

CG_HW2

功能介紹與實作:

z,x,c,w,h: 上次作業不再贅述。

i: 顯示目前話的 model 跟 mode，mode 方面用一個變數紀錄，並用 if else if 判斷即可

上次作業的 normalize 這次是透過 matrix 實作，translate 然後 scale，公式跟上次一樣。

```
nx = -(maxVal[0] + minVal[0]) / 2, ny = -(maxVal[1] + minVal[1]) / 2, nz = -(maxVal[2] + minVal[2]) / 2;
```

```
N = Matrix4(
    1, 0, 0, nx,
    0, 1, 0, ny,
    0, 0, 1, nz,
    0, 0, 0, 1
);

scale = max((maxVal[0] - minVal[0]), (maxVal[1] - minVal[1]));
scale = max(scale, (maxVal[2] - minVal[2]));

S = Matrix4(
    2/scale*sx, 0, 0, 0,
    0, 2/scale/sy, 0, 0,
    0, 0, 2/scale, 0,
    0, 0, 0, 1
);
```

s:object scaling mode，透過滑鼠向外拉放大，向內縮小，並將放大倍率乘進去 S(Matrix4)裡，這裡要注意的部分就是因為 y 是越往上面越小，所以是要乘上在 y 的倍率的倒數。

```
S = Matrix4(
    2/scale*sx, 0, 0, 0,
    0, 2/scale/sy, 0, 0,
    0, 0, 2/scale, 0,
    0, 0, 0, 1
);
```

t:object translate mode，預設模式，透過滑鼠左右移動去移動物體，透過 T

```

T = Matrix4(
    1, 0, 0, tx,
    0, 1, 0, ty+0.05,
    0, 0, 1, 0,
    0, 0, 0, 1

```

來達成。);

在 onMouseMotion 裡去判斷跟上一次位置的差距來進行移動，再將值給 tx,ty，然後 tx,ty 會在上面的 T 矩陣用到。

```

tx += (x - tempx) / 500;
ty += -(y - tempy) / 500;
tempx = x;
tempy = y;

```

r:object rotation mode,在 onMouseMotion 裡一樣去偵測滑鼠移動方向。

```

else if (mode == 3) {
    rotx += (x - tempx)/100;
    roty -= (y - tempy)/100;

    tempx = x;
    tempy = y;
}

```

在 onMouse 裡的 GLUT_WHEEL_DOWN 和 GLUT_WHEEL_UP 處理滾輪事件，並套用在 Z 軸的旋轉上。

```

if(mode==3)rotx += 0.04;

```

```

if(mode==3)roty -= 0.04;

```

最後在 onDisplay 中套用課本中講的公式，分別算出 X,Y,Z 軸的旋轉，之後乘起來並乘進 MVP 裡。

```

float sinx = sin(rotx);
float cosx = cos(rotx);
Matrix4 r1 = Matrix4(
    1, 0, 0, 0,
    0, cosx, -sinx, 0,
    0, sinx, cosx, 0,
    0, 0, 0, 1);
float siny = sin(roty);
float cosy = cos(roty);

Matrix4 r2 = Matrix4(
    cosy, 0, siny, 0,
    0, 1, 0, 0,
    -siny, 0, cosy, 0,
    0, 0, 0, 1);
float sinz = sin(rotx);
float cosz = cos(rotx);

```

e:object eye translate mode，這是 mvp 裡的 v，也就是眼睛看的位置，跟眼睛所在位置，e 是期望讓眼睛移動，e 移動的話，center 不會變，但法向量也就是到 up 方向會變，於是乎我們可以透過 HW2_Supplement 給的公式來算出新的法向量，然後一樣在 onMouseMotion 跟 onMouse 裡計算滾輪事件以及 x,y 軸移動方向來控制，最後套用講義 71 頁的公式算出 T、R 相乘得到 V。

```
else if (mode == 4) {
    ex = -(x - tempx) / 800;
    ey = (y - tempy) / 800;
    tempx = x;
    tempy = y;
}

else if (mode == 4) ez += 0.04;

else if (mode == 4) {
    ez -= 0.04;
    printf("%f\n",ez);
}

Tv = Matrix4(
    1, 0, 0, -eye.x,
    0, 1, 0, -eye.y,
    0, 0, 1, -eye.z,
    0, 0, 0, 1
);
Rv = 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
);

eye += Vector3( ex, ey, ez);
center += Vector3(cx, cy, cz);
Forward = center - eye;
Right = Forward.cross(up);
upn = Right.cross(Forward);
upn = upn;
up = eye + upn;
Vector3 Rx, Ry, Rz;
Rz = (center - eye) / (center - eye).length();
Rx = (center - eye).cross(up - eye) / ((center - eye).cross(up - eye)).length();
Ry = Rx.cross(Rz);
```

l:object center mode，這個模式是期望讓眼睛看的位置改變，改變 center 並不會改變 eye、up 的位置，所以只要去計算新的 forward 即可，同上套用 71 頁公式算出新的 V，然後如慣例 onMouseMotion 偵測 x,y 也就是 center 的移動

量。

```
else if (mode == 5) {
    cx = (x - tempx) / 800;
    cy = (y - tempy) / 800;
    tempx = x;
    tempy = y;
}
```

o:改變 projection 的方式，平行投影或垂直投影，套用講義 125 頁的公式即可。

```
if (fin) {
    P = Matrix4(
        2 / (right - left), 0, 0, -(right + left) / (right - left),
        0, 2 / (top - bottom), 0, -(top + bottom) / (top - bottom),
        0, 0, -2 / (fa - nea), -(fa + nea) / (fa - nea),
        0, 0, 0, 1
    );
}
else {
    P = Matrix4(
        2*nea / (right - left), 0, (right+left)/(right-left), 0,
        0, 2 *nea/ (top - bottom), (top + bottom) / (top - bottom), 0,
        0, 0, -(fa + nea) / (fa - nea), -2*fa*nea / (fa - nea),
        0, 0, -1, 0
    );
}
```

p:一樣在 onMouseMotion 處理 x,y 的位移量，x 的位移量反應在 right，left 上，y 反應在 top，bottom 上，這個跟單純 scale 的差別在於連地板也會變小，因為我們改變的是視窗大小，而不是單改模組的大小。

```
else if (mode == 7) {

    px = + ((x - tempx) / 100);
    py = - ((y - tempy) / 100);
    right += px;
    left -= px;
    top += -py;
    bottom -= -py;

    tempx = x;
    tempy = y;
}
```

```
else if (mode == 7) {  
    pz += 0.04;  
    nea += pz;  
    fa += -pz;  
    printf("%f %f\n", nea, fa);  
}  
  
else if (mode == 7) {  
    pz -= 0.04;  
    nea += -pz;  
    fa += pz;  
    printf("%f %f\n", nea, fa);  
}
```

地板跟讀 model 一樣，只不過 mvp 中的 m 是不一樣的，其他都跟 model 一樣。