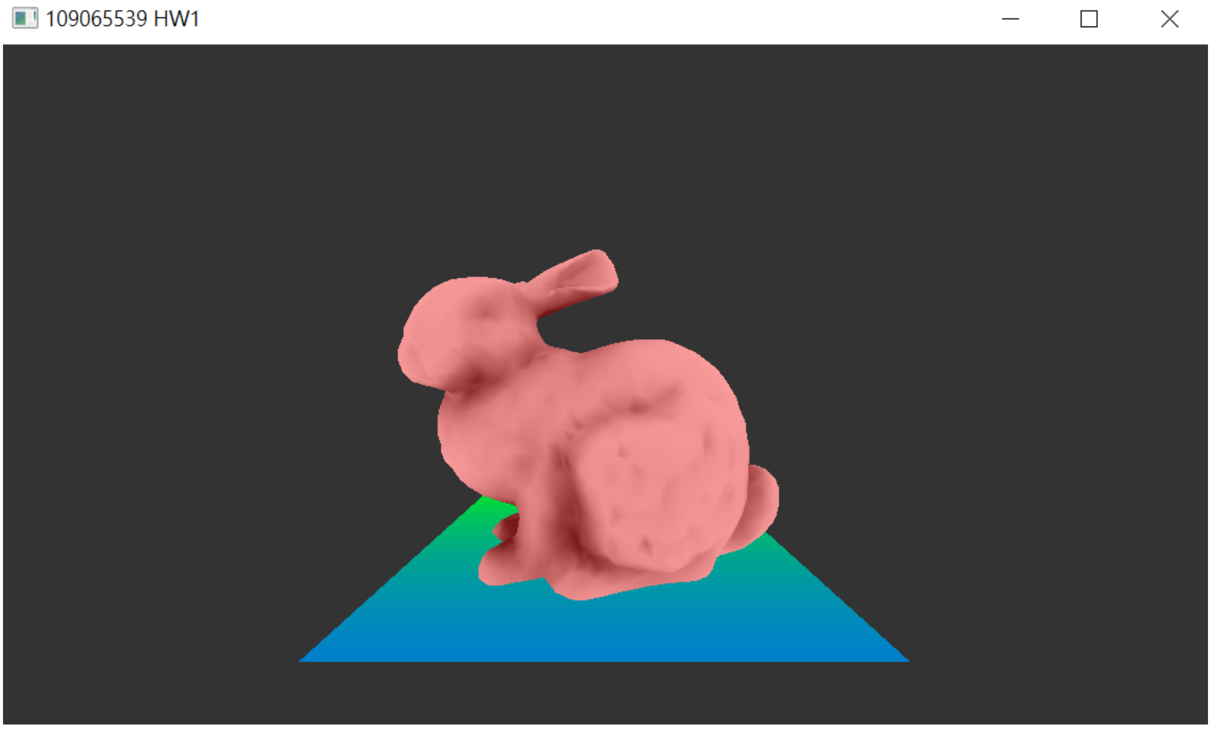
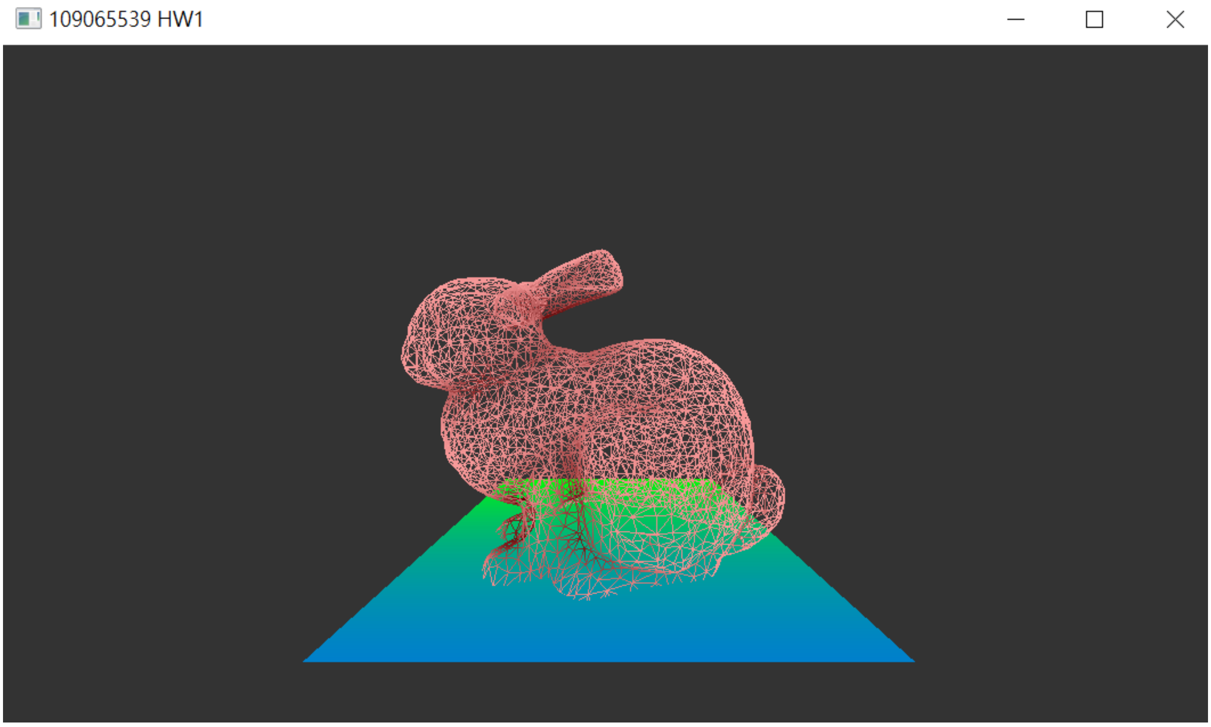
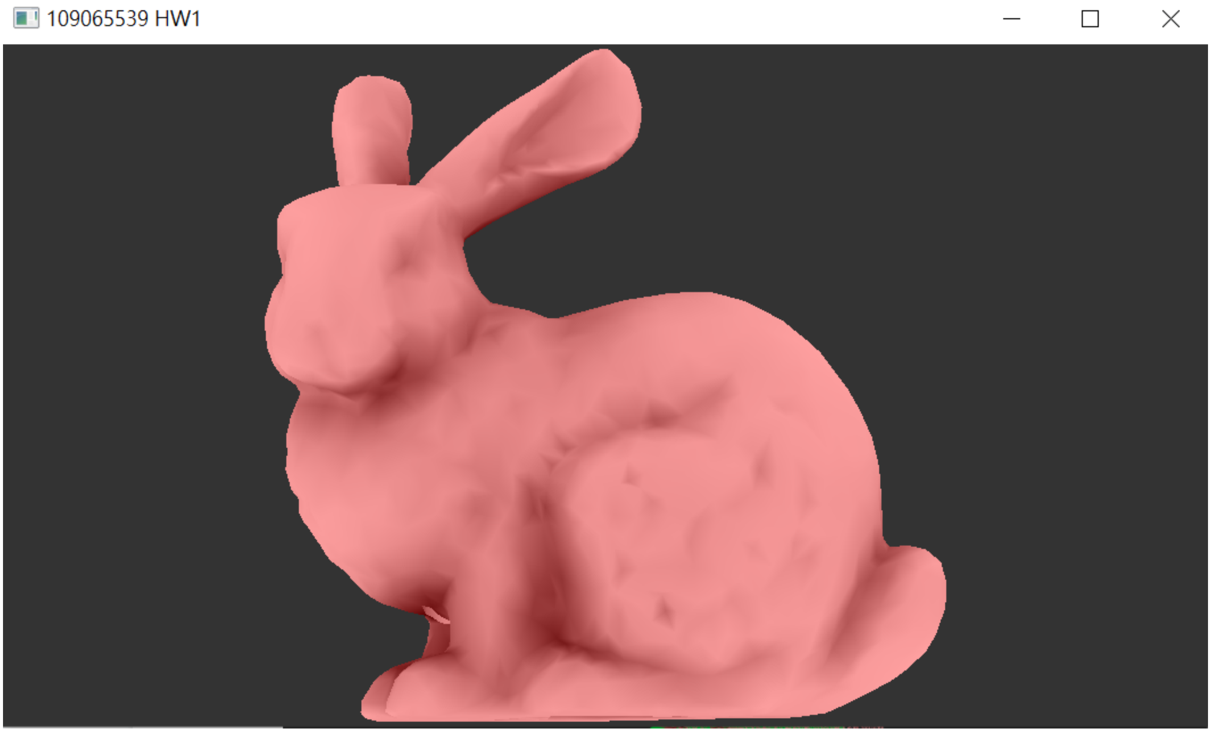
HW1_Report

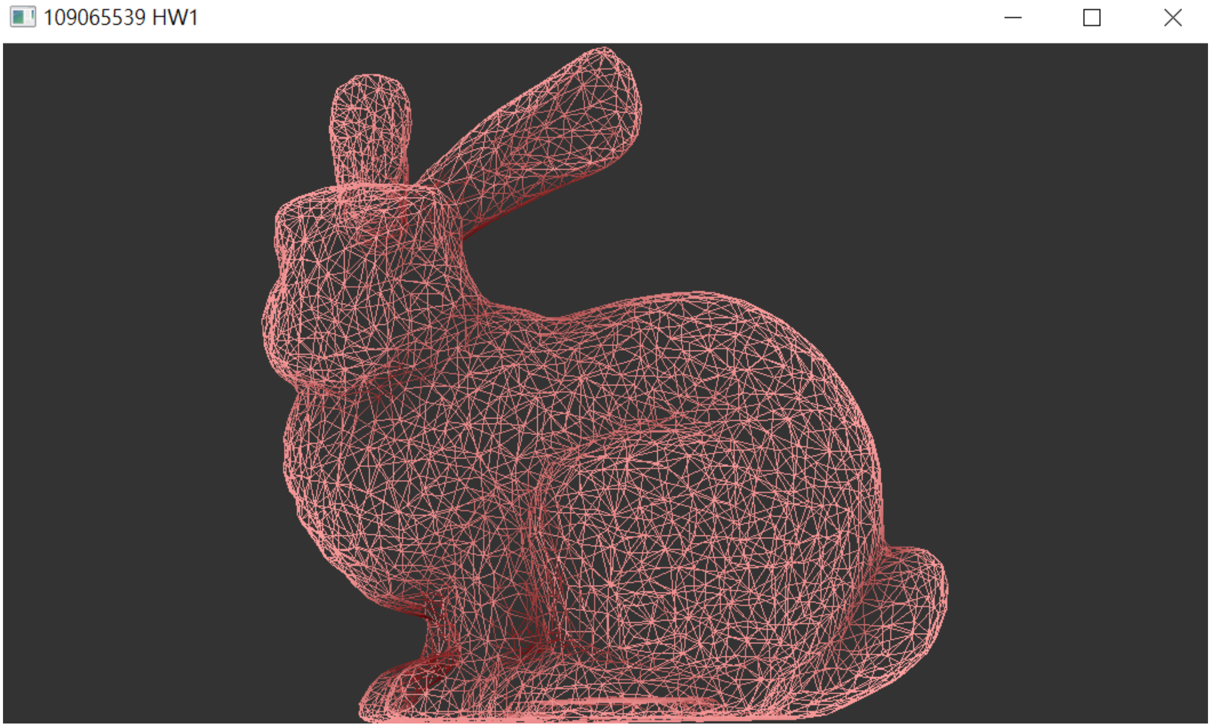
109065539 韓承翰

Demo：



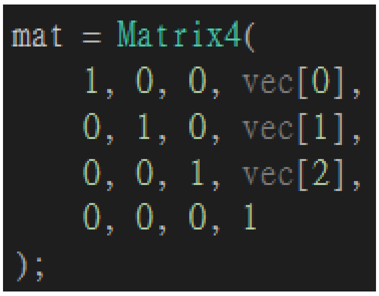



109065539 HW1


109065539 HW1

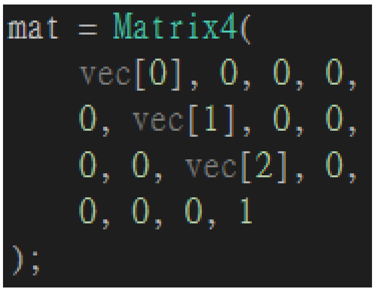
作業要求code說明：

- Matrix4 translate:參考PPT CG06 第39頁



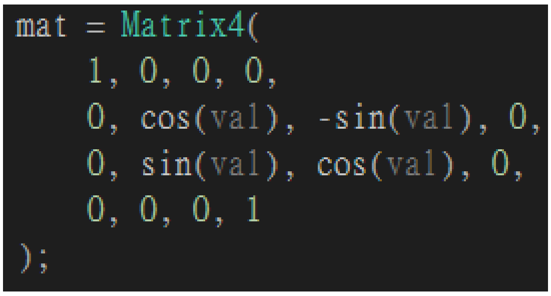
```
mat = Matrix4(
    1, 0, 0, vec[0],
    0, 1, 0, vec[1],
    0, 0, 1, vec[2],
    0, 0, 0, 1
);
```

- Matrix4 scaling:參考PPT CG06 第40頁



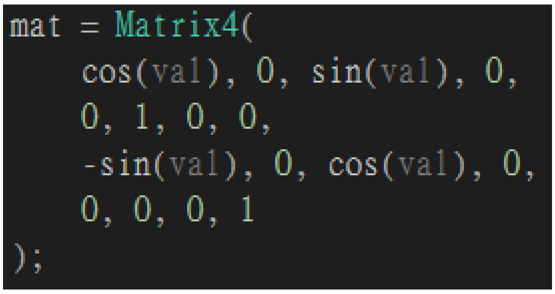
```
mat = Matrix4(
    vec[0], 0, 0, 0,
    0, vec[1], 0, 0,
    0, 0, vec[2], 0,
    0, 0, 0, 1
);
```

- Matrix4 rotationX:參考PPT CG06 第41頁



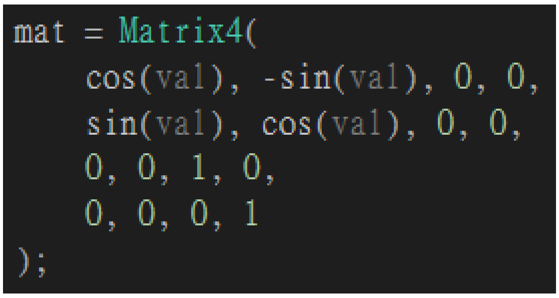
```
mat = Matrix4(
    1, 0, 0, 0,
    0, cos(val), -sin(val), 0,
    0, sin(val), cos(val), 0,
    0, 0, 0, 1
);
```

- Matrix4 rotationY:參考PPT CG06 第42頁



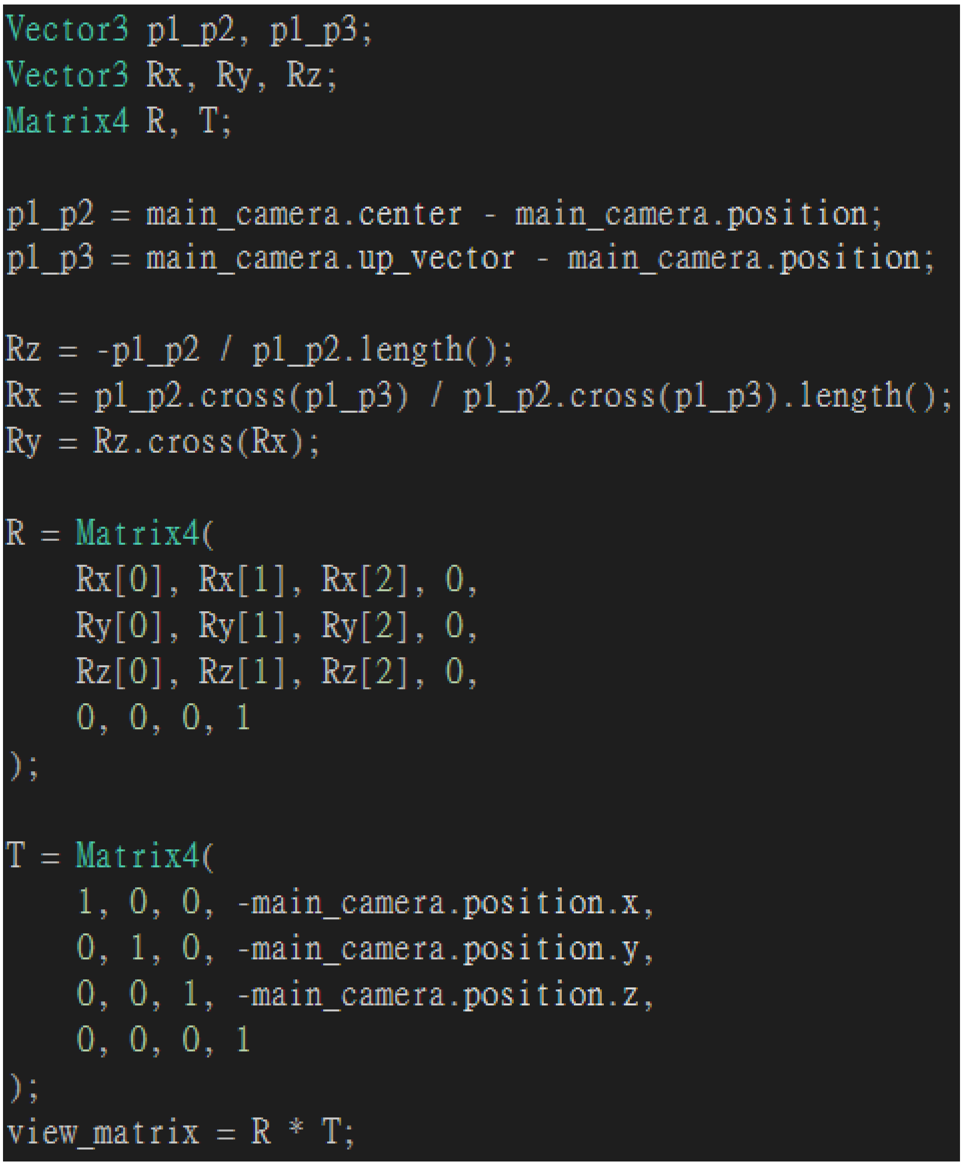
```
mat = Matrix4(
    cos(val), 0, sin(val), 0,
    0, 1, 0, 0,
    -sin(val), 0, cos(val), 0,
    0, 0, 0, 1
);
```

- Matrix4 rotationZ：參考PPT CG06 第43頁



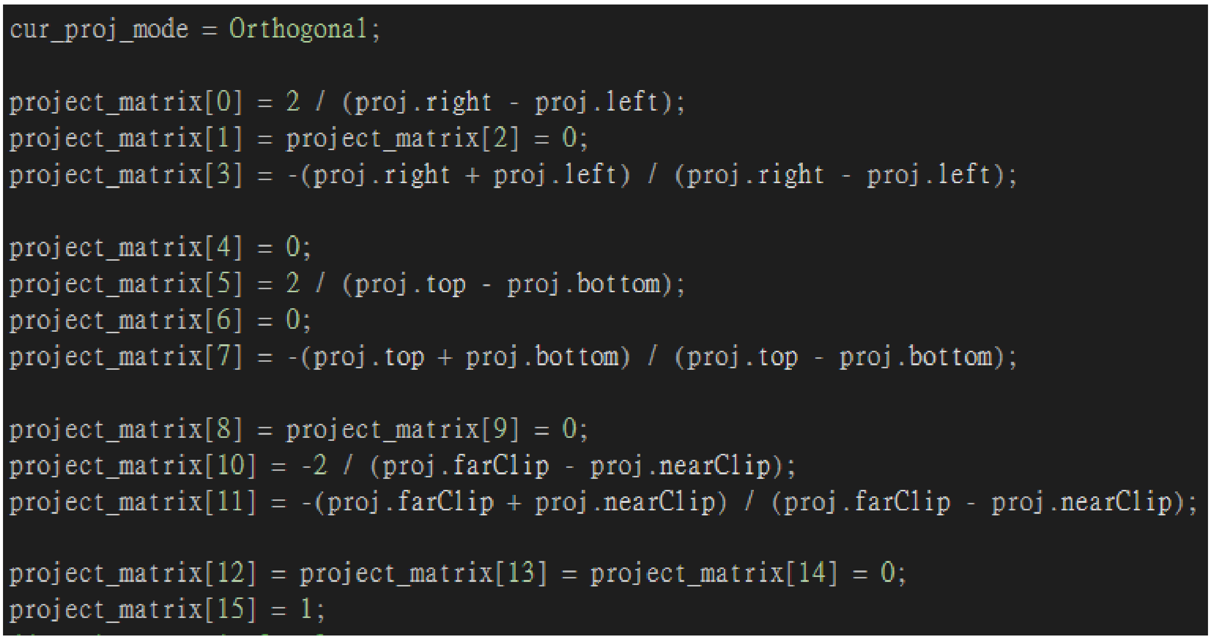
```
mat = Matrix4(
    cos(val), -sin(val), 0, 0,
    sin(val), cos(val), 0, 0,
    0, 0, 1, 0,
    0, 0, 0, 1
);
```

- setViewingMatrix()：參考PPT CG06 第71-74頁



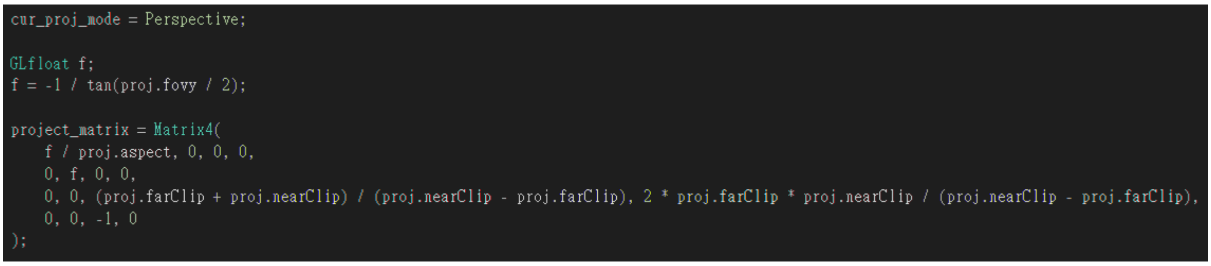
```
Vector3 p1_p2, p1_p3;
Vector3 Rx, Ry, Rz;
Matrix4 R, T;

p1_p2 = main_camera.center - main_camera.position;
p1_p3 = main_camera.up_vector - main_camera.position;

Rz = -p1_p2 / p1_p2.length();
Rx = p1_p2.cross(p1_p3) / p1_p2.cross(p1_p3).length();
Ry = Rz.cross(Rx);

R = Matrix4(
    Rx[0], Rx[1], Rx[2], 0,
    Ry[0], Ry[1], Ry[2], 0,
    Rz[0], Rz[1], Rz[2], 0,
    0, 0, 0, 1
);

T = Matrix4(
    1, 0, 0, -main_camera.position.x,
    0, 1, 0, -main_camera.position.y,
    0, 0, 1, -main_camera.position.z,
    0, 0, 0, 1
);
view_matrix = R * T;
```

- setOrthogonal():參考PPT CG06 第127頁



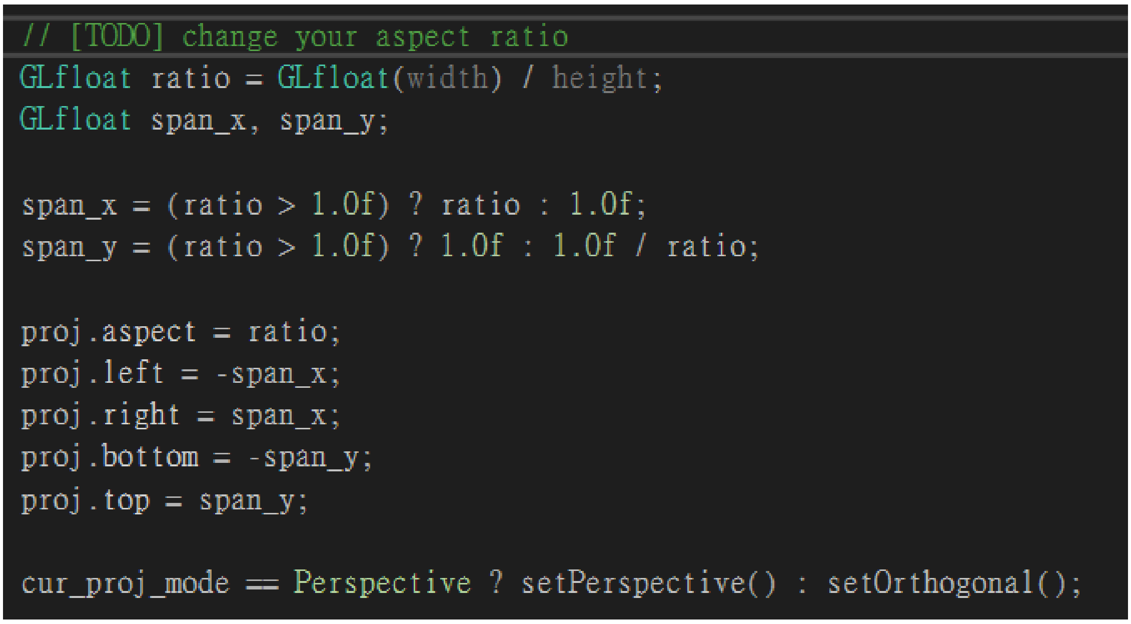
```
cur_proj_mode = Orthogonal;

project_matrix[0] = 2 / (proj.right - proj.left);
project_matrix[1] = project_matrix[2] = 0;
project_matrix[3] = -(proj.right + proj.left) / (proj.right - proj.left);

project_matrix[4] = 0;
project_matrix[5] = 2 / (proj.top - proj.bottom);
project_matrix[6] = 0;
project_matrix[7] = -(proj.top + proj.bottom) / (proj.top - proj.bottom);

project_matrix[8] = project_matrix[9] = 0;
project_matrix[10] = -2 / (proj.farClip - proj.nearClip);
project_matrix[11] = -(proj.farClip + proj.nearClip) / (proj.farClip - proj.nearClip);

project_matrix[12] = project_matrix[13] = project_matrix[14] = 0;
project_matrix[15] = 1;
```

- setPerspective():參考PPT CG06 第122頁



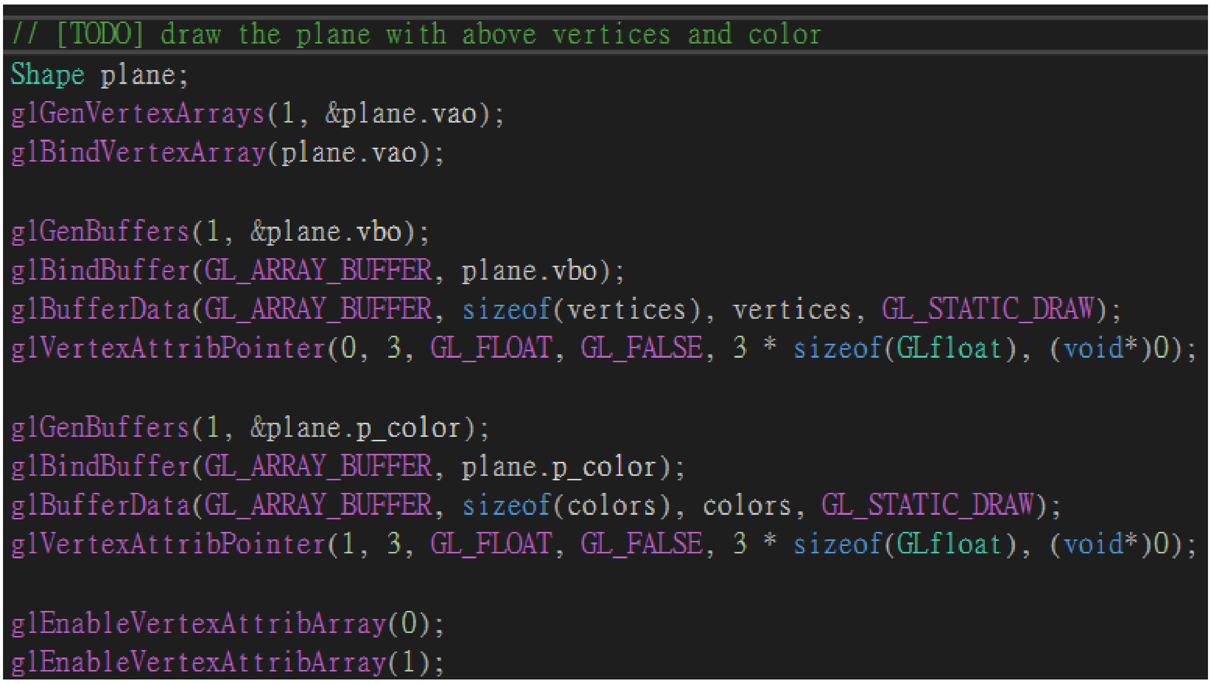
```
cur_proj_mode = Perspective;

GLfloat f;
f = -1 / tan(proj.fovy / 2);

project_matrix = Matrix4(
    f / proj.aspect, 0, 0, 0,
    0, f, 0, 0,
    0, 0, (proj.farClip + proj.nearClip) / (proj.nearClip - proj.farClip), 2 * proj.farClip * proj.nearClip / (proj.nearClip - proj.farClip),
    0, 0, -1, 0
);
```

- ChangeSize():參考https://stackoverflow.com/a/50329737

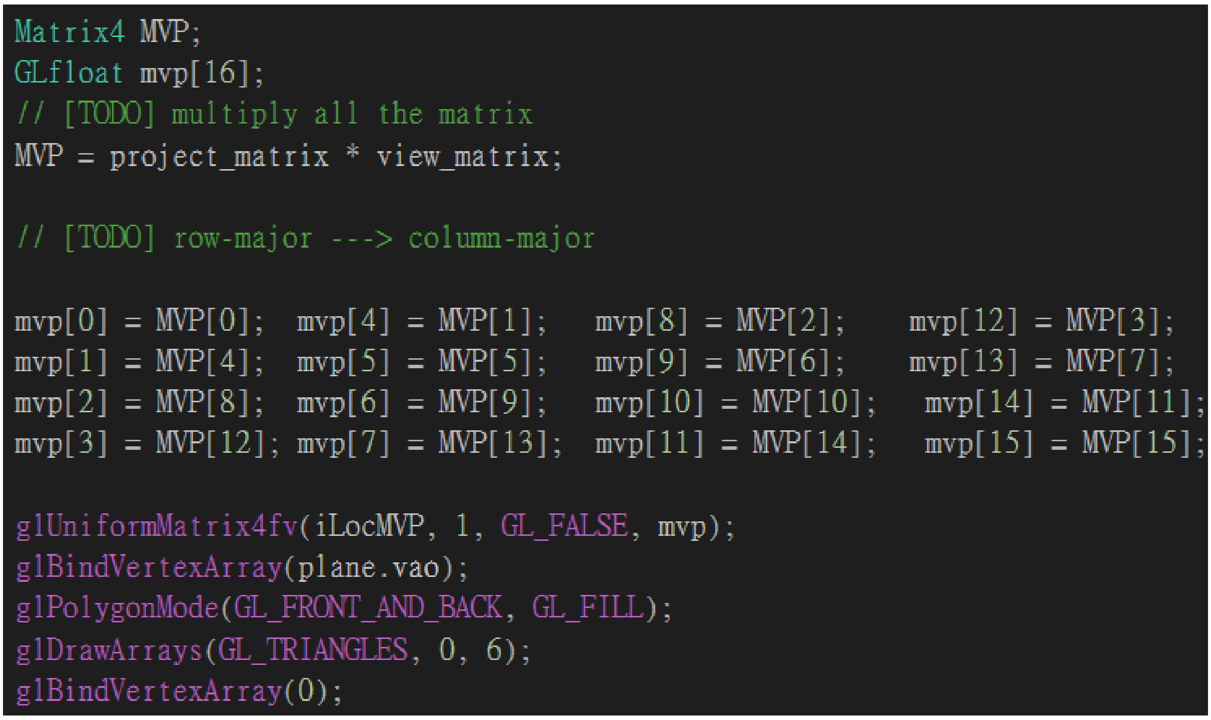


```
// [TODO] change your aspect ratio
GLfloat ratio = GLfloat(width) / height;
GLfloat span_x, span_y;

span_x = (ratio > 1.0f) ? ratio : 1.0f;
span_y = (ratio > 1.0f) ? 1.0f : 1.0f / ratio;

proj.aspect = ratio;
proj.left = -span_x;
proj.right = span_x;
proj.bottom = -span_y;
proj.top = span_y;

cur_proj_mode == Perspective ? setPerspective() : setOrthogonal();
```

- drawPlane(): 參考loadmodels()



```
// [TODO] draw the plane with above vertices and color
Shape plane;
glGenVertexArrays(1, &plane.vao);
glBindVertexArray(plane.vao);

glGenBuffers(1, &plane.vbo);
glBindBuffer(GL_ARRAY_BUFFER, plane.vbo);
glBufferData(GL_ARRAY_BUFFER, sizeof(vertices), vertices, GL_STATIC_DRAW);
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 3 * sizeof(GLfloat), (void*)0);

glGenBuffers(1, &plane.p_color);
glBindBuffer(GL_ARRAY_BUFFER, plane.p_color);
glBufferData(GL_ARRAY_BUFFER, sizeof(colors), colors, GL_STATIC_DRAW);
glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 3 * sizeof(GLfloat), (void*)0);

glEnableVertexAttribArray(0);
glEnableVertexAttribArray(1);
```



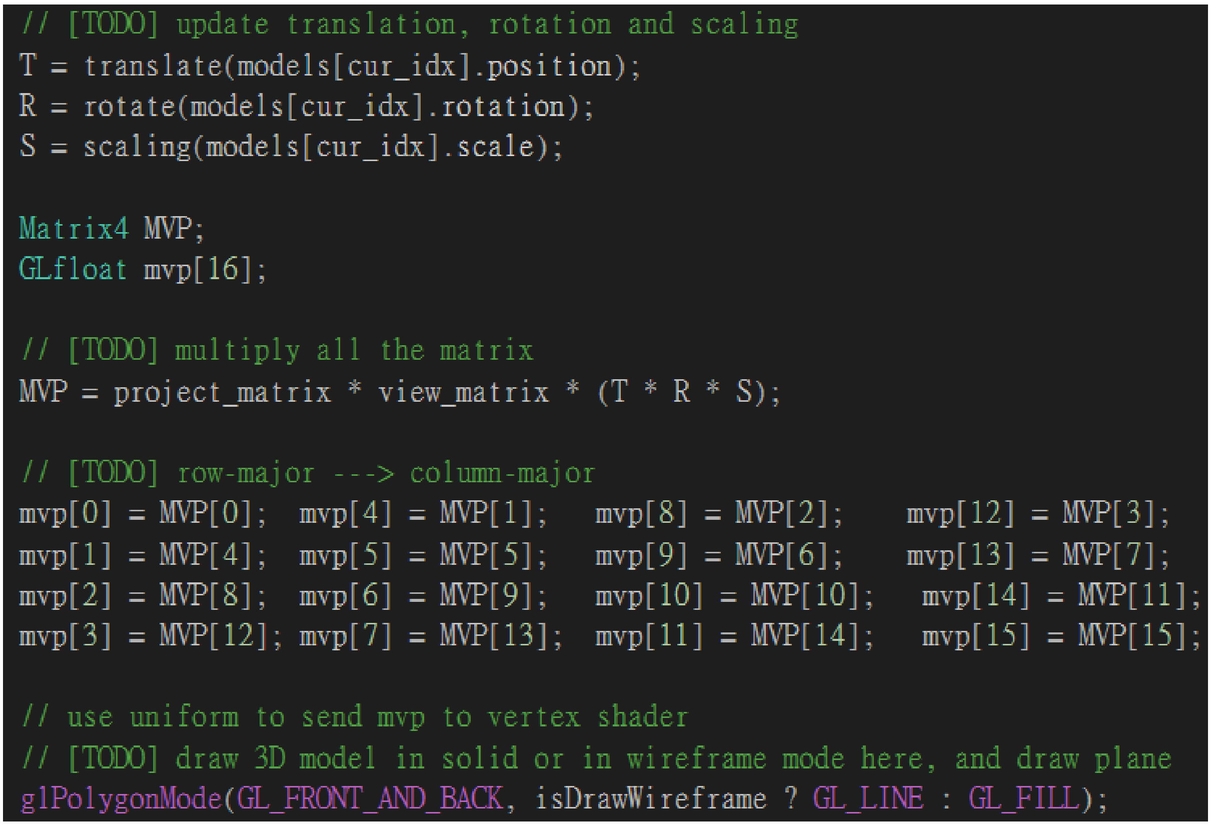
```
Matrix4 MVP;
GLfloat mvp[16];
// [TODO] multiply all the matrix
MVP = project_matrix * view_matrix;

// [TODO] row-major ---> column-major

mvp[0] = MVP[0];  mvp[4] = MVP[1];  mvp[8] = MVP[2];    mvp[12] = MVP[3];
mvp[1] = MVP[4];  mvp[5] = MVP[5];  mvp[9] = MVP[6];    mvp[13] = MVP[7];
mvp[2] = MVP[8];  mvp[6] = MVP[9];  mvp[10] = MVP[10];  mvp[14] = MVP[11];
mvp[3] = MVP[12]; mvp[7] = MVP[13]; mvp[11] = MVP[14];  mvp[15] = MVP[15];

glUniformMatrix4fv(iLocMVP, 1, GL_FALSE, mvp);
glBindVertexArray(plane.vao);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glDrawArrays(GL_TRIANGLES, 0, 6);
glBindVertexArray(0);
```

註:

1. MVP沒有乘上R, T, S 使得平面固定

2. 在DrawArray之前先改回solid模式(GL_FILL), 不會因為模型為wireframe模式而使平面也變成wireframe模式

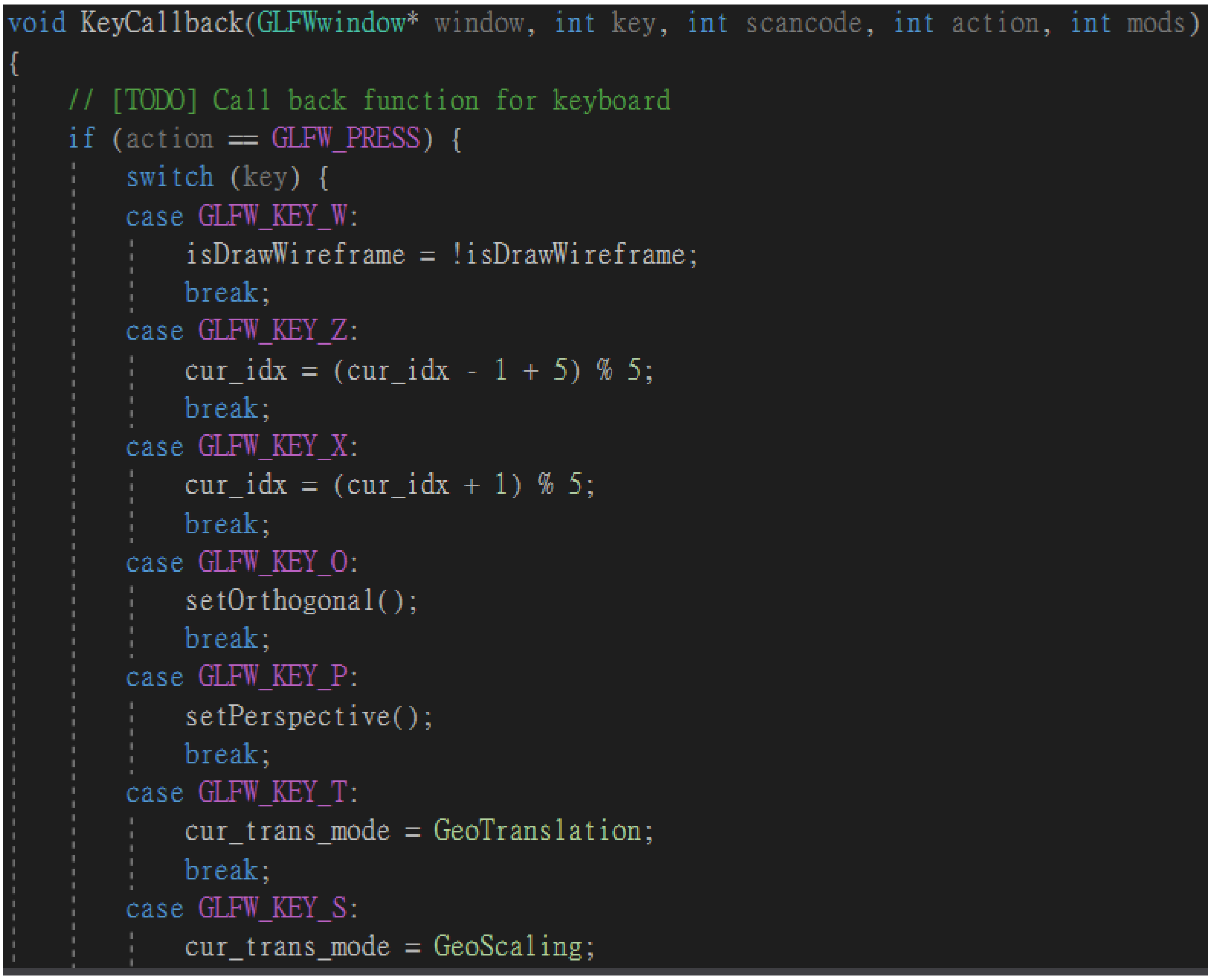
- RenderScene():照著TODO說明打...



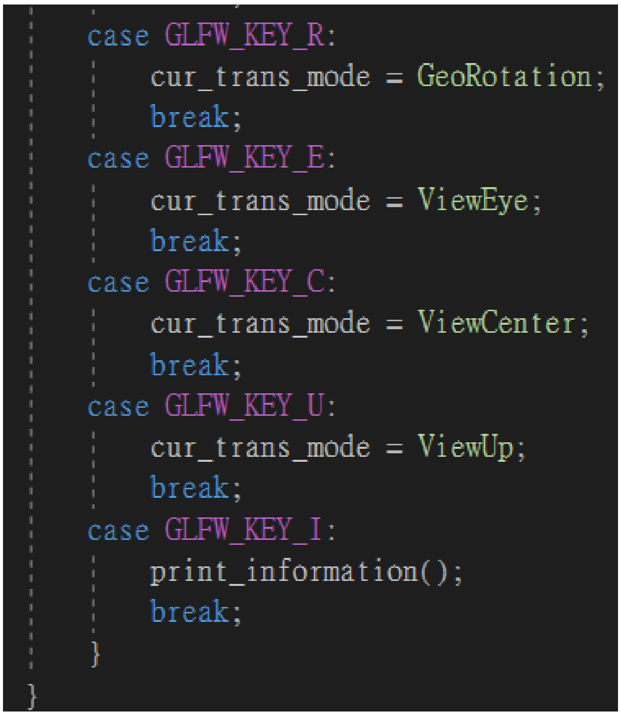
```
// [TODO] update translation, rotation and scaling
T = translate(models[cur_idx].position);
R = rotate(models[cur_idx].rotation);
S = scaling(models[cur_idx].scale);

Matrix4 MVP;
GLfloat mvp[16];

// [TODO] multiply all the matrix
MVP = project_matrix * view_matrix * (T * R * S);

// [TODO] row-major ---> column-major
mvp[0] = MVP[0];  mvp[4] = MVP[1];   mvp[8] = MVP[2];    mvp[12] = MVP[3];
mvp[1] = MVP[4];  mvp[5] = MVP[5];   mvp[9] = MVP[6];    mvp[13] = MVP[7];
mvp[2] = MVP[8];  mvp[6] = MVP[9];   mvp[10] = MVP[10];   mvp[14] = MVP[11];
mvp[3] = MVP[12]; mvp[7] = MVP[13];  mvp[11] = MVP[14];   mvp[15] = MVP[15];

// use uniform to send mvp to vertex shader
// [TODO] draw 3D model in solid or in wireframe mode here, and draw plane
glPolygonMode(GL_FRONT_AND_BACK, isDrawWireframe ? GL_LINE : GL_FILL);
```

註:根據isDrawWireframe來調整模式

- KeyCallback():參考助教提供的glfw Event handings文件

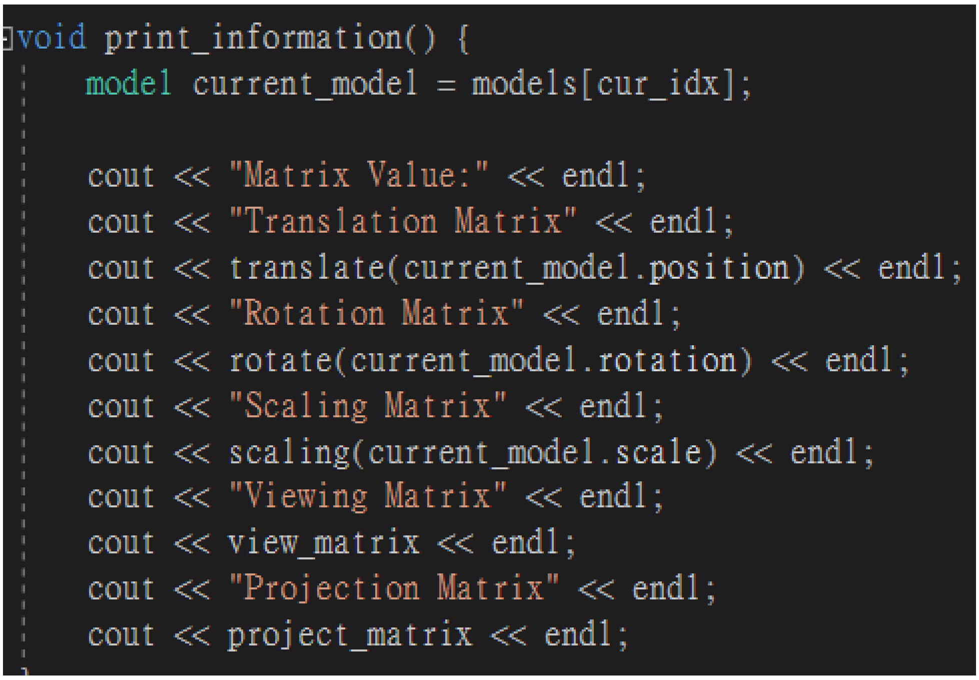


```
void KeyCallback(GLFWwindow* window, int key, int scancode, int action, int mods)
{
    // [TODO] Call back function for keyboard
    if (action == GLFW_PRESS) {
        switch (key) {
        case GLFW_KEY_W:
            isDrawWireframe = !isDrawWireframe;
            break;
        case GLFW_KEY_Z:
            cur_idx = (cur_idx - 1 + 5) % 5;
            break;
        case GLFW_KEY_X:
            cur_idx = (cur_idx + 1) % 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;
```



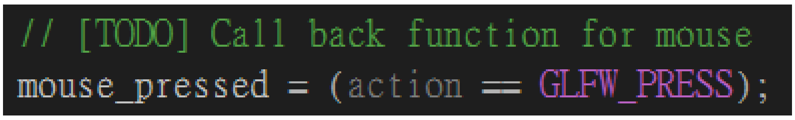
```
        case GLFW_KEY_R:
            cur_trans_mode = GeoRotation;
            break;
        case GLFW_KEY_E:
            cur_trans_mode = ViewEye;
            break;
        case GLFW_KEY_C:
            cur_trans_mode = ViewCenter;
            break;
        case GLFW_KEY_U:
            cur_trans_mode = ViewUp;
            break;
        case GLFW_KEY_I:
            print_information();
            break;
        }
    }
```

- print_information()：使用者按I後顯示matrix資訊



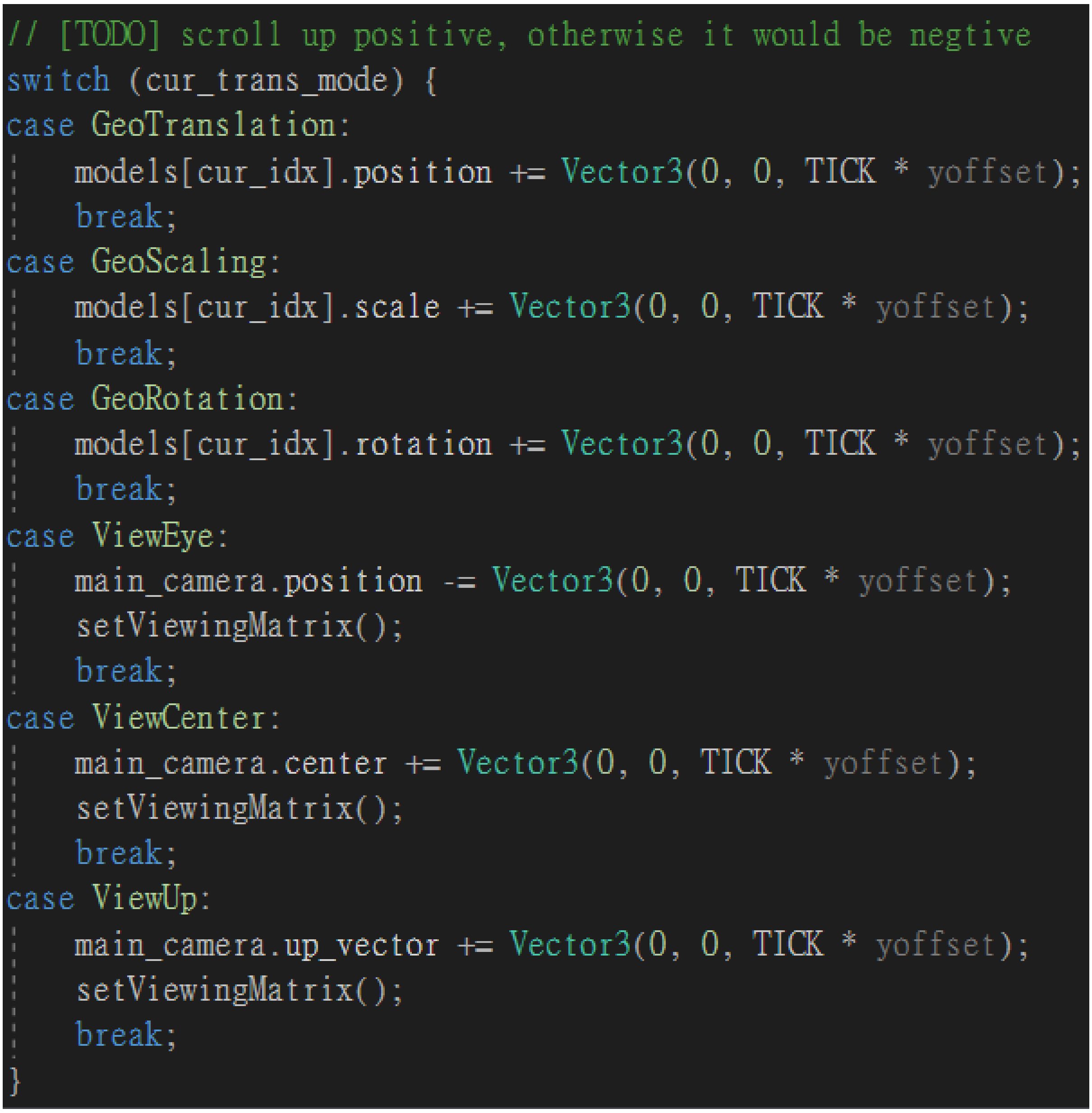
```
void print_information() {
    model current_model = models[cur_idx];

    cout << "Matrix Value:" << endl;
    cout << "Translation Matrix" << endl;
    cout << translate(current_model.position) << endl;
    cout << "Rotation Matrix" << endl;
    cout << rotate(current_model.rotation) << endl;
    cout << "Scaling Matrix" << endl;
    cout << scaling(current_model.scale) << endl;
    cout << "Viewing Matrix" << endl;
    cout << view_matrix << endl;
    cout << "Projection Matrix" << endl;
    cout << project_matrix << endl;
```

- mouse_button_callback()：參考助教提供的glfw Event handings文件



```
// [TODO] Call back function for mouse
mouse_pressed = (action == GLFW_PRESS);
```

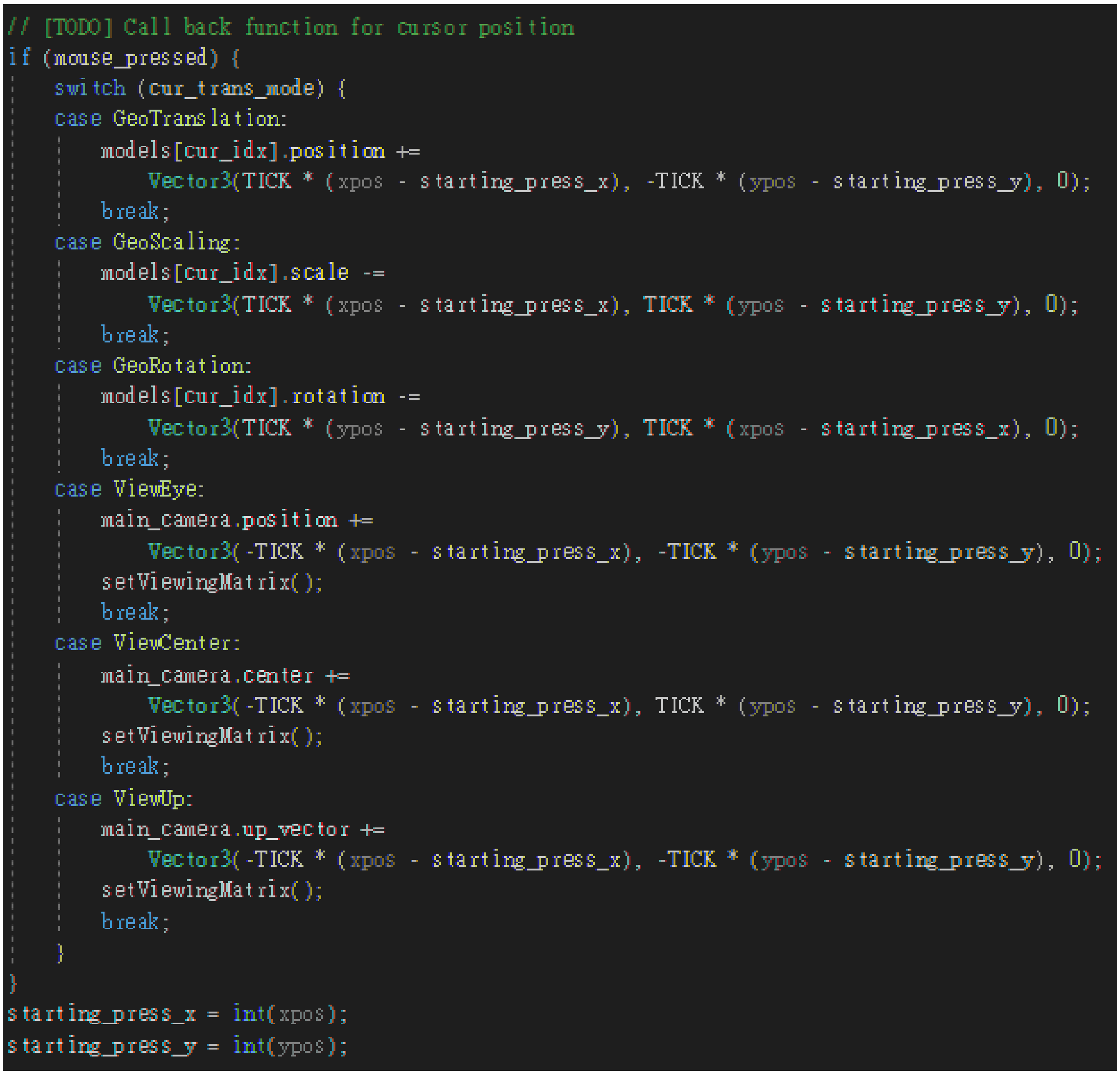
- scroll_callback():參考助教提供的glfw Event handings文件



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

調整正負號與Demo需求相符
TICK為自行設定之全域常數，目前設為0.01

- cursor_pos_callback():參考助教提供的glfw Event handings文件



```
// [TODO] Call back function for cursor position
if (mouse_pressed) {
    switch (cur_trans_mode) {
    case GeoTranslation:
        models[cur_idx].position +=
            Vector3(TICK * (xpos - starting_press_x), -TICK * (ypos - starting_press_y), 0);
        break;
    case GeoScaling:
        models[cur_idx].scale -=
            Vector3(TICK * (xpos - starting_press_x), TICK * (ypos - starting_press_y), 0);
        break;
    case GeoRotation:
        models[cur_idx].rotation -=
            Vector3(TICK * (ypos - starting_press_y), TICK * (xpos - starting_press_x), 0);
        break;
    case ViewEye:
        main_camera.position +=
            Vector3( -TICK * (xpos - starting_press_x), -TICK * (ypos - starting_press_y), 0);
        setViewingMatrix();
        break;
    case ViewCenter:
        main_camera.center +=
            Vector3( -TICK * (xpos - starting_press_x), TICK * (ypos - starting_press_y), 0);
        setViewingMatrix();
        break;
    case ViewUp:
        main_camera.up_vector +=
            Vector3( -TICK * (xpos - starting_press_x), -TICK * (ypos - starting_press_y), 0);
        setViewingMatrix();
        break;
    }
}
starting_press_x = int(xpos);
starting_press_y = int(ypos);
```

調整正負號與Demo需求相符
TICK為自行設定之全域常數，目前設為0.01

- 其他：
  參考討論區提出的問題：修改WINDOW_WIDTH為600