Tutorial 2

Transformations

CS4052 Computer Graphics

Alexandru Sulea
D Stream
#12315152
18 October 2016

1 Overview

I used the math file provided for the identity, translation, rotation and scaling matrices.As well as combined transformations and individual transformations. The Project2.zip file attached with this tutorial comes up as being infected but there should be no viruses as far as i know. The project works fine as long as 8, 9, 6 is pressed at the start of the window. due to some maths calcs, if those numbers are not pressed the location of the triangles is never set at the origin.

2 X-Y-Z Translations

For this part of the assignmeant I used the W and S keys for y-directions. A and D keys for x-direction. The Q and E keys for z-direction. To show the z-direction easier I added a randomized function for color change for each translation to show the z-translations more clearly. NOTE: May afffect people with epilepsy.

```
mat4 translationmatrix = identity_mat4();
translationmatrix.m[3] = Xt;
translationmatrix.m[7] = Yt;
translationmatrix.m[11] = Zt;
```

3 X-Y-Z Rotations

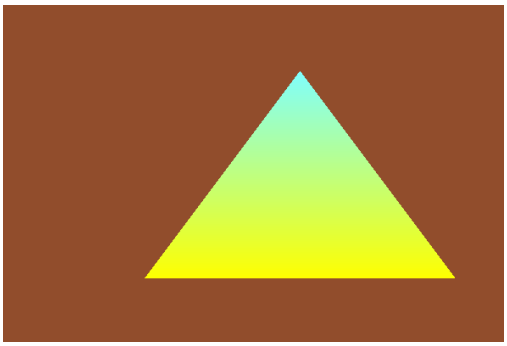
For this part of the assignmeant I used the 4 and 6 keys for y-rotations. 8 and 2 keys for x-rotation. The 7 and 9 keys for z-rotation.

```
mat4 rotationmatrix = identity_mat4();
/*z-axis*/
rotationmatrix.m[1] = -Z2r;
rotationmatrix.m[4] = Z2r;
/*y-axis*/
rotationmatrix.m[0] = X1r*Z1r;
rotationmatrix.m[2] = X2r;
rotationmatrix.m[8] = -X2r;
/*x-axis*/
rotationmatrix.m[5] = Y1r*Z1r;
rotationmatrix.m[6] = -Y2r;
rotationmatrix.m[9] = Y2r;
rotationmatrix.m[10] = Y1r*X1r;
```

4 Uniform and Non-Uniform Scaling

For this part of the assignmeant I used the P and L keys for uniform scaling. The O and K keys for non-uniform scaling and the R key for resetting the dimensions back to their original place.

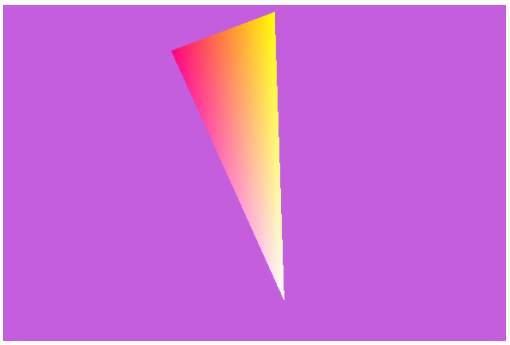
```
mat4 scalingmatrix = identity_mat4();
/*non- uniform/ uniform scaling*/
scalingmatrix.m[0] = Su;
scalingmatrix.m[5] =Su*Snu;
scalingmatrix.m[10] = Su;
```



5 Combined Transformations

```
void main()                          \n\
{                          \n\
                /*T*R*S*T*/ \n\
                                     \n\
    gl_Position =tMatrix * rMatrix* sMatrix *iMatrix * vec4(vPosition.x/4, vPosition.y/4,
        vPosition.z/4, 1.0); \n\
   color = vColor;                    \n\
}";

   if (key == 'o')
   {color();
      Snu = Snu + temp *2;
      //Su= Su = Su + temp;
      printf(" scaling:%c : %f",key, Snu);}
   if (key == 'k')
   {     if (Snu > 0.01) {
         Snu = Snu - temp;
         printf(" scaling:%c : %f", key, Snu);}
      else {
         Snu = Snu + temp;
         printf(" scaling:%c : %f", key, Snu);}}
```



6 Individual Transformations

```
glUniformMatrix4fv(matrixtranslation, 1, GL_TRUE, &translationmatrix.m[0]);
glUniformMatrix4fv(matrixrotation, 1, GL_TRUE, &rotationmatrix.m[0]);
glUniformMatrix4fv(matrixscaling, 1, GL_TRUE, &scalingmatrix.m[0]);
glUniformMatrix4fv(matrixidentity, 1, GL_TRUE, &identitymatrix.m[0]);

glDrawArrays(GL_TRIANGLES, 0, 3); /*pattern 0,3/3,3//6,3*/ glUniformMatrix4fv(matrixrotation, 1,
    GL_TRUE, &identitymatrix.m[0]);
glUniformMatrix4fv(matrixidentity, 1, GL_TRUE, &identitymatrix.m[0]);
glUniformMatrix4fv(matrixidentity, 1, GL_TRUE, &identitymatrix.m[0]);
glUniformMatrix4fv(matrixidentity, 1, GL_TRUE, &identitymatrix.m[0]);
```