Introduction to Computer Graphics with WebGL

Ed Angel

Square Program
Part 5

1

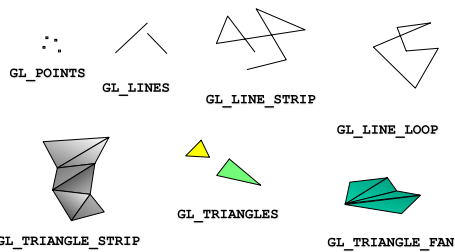**WebGLPrimitives**

GL_POINTS

GL_LINES

GL_LINE_STRIP

GL_LINE_LOOP

GL_TRIANGLES

GL_TRIANGLE_STRIP

GL_TRIANGLE_FAN

2

Why triangles?

• All triangles are
  - simple: edges can't cross
  - flat: a triangle defines a unique plane
  - convex: any line segment connecting two points in the triangle is entirely in the triangle
  - easy to render
  - degenerate triangles in which all three vertices lie in a line are not a problem for rasterizer

3

Coordinate Systems

- The units in **points** are determined by the application and are called *object, world, model* or *problem coordinates*

- Viewing specifications usually are also in object coordinates

- Eventually pixels will be produced in *window coordinates*

- WebGL also uses some internal representations that usually are not visible to the application but are important in the shaders

- Most important is *clip coordinates*

Coordinate Systems and Shaders

- Vertex shader must output in clip coordinates

- Input to fragment shader from rasterizer is in window coordinates

- Application can provide vertex data in any coordinate system but shader must eventually produce gl_Position in clip coordinates

- Simple example uses clip coordinates
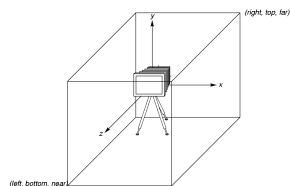
WebGL Camera

- WebGL places a camera at the origin in object space pointing in the negative z direction
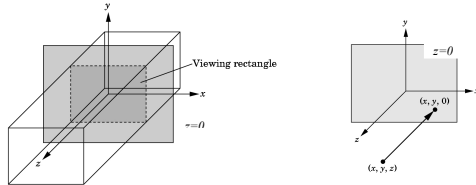- The default viewing volume is a box centered at the origin with sides of length 2

Orthographic Viewing

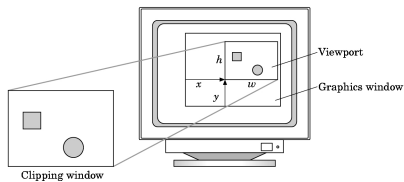In the default orthographic view, points are projected forward along the z axis onto the plane $z=0$

Viewports

• Do not have use the entire window for the image:
`gl.viewport(x,y,w,h)`

• Values in pixels (window coordinates)

Transformations and Viewing

• In WebGL, we usually carry out projection using a projection matrix (transformation) before rasterization

• Transformation functions are also used for changes in coordinate systems

• Pre 3.1 OpenGL had a set of transformation functions which have been deprecated

• Three choices in WebGL
  - Application code
  - GLSL functions
  - MV.js