Announcements

• HW2 and A2 due tonight.

• Artifacts due tomorrow; showcase later this week.
Viewing Transformations: Overview

A standard sequence of transforms to go from object (model) space to screen (image) space
Wireframe Rendering: Algorithm

1. Form matrices $M_{vp}, M_{proj}, M_{cam}, M_{model}$

2. Let $M = M_{vp} \times M_{proj} \times M_{cam} \times M_{model}$

3. For each line segment $a_i, b_i$:
   
   1. $p = Ma_i$
   2. $q = Mb_i$
   3. $\text{draw\_line}(p, q)$

Aside: Why wireframe? Depth
Overview

• https://www.cs.cornell.edu/courses/cs4620/2019fa/demos/view_explore/view_explore_ortho.html
Model Matrix

**Input**: Scene in model coordinates

**Parameters**: Pose, scale, etc of model in scene

**Output**: Scene in world coordinates
Camera Matrix

**Input:** Scene in world coordinates

**Parameters:** Camera frame \((\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{e})\)

**Output:** Scene in camera coordinates
Projection Matrix - Orthographic

**Input:** Scene in (canonically-posed) camera coordinates

**Parameters:** Orthographic viewport dimensions

**Output:** Normalized device coordinates
Viewport Matrix

**Input:** Scene in the canonical view volume

**Parameters:** $W, H$ (image dimensions in pixels)

**Output:** $(x, y)$ in pixel coordinates; $z$ unchanged
Let's build it
Viewport Matrix

Input: Scene in the canonical view volume
Parameters: $W, H$ (image dimensions in pixels)
Output: $(x, y)$ in pixel coordinates; $z$ unchanged
Projection Matrix - Orthographic

**Input:** Scene in (canonically-posed) camera coordinates

**Parameters:** Orthographic viewport dimensions

**Output:** Normalized device coordinates
Camera Matrix

Input: Scene in world coordinates
Parameters: Camera frame \((u, v, w, e)\)
Output: Scene in camera coordinates
Model Matrix

**Input:** Scene in model coordinates

**Parameters:** Pose, scale, etc of model in scene

**Output:** Scene in world coordinates
What about perspective cameras?

Perspective Projection

Exercise:
Find \( y_s \), the y coordinate of the point where \((x, y, z)\) projects onto the viewport.