Display Lists, Menus, and Picking

Tom Kelliher, CS 320

Feb. 18, 2005

1 Administrivia

Announcements

Assignment

Read `paint.c`, Section 3.8.

From Last Time

Input devices and interaction introduction.

Outline

1. Display lists.
2. Menus.
3. Pick selection.

Coming Up

`paint.c`

2 Display Lists, Menus, and Picking

Display lists and distributed computing.

2.1 Text Display

Idea:

1. Specify starting location of text (world coordinates).
2. Start writing, specifying font and character.

Example:

```
void renderString(GLdouble x, GLdouble y, void *font, char *text)
{
   glRasterPos2d(x, y);

   while (text)
   {
      glutBitmapCharacter(font, *text);
      ++text;
   }
}
```

See man page for `glutBitmapCharacter` for list of available bitmap fonts. Example:

```
renderString(0.0, 0.0, GLUT_BITMAP_9_BY_15, "OpenGL rocks!");
```

Idea similar to font cache: display lists. Program example:

```
base = glGenLists(128);

for(i=0;i<128;i++)
{
   glNewList(base+i, GL_COMPILE);
   glutBitmapCharacter(GLUT_BITMAP_9_BY_15, i);
   glEndList();
}

glListBase(base);
```

Use:

```
// Dump time string into out.

glRasterPos2i(ww-80, wh-15);  // Window units = world units.
glColor3f(0.0,0.0,0.0);
glBegin(GL_QUADS);            // Erase current time.
   glVertex2i(ww-80, wh-15);
   glVertex2i(ww, wh-15);
   glVertex2i(ww, wh);
   glVertex2i(ww-80, wh);
glEnd();
glColor3f(1.0,1.0,1.0);
glCallLists(strlen(out), GL_BYTE, out);
```

Another example:

```
list = glGenLists(1)

glNewList(list, GL_COMPILE);
   glBegin(GL_POLYGON);
      glVertex2f(...);
      ...
   glEnd();
glEndList();

...

glCallList(list);
```

2.2 Menus and Sub-Menus

Why do we use menus? — What's the alternative?

```
int main(int argc, char** argv)
{
    int c_menu, p_menu, f_menu;

    // ...

    c_menu = glutCreateMenu(color_menu);
    glutAddMenuEntry("Red",1);
    glutAddMenuEntry("Green",2);
    glutAddMenuEntry("Blue",3);
    glutAddMenuEntry("Cyan",4);
    glutAddMenuEntry("Magenta",5);
    glutAddMenuEntry("Yellow",6);
    glutAddMenuEntry("White",7);
    glutAddMenuEntry("Black",8);
    p_menu = glutCreateMenu(pixel_menu);
    glutAddMenuEntry("increase pixel size", 1);
    glutAddMenuEntry("decrease pixel size", 2);
    f_menu = glutCreateMenu(fill_menu);
    glutAddMenuEntry("fill on", 1);
    glutAddMenuEntry("fill off", 2);
    glutCreateMenu(right_menu);
    glutAddMenuEntry("quit",1);
    glutAddMenuEntry("clear",2);
    glutAttachMenu(GLUT_RIGHT_BUTTON);
    glutCreateMenu(middle_menu);
    glutAddSubMenu("Colors", c_menu);
    glutAddSubMenu("Pixel Size", p_menu);
    glutAddSubMenu("Fill", f_menu);
    glutAttachMenu(GLUT_MIDDLE_BUTTON);

    // ...
}
```

```
void color_menu(int id)
{
   if(id == 1) {r = 1.0; g = 0.0; b = 0.0;}
   else if(id == 2) {r = 0.0; g = 1.0; b = 0.0;}
   else if(id == 3) {r = 0.0; g = 0.0; b = 1.0;}
   else if(id == 4) {r = 0.0; g = 1.0; b = 1.0;}
   else if(id == 5) {r = 1.0; g = 0.0; b = 1.0;}
   else if(id == 6) {r = 1.0; g = 1.0; b = 0.0;}
   else if(id == 7) {r = 1.0; g = 1.0; b = 1.0;}
   else if(id == 8) {r = 0.0; g = 0.0; b = 0.0;}
}
```

2.3 Coordinate systems

1. Rendering origin for OpenGL.
2. Window and mouse coordinate origin for the window system.
3. Translating.

2.4 Pick Determination and Drawing States

What's the problem here?

1. Mouse function called into play here.
2. Note that right button taken out of action.
3. Canvas icons: five in row, upper-left. Dimensions?

```
int pick(int x, int y)
{
    y = wh - y;
    if(y < wh-ww/10) return 0;
    else if(x < ww/10) return LINE;
    else if(x < ww/5) return RECTANGLE;
    else if(x < 3*ww/10) return TRIANGLE;
    else if(x < 2*ww/5) return POINTS;
    else if(x < ww/2) return TEXT;
    else return 0;
}


void mouse(int btn, int state, int x, int y)
{
    static int count;
    int where;
    static int xp[2],yp[2];
    if(btn==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
    {
       glPushAttrib(GL_ALL_ATTRIB_BITS);

       where = pick(x,y);
       glColor3f(r, g, b);
       if(where != 0)
       {
          count = 0;
          draw_mode = where;
       }
       else switch(draw_mode)
       {
         case(LINE):
          if(count==0)
          {
              count++;
              xp[0] = x;
              yp[0] = y;
          }
          else
          {
              glBegin(GL_LINES);
                 glVertex2i(x,wh-y);
                 glVertex2i(xp[0],wh-yp[0]);
              glEnd();
              draw_mode=0;
              count=0;
          }
          break;
        case(RECTANGLE):
          if(count == 0)
          {
              count++;
              xp[0] = x;
              yp[0] = y;
          }
          else
          {
              if(fill) glBegin(GL_POLYGON);
              else glBegin(GL_LINE_LOOP);
                 glVertex2i(x,wh-y);
                 glVertex2i(x,wh-yp[0]);
                 glVertex2i(xp[0],wh-yp[0]);
                 glVertex2i(xp[0],wh-y);
              glEnd();
              draw_mode=0;
              count=0;
          }
          break;
        case (TRIANGLE):
          switch(count)
          {
            case(0):
              count++;
              xp[0] = x;
              yp[0] = y;
              break;
            case(1):
              count++;
              xp[1] = x;
              yp[1] = y;
              break;
            case(2):
              if(fill) glBegin(GL_POLYGON);
              else glBegin(GL_LINE_LOOP);
                 glVertex2i(xp[0],wh-yp[0]);
                 glVertex2i(xp[1],wh-yp[1]);
                 glVertex2i(x,wh-y);
              glEnd();
              draw_mode=0;
              count=0;
          }
          break;
        case(POINTS):
          {
             drawSquare(x,y);
             count++;
          }
        break;
      case(TEXT):
        {
          rx=x;
          ry=wh-y;
          glRasterPos2i(rx,ry);
          count=0;
        }
       }

       glPopAttrib();
       glFlush();
     }
}
```