```
 1: // $Id: translate.cpp,v 1.43 2019-02-22 17:45:37-08 - - $
 2:
 3: #include <iomanip>
 4: #include <iostream>
 5: #include <sstream>
 6: #include <string>
 7: using namespace std;
 8:
 9: #include <GL/freeglut.h>
10: #include <libgen.h>
11:
12: struct {
13:    string name;
14:    int width;
15:    int height;
16: } window;
17:
18: struct rgbcolor {
19:    union {
20:       GLubyte ubvec[3];
21:       struct {
22:          GLubyte red;
23:          GLubyte green;
24:          GLubyte blue;
25:       } rgb;
26:    };
27: };
28: const rgbcolor Red     {0xFF, 0x00, 0x00};
29: const rgbcolor Green   {0x00, 0xFF, 0x00};
30: const rgbcolor Blue    {0x00, 0x00, 0xFF};
31: const rgbcolor Cyan    {0x00, 0xFF, 0xFF};
32: const rgbcolor Magenta {0xFF, 0x00, 0xFF};
33: const rgbcolor Yellow  {0xFF, 0xFF, 0x00};
34: const rgbcolor White   {0xFF, 0xFF, 0xFF};
35: const rgbcolor Black   {0x00, 0x00, 0x00};
36:
37: string to_string (const rgbcolor& color) {
38:    ostringstream result;
39:    result << "0x"
40:           << hex << setiosflags (ios::uppercase) << setfill ('0')
41:           << setw(2) << unsigned (color.rgb.red)
42:           << setw(2) << unsigned (color.rgb.green)
43:           << setw(2) << unsigned (color.rgb.blue);
44:    return result.str();
45: }
46:
```

```
47:
48: void draw_rectangle (const rgbcolor& color, const string& name,
49:                      GLfloat xcenter, GLfloat ycenter) {
50:    cout << __PRETTY_FUNCTION__ << ":" << endl;
51:    cout << to_string (color) << ", "
52:         << xcenter << ", " << ycenter << ")" << endl;
53:    GLfloat delta_x = window.width / 8;
54:    GLfloat delta_y = window.height / 4;
55:    glPushMatrix();
56:    glTranslatef (xcenter,  ycenter, 0);
57:    glBegin (GL_POLYGON);
58:    glColor3ubv (color.ubvec);
59:    glVertex2f (-delta_x, -delta_y);
60:    glVertex2f (+delta_x, -delta_y);
61:    glVertex2f (+delta_x, +delta_y);
62:    glVertex2f (-delta_x, +delta_y);
63:    glEnd();
64:    rgbcolor inverse = {GLubyte (0xFF - color.rgb.red),
65:                        GLubyte (0xFF - color.rgb.green),
66:                        GLubyte (0xFF - color.rgb.blue)};
67:    glColor3ubv (inverse.ubvec);
68:    void* font = GLUT_BITMAP_TIMES_ROMAN_24;
69:    auto gl_name = reinterpret_cast<const GLubyte*> (name.c_str());
70:    float xpos = - glutBitmapLength (font, gl_name) / 2;
71:    float ypos = - glutBitmapHeight (font) / 2;
72:    glRasterPos2f (xpos, ypos);
73:    glutBitmapString (font, gl_name);
74:    glPopMatrix();
75:    glutSwapBuffers();
76: }
77:
78: void display() {
79:    GLfloat width = window.width;
80:    GLfloat height = window.height;
81:    glClear (GL_COLOR_BUFFER_BIT);
82:    draw_rectangle (Red    , "Red"    ,  width * 0.125, height * 0.75);
83:    draw_rectangle (Green  , "Green"  ,  width * 0.375, height * 0.75);
84:    draw_rectangle (Blue   , "Blue"   ,  width * 0.625, height * 0.75);
85:    draw_rectangle (White  , "White"  ,  width * 0.875, height * 0.75);
86:    draw_rectangle (Cyan   , "Cyan"   ,  width * 0.125, height * 0.25);
87:    draw_rectangle (Magenta, "Magenta",  width * 0.375, height * 0.25);
88:    draw_rectangle (Yellow , "Yellow" ,  width * 0.625, height * 0.25);
89:    draw_rectangle (Black  , "Black"  ,  width * 0.875, height * 0.25);
90: }
91:
```

```
 92:
 93: void reshape (int width, int height) {
 94:    cout << __PRETTY_FUNCTION__ << ": "
 95:         << width << ", " << height << endl;
 96:    window.width = width;
 97:    window.height = height;
 98:    ostringstream title;
 99:    title << window.name << "(" << window.width << ","
100:          << window.height << ")";
101:    glutSetWindowTitle (title.str().c_str());
102:    glutSetIconTitle (title.str().c_str());
103:    glMatrixMode (GL_PROJECTION);
104:    glLoadIdentity();
105:    gluOrtho2D (0, window.width, 0, window.height);
106:    glViewport (0, 0, window.width, window.height);
107:    glClearColor (0.5, 0.5, 0.5, 1.0);
108: }
109:
110: int main (int argc, char** argv) {
111:    window.name = basename (argv[0]);
112:    glutInit (&argc, argv);
113:    glutInitWindowSize (480, 360);
114:    glutCreateWindow (window.name.c_str());
115:    glutDisplayFunc (display);
116:    glutReshapeFunc (reshape);
117:    glutMainLoop();
118:    return 0;
119: }
120:
121: //TEST// mkpspdf translate.ps translate.cpp*
122:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting translate.cpp
    2: checksource translate.cpp
    3: ident translate.cpp
    4: translate.cpp:
    5:      $Id: translate.cpp,v 1.43 2019-02-22 17:45:37-08 - - $
    6: cpplint.py.perl translate.cpp
    7: Done processing translate.cpp
    8: g++ -g -O0 -Wall -Wextra -Werror -Wpedantic -Wshadow -fdiagnostics-color
=never -std=gnu++17 -Wold-style-cast translate.cpp -o translate -lm -lglut -lGL
U -lGL -lX11 -ldrm -lm
    9: rm -f translate.o
   10: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished translate.cpp
```