Orthographic Projection
CS 418: Interactive Computer Graphics
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Eric Shaffer
Projections

In computer graphics, eventually we need to move from 3D space to 2D space.

More accurately:
from 4D homogeneous coordinates to 3D homogeneous coordinates

A **projection** is a transformation that maps from a high-dimensional space into a lower-dimensional space.

We will look at some common projections...and then we will discuss projection within WebGL.
Planar Geometric Projections

Standard projections project onto a plane

*Projectors* are lines that either
- converge at a center of projection
- are parallel

Such projections preserve lines
- but not necessarily angles

Non-planar projections are often used in map construction
Perspective Projection
Orthographic Projection
Oblique Projections

Oblique parallel projection     Oblique perspective projection

Linear projections can be categorized
• By whether the projectors are parallel
• By whether the projectors are orthogonal to the view plane

We will not use these.....
**Definition to Know: Foreshortening**

**Foreshortening** is the visual effect or optical illusion that causes an object or distance to appear shorter than it actually is because it is angled toward the viewer.

i.e. projections squash receding surfaces

Andrea Mantegna
The Lamentation over the Dead Christ

Can foreshortening happen in orthographic projection?
Clip Space View Volume

Before looking at projections in more detail...
...let’s experimentally determine what the default view and view volume are.

So, if we don’t do any transformations...where are we and what direction are we looking?

Take a look at

http://jsfiddle.net/2x03hdc8/

It is a simple WebGL program that lets you draw rectangles

No transformations

By altering the rectangle coordinates, you can figure out the view....
Clip Space View Volume

- Something new – the code enables hidden surface removal

```javascript
38 // Tell WebGL to test the depth when drawing, so if a square is behind
39 // another square it won't be drawn
40 gl.enable(gl.DEPTH_TEST);
```

Side note - the overall code uses different conventions than used in our course (e.g. no glmatrix library)

So, probably don't base your assignments on this code
Clip Space View

• Red rectangle has z=0
• Green has z=0.5

What direction are we looking?
Clip Space View

We add a blue rectangle

It’s at z=-1

Where is the near clipping plane?
Clip Space View and View Volume

View is looking down Z+
Eyepoint is effectively at (0,0,-1)
This is a left-handed coordinate system

This is a little-known fact....
People often learn to use WebGL without ever learning this.
We’ll see why in a few slides
WebGL Style View

WebGL/OpenGL convention is to assume a right-handed world coordinate system.

The **ortho** matrix is used to flip this coordinate system by scaling Z by -1. It then matches WebGL clip space.
More WebGL Secrets

WebGL only performs an orthogonal projection

• Everything is projected to the z=0 plane in the normalized view volume
• But you can distort your geometry to achieve a perspective projection

The projection occurs when the geometry is in clip space (NDC)

• Even then, depth information is kept around to do hidden surface removal
• Depth information means transformed z coordinates
Orthographic Projection

- Foreshortens
- No change in size by depth
- Classic Orthographic Projection matrix simply zeros the z-coordinate

\[
\begin{bmatrix}
W2V & 1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
View & Model
\end{bmatrix}
\]

- \text{mat4.ortho(out,left,right,bottom,top,near,far)}

\[
\begin{bmatrix}
\frac{2}{r - l} & 0 & 0 & -\frac{r + l}{r - l} \\
0 & \frac{2}{t - b} & 0 & -\frac{r + l}{t - b} \\
0 & 0 & -\frac{2}{f - n} & -\frac{f + n}{f - n} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Does the ortho matrix perform a projection?
GLMatrix ortho matrix

\[
\text{ortho}(\text{left}, \text{right}, \text{bottom}, \text{top}, \text{near}, \text{far})
\]

- \text{near} and \text{far} are distances down the \(-z\) axis from origin
- \text{l}, \text{r}, \text{b}, \text{t} are coordinates of the bounding planes
- what does the matrix do?
GLMatrix ortho matrix

\[
\text{ortho}(\text{left}, \text{right}, \text{bottom}, \text{top}, \text{near}, \text{far})
\]

Imagine the eye is at (0,0,0)
We look down the \(-z\) axis

The view volume is:
\[
[\text{left}, \text{right}] = [-1, 1] \\
[\text{bottom}, \text{top}] = [-1, 1]
\]
near = 0
far = 2

What does the matrix look like?
What does it do?
GLMatrix ortho matrix

\[
\text{ortho}(\text{left}, \text{right}, \text{bottom}, \text{top}, \text{near}, \text{far})
\]

Imagine the eye is at (0,0,0)
We look down the –z axis

The view volume is:
[\text{left, right}] = [-1, 1]
[\text{bottom, top}] = [-1, 1]
near = 0
far = 2

Where does (0,0,0) get moved to?
What else happens?

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & -1 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]