

```
1: // $Id: color-wheel.cpp,v 1.25 2019-02-22 15:36:02-08 - - $
2:
3: #include <cmath>
4: #include <iostream>
5: #include <memory>
6: #include <string>
7: #include <vector>
8: using namespace std;
9:
10: #include <GL/freeglut.h>
11: #include <libgen.h>
12:
13: // Characteristics of the window.
14: struct window {
15:     string name;
16:     int width {640};
17:     int height {480};
18: } window;
19:
20: const GLubyte WHITE[]    {255, 255, 255};
21: const GLubyte BLACK[]   { 0,   0,   0};
22: const GLubyte RED[]      {255,  0,   0};
23: const GLubyte GREEN[]    { 0, 255,  0};
24: const GLubyte BLUE[]     { 0,  0, 255};
25: const GLubyte CYAN[]     { 0, 255, 255};
26: const GLubyte MAGENTA[]  {255,  0, 255};
27: const GLubyte YELLOW[]   {255, 255,  0};
28:
29: vector<const GLubyte*> color_wheel {
30:     YELLOW, RED, MAGENTA, BLUE, CYAN, GREEN,
31: };
32:
33: void draw_6th_wedge (GLenum mode, const GLubyte* color,
34:                     const GLfloat angle, const GLfloat radius) {
35:     const GLfloat max_angle = angle + M_PI / 3.0;
36:     const GLfloat delta = M_PI / 30.0;
37:     glBegin (mode);
38:     glColor3ubv (color);
39:     glVertex2f (0.0, 0.0);
40:     for (GLfloat theta = angle; theta <= max_angle; theta += delta) {
41:         GLfloat xpos = radius * cos (theta);
42:         GLfloat ypos = radius * sin (theta);
43:         glVertex2f (xpos, ypos);
44:     }
45:     glEnd();
46: }
47:
48: void draw_6th_pie (const GLubyte* color,
49:                   const GLfloat angle, const GLfloat radius) {
50:     draw_6th_wedge (GL_POLYGON, color, angle, radius);
51:     glLineWidth (5.0);
52:     draw_6th_wedge (GL_LINE_LOOP, BLACK, angle, radius);
53: }
54:
```

```
55:
56: void display() {
57:     glClearColor (1.0, 1.0, 1.0, 1.0);
58:     glClear (GL_COLOR_BUFFER_BIT);
59:
60:     for (size_t color = 0; color != color_wheel.size(); ++color) {
61:         draw_6th_pie (color_wheel[color], M_PI / 3.0 * color,
62:                     min (window.width, window.height) * 0.4);
63:     }
64:
65:     glutSwapBuffers();
66: }
67:
68: void reshape (int width, int height) {
69:     window.width = width;
70:     window.height = height;
71:     glMatrixMode (GL_PROJECTION);
72:     glLoadIdentity();
73:     gluOrtho2D (- window.width / 2.0, + window.width / 2.0,
74:               - window.height / 2.0, + window.height / 2.0);
75:     glMatrixMode (GL_MODELVIEW);
76:     glViewport (0, 0, window.width, window.height);
77:     glutPostRedisplay();
78: }
79:
80: int main (int argc, char** argv) {
81:     window.name = basename (argv[0]);
82:     glutInit (&argc, argv);
83:     glutInitDisplayMode (GLUT_RGBA | GLUT_DOUBLE);
84:     glutInitWindowSize (window.width, window.height);
85:     glutCreateWindow (window.name.c_str());
86:     glutDisplayFunc (display);
87:     glutReshapeFunc (reshape);
88:     glutMainLoop();
89:     return 0;
90: }
91:
92: //TEST// mkpspdf color-wheel.ps color-wheel.cpp*
93:
```

[illegible]