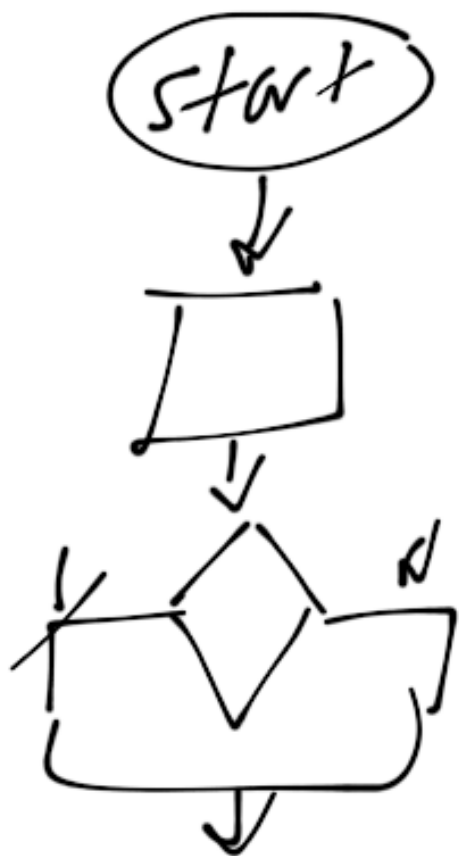
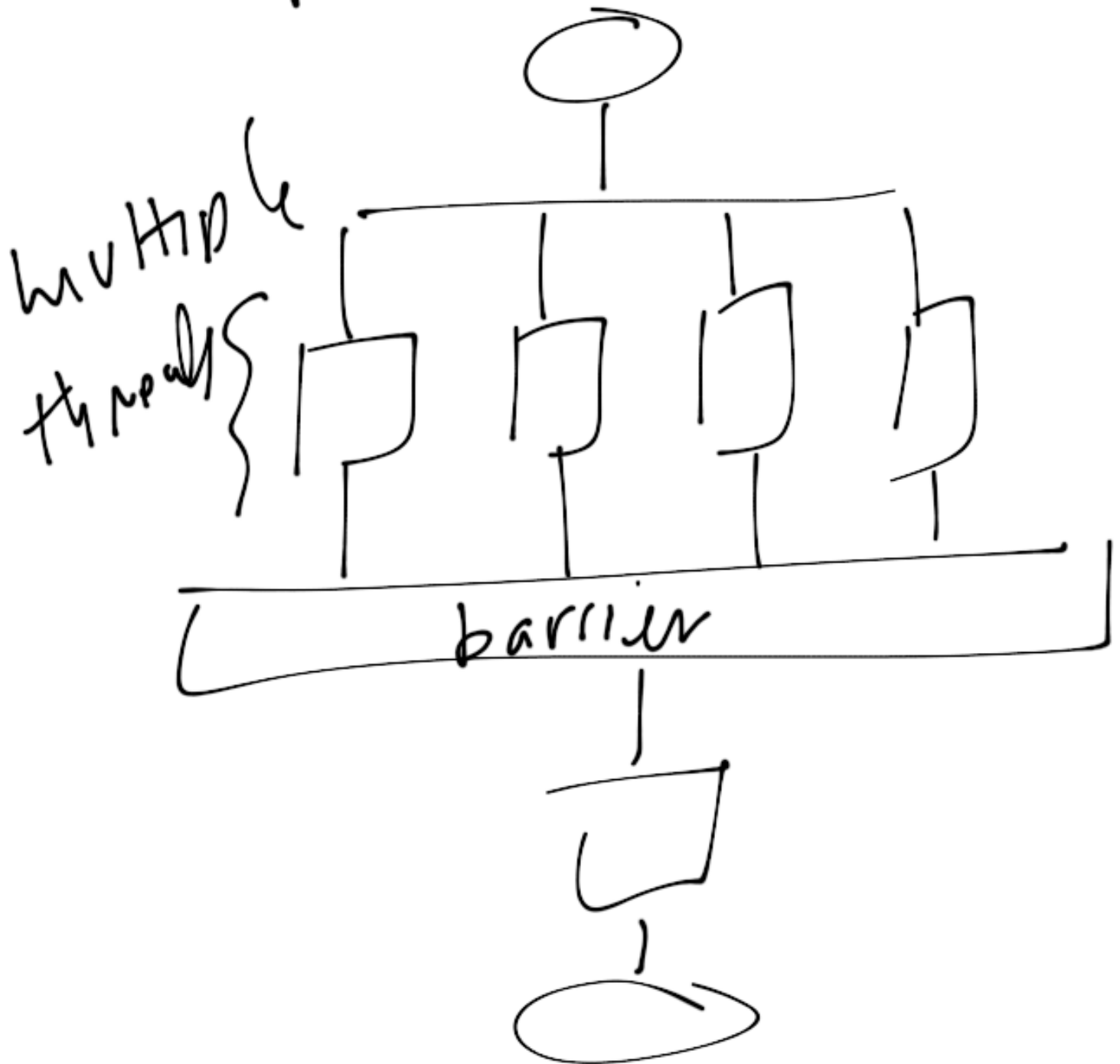


Procedural / Functional Programming:

- the "regular" flow of control
- start in main()



- Concurrent Programming: (OpenMP)

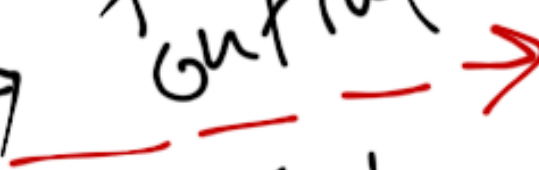


Event-driven programming (GUI)

your Gcd



flow of
control



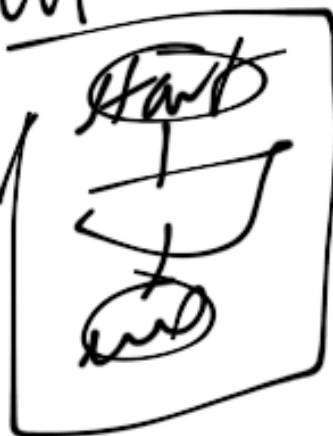
GUI



transf.
to GUI
Kbd
event

event

handler



listening
for events

(mouse, button, kbd)

Qt:

- signals:
issued when
events occur

- slots

public member fns
that process events:
event handlers

- the img - texture code

File

blank
drawable

empty
window that
accepts
OpenGL
(GL)

glTexObj;

glWinObj;

main obj;

contains menu bar, glTexObj;

- OpenGL: Graphics
Library

- Think of it as a
state started with

e.g. of color (1, 0, 0, 1)

sets color to red, fully

opaque

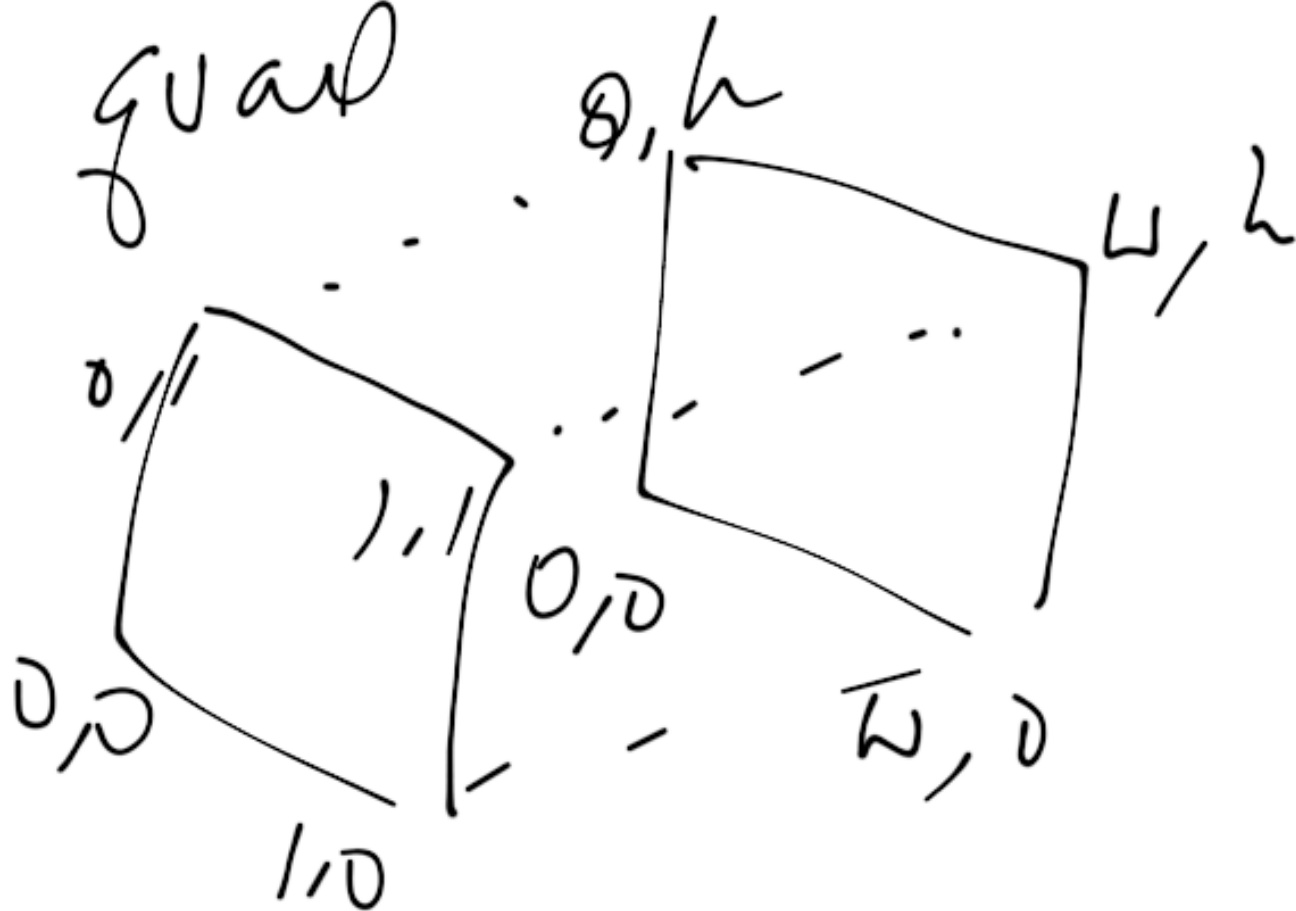
(transparent) alpha
channel

- In 3D texture:

- texture map
(image) to

colorize a sample

quad



- Qt photon visualizer:

- gltkoxy; needs
to maintain (private
data member) ~

`std::vector< photon_t * >`

- gltkoxy, need to
read in 'photons.pt'

- glTexCoord1i:

- in part 64():

loop through all

photon_t *

i

use glBegin(64_POINTS);

glVertex3f(

(*photon[i])[0],

(*photon[i])[1],

(*photon[i])[2]);

glEnd();

- ahead of point rendering
setup camera view:

glMatrixMode (GL_MODELVIEW),
glLoadIdentity(),

gluLookAt(4, 3, 6, (eye))

4, 3, 0, (up)

0, 1, 0); (right)

all in part 6L()

in ResizeGL (int w, int h)

glViewport (0, 0, w, h)

glMatrixMode (GL_PROJECTION)

glLoadIdentity();

gluPerspective (45,



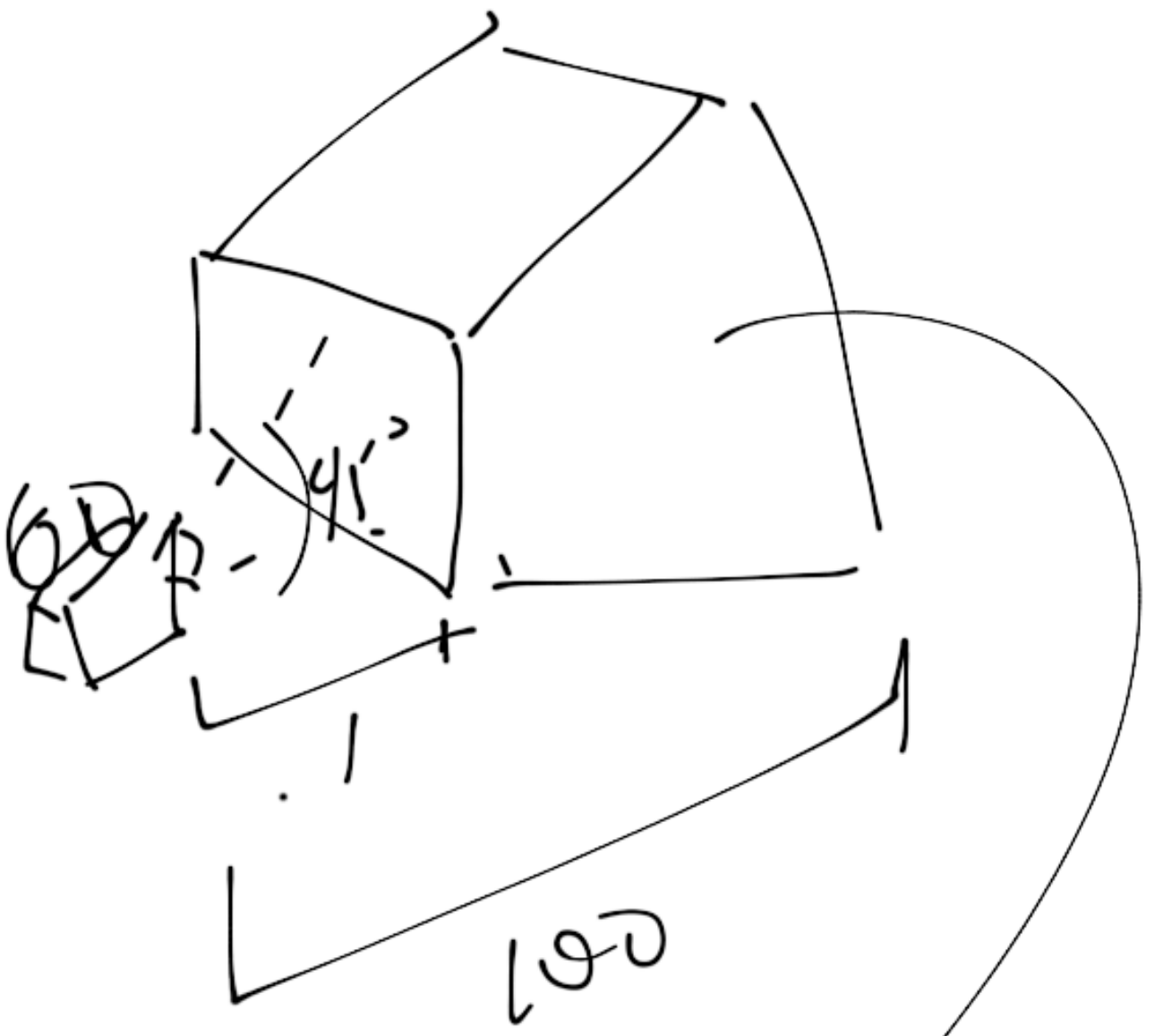
(fovy)

8/6

(aspect)

near, far
ViewPlane

(.1, 100)



Viewing
frustum