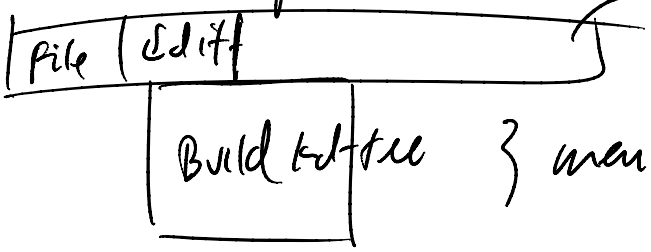
Thursday, November 12, 2009
3:39 PM

Last time:
- point generation
  - every left mouse click } mouseEventHandler
    generated a new point
    (point pointer)
- drawing points } paintGL

This time: build kd-tree: from a menu item

- in glwindobj.cpp (see img-texture example)
  want an Edit pop menu



menu item (like a button)

- in GLobjectWindow constructor, we have

```
menubar = new QMenuBar(this);        // parent object
QPopupMenu* edit = new QPopupMenu(menubar);   // parent; "new"
menubar->insertItem("Edit", edit);   // label; pop new (empty)
edit->insertItem("Build kd-tree", textwin,
    SLOT(grabkdTree()), CTRL+Key_k);
                                     // hot key
```

a function (slot) that gets called

hot key

- in glteapots.cpp we have

if of this type (teapotPtr = new GLTeapots)

```
void GLTeapots::growKdTree()
{
    kdtree.insert(pts, Point(-width(), -height()),
                       Point(width(), height()));
}
```

same as in A16 main.cpp

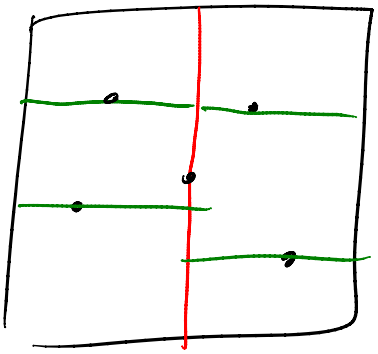
data

have to be a member of GLTeapots;

in glteapots.h

```
(std::vector<Point*> pts;)
kdTree<Point, Point*, PointAxisCompare> kdtree;
```

- so now we have:
  - point generation
  - point drawing
  - kdtree creation

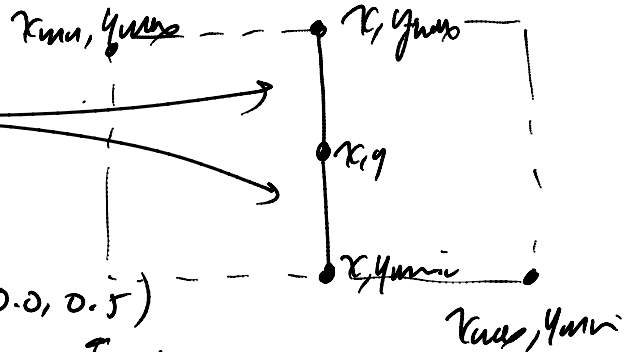


- how about drawing the kdTree
  - we can let the kdtree draw itself.

kdtree.cpp:

```
#include <GL/gl.h>

template <typename T, typename P, int C>
void kdTree<T,P,C>::render(kdNode* t)
{
  // note that there is a public version of render()
  // with no arguments
  if(!t) return; // (stop recursion if t is NULL)
  // draw via in-order traversal
  render(t->left);
  if(!t->axis) { // node splits on x-coord
    // draw three two
    // lines
    // just provide two
    // end points
    glColor4f(1.0, 0.0, 0.0, 0.5)
```



or just draw one line between (x, ymin) & (x, ymax)

```
glBegin(GL_LINES)                    // temporary
    glVertex2f((rt->data)[0], (rt->data)[1]);
    glVertex2f((rt->data)[0], t->min[1]);
glEnd();
// similarly for line from (x,y) to (x, ymax)
```

```
} else {  // split on y,
    glColor4f(0,1,0,.5)
    // draw line between (xmin,y) & (xmax,y)
}
render(t->right);
}
```

Thursday, November 12, 2009
4:18 PM

- now we have:
  - points,
  - kdtree,
  - drawing kdtree
  - drawing points

- queries:
  - single point query: middle mouse button
  - k point query: CTRL middle mouse button

```
mouseReleaseEvent(...)
{
    // mapping of x,y coords with (0,0) at center
    if (e->button() & MidButton) {
        // create new query point at xy coords
        if (e->state() & ControlButton) {
            // do knn query
            kNearest.clear();
            kdtree.knn(query, kNearest, r, k)
        } else {
            kdtree.nn(query, nearest, r)
        }
```

just like in nn.cpp

- next time, drawing bars

OpenGL odds & ends, e.g. initialization