T10/11/2011

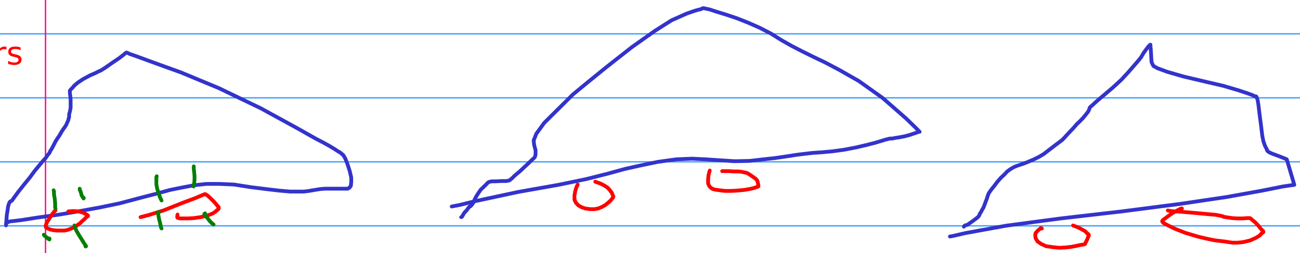
Problems with picking
1. What do you want?   Maybe the cursor overlaps several parts.
2. What's a part?   Upper left arm?   face 3 of that?   ...
3. There may be many instances of the part.
4. Even returning the line of code that generated what you picked may be insufficient.
5. OpenGL as we've seen it so far goes from data structures to the framebuffer. We want, sort of, to reverse that.

Quick and dirty method:
1. Color each object (part, ...) differently.
1a. Redraw the scene.
2. Read the color of the pixel under the cursor.
3. Use different colors only when you want to know that the pick was one or the other of those parts.
4. When drawn with these colors, the scene will not look realistic, so maybe draw into a separate buffer.   Use the back color buffer?
5. If several objects are on top of each other, you'll get the front one. That's probably what you want.
See text p 169, picking by color coding.

Method 2:   selection.cpp p 162
Just as you can set a color before drawing an object, you can also set a 'name'.
Selection returns the names of the objects inside a given volume.
Typically, that volume is a long thin box going back from the pixel you selected.

A 'name' you define applies to all future objects until you change it.
You can have a stack of names.

3 cars



Each car has 4 wheels

Each wheel has 4 nuts.

You select a nut. You want to know, say, "nut 3 of wheel 2 of car 1".
There are 48 nuts.   The name stack could record the car #, and the wheel # and the nut #.

One view of this section method:
1. Create a view volume around the cursor.
2. Clip the scene with that view volume.
3. Return the names of object intersecting that view volume.
4. Restore the old view volume and return from selection to viewing mode.
5. You draw your scene in normal view mode and draw it separately in selection mode, that time with names.
6. You can choose how fine grained your selection is, by setting size of selection view volume.
7. All objects at any depth intersecting it are returned.

'Picking' is a 3rd method, adds a little.   See p 168 ballandtorus picking.

This goes through the list of objects intersecting the pick region and highlights the closest.

Yes, selecting is complicated.

What's next?   Less programming, more math, e.g., transformations.
The point of chapter 4 was lots of OpenGL programming techniques.

ECSE intends to offer ECSE-6800 Advanced Graphics in the spring. It will continue where this course ends. I'll teach it.