**Worksheet 3 – Part 3: Theory**

1. **List the transformation matrices that you used in Parts 1 and 2:**

For the first part I used modelViewMatrix (computed using LookAt function):

mat4 LookAt(vec3 eye, vec3 at, vec3 up){

vec3 zaxis = normalize(at - eye);

vec3 xaxis = normalize(cross(zaxis, up));

vec3 yaxis = cross(xaxis, zaxis);

negate(zaxis);

mat4 viewMatrix = {

vec4(xaxis.x, xaxis.y, xaxis.z, -dot(xaxis, eye)),

vec4(yaxis.x, yaxis.y, yaxis.z, -dot(yaxis, eye)),

vec4(zaxis.x, zaxis.y, zaxis.z, -dot(zaxis, eye)),

vec4(0, 0, 0, 1)};

return viewMatrix; }

For the second part I used projectMatrix (computed using perspective function) and a translation matrix:

mat4 perspective(fovy, aspect, near, far)

{

mat4 perspective = {

vec4((1/aspect) cot (fovy/2), 0, 0, 0),

vec4(0, cot fovy/2, 0, 0),

vec4(0, 0, (near + far) / (near -far), 2\*near\*far/(near - far) ),

vec4(0, 0, -1, 0)

}

return perspective; }

mat4 translate(X, Y, Z) = {

vec4(1, 0, 0, X),

vec4(0, 1, 0, Y),

vec4(0, 0, 1, Z),

vec4(0, 0, 0, 1) }

, where X, Y, Z are the values that you want to add to your position.

x' = x + X ,

y' = y + Y ,

z' = z + Z

1. **For each cube, write down a formula showing how the matrices were concatenated to become the current transformation matrix (CTM) that was used to transform the vertices in the vertex shader.**

Part 1.1 & 1.2: CTM - viewMatrix \* vertex

Part 2.1: CTM - perspective \* translate \* vertex

Part 2.2: CTM - perspective \* viewMatrix \* translate \* vertex