

on linux:

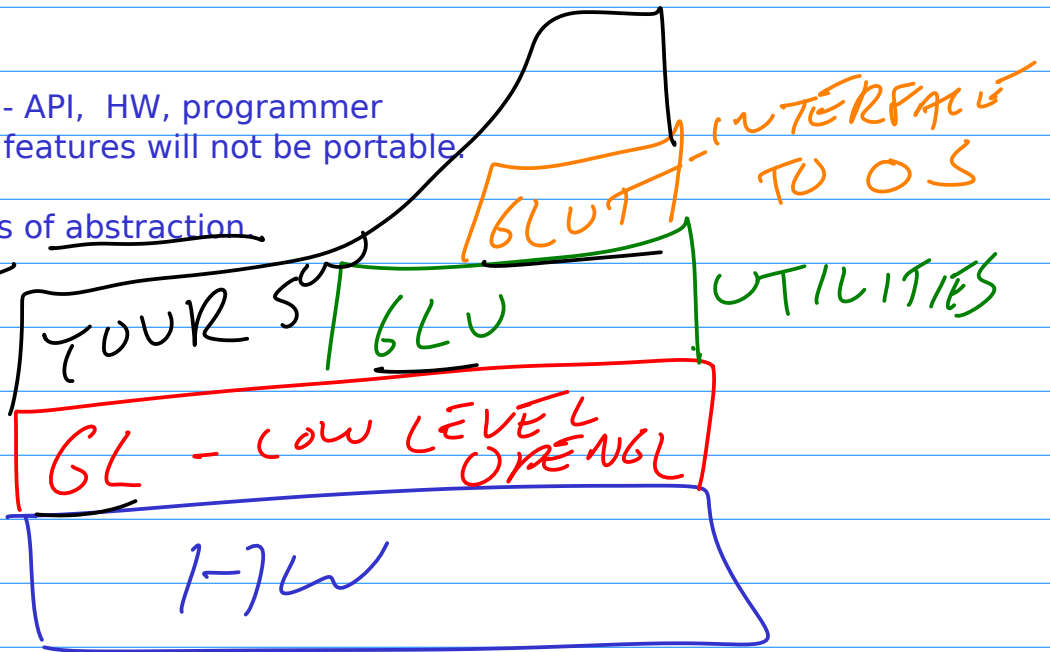
LDLIBS='-IGLEW -lglut -lGLU -lGL'

make box

OpenGL: portable - API, HW, programmer  
bleeding edge HW features will not be portable

OpenGL uses levels of abstraction

TO HIDE  
DETAILS



### EX OF HIDING DETAIL

- LOGICAL KEYBOARD SENDS CHARS TO CPU

2

OpenGL has an event loop. When an event occurs, it calls your callback routine (if given) then continues looping.

To draw an object:

1. exec GL\_BEGIN
2. exec vertex routines etc
3. exec GL\_END

glVertex3f It takes 3 floats as args

OpenGL has some internal STATE info, like vertex color.

When you set it, it applies to all future vertices, until changed.

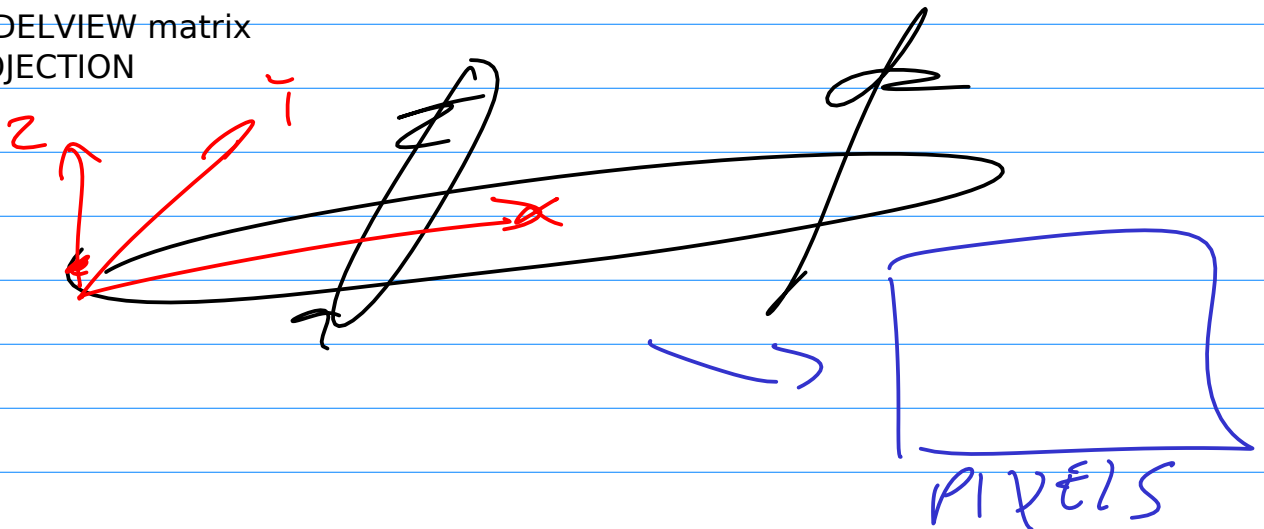
2 different types of coordinate systems:

1. When interacting with window manager, origin is TOP LEFT, units are pixels.
2. inside OpenGL, origin is BOTTOM LEFT, units are variable.

Sometimes you need to convert, e.g., with mouse location.

OpenGL uses 2 matrices to convert from user coordinate system to screen coordinates, via camera coordinates.

1. MODELVIEW matrix
2. PROJECTION



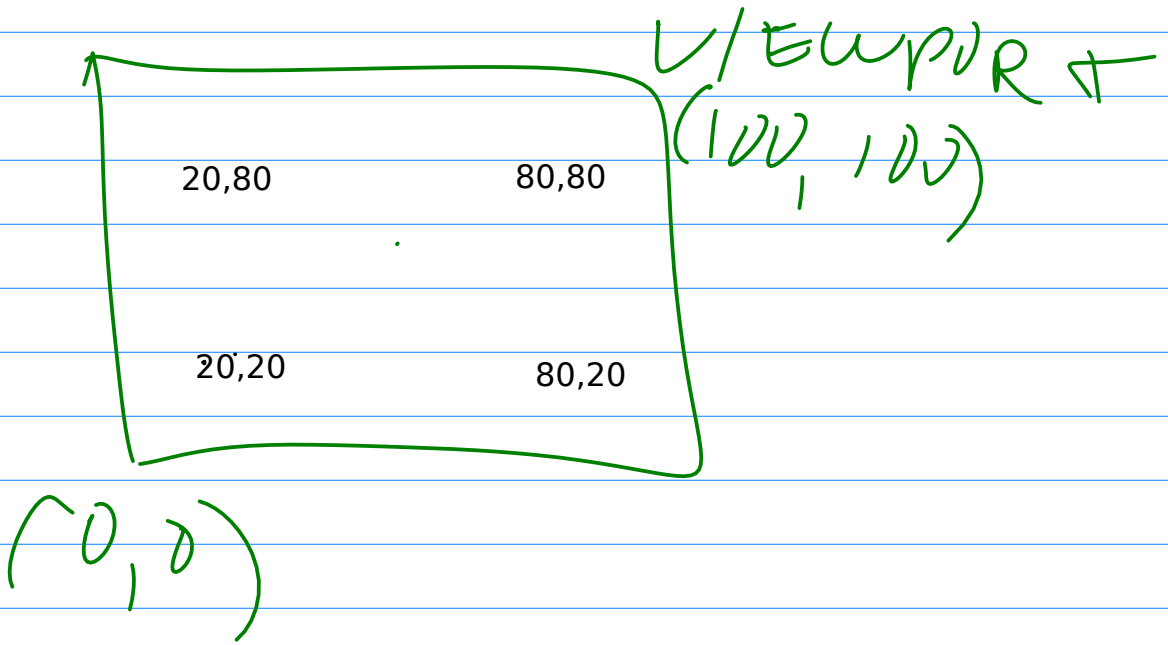
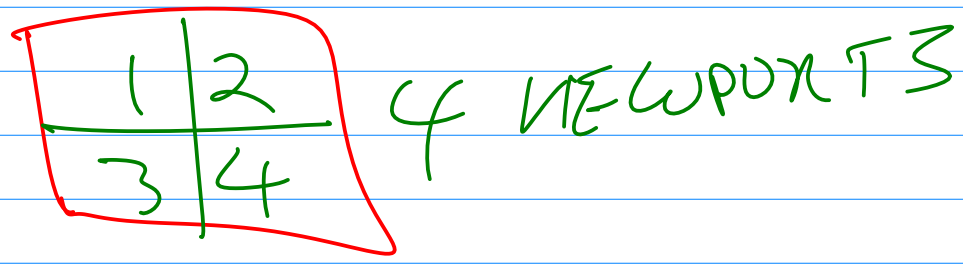
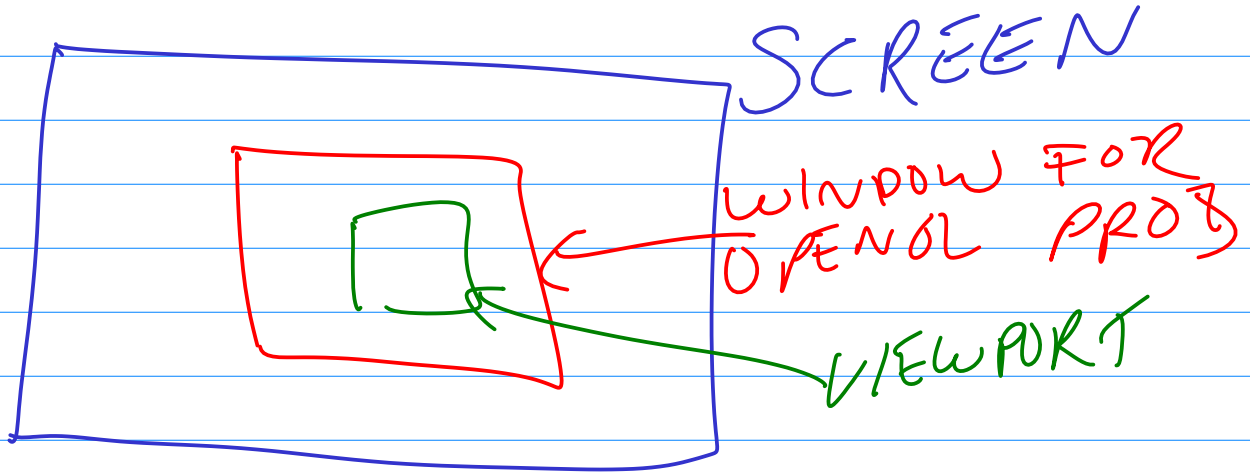
Callback routine call sequence is fixed by OpenGL.

You can store info in a callback routine with a static local variable.

You can use a different callback system, such as SFML or SDL.  
You'd get more modern stuff.

You can get info into any routine with a global variable.

· windows, viewports, etc. - terminology not standardized



\*\*\* glOrtho modifies the current matrix value by multiplying it. You almost certainly want to make the matrix to the identity first.

The default camera is at (0,0,0) looking to (0,0,-infinity).

No lab next Wed.