Computer Graphics HW#1

#1 Model Matrix

Translation

- Translate Model given a vector in Homogeneous Coordinates

```
mat = Matrix4(
    1, 0, 0, vec.x,
    0, 1, 0, vec.y,
    0, 0, 1, vec.z,
    0, 0, 0, 1
);
```

- Scale Model given a vector in Homogeneous Coordinates

```
mat = Matrix4(
    1, 0, 0, vec.x,
    0, 1, 0, vec.y,
    0, 0, 1, vec.z,
    0, 0, 0, 1
);
```

- Rotate Model given a vector
  - Rotate alone axis X

```
GLfloat cosX = cos(val);
GLfloat sinX = sin(val);

mat = Matrix4(
    1, 0, 0, 0,
    0, cosX, -sinX, 0,
    0, sinX, cosX, 0,
    0, 0, 0, 1
);
```

  - Rotate alone axis Y

```
GLfloat cosX = cos(val);
GLfloat sinX = sin(val);

mat = Matrix4(
    cosX, 0, sinX, 0,
    0, 1, 0, 0,
    -sinX, 0, cosX, 0,
    0, 0, 0, 1
);
```

- Rotate alone axis Z

```
GLfloat cosX = cos(val);
GLfloat sinX = sin(val);

mat = Matrix4(
    cosX, -sinX, 0, 0,
    sinX, cosX, 0, 0,
    0, 0, 1, 0,
    0, 0, 0, 1
);
```

#2 Viewing Matrix

- 按照講義上的公式

```
Vector3 P1P2 = main_camera.center - main_camera.position;
Vector3 P1P3 = main_camera.up_vector;
Vector3 Rz = -P1P2.normalize();
Vector3 Rx = P1P2.cross(P1P3).normalize();
Vector3 Ry = Rz.cross(Rx);

Matrix4 R = Matrix4(
    Rx.x, Rx.y, Rx.z, 0,
    Ry.x, Ry.y, Ry.z, 0,
    Rz.x, Rz.y, Rz.z, 0,
    0, 0, 0, 1
);

Matrix4 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;
```

#3 Projection Matrix

- Orthogonal Projection
  - 我用講義第125頁和Normalize結合起來的Matrix

```
GLfloat x_s = proj.right - proj.left;
GLfloat x_p = proj.right + proj.left;
GLfloat y_s = proj.top - proj.bottom;
GLfloat y_p = proj.top + proj.bottom;
GLfloat z_s = proj.farClip - proj.nearClip;
GLfloat z_p = proj.farClip + proj.nearClip;

project_matrix = Matrix4(
    2/x_s, 0, 0, -1*x_p/x_s,
    0, 2/y_s, 0, -1*y_p/y_s,
    0, 0, -2/z_s, -1*z_p/z_s,
    0, 0, 0, 1
);
Projection_Mode = 0;    //for printing information
```

- Perspective Projection
  - 和上面一樣用有Normalize的Matrix(127頁)

```
GLfloat x_s = proj.right - proj.left;
GLfloat x_p = proj.right + proj.left;
GLfloat y_s = proj.top - proj.bottom;
GLfloat y_p = proj.top + proj.bottom;
GLfloat z_s = proj.farClip - proj.nearClip;
GLfloat z_p = proj.farClip + proj.nearClip;

project_matrix = Matrix4(
    2 * proj.nearClip / x_s, 0, x_p / x_s, 0,
    0, 2 * proj.nearClip / y_s, y_p / y_s, 0,
    0, 0, -z_p / z_s, -2 * proj.farClip*proj.nearClip / z_s,
    0, 0, -1, 0
);

Projection_Mode = 1;    //for printing information
```

#4 Control

Modes

- X：下一個model
- Z：上一個model
- W：wire mode
- O：設成orthogonal視角
- P：設成perspective視角
- E：調整eye的位置
- C：調整center的位置
- T：translate模式

- S：scaling模式
- R：rotation模式
- I：印出model和camera還有Projection Mode的資訊

Usage

- 按字母選模式之後，滑鼠可以控制x、y軸，滾輪可以控制z軸