

- initialize GLC()
- resize GLC()
- paint GLC()
- keyPressed Event()
- mouseReleaseEvent()
- mouseMoveEvent()

in gltextobj.cpp

asg05 + lab 11
(a^t vis.) (kd. true)

glwindj.cpp

- for the GL window,

set:

c → setMouseTracking(true)

or

texwin (GLTexobj*)

- File, Edit menus



read in/out

editme

points. in

(build time
action)

(points. pts)

qtexchj.cpl :

- GL window object,

contains :

double x, y : input point
(mouse)

double $x_1, x_2,$
 y_1, y_2 : range box
coords

vector<point_t *> pts;

```
kdtree_t < point_t, point_t *,  
point_c > kdTree;
```

```
point_t * query;
```

```
point_t * nearest; // na query result
```

```
double radius
```

```
int k (set to 3)
```

```
vector < point_t * > kNearest
```

```
// kth query result
```

```
vector < point_t * > range
```

initialize GL():

glClearColor(0, 0, 0, 0)

resize GL():

glViewport(0, 0, w, h)

glMatrixMode(GL_PROJECTION)

glLoadIdentity()

gluOrtho2D(-w, w, -h, h)

glMatrixMode(GL_MODELVIEW)

glLoadIdentity()

paintGL() :

// draws mouse point

// draws input points
(if any)

// draw kd-tree (if !empty)

// draw query point

// draw circle of radius r

// nearest point if there

// bounding box

// k nearest points (if any)

glDrawBuffer (GL_BACK)

glClear (GL_COLOR_BUFFER_

BIT |

GL_DEPTH_BUFFER_BIT);

// draw little white box at
mark point

glColor4f(1, 1, 1, 1)

glBegin(GL_QUADS)

glVertex2f(x-5, y-5)

1. (x+5, y-5)

|| (x+5, y+5)

|| (x-5, y+5)

glEnd()

- drawing circle:

glPushMatrix()

glTranslatef(x, y, z)

coords of given pt

$(x_{given})[0], (x_{given})[1]$

gluDisk($gobj, radius - z.o,$
 $radius, 360, 1$)

GLUquadricObj \times $gobj;$

$gobj =$
~~glPopMatrix()~~

// new quadric

range grey box:

`glRectd(x1, y1, x2, y2)`

ahead of this,

`glPolygonMode` (

`GL_FRONT_AND_BACK,`

`GL_LINE`)

(see man page for options)

- MouseEvent (QMouseEvent * e)

MapCoords (e → x(), e → y())
{
 ex
 ey

GLtexobj; :: x = ex;

GLtexobj; :: y = height() - ey;

double dx = x / width();

double dy = y / height();

x = width() * (2.0 * dx - 1.0)

y = height() * (2.0 * dy - 1.0)

→ (0,0) at center

```
if (e->button() ==  
    Qt::LeftButton) {
```

```
    if (e->modifiers() ==  
        Qt::NoModifier)
```

```
        pts.push_back
```

```
            (new point_t(x, y))
```

```
    else
```

```
        if (e->modifiers() ==
```

```
            Qt::ShiftModifier) {
```

```
            drawingBox = true;
```

```
            x1 = x; y1 = y;
```

```
            x2 = x; y2 = y;
```

- rest of actions happen
in mouseMoveEvent()
mouseReleaseEvent()



mapGoals(e → x(), e → y());

else :

if (e → modifiers() ==

(Qt::AltModifier | Qt::ShiftModifier)

// keyboard given

query = new Point(x, y)

kdTree.knn(query, kNearest, r, k)

if (e → modif/der) ==

at: i shift Modif/der {

$x_2 = x; y_2 = y_i$

point_t min, max;

$\text{min}[0] = x_1 < x_2 ? x_1 : x_2;$

$\text{min}[1] = y_1 < y_2 ? y_1 : y_2;$

$\text{max}[0] = x_1 > x_2 ? x_1 : x_2;$

$\text{max}[1] = y_1 > y_2 ? y_1 : y_2;$

drumrigBox = false

KdTree. range (min, max, range)

3

mouse Move Event :

mapCoords($e \rightarrow x()$, $e \rightarrow y()$)

$x2 = x$

$y2 = y$

$e \rightarrow \text{accept}()$

update GL() // redraw

(issue event)

- KdTree.render() :

- render via inOrder
+ recursive :

render(t → left)

// draw self

render(t → right)

- drawing & draw mode :

- figure out which axis
(horiz or vert. line)

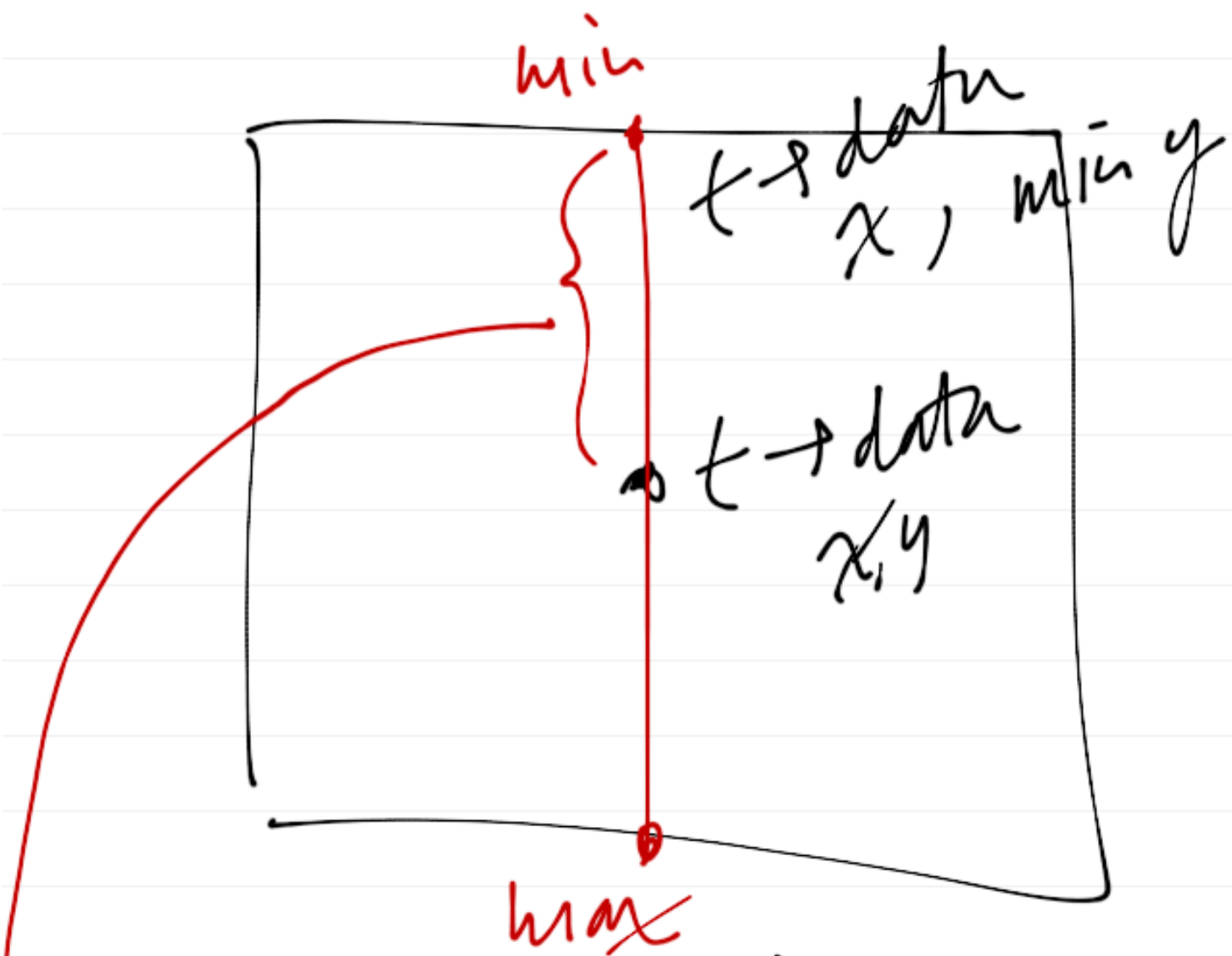
- draw line from
(min, pos)

to

(pos, max)

x1, y1

x2, y2



glBegin(64_LINES)

glVertex2f (*t -> data[0],
(*t -> data)[1])

glVertex2f (*t -> data[0],

t -> min[1])

glEnd()