

```
1: // $Id: showtext.cpp,v 1.29 2015-02-19 19:39:51-08 - - $
2:
3: // Display text using OpenGL.
4: // Text is from argv.
5:
6: #include <iostream>
7: #include <string>
8: #include <unordered_map>
9: using namespace std;
10:
11: #include <GL/freeglut.h>
12: #include <libgen.h>
13: #include <unistd.h>
14:
15: unordered_map<string,void*> font_map {
16:     {"f13", GLUT_BITMAP_8_BY_13 },
17:     {"f15", GLUT_BITMAP_9_BY_15 },
18:     {"h10", GLUT_BITMAP_HELVETICA_10 },
19:     {"h12", GLUT_BITMAP_HELVETICA_12 },
20:     {"h18", GLUT_BITMAP_HELVETICA_18 },
21:     {"t10", GLUT_BITMAP_TIMES_ROMAN_10},
22:     {"t24", GLUT_BITMAP_TIMES_ROMAN_24},
23: };
24:
25: char* progname = nullptr;
26: const GLubyte GREEN[] = {0, 255, 0};
27: struct {
28:     int width {640};
29:     int height {128};
30:     void* font = GLUT_BITMAP_TIMES_ROMAN_24;
31:     string text = "NO TEXT";
32: } window;
33:
34: void init_message (int argc, char**argv) {
35:     for (;;) {
36:         int opt = getopt (argc, argv, "f:h:t:");
37:         if (opt == EOF) break;
38:         string font_code = static_cast<char> (opt) + string (optarg);
39:         cout << progname << ": font = " << font_code << endl;
40:         const auto& entry = font_map.find (font_code);
41:         if (entry != font_map.end()) window.font = entry->second;
42:     }
43:     if (optind == argc) return;
44:     window.text = argv[optind];
45:     for (char** arg = &argv[optind + 1]; arg < &argv[argc]; ++arg) {
46:         (window.text += " ") += *arg;
47:     }
48:     cout << progname << ": " << window.text << endl;
49: }
50:
```

```
51:
52: void display() {
53:     glClearColor (0.2, 0.2, 0.2, 1.0);
54:     glClear (GL_COLOR_BUFFER_BIT);
55:     auto text = reinterpret_cast<const GLubyte*> (window.text.c_str());
56:     size_t width = glutBitmapLength (window.font, text);
57:     size_t height = glutBitmapHeight (window.font);
58:     glColor3ubv (GREEN);
59:     float xpos = window.width / 2.0 - width / 2.0;
60:     float ypos = window.height / 2.0 - height / 2.0;
61:     glRasterPos2f (xpos, ypos);
62:     glutBitmapString (window.font, text);
63:     glutSwapBuffers();
64: }
65:
66: void reshape (int width, int height) {
67:     window.width = width;
68:     window.height = height;
69:     glMatrixMode (GL_PROJECTION);
70:     glLoadIdentity();
71:     gluOrtho2D (0, window.width, 0, window.height);
72:     glMatrixMode (GL_MODELVIEW);
73:     glViewport (0, 0, window.width, window.height);
74:     glutPostRedisplay();
75: }
76:
77: int main (int argc, char** argv) {
78:     progname = basename (argv[0]);
79:     glutInit (&argc, argv);
80:     glutInitWindowSize (window.width, window.height);
81:     glutCreateWindow (progname);
82:     glutDisplayFunc (display);
83:     glutReshapeFunc (reshape);
84:     init_message (argc, argv);
85:     glutMainLoop();
86:     return 0;
87: }
88:
```