9/22/2020

glOrtho(-1, 1, -1, 1,-1, 1) // Cube centered in the origin with edges of length 2.

Plotting a

glOrtho(l, r, b, t, n, f); glOrtho(-3,3,-3,3,-3,3) cube; glOrtho(-3,3,-3,3,-1,1) box

~~glvertex3f(-5, 1, 0) –~~ will not be rendered. It is outside the view volume

glOrtho() defines a volume → the View volume
Type? Rectangular prism
Meaning parallel projection orthographic projection
Your vertices (glVertex())mist be inside the view volume to be seen or they are
Clipped

Plotting functions

Explicit, implicit, parametric, using approximation
Implicit has to be converted to parametric or explicit

Explain the difference between
GL_POINTS, LINE_STRIP, and QUAD_STRIP in this context

Number of arguments 2 (2D) 3 (3D)

For explicit with 1 – 2D

Use points or carefully use line_strip (line loop)
Loop on x from X0 to X1 in increments of dx{
Compute y=f(x);
Use glVertex2*(x, y)}

For explicit with 2 3-d

Use points or carefully use line_strip or quad_strip
Loop on x from X0 to X1 in increments of dx{
Loop on y from y0 to y1 in increments of dy {
Compute z=f(x, y);
Use glVertex3*(x, y, z)}}

For parametric with 1 variables 2D

Use points or carefully use line_strip (line loop)
Loop on v with dv {

Compute P(u) = [x(u), y(u)]^T = $\begin{bmatrix} x(u) \\ y(u) \end{bmatrix}$ often use α, or t as parameters
circle x=r*cos(v), y=r*sin(v)
Use glVertex2*(x, y)}

For parametric with 2 variables 3D

Use points or line_strip, or quad_strip
Loop on v with dv {
Loop on u with du {
Compute P(u, v) = [x(u, v), y(u, v), z(u, v)]^T = $\begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix}$ often use α, β as parameters
}}

The slides include parametric equation of: line, triangle, plan, circle,
sphere (sample program on TRACS). Google knows it already.
Moving to Chapter 4 Transformations

Two types

1) The projection transformation for setting/changing view volume

Can be parallel (using glOrtho()) now or
Perspective (use glFrustum or gluPerspective)

2) The ModelView Transformation (change your model)
   Mainly interested in translation, rotation, scaling
   Get a library of objects and duplicate / reproduce objects using transformations

Study, CG08, CG09

Moving to CG10 and above
Theory
OGL Transformations

Linear space
Affine space
Affine homogenous coordinates – in these coordinates all of OGL transformations can be implemented
Using matrix multiplication 4x4 by 4x4 or 4x4 by 4x1
Coordinate system
3-D coordinates
Volume
Try at home z=(sin(x)/x*sin(y),y)