实验报告 1

姓名：蒋卓洋
学号：59119125

1、实现

（1）函数位置：main.cpp
（2）函数实现：

A.get_model_matrix()函数

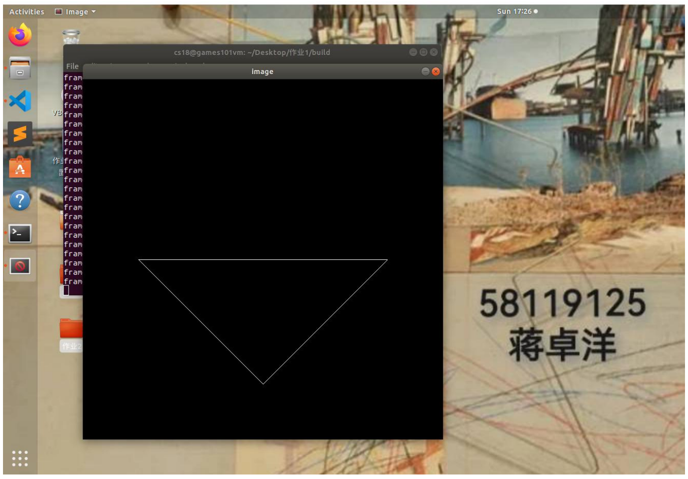
```
Eigen::Matrix4f get_model_matrix(float rotation_angle)
{
    Eigen::Matrix4f model = Eigen::Matrix4f::Identity();

    // TODO: Implement this function
    // Create the model matrix for rotating the triangle around the Z axis.
    // Then return it.
    ////////////////Solution//////////////////
    ////Name:JiangZhuoyang
    ////StudentID:58119125
    ////FinishDate:21/9/30

    //1.Translation to origin: no movement
    Eigen::Matrix4f translate_pos;
    translate_pos << 1,0,0,0,
                     0,1,0,0,
                     0,0,1,0,
                     0,0,0,1;

    //2.Scale: no need
    Eigen::Matrix4f scale;
    scale << 1,0,0,0,
             0,1,0,0,
             0,0,1,0,
             0,0,0,1;

    //3.Rotation: main work WITH Z
    Eigen::Matrix4f rotation;
    float rotation_radian = rotation_angle * MY_PI / 180.0;//angle to radian
    rotation << cos(rotation_radian),-sin(rotation_radian),0,0,
                sin(rotation_radian),cos(rotation_radian),0,0,
                0,0,1,0,
                0,0,0,1;


    //4.Translation back: no movement
    Eigen::Matrix4f translate_neg;
    translate_neg << 1,0,0,0,
                    0,1,0,0,
                    0,0,1,0,
                    0,0,0,1;

    //5.Return model
    model = translate_neg * rotation * scale * translate_pos;
    ///////////////////////////////////////////

    return model;
}
```

B.get_projection_matrix()函数

```
Eigen::Matrix4f get_projection_matrix(float eye_fov, float aspect_ratio,
                                      float zNear, float zFar)
{
    // Students will implement this function

    Eigen::Matrix4f projection = Eigen::Matrix4f::Identity();

    // TODO: Implement this function
    // Create the projection matrix for the given parameters.
    // Then return it.
    /////////////////Solution//////////////////
    ////Name:JiangZhuoyang
    ////StudentID:58119125
    ////FinishDate:21/9/30
    //1.Definition
    Eigen::Matrix4f perspective = Eigen::Matrix4f::Identity();//perspective projection
     Eigen::Matrix4f persp_to_ortho = Eigen::Matrix4f::Identity();//turn the perspective projection to orthographic projection
    Eigen::Matrix4f orthographic = Eigen::Matrix4f::Identity();//orthographic projection

    //2.Construction
    //2.1.P -> O
    persp_to_ortho << zNear,0,0,0,
                0,zNear,0,0,
                0,0,zNear+zFar,-zNear*zFar,
                0,0,1,0;

    //2.2.Orthographic
    //(1)Get edges: implicitly, do the first translation with the use of eye fov and aspect ration directly.
    float yTop = zNear * tan( (eye_fov/2) * MY_PI / 180 );
    float yBottom = (-1) * yTop;
    float xLeft = yTop * aspect_ratio;
    float xRigth = (-1) * xLeft;
    //(2)Orthographic translate:
    Eigen::Matrix4f ortho_trans = Eigen::Matrix4f::Identity();
    ortho_trans << 1, 0, 0, -(xRigth + xLeft) / 2, //(1,0,0,0)
              0, 1, 0, -(yTop + yBottom) / 2, //(0,1,0,0)
              0, 0, 1, -(zNear + zFar) / 2,
              0, 0, 0, 1;
    //(3)Orthographic scale:
    Eigen::Matrix4f ortho_scale = Eigen::Matrix4f::Identity();
    ortho_scale << 2/(xRigth - xLeft), 0, 0, 0, //(1,0,0,0)
              0, 2/(yTop - yBottom), 0, 0, //(0,1,0,0)
              0, 0, 2/(zNear - zFar), 0,
              0, 0, 0, 1;
    //(3)Orthographic Matrix:
    orthographic = ortho_scale * ortho_trans;

    //2.3.Perspective:
    perspective = orthographic * persp_to_ortho;

    //3.projection:
    projection = perspective;
    ////////////////////////////////////////////
    return projection;
}
```

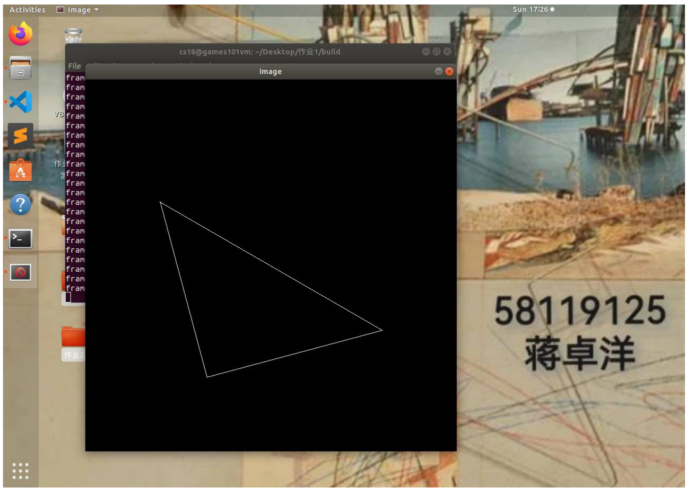
2、时结果

实验结果如下：



```
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
```

图 1.作业一结果--初始显示



```
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
fram
```

图 1.作业一结果--按三次 A 后显示