**CSCI 4250/5250 Homework 5 (Due beginning of class, Tuesday Oct 31st)**

1) Given the 3D cube example in programs: ortho.js and ortho.html (available on the course web page), if the view position and the orthographic viewing volume is changed into each of the following situations, how will the final 2D image change from its original image? Justify your answer.

**a. mvMatrix=lookAt(vec3(-4, 0, 0), at, up); // pMatrix does not change**

Ans :We can only view right side of the cube because we first moved from eye’s initial position vec3(4, 4, -4) ) to eye’s new position vec3(-4, 0, 0).Change in y axis is that from 4 to 0 meaning we cannot see up side so we can only see back and right side . Now change in z axis from -4 to 0 will hide the back side and we can only view right side .

**b. mvMatrix=lookAt(vec3(3, 3, 3), at, up); // pMatrix does not change**

Ans:We can see the top part , front part and right part. This happens because there is not much change in eye’s x and y coordinate i.e from (4,4) to (3,3) but z’s coordinate changed from negative to positive ie -4 to 3 . So that means we moved to front side of the cube and can view top , front and right side of the cube.

**c. mvMatrix=lookAt(vec3(3, 3, 3, at, up); pMatrix=ortho(-3, 3, -3, 3, -1, 1);**

Ans : we cannot see anything as we are further then far plane i.e z=1 is far plane and our eye is located at (3,3,3) so we are further then far plane .

**d. pMatrix= ortho(-6, 6, -3, 3, 2, 10); // mvMatrix does not change**

**ans:** Here the view volume has been increased i.e left and right changed from (-2,2) to (-6,6) so we can see that cube has shrink to adjust in new volume matrix .

**e. pMatrix=ortho(0, 4, 0, 3, 2, 10); // mvMatrix does not change**

**Ans:** I can only see some part of top and back sides of the cube. Here left and right of view volume are being moved from -2,2 to 0,4 and since I’m still located at eye(4,4,4) so I cannot see any of the left and right side. And bottom and top sides of view volume are being moved from(-4,4) to (0,3) and eyes are at (4,4,4) so I can see a little of top . and there is not really much change in near and far plane so I can see a little of back. Basically here view volume is moved a lot to my left and up direction of my eye’s coordinates.

**2) Given:**

**mvMatrix=lookAt(vec3(4, 4, - 4), at, up);**

**pMatrix=ortho(-2, 2, -4, 4, -10, 10);**

**show:**

**• the mvMatrix**

**• the pMatrix**

**• the coordinates of a point F(1, 1, -1) when converted into the final clip coordinates. (show intermediate steps in deriving the results)**

**Ans:**

**• the mvMatrix**

n=eye-look = (4,4,-4) – (0,0,0)= (4,4-4)

u = up x n = (0, 1, 0) \* (4, 4,-4) = ( -4, 0, -4 )

v= n x u = (-16, 32,16)

Normalize n, u, and v

n= (0.577, 0.577, -0.577)

u= (- 0.707, 0, -0.707)

v= (- 0.408, 0.816, 0.408)

dx = -eye.u = (-4,-4,4) . (- 0.707, 0, -0.707) = 0

dy=-eye.v = (-4,-4,4) . (- 0.408, 0.816, 0.408) =0

dz = -eye .n = (-4,-4,4) . (0.577, 0.577, -0.577) = -6.924

v =| ux uy uz dx |

| vx vy vz dy |

| nx ny nz dz |

| 0 0 0 1 |

v =| - 0.707 0 - 0.707 0 |

| - 0.408 0.816 0.408 0 |

| 0.577 0.577 - 0.577 -6.924 |

| 0 0 0 1 |

**• the pMatrix**

p =| 2/ (right-left) 0 0 -(left+right/right-left) |

| 0 2/ (top-bottom) 0 -(top+bottom/top-bottom) |

| 0 0 -2/(far-near) -(near+far/far-near) |

| 0 0 0 1 |

pMatrix=ortho(-2, 2, -4, 4, -10, 10) // ortho(left, right, bottom, top, near, far)

p =| 0.5 0 0 0 |

| 0 0.25 0 0 |

| 0 0 -0.1 0 |

| 0 0 0 1 |

**• the coordinates of a point F(1, 1, -1) when converted into the final clip coordinates. (show intermediate steps in deriving the results)**

C’=P\*V\*C

p\*v=

| 0.5 0 0 0 | \* | - 0.707 0 - 0.707 0 |

| 0 0.25 0 0 | | - 0.408 0.816 0.408 0 |

| 0 0 -0.1 0 | | 0.577 0.577 - 0.577 -6.924 |

| 0 0 0 1 | | 0 0 0 1 |

= |-0.3535 0 -0.3535 0 |

| -0.102 0.204 0.102 0 |

| -0.0577 -0.0577 0.0577 0.6924 |

| 0 0 0 1 |

C’=P\*V\*C

= |-0.3535 0 -0.3535 0 |\* | 1 |

| -0.102 0.204 0.102 0 | | 1 |

| -0.0577 -0.0577 0.0577 0.6924 | |-1|

| 0 0 0 1 | | 1 |

= | 0 |

| 0 |

| 0.5193 |

| 0 |

**3) Changing the orthographic viewing volume in problem 2) to a frustum with left=-2, right=2, bottom=-4, top=4 for the near plane, and the near plane at distance 4 and far plane at distance 10 from the eye/camera. How would you call the perspective function to set up the corresponding pMatrix in the .js program?**

**Ans :**

*var eye = vec3(4, 4, -4);*

*var at = vec3(0, 0, 0);*

*var up = vec3(0, 1, 0);*

*var near = 4;*

*var far = 10;*

*var left= -2;*

*var right =2;*

*var top\_cube = 4;*

*var bottom = -4;*

*var render = function() {*

*var aspect = (right-left)/(top\_cube-bottom);*

*var viewAngle=2\* Math.atan(1/2\*(top\_cube-bottom)/near) ;*

*gl.clear( gl.COLOR\_BUFFER\_BIT | gl.DEPTH\_BUFFER\_BIT);*

*mvMatrix = lookAt(eye, at, up);*

*pMatrix =perspective(viewAngle,aspect,near,far)*

*var t=mult(pMatrix, mvMatrix);*

*console.log("projection\*modelview");*

*console.log(flatten(t));*

*gl.uniformMatrix4fv( modelView, false, flatten(mvMatrix) );*

*gl.uniformMatrix4fv( projection, false, flatten(pMatrix) );*

*gl.drawArrays( gl.TRIANGLES, 0, numVertices );*

*}*

**4) With the perspective viewing volume defined in problem 3), what will be the x and y coordinates of the two points F(1, 1, -1) and B(1, 1, 1) when projected onto the near plane?**

F’(f’x ,f’y) and B’(b’x,b’y) are new coordinates for F(fx,fy,fz) and B (bx,by,bz)

f’x = N\*fx/(-fz)= 4 \*1/1=4

f’y = N\*fy/(-fz)=4\*1/1 =4

b’x = N\*bx/(-bz)=4\*1/(-1)= -4

b’y = N\*by/(-bz)=4\*1/(-1)= - 4

F’(f’x,f’y) is (4,4)

B’(b’x,b’y) is (-4,-4)