10/15/2020

Moving the Camera (changing the relative location of object).

No functions in OGL

Camera is at the origin (0, 0, 0, 1) cannot move.
Pointing to the negative-Z direction.

To get the effect of moving the camera we move objects.
Moving objects means applying model-view transformations on objects.

In my setting to see pictures behind me. I would
translate them to be in front of me and rotate.
This can be done by Model view transformations or using gluLookat
Which does model view transformations as in S2.

CG11-23
Moving the camera to be in front of objects
Translating the objects by -d so that all the objects are in the
View of the camera.

CG16-6, 7, 8 Moving the camera
9 OGL code

CG16 10,11 The GL Utility function gluLookat.
Describe the desired location and orientation of the camera.
Glu look at uses the appropriate model view transformations to achieve the effect.
9 parameters,

Eye-x, eye-y eye-z desired camera location (point).
at-x, at-y at-z : the camera is directed towards a specific point.
Upx, upy, upz – define the camera orientation
[1, 0, 0] means protrait
[0, 1, 0] landscape

In Unity(
Graphical User interface
You place the camera via GUI
Unity is calling the equivalent of GLU look at.

Moving to OGL 4
In gl4

Complains that this is depreciated

OGL4 has only to relevant function
glLoadMatrix, ModelView ← A where you define A
glMutiplyMatrix ModelView ← ModelView * A post multiply
You define A

What can you do?

1) Enforce OGL4 to accept deprecated functions
2) Define the Rotation Matrix A and use glMutiplyMatrix

Using 2.x glRotate(45, 0,1,0)

glLookAt(eyex, eyey, eyez, atx, aty, atz, upx, upy, upz)

Back to CG10

Skip 9.
Make sure you understand how to draw objects using explicit and parametric Representation.
Specifically, parametric lines, circles and triangles
Skip 17,
Understand what is a convex Hull. The algorithm for finding it was not covered yet.

Using slide 20.
Slide 21 shows a rudimentary way to define a parametric equation of a triangle.

We produce the parametric representation of a point S(a) located on the PQ vertex.
Using parametric equation of a line (slide 14)

Fix a
Produce the parametric representation of the line from R to S(a) T(a,b)

Check Shoaff and Google for an alternative representation of T(a, b)

Slide 22 (skip) provide methods for finding a normal to a plan

CG11
Understand the definition
Skip the proof of linear independence
Skip 6, 7
General understanding of 10
Skip 15 to 20 The message for now:

Our transformations can be done via matrix multiplication as in set CG12
In homogeneous coordinates

Consider [x, y, z, 0] vector

[x, y, z, w] w <> 0 P1 = [x1, y1, z1, 2] point
P2 = [x2, y2, z2, 1] normal representation of a point
How to normalize P1
P1 normalized is [X1/2, y1/2, Z1/2, 1] check slide 13

CG 4,5,6 related to assignment 1 do not skip the recursion

Need to understand the parallel and perspective projection
And the implied view volume
For now concentrate on glortho in OGL and its effect on the view volume
Understand valid polygons in OGL (more about it in in CG10 or CG11
CG 7, 8,
Self-study on material relevant to the assignments
Skip animation

Skip CG9