```
 1: // $Id: glclock.cpp,v 1.10 2019-02-22 15:37:17-08 - - $
 2:
 3: // Show a real-time analog clock.
 4:
 5: #include <cmath>
 6: #include <iostream>
 7: using namespace std;
 8:
 9: #include <GL/freeglut.h>
10: #include <libgen.h>
11: #include <time.h>
12:
13: struct {
14:    int width = 256;
15:    int height = 256;
16: } window;
17:
18: string program_name;
19: static const float RADIUS = 0.9;
20: static const GLubyte FOREGROUND[] {0x2F, 0x2F, 0x2F};
21:
22: struct calend {
23:    time_t clock;
24:    struct tm localtime;
25:    char sdate[64];
26:    char stime[64];
27:    void set() {
28:       clock = time (nullptr);
29:       localtime_r (&clock, &localtime);
30:       strftime (sdate, sizeof sdate, "%a %b %e", &localtime);
31:       strftime (stime, sizeof stime, "%T", &localtime);
32:    }
33: } calend;
34:
35: const GLubyte* to_ubytes (const char* cstring) {
36:    return reinterpret_cast<const GLubyte*> (cstring);
37: }
38:
39: void show_time() {
40:    void* font = GLUT_BITMAP_HELVETICA_18;
41:    glRasterPos2f (-0.95, -0.95);
42:    glutBitmapString (font, to_ubytes (calend.sdate));
43:    float timewidth = glutBitmapLength (font, to_ubytes (calend.stime));
44:    float timexpos = 0.95 - 2 * timewidth / window.width;
45:    glRasterPos2f (timexpos, -.95);
46:    glutBitmapString (font, to_ubytes (calend.stime));
47: }
48:
```

```
49:
50: void draw_dots (int points, int count) {
51:    glEnable (GL_POINT_SMOOTH);
52:    glPointSize (points);
53:    glBegin(GL_POINTS);
54:    glColor3ubv (FOREGROUND);
55:    for (float theta = 0; theta < 2 * M_PI; theta += 2 * M_PI / count) {
56:       float xdot = 0.9 * RADIUS * cos (theta);
57:       float ydot = 0.9 * RADIUS * sin (theta);
58:       glVertex2f (xdot, ydot);
59:    }
60:    glEnd();
61: }
62:
63: void draw_hand (GLfloat width, GLfloat length, GLfloat clock) {
64:    glEnable (GL_LINE_SMOOTH);
65:    glEnable (GL_POLYGON_SMOOTH);
66:    glPushMatrix();
67:    glRotatef (-clock * 6, 0, 0, 1);
68:    glColor3ubv (FOREGROUND);
69:    glBegin (GL_POLYGON);
70:    glVertex2f (-width / 2 * RADIUS, 0);
71:    glVertex2f (+width / 2 * RADIUS, 0);
72:    glVertex2f (+width / 8, length * RADIUS);
73:    glVertex2f (-width / 8, length * RADIUS);
74:    glEnd();
75:    glPopMatrix();
76: }
77:
78: void display() {
79:    glClear (GL_COLOR_BUFFER_BIT);
80:    glColor3ubv (FOREGROUND);
81:    draw_dots (2, 60);
82:    draw_dots (5, 12);
83:    calend.set();
84:    float second = calend.localtime.tm_sec;
85:    float minute = calend.localtime.tm_min + second / 60;
86:    float hour = calend.localtime.tm_hour + minute / 60;
87:    draw_hand (0.2, 0.5, hour * 5);
88:    draw_hand (0.1, 0.75, minute);
89:    draw_hand (0.05, 0.95, second);
90:    show_time();
91:    glutSwapBuffers();
92: }
93:
94: const float frequency_msec = 500;
95: void timer (int) {
96:    glutTimerFunc (frequency_msec, timer, 100);
97:    glutPostRedisplay();
98: }
99:
```

```
100:
101: void reshape (int width, int height) {
102:    cout << "reshape(width=" << width << ", height=" << height << endl;
103:    window.width = width;
104:    window.height = height;
105:    glMatrixMode (GL_PROJECTION);
106:    glLoadIdentity();
107:    gluOrtho2D (-1, +1, -1, +1);
108:    glMatrixMode (GL_MODELVIEW);
109:    glHint (GL_POINT_SMOOTH_HINT, GL_NICEST);
110:    glHint (GL_LINE_SMOOTH_HINT, GL_NICEST);
111:    glHint (GL_POLYGON_SMOOTH_HINT, GL_NICEST);
112:    glViewport (0, 0, window.width, window.height);
113:    glClearColor (0x2Fp0/255, 0x9Fp0/255, 0x2Fp0/255, 1.0);
114: }
115:
116: int main (int argc, char** argv) {
117:    program_name = basename (argv[0]);
118:    glutInit (&argc, argv);
119:    glutInitDisplayMode (GLUT_RGBA | GLUT_DOUBLE);
120:    glutInitWindowSize (window.width, window.height);
121:    glutCreateWindow (program_name.c_str());
122:    glutDisplayFunc (display);
123:    glutReshapeFunc (reshape);
124:    glutTimerFunc (frequency_msec, timer, 100);
125:    glutMainLoop();
126:    return 0;
127: }
128:
129: //TEST// mkpspdf glclock.ps glclock.cpp*
130:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting glclock.cpp
    2: checksource glclock.cpp
    3: ident glclock.cpp
    4: glclock.cpp:
    5:      $Id: glclock.cpp,v 1.10 2019-02-22 15:37:17-08 - - $
    6: cpplint.py.perl glclock.cpp
    7: Done processing glclock.cpp
    8: g++ -g -O0 -Wall -Wextra -Werror -Wpedantic -Wshadow -fdiagnostics-color
=never -std=gnu++17 -Wold-style-cast glclock.cpp -o glclock -lm -lglut -lGLU -l
GL -lX11 -ldrm -lm
    9: rm -f glclock.o
   10: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished glclock.cpp
```