計算機圖學 HW1 Report 111062678 卓榮祥

完成清單：

- ✓ Correctly render model in Orthogonal projection
- ✓ Correctly render model in NDC perspective
- ✓ Translation, Rotation, Scaling models
- ✓ Camera Control
- ✓ Switch models (5 models in Line 566 of main.cpp)
- ✓ render quad
- ✓ Switch between solid and wireframe mode
- ✓ Print information
- ✓ Window resize
- ✓ Report

Viewing

Viewing matrix 的實作：

仿造 Transformation 講義中 p72 頁的”Fast Viewing Matrix Derivation”的 matrix 實作。

```
Vector3 f = main_camera.center - main_camera.position ;
Vector3 f_p = f;
f_p.normalize();
Vector3 u_p = main_camera.up_vector;
u_p.normalize();

Vector3 s = f_p.cross(u_p);
Vector3 s_p = s;
s_p.normalize();

Vector3 u_pp = s_p.cross(f_p);
```

```
Matrix4 R = Matrix4(
    s[0], s[1], s[2], 0,
    u_pp[0], u_pp[1], u_pp[2], 0,
    (-1) * f_p[0], (-1) * f_p[1], (-1) * f_p[2], 0,
    0, 0, 0, 1
);

Matrix4 T = Matrix4(
    1, 0, 0, (-1) * main_camera.position[0],
    0, 1, 0, (-1) * main_camera.position[1],
    0, 0, 1, (-1) * main_camera.position[2],
    0, 0, 0, 1
);


view_matrix = R * T;
```

➢ Projection

Perspective projection 的實作：

仿造 gluPerspective(fovy, aspect, near, far) 矩陣的實作。

```
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 );
```

Orthographic Projection 的實作：

仿造 glOrtho(left, right, bottom, top, near, far) 矩陣的實作。

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

- **Window resize**

Perspective projection 的實作：

除了將 aspect 重新計算外，還須更新 projection matrix 來設定

Projection plane 的比例。

```
proj.aspect = (float)width / (float)height;
if (cur_proj_mode == Perspective) {
    setPerspective();
}
else if (cur_proj_mode == Orthogonal) {
    setOrthogonal();
}
```

- **multiply all the matrix**

以下式子等同於先做 scaling、rotation，再做 translation。

若先做 translation，則 scaling、rotation 會在 translation 後的位置以原

點進行伸縮、旋轉，但這並非正確的行為。

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

- **callback function**

KeyCallback、scroll_callback 實作上來說較為直觀，KeyCallback 中主要是根據按下的 key 類別來判斷需要執行的 case，scroll_callback 的話是利用滾輪的移動產生 yoffset，因此可以透過 yoffset 來改變 Translation 沿著 Z 軸的移動和 ViewCenter 沿著 Z 軸的移動等等。

而在處理滑鼠的拖拉時，需要先判斷滑鼠是否已經按下，若將滑鼠按下，則需要立即記錄滑鼠按下的位置以便計算拖拉後的相對距離來更新 Geometrical Transformation Matrix 的狀態以及 Viewing Matrix 的狀態。以下為實作的細節：

Mouse button call back function：

```
if (button == GLFW_MOUSE_BUTTON_LEFT && action == GLFW_PRESS) {

    mouse_pressed = true;
    //cout << "pressed" << endl;

}
else {
    mouse_pressed = false;
    starting_press_x = -1;
    starting_press_y = -1;
}
```

Cursor pos callback function：

```
if (mouse_pressed == true ) {
    if (starting_press_x == -1) {
        starting_press_x = xpos;
        starting_press_y = ypos;
    }
    Vector3 move = Vector3((xpos - starting_press_x), (starting_press_y - ypos), 0);
    move *= 0.05;
    switch (cur_trans_mode) {
        case GeoTranslation:
            models[cur_idx].position = models[cur_idx].position + move;
            break;
        case GeoRotation:
            models[cur_idx].rotation = models[cur_idx].rotation + move;
            break;
        case GeoScaling:
            models[cur_idx].scale = models[cur_idx].scale + move;
            break;
        case ViewCenter:
            main_camera.center = main_camera.center + move;
            setViewingMatrix();
            break;
        case ViewEye:
            main_camera.position = main_camera.position + move;
            setViewingMatrix();
            break;
        case ViewUp:
            main_camera.up_vector = main_camera.up_vector + move;
            setViewingMatrix();
            break;



    }

    starting_press_x = xpos;
    starting_press_y = ypos;
```