Aaron Itzkovitz

Dr. Li

CSCI 4250/5250 Homework 5

10/31/17

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

The eye coordinates changed so that we were level with the cube on the y-axis and looking directly at the other side (left) of the object.

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

The eye moves to the top right of the cube so it can see the right blue, the front red, and the top cyan.

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

The cube doesn’t show up because it’s not between the near and far plane.

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

Now, because the range between the near and far planes is increased, we can see the object. Because the viewing area is now much bigger, scaling it down to a CVV also squishes the cube down.

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

Now, because we changed the first 4 arguments of ortho, left, right, bottom, and top, we are clipping more and viewing a smaller area. Because of this, we can only see a small portion of the square.

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)

mvMatrix =

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 4 |
| 0 | 0 | 1 | 4 |
| 0 | 0 | 0 | 1 |

pMatrix =

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 0 |  |
| 0 |  | 0 |  |
| 0 | 0 |  |  |
| 0 | 0 | 0 | 1 |

pMatrix =

|  |  |  |  |
| --- | --- | --- | --- |
| .5 | 0 | 0 | 0 |
| 0 | .25 | 0 | 0 |
| 0 | 0 | -.1 | 0 |
| 0 | 0 | 0 | 1 |

pMatrix • vMatrix =

|  |  |  |  |
| --- | --- | --- | --- |
| .5 | 0 | 0 | 2 |
| 0 | .25 | 0 | 1 |
| 0 | 0 | -.1 | -.4 |
| 0 | 0 | 0 | 1 |

pMatrix • vMatrix • F = ( 2.5, 1.25, -0.3, 1)

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?

p = perspective(1.57, 0.5, 4, 10 )

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?

P =

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 7/3 | -40/3 |
| 0 | 0 | -1 | 0 |

F’ = ( 2, 1 )

B’ = ( 2, 1 )