9/15/2020

Strict instructions for Participation Credit

It must be related to class questions.

The only format accepted is:
The email has subject line only with the format:

<name>, <class>, <date>, participated.
Example:
Dan Tamir CS3348 20200915 Participated

No contents inside
Must be received on the same day
Otherwise it will not be accepted for participation credit.

9/10/20

Key-board interaction example in big_deep.cpp
Interaction with the mouse in slide-set CG08

For assignment 1 – define the camera using:

```
        glMatrixMode (GL_PROJECTION);
        glLoadIdentity ();
        glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);  // Setting the view in this case it is setting to the default
// Setting the camera to Parallel Projection
```

Note that gluOrtho2D() (omitting the Z coordinates) is not necessarily supported in each version of the vm. Use glOrtho()
e.g., glOrtho(-8, 8, -1, 1, -1, 1) instead of gluOrtho2D(-8,1,-1,1))

Assignment view volume is different
Make sure that your vertices of interest are inside your own view volume.
Sinc.cpp includes an example of Keyboard interaction
Slide XYZ includes example of mouse interaction. Qt includes more rich methods for GUI with GL

Drawing an n-gon
Rectangular vs square (4-gon) the 4-gon has 4 equal edges.
n-gon – has n equal edges. There is an equation to plot a general n-gon.
If you know n you can draw a specific n-gon.

A very good tutorial in NeHe (use the 2.x tutorial)
Initially your program should only place the triangle.

You have to send one screenshot per interaction
(for each object one screenshot)

Requirement code (cpp) and screenshots

Generate objects using variables in glVertex() – for dynamic
How should we feed the values into variables?
GLFLOAT x, y, z ;

Read from a file,
Get from the user,
Get from the KBD,
Get from the Mouse,

Use a loop or a recursion
To generate the values and place into the variable
Example in sinc.cpp.

Place in a data structure; use the data

glVertex (x, y, z)

Set of slides 6 shows an example of dynamic object
creation using recursion.

The object is generated in dynamic time using recursion.

A problem that might occur with 3-D rendering is that the order of
Object generation may provide a different view than the “natural” view.

Objects that are closer to the eye might hide objects that are more distant

Hidden surface removal.
If object A is hiding [a part of] object B (it is closer to the eye/camera

Then 1) We want to make sure that the right part is hidden (B should no hide A)
2) Avoid rendering parts of objects that are invisible due to being hidden.

Open GL uses the Z buffer algorithm the buffer saves the z coordinates of vertices
And enable removing (not rendering) hidden [parts of] objects.

To enable use slide 6-21