

Lab 04: Open GL Viewing (Chapter 8, 10)

I. 2D - viewing projection with GL utility (GLU)

`gluOrtho2D(xwmin, xwmax, ywmin, ywmax)` (Projection matrix!)

II. View port

`glViewport(xvmin, yvmin, vpWidth, vpHeight)`

see chapter 8 for glut window method details, but multiple viewports/windows allowed w/ gl+glut

III. 3D view

`gluLookAt(xo, yo, zo, xref, yref, zref, vx, vy, vz)`; (model/view matrix)

Note glu functions just manually set matrix we learned in gl.

IV. 3D projection

A. Orthogonal Projection

`glOrtho(xwmin, xwmax, ywmin, ywmax, dnear, dfar)` (Projection matrix!)
 $z_{near} = -d_{near}$
 $z_{far} = -d_{far}$ for projection

Near clip plane is always the view plane! For simplicity in computations.

Note there is no oblique projection w/ OpenGL. would need to manually set matrix

B. Symmetric Perspective Projection

`gluPerspective(theta, aspect, dnear, dfar)`

$z_{near} = -d_{near}$, $z_{far} = -d_{far}$ for projection
Near plane is the view plane

C. General Perspective Projection

`glFrustum(xwmin, xwmax, ywmin, ywmax, dnear, dfar)`

$z_{near} = -d_{near}$, $z_{far} = -d_{far}$ for projection
near plane is the view plane

V. Can specify additional clip planes in scene - see ch. 10-12.

degrades performance {
 `glClipPlane(id, planeParams)`
 es. `GL_CLIP_PLANE0` ↑ array of 4 params A, B, C, D defining plane
 `glEnable(id)` to enable `glDisable(id)` to turn off