Tuesday, November 10, 2009
3:31 PM

- Goal: move code out main.cpp into glutbody.cpp
  (kdtree assignment)

taking note of how Asn. 5 data (the points)
was created, queried, obtained for kdtree
(retrieval)

data read from file

- Now, get user to click on GL window
  to generate points (left mouse button)

query point hardcoded

- generated interactively w/ middle mouse button
  (issue query to kdtree)

output just printed out

- draw selected points in different color
  (e.g. blue vs. green)

Tuesday, November 10, 2009
3:36 PM

- Getting list (vector) of data points
  (the vector of Point's that holds data)
- each left mouse button click gives a new point at coordinates of mouse during click.
  ⟹ create new Point, add to list, issue redraw event (draw list of points)
  } mouse click event handler
  } paintGL (redraw routine)

mouse click event handler signals app that redraw is needed —
NEVER CALL paintGL() DIRECTLY,
use updateGL() (signal)

- there are 3 mouse button events:
  mousePressEvent(QMouseEvent * e)
  mouseReleaseEvent(")
  mouseMoveEvent(?) (look this up)

Tuesday, November 10, 2009
3:45 PM

- various variables (Point* ptp; vector<Point*> pts, ...)
now get stored in GLTexobj, so in
gltexobj.h:

```
class GLTexobj : public QGLWidget
{
        Q_OBJECT
            :
    public slots:
        void mousePressEvent(QMouseEvent *e);
        :
    private:
        vector<Point*> pts;
        :
}
```

- in gltutorial.cpp

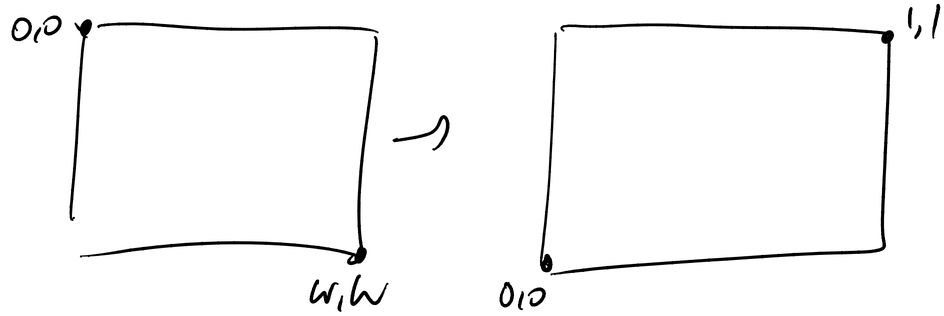
in mousePressEvent (QMouseEvent *e)

{

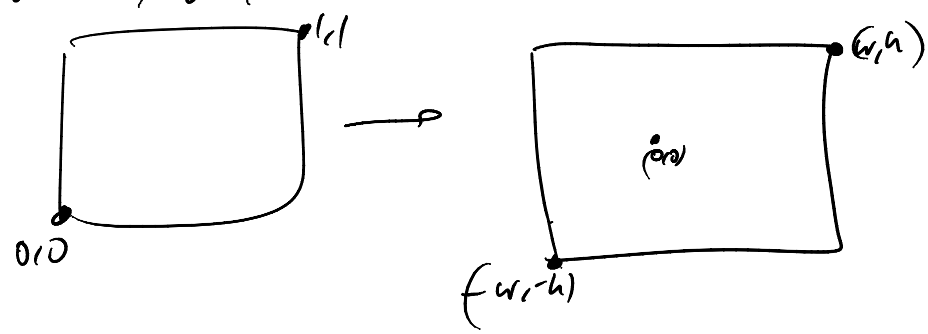
// normalize (x,y) & flip y-coord

double x = (double)e->x() / (double) width();

double y = ((double) height() - (double)e->y()) / ((double)height());



// scale & shift s.t. (0,0) is at center



x = width() * (2.0*x - 1.0) // [-w,w]

y = height() * (2.0*y - 1.0) // [-h,h]

← scaled & shifted mouse coordinates

- now create point & add it to list

ptp = new Point(x,y)

pts.push_back(ptp);

do this only when left mouse button is pressed

```
if (e->button() & LeftButton) {
```

logical and

```
}
```

- at the end of mouseMoveEvent,

```
e->accept(); //— event processed
update(); //— ignore signal to redraw
```

Tuesday, November 10, 2009
4:00 PM

- now, in paintGL, draw the points

- GLTest::paintGL()

for animation

```
}
    glDrawBuffer(GL_BACK);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor4f(0.0, 1.0, 0.0, 1.0);
    for (int i=0; i< (int)pts.size(); i++) {
        glBegin(GL_QUADS)
            glVertex2f((*pts)[i])[0]-5.0, (*pts)[i])[1]-5.0);
                //          +5.0        -5.0
                //          +5.0        +5.0
                //          -5.0        +5.0
        glEnd();
    }
    swapBuffers();
}
```

x-5, y+5 — x+5, y+5

x-5, y-5 — x+5, y-5

ccw

Tuesday, November 10, 2009
4:08 PM

- so far, data point generation,
  " " rendering

- next, need to build the kdTree

- again, let GLTextArea contain the kdTree object

kdTree<Point, PointInfo, PointAxis, Compare> kdtree

- now, let user press "build" tree button to create the kdTree — this will be done via Edit menu item
  - the callback for this menu item will have invoke:
    kdtree.insert(pts, Point(-width(), -height()), Point(width(), height()))

    ↓

    note, now have to match coordinate reference frame of all points

- build button will be an item in the Edit popup menu in the window