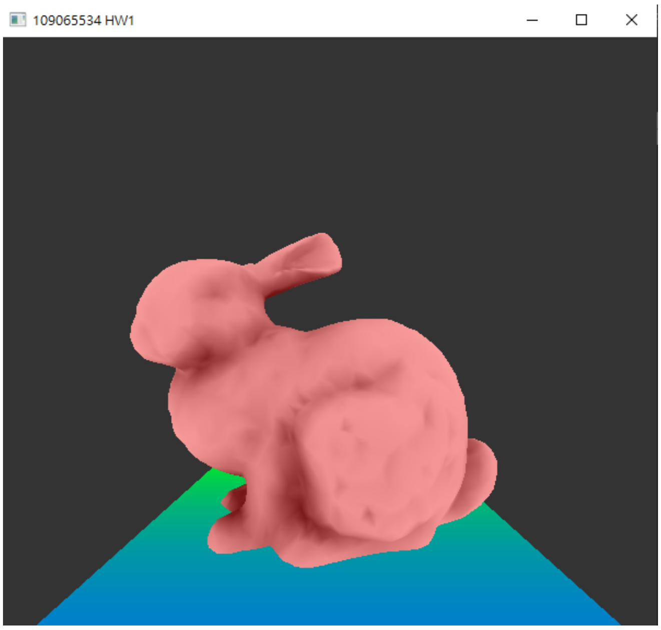
CG HW1 Report

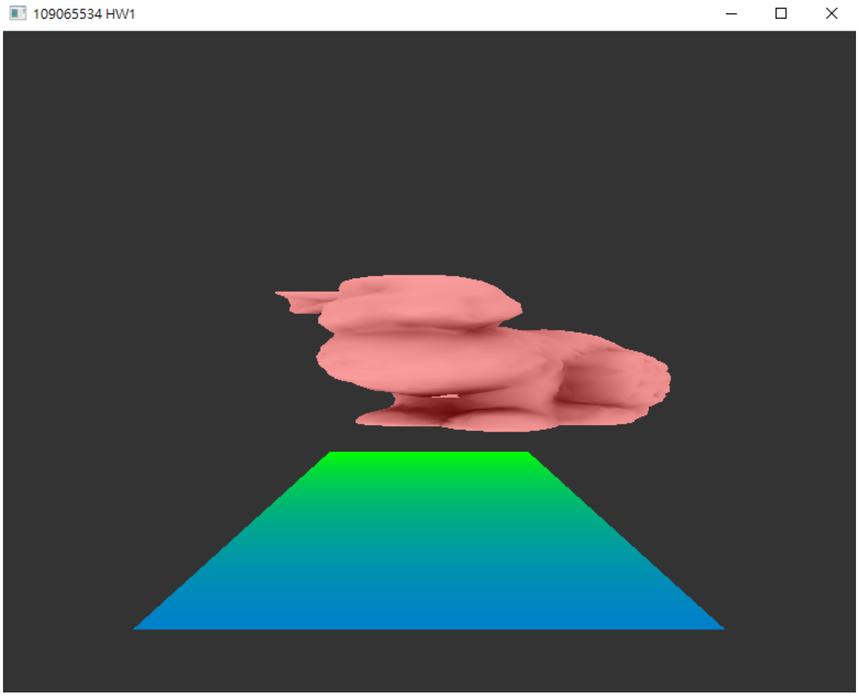
花了一點時間理解各個指令的功能，按照上課第六章的投影片一步一步實作矩陣。MVP矩陣的概念有比較難弄懂，分離物件跟平面多花了一點時間做嘗試。

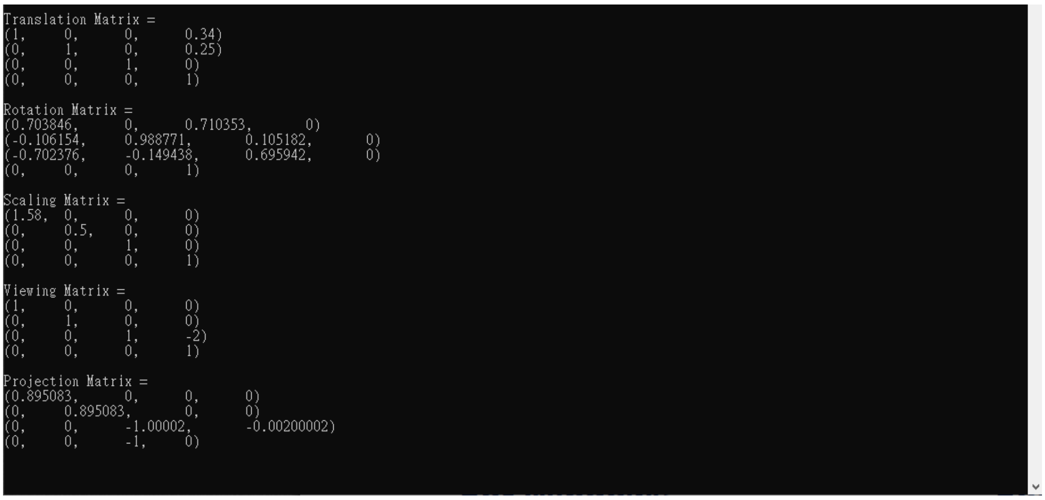
原始擺設：



information:

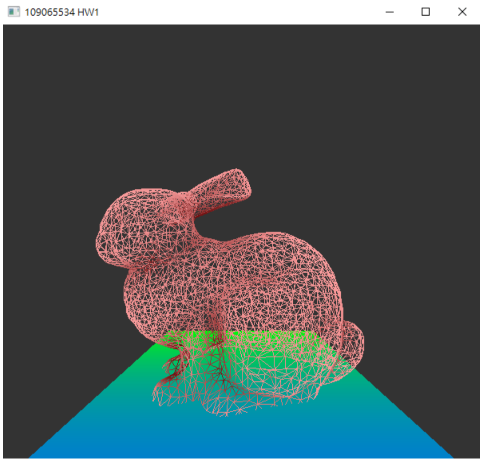
確認矩陣資訊跟拉伸視窗會不會形變



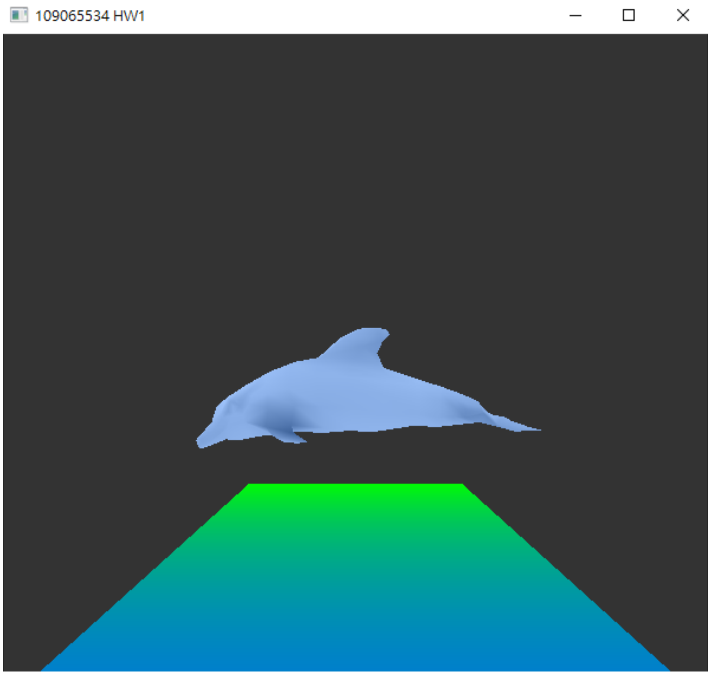


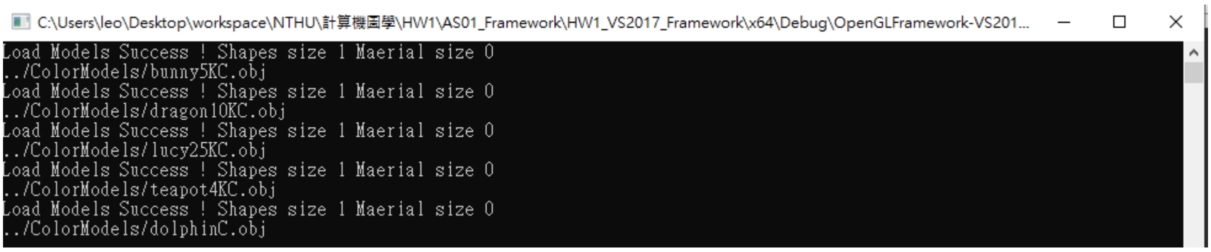
```
Translation Matrix =
(1,     0,      0,      0.34)
(0,     1,      0,      0.25)
(0,     0,      1,      0)
(0,     0,      0,      1)

Rotation Matrix =
(0.703846,      0,      0.710353,       0)
(-0.106154,     0.988771,       0.105182,       0)
(-0.702376,     -0.149438,      0.695942,       0)
(0,     0,      0,      1)

Scaling Matrix =
(1.58,  0,      0,      0)
(0,     0.5,    0,      0)
(0,     0,      1,      0)
(0,     0,      0,      1)

Viewing Matrix =
(1,     0,      0,      0)
(0,     1,      0,      0)
(0,     0,      1,      -2)
(0,     0,      0,      1)

Projection Matrix =
(0.895083,      0,      0,      0)
(0,     0.895083,       0,      0)
(0,     0,      -1.00002,       -0.00200002)
(0,     0,      -1,     0)
```

網狀 :



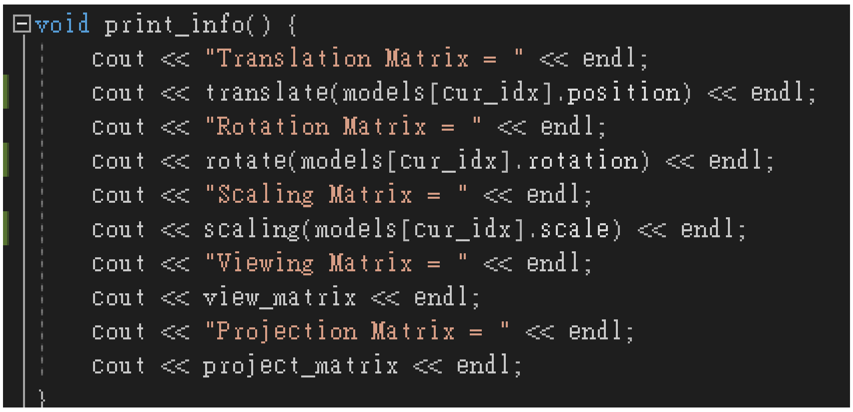
載入其他model:



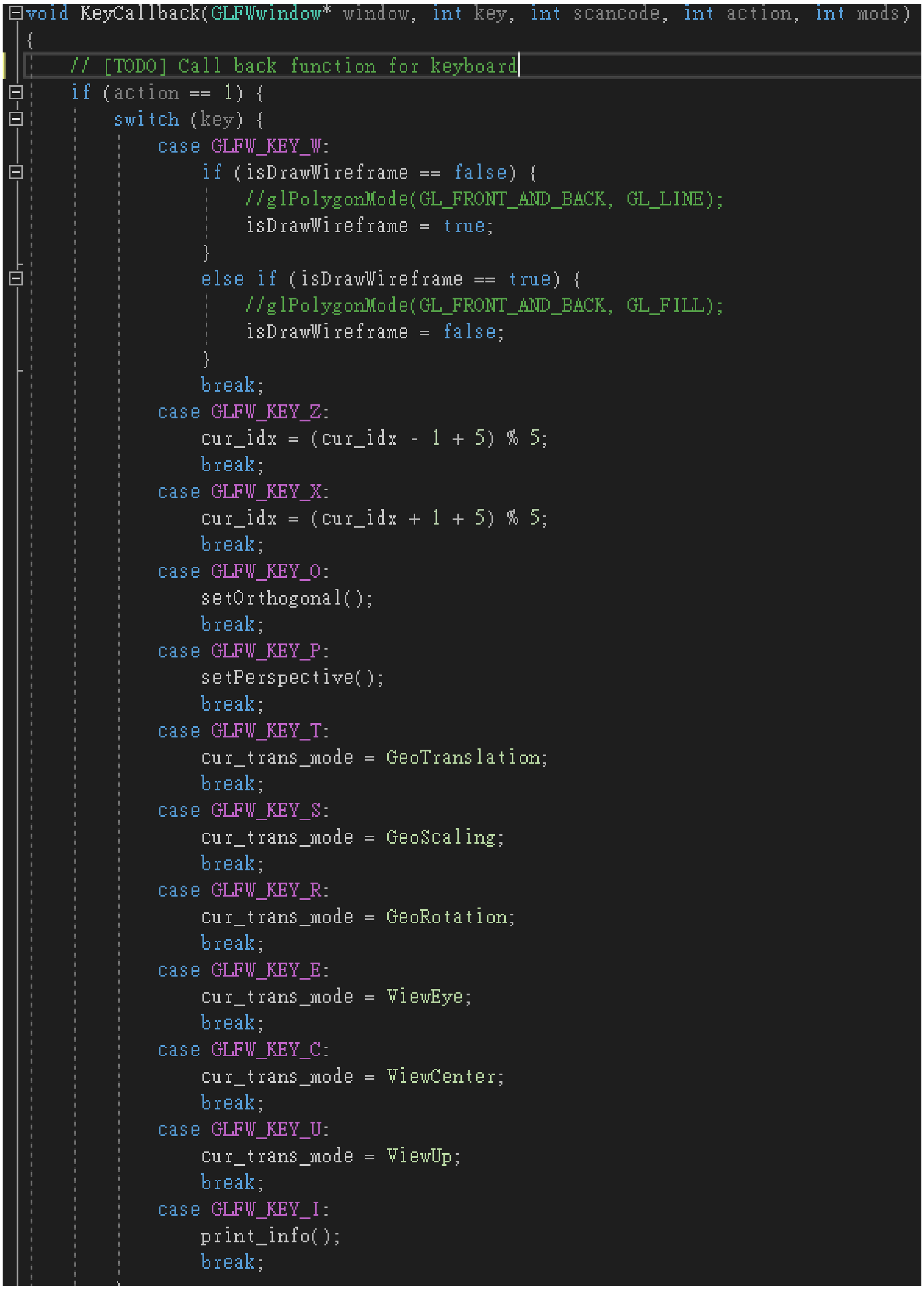

C:\Users\leo\Desktop\workspace\NTHU\計算機圖學\HW1\AS01_Framework\HW1_VS2017_Framework\x64\Debug\OpenGLFramework-VS201...

```
Load Models Success ! Shapes size 1 Maerial size 0
../ColorModels/bunny5KC.obj
Load Models Success ! Shapes size 1 Maerial size 0
../ColorModels/dragon10KC.obj
Load Models Success ! Shapes size 1 Maerial size 0
../ColorModels/lucy25KC.obj
Load Models Success ! Shapes size 1 Maerial size 0
../ColorModels/teapot4KC.obj
Load Models Success ! Shapes size 1 Maerial size 0
../ColorModels/dolphinC.obj
```

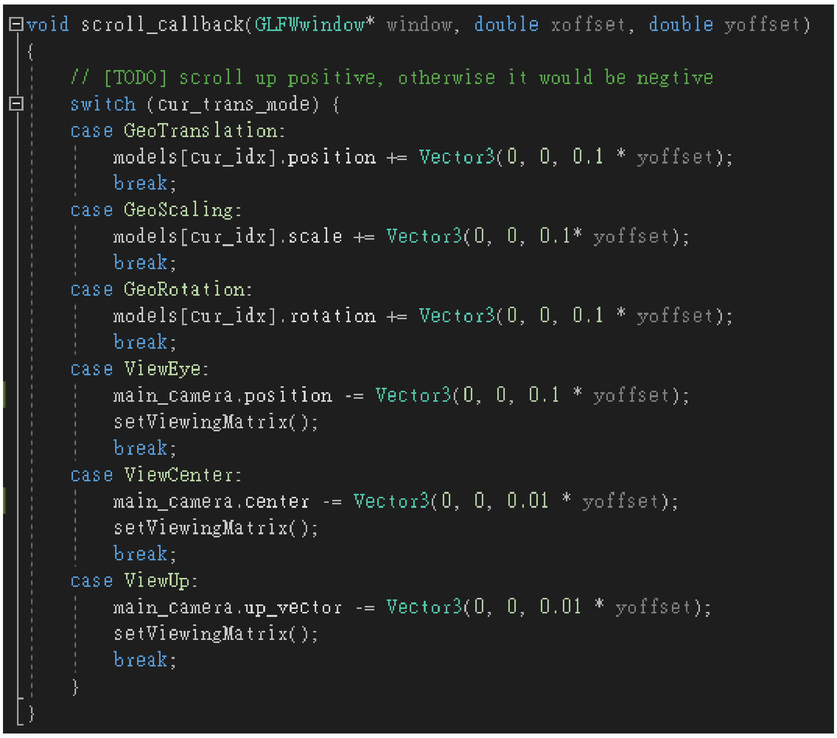
滑鼠功能code:



```
void print_info() {
    cout << "Translation Matrix = " << endl;
    cout << translate(models[cur_idx].position) << endl;
    cout << "Rotation Matrix = " << endl;
    cout << rotate(models[cur_idx].rotation) << endl;
    cout << "Scaling Matrix = " << endl;
    cout << scaling(models[cur_idx].scale) << endl;
    cout << "Viewing Matrix = " << endl;
    cout << view_matrix << endl;
    cout << "Projection Matrix = " << endl;
    cout << project_matrix << endl;
}
```



```
void KeyCallback(GLFWwindow* window, int key, int scancode, int action, int mods)
{
    // [TODO] Call back function for keyboard
    if (action == 1) {
        switch (key) {
            case GLFW_KEY_W:
                if (isDrawWireframe == false) {
                    //glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
                    isDrawWireframe = true;
                }
                else if (isDrawWireframe == true) {
                    //glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
                    isDrawWireframe = false;
                }
                break;
            case GLFW_KEY_Z:
                cur_idx = (cur_idx - 1 + 5) % 5;
                break;
            case GLFW_KEY_X:
                cur_idx = (cur_idx + 1 + 5) % 5;
                break;
            case GLFW_KEY_O:
                setOrthogonal();
                break;
            case GLFW_KEY_P:
                setPerspective();
                break;
            case GLFW_KEY_T:
                cur_trans_mode = GeoTranslation;
                break;
            case GLFW_KEY_S:
                cur_trans_mode = GeoScaling;
                break;
            case GLFW_KEY_R:
                cur_trans_mode = GeoRotation;
                break;
            case GLFW_KEY_E:
                cur_trans_mode = ViewEye;
                break;
            case GLFW_KEY_C:
                cur_trans_mode = ViewCenter;
                break;
            case GLFW_KEY_U:
                cur_trans_mode = ViewUp;
                break;
            case GLFW_KEY_I:
                print_info();
                break;
```



```
void scroll_callback(GLFWwindow* window, double xoffset, double yoffset)
{
    // [TODO] scroll up positive, otherwise it would be negtive
    switch (cur_trans_mode) {
    case GeoTranslation:
        models[cur_idx].position += Vector3(0, 0, 0.1 * yoffset);
        break;
    case GeoScaling:
        models[cur_idx].scale += Vector3(0, 0, 0.1* yoffset);
        break;
    case GeoRotation:
        models[cur_idx].rotation += Vector3(0, 0, 0.1 * yoffset);
        break;
    case ViewEye:
        main_camera.position -= Vector3(0, 0, 0.1 * yoffset);
        setViewingMatrix();
        break;
    case ViewCenter:
        main_camera.center -= Vector3(0, 0, 0.01 * yoffset);
        setViewingMatrix();
        break;
    case ViewUp:
        main_camera.up_vector -= Vector3(0, 0, 0.01 * yoffset);
        setViewingMatrix();
        break;
    }
}
```