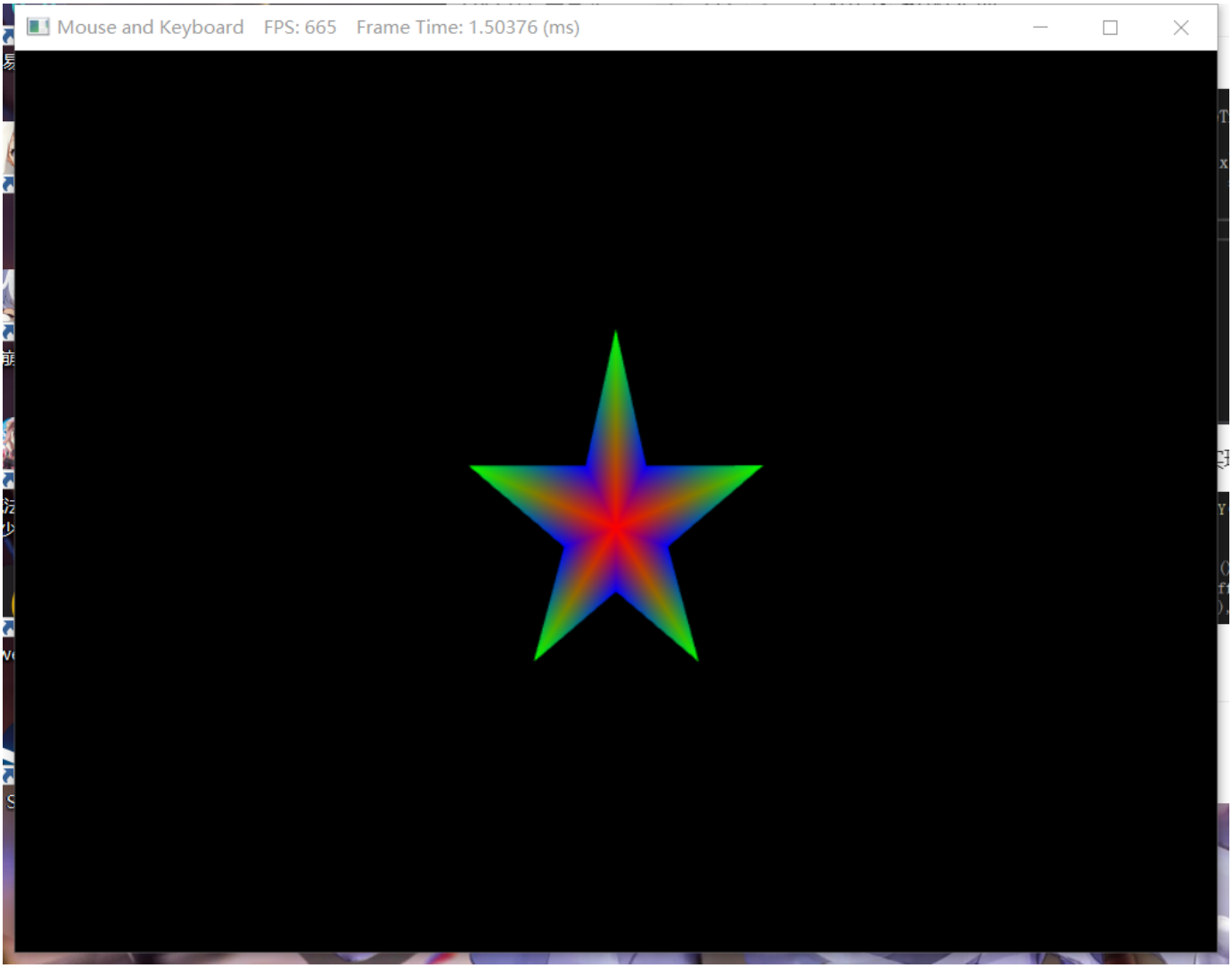
根据鼠标和键盘的状态获取旋转量

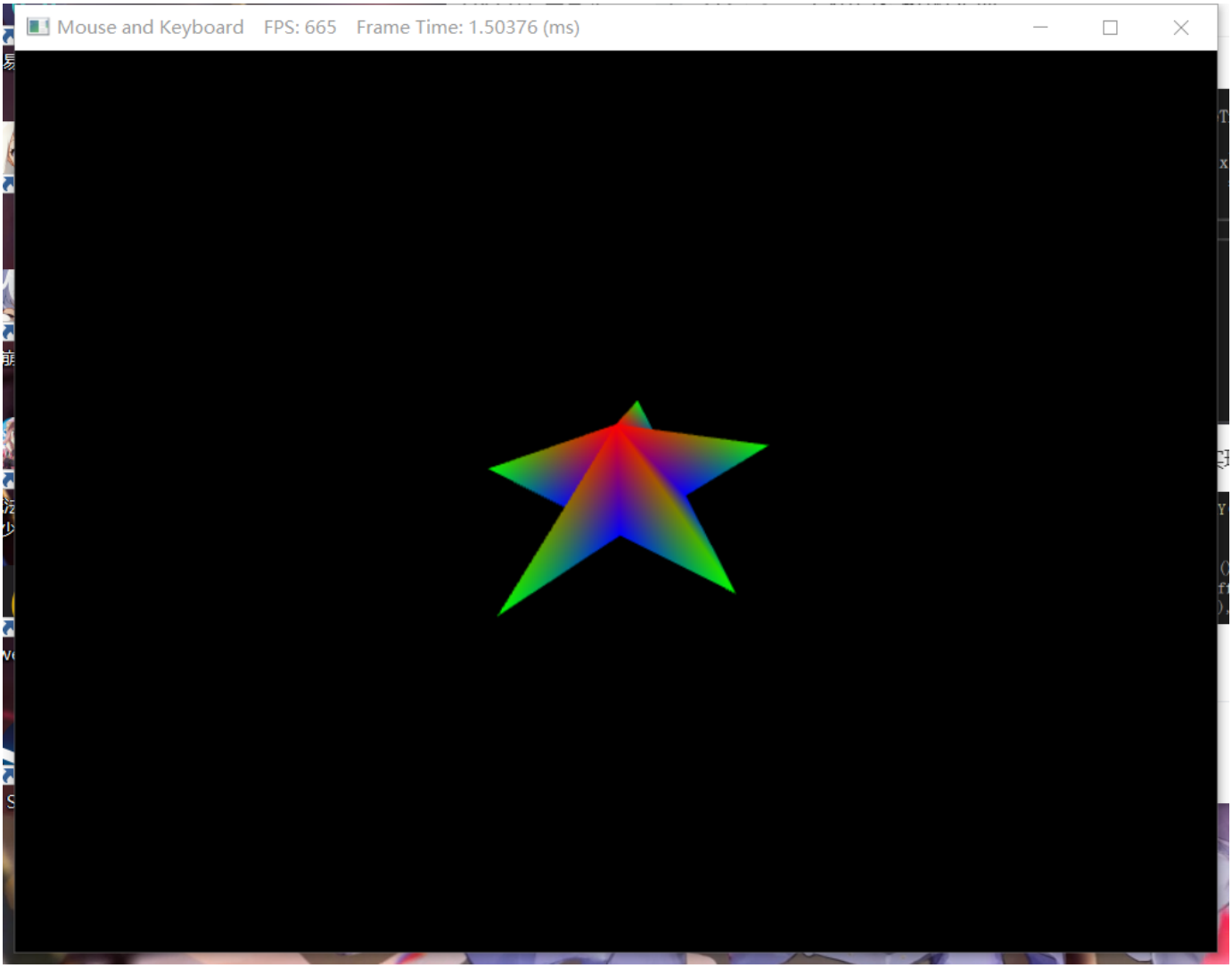
```
static float Phi = 0.0f, Theta = 0.0f;
// 获取鼠标状态
Mouse::State mouseState = m_pMouse->GetState();
Mouse::State lastMouseState = m_MouseTracker.GetLastState();
// 获取键盘状态
Keyboard::State keyState = m_pKeyboard->GetState();
Keyboard::State lastKeyState = m_KeyboardTracker.GetLastState();
// 更新鼠标按钮状态跟踪器，仅当鼠标按住的情况下才进行移动
m_MouseTracker.Update(mouseState);
m_KeyboardTracker.Update(keyState);
if (mouseState.leftButton == true && m_MouseTracker.leftButton == m_MouseTracker.HELD)
{
    Theta -= (mouseState.x - lastMouseState.x) * 0.01f;
    Phi -= (mouseState.y - lastMouseState.y) * 0.01f;
}
//按下对应按键时,旋转量改变
if (keyState.IsKeyDown(Keyboard::W))
    Phi += dt * 2;
if (keyState.IsKeyDown(Keyboard::S))
    Phi -= dt * 2;
if (keyState.IsKeyDown(Keyboard::A))
    Theta += dt * 2;
if (keyState.IsKeyDown(Keyboard::D))
    Theta -= dt * 2;
```

将旋转量传入旋转矩阵并更新常量缓冲区,从而实现物体的旋转.

```
m_CBuffer.world = XMMatrixTranspose(XMMatrixRotationY(Theta) * XMMatrixRotationX(Phi));
// 更新常量缓冲区，让物体转起来
D3D11_MAPPED_SUBRESOURCE mappedData;
HR(m_pd3dImmediateContext->Map(m_pConstantBuffer.Get(), 0, D3D11_MAP_WRITE_DISCARD, 0, &mappedData));
memcpy_s(mappedData.pData, sizeof(m_CBuffer), &m_CBuffer, sizeof(m_CBuffer));
m_pd3dImmediateContext->Unmap(m_pConstantBuffer.Get(), 0);
```

结果


Mouse and Keyboard FPS: 665 Frame Time: 1.50376 (ms)


Mouse and Keyboard FPS: 665 Frame Time: 1.50376 (ms)