



---

# Tutorial 3

---

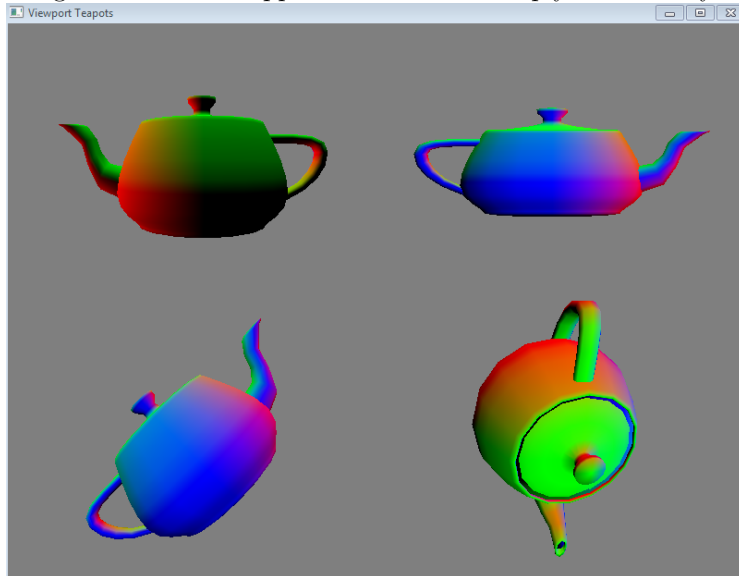
VIEWS

CS4052 COMPUTER GRAPHICS

ALEXANDRU SULEA  
D STREAM  
#12315152  
25 OCTOBER 2016

# 1 Overview

I used the mathfuncs and the sample code provided for the lab to achieve the initial views. The orthographic projection function was written into mathfuncs.cpp and the lookat function was also used from the given mathfuncs.cpp. Rotations was simply achieved by incrementally increasing the angle of rotation.



## 2 Rotation

For this part of the assignment I used the rotation functions provided by mathfunc.cpp

```
if (i > 359)
{
    i = 0;
}
else
{
    i+=0.03;
}
mat4 model1 = rotate_y_deg(rotate_z_deg(rotate_x_deg(identity_mat4(), 3*i),i),5*i);
```

## 3 Static

For this part of the assignment I didn't really have to do anything as the default position of the teapots was non-spinning.

## 4 Two different Perspectives

For this part of the assignment I changed the variables in one of the perspective functions to achieve a different overall perspective

```
mat4 persp_proj = perspective(45.0, (float)width / (float)height, 0.1, 100.0);
mat4 persp_proj1 = perspective(90.0, (float)width / (float)height, 0.1, 100.0);
```

## 5 Orthographic Projection

For this part of the assignment I wrote an orthographic function in mathfunc.cpp based on the matrix given in the slides. The function simply implements that matrix and multiplies it depending on the inputs given.

```
mat4 Ort = orthograp(-20,20,-20,20,20,-20);
mat4 orthograp(float left, float right, float near, float far, float top, float bottom ) {
    mat4 m = zero_mat4(); // make sure bottom-right corner is zero
    m.m[0] = 2/(right-left);
    m.m[5] = 2/(top-bottom);
    m.m[10] = -2/(far-near);
    m.m[3] = -(right + left) / (right-left);
    m.m[7] = -(top + bottom) / (top - bottom);
    m.m[11] = -(far + near) / (far - near);
    m.m[15] = 1.0f;
    return m;
}
```

## 6 Look At Function

Look at Function is implemented using the camera position the teapot position and the direction in which

```
vec3 cam_pos = { 0.0, 0.0, -40.0 };
vec3 teapot_pos = { 0.0,0.0,0.0 };
vec3 up = {0.0,1.0,0.0};
mat4 La = look_at(cam_pos, teapot_pos, up);
```