```
 1: // $Id: keyboard.cpp,v 1.12 2014-07-22 16:38:08-07 - - $
 2:
 3: // Respond to keystrokes.
 4:
 5: #include <iostream>
 6: #include <string>
 7: #include <unordered_map>
 8: using namespace std;
 9:
10: #include <GL/freeglut.h>
11: #include <libgen.h>
12:
13: unordered_map<int,string> special_keys {
14:    {GLUT_KEY_F1       , "GLUT_KEY_F1"       },
15:    {GLUT_KEY_F2       , "GLUT_KEY_F2"       },
16:    {GLUT_KEY_F3       , "GLUT_KEY_F3"       },
17:    {GLUT_KEY_F4       , "GLUT_KEY_F4"       },
18:    {GLUT_KEY_F5       , "GLUT_KEY_F5"       },
19:    {GLUT_KEY_F6       , "GLUT_KEY_F6"       },
20:    {GLUT_KEY_F7       , "GLUT_KEY_F7"       },
21:    {GLUT_KEY_F8       , "GLUT_KEY_F8"       },
22:    {GLUT_KEY_F9       , "GLUT_KEY_F9"       },
23:    {GLUT_KEY_F10      , "GLUT_KEY_F10"      },
24:    {GLUT_KEY_F11      , "GLUT_KEY_F11"      },
25:    {GLUT_KEY_F12      , "GLUT_KEY_F12"      },
26:    {GLUT_KEY_LEFT     , "GLUT_KEY_LEFT"     },
27:    {GLUT_KEY_UP       , "GLUT_KEY_UP"       },
28:    {GLUT_KEY_RIGHT    , "GLUT_KEY_RIGHT"    },
29:    {GLUT_KEY_DOWN     , "GLUT_KEY_DOWN"     },
30:    {GLUT_KEY_PAGE_UP  , "GLUT_KEY_PAGE_UP"  },
31:    {GLUT_KEY_PAGE_DOWN, "GLUT_KEY_PAGE_DOWN"},
32:    {GLUT_KEY_HOME     , "GLUT_KEY_HOME"     },
33:    {GLUT_KEY_END      , "GLUT_KEY_END"      },
34:    {GLUT_KEY_INSERT   , "GLUT_KEY_INSERT "  },
35: };
36:
37: unordered_map<int,string> control_chars {
38:    {  0, "NUL"}, {  1, "SOH"}, {  2, "STX"}, {  3, "ETX"},
39:    {  4, "EOT"}, {  5, "ENQ"}, {  6, "ACK"}, {  7, "BEL"},
40:    {  8, "BS" }, {  9, "HT" }, { 10, "LF" }, { 11, "VT" },
41:    { 12, "FF" }, { 13, "CR" }, { 14, "SO" }, { 15, "SI" },
42:    { 16, "DLE"}, { 17, "DC1"}, { 18, "DC2"}, { 19, "DC3"},
43:    { 20, "DC4"}, { 21, "NAK"}, { 22, "SYN"}, { 23, "ETB"},
44:    { 24, "CAN"}, { 25, "EM" }, { 26, "SUB"}, { 27, "ESC"},
45:    { 28, "FS" }, { 29, "GS" }, { 30, "RS" }, { 31, "US" },
46:    {127, "DEL"},
47: };
48:
```

```
 49:
 50: using uchar = unsigned char;
 51:
 52: // Characteristics of the window.
 53: struct window {
 54:    string name;
 55:    int width {256};
 56:    int height {192};
 57: } window;
 58:
 59: // Called by glutMainLoop to display window contents.
 60: void display() {
 61:    cout << __func__ << "()" << endl;
 62:    glClearColor (0.25, 0.25, 0.25, 1.0);
 63:    glClear (GL_COLOR_BUFFER_BIT);
 64:    glutSwapBuffers();
 65: }
 66:
 67: void print_special_key (int key) {
 68:    const auto& keyname = special_keys.find (key);
 69:    if (keyname == special_keys.end()) cout << "Unknown GLUT_KEY";
 70:                                  else cout << keyname->second;
 71: }
 72:
 73: void print_keyboard_key (int key) {
 74:    if (isgraph (key)) cout << "'" << (uchar)key << "'";
 75:    else {
 76:       const auto& control = control_chars.find (key);
 77:       if (control != control_chars.end()) cout << control->second;
 78:    }
 79: }
 80:
 81: void special (int key, int x, int y) {
 82:    cout << __func__ << "(" << key << "," << x << "," << y << "): ";
 83:    print_special_key (key);
 84:    cout << endl;
 85: }
 86:
 87: void specialup (int key, int x, int y) {
 88:    cout << __func__ << "(" << key << "," << x << "," << y << "): ";
 89:    print_special_key (key);
 90:    cout << endl;
 91: }
 92:
 93: void keyboard (uchar key, int x, int y) {
 94:    cout << __func__ << "(" << (int)key << "," << x << "," << y << "): ";
 95:    print_keyboard_key (key);
 96:    cout << endl;
 97: }
 98:
 99: void keyboardup (uchar key, int x, int y) {
100:    cout << __func__ << "(" << (int)key << "," << x << "," << y << "): ";
101:    print_keyboard_key (key);
102:    cout << endl;
103: }
104:
```

```
105:
106: void reshape (int width, int height) {
107:    cout << __func__ << "(" << width << "," << height << ")" << endl;
108:    window.width = width;
109:    window.height = height;
110:    glMatrixMode (GL_PROJECTION);
111:    glLoadIdentity();
112:    glOrtho (0, window.width, 0, window.height, -1, +1);
113:    glMatrixMode (GL_MODELVIEW);
114:    glViewport (0, 0, window.width, window.height);
115:    glutPostRedisplay();
116: }
117:
118: void close() {
119:    cout << __func__ << "()" << endl;
120: }
121:
122: void entry (int state) {
123:    cout << __func__ << "(";
124:    switch (state) {
125:       case GLUT_LEFT: cout << "GLUT_LEFT"; break;
126:       case GLUT_ENTERED: cout << "GLUT_ENTERED"; break;
127:       default: cout << state; break;
128:    }
129:    cout << ")" << endl;
130: }
131:
132: int main (int argc, char** argv) {
133:    window.name = basename (argv[0]);
134:    glutInit (&argc, argv);
135:    glutInitDisplayMode (GLUT_RGBA | GLUT_DOUBLE);
136:    glutInitWindowSize (window.width, window.height);
137:    glutCreateWindow (window.name.c_str());
138:    glutDisplayFunc (display);
139:    glutReshapeFunc (reshape);
140:    glutEntryFunc (entry);
141:    glutCloseFunc (close);
142:    glutKeyboardFunc (keyboard);
143:    glutKeyboardUpFunc (keyboardup);
144:    glutSpecialFunc (special);
145:    glutSpecialUpFunc (specialup);
146:    glutMainLoop();
147:    return 0;
148: }
149:
```

```
1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting keyboard.cpp
2: keyboard.cpp:
3:      $Id: keyboard.cpp,v 1.12 2014-07-22 16:38:08-07 - - $
4: g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++14 keyboard.cpp
5:          -o keyboard -lglut -lGLU -lGL -lX11 -lrt -lm
6: rm -f keyboard.o
7: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished keyboard.cpp
```