```
// $Id: glclock.cpp,v 1.15 2016-07-20 14:20:55-07 - - $

// Show a real-time analog clock.

#include <cmath>
#include <iostream>
using namespace std;

#include <GL/freeglut.h>
#include <libgen.h>
#include <time.h>

struct {
   int width = 256;
   int height = 256;
} window;

string program_name;
float radius = 0.9;

struct calend {
   time_t clock;
   struct tm localtime;
   char sdate[64];
   char stime[64];
   void set() {
      clock = time (NULL);
      localtime_r (&clock, &localtime);
      strftime (sdate, sizeof sdate, "%a %b %e", &localtime);
      strftime (stime, sizeof stime, "%T", &localtime);
   }
} calend;

void show_time() {
   void* font = GLUT_BITMAP_HELVETICA_12;
   glRasterPos2f (-0.95, -0.95);
   glutBitmapString (font, (GLubyte*) calend.sdate);
   float timewidth = glutBitmapLength (font, (GLubyte*) calend.stime);
   float timexpos = 0.95 - 2 * timewidth / window.width;
   glRasterPos2f (timexpos, -.95);
   glutBitmapString (font, (GLubyte*) calend.stime);
}

void draw_dots (int points, int count) {
   glEnable (GL_POINT_SMOOTH);
   glPointSize (points);
   glBegin(GL_POINTS);
   for (float theta = 0; theta < 2 * M_PI; theta += 2 * M_PI / count) {
      float xdot = 0.9 * radius * cos (theta);
      float ydot = 0.9 * radius * sin (theta);
      glVertex2f (xdot, ydot);
   }
   glEnd();
}

```

```cpp

void draw_hand (GLfloat width, GLfloat length, GLfloat clock) {
   glEnable (GL_LINE_SMOOTH);
   glEnable (GL_POLYGON_SMOOTH);
   glPushMatrix();
   glRotatef (-clock * 6, 0, 0, 1);
   glColor3ub (0x2F, 0xFF, 0x2F);
   glBegin (GL_POLYGON);
   glVertex2f (-width / 2 * radius, 0);
   glVertex2f (+width / 2 * radius, 0);
   glVertex2f (+width / 8, length * radius);
   glVertex2f (-width / 8, length * radius);
   glEnd();
   glPopMatrix();
}

void display() {
   glClear (GL_COLOR_BUFFER_BIT);
   glColor3ub (0x2F, 0xFF, 0x2F);
   draw_dots (2, 60);
   draw_dots (5, 12);
   calend.set();
   float second = calend.localtime.tm_sec;
   float minute = calend.localtime.tm_min + second / 60;
   float hour = calend.localtime.tm_hour + minute / 60;
   draw_hand (0.2, 0.5, hour * 5);
   draw_hand (0.1, 0.75, minute);
   draw_hand (0.05, 0.95, second);
   show_time();
   glutSwapBuffers();
}

const float frequency = 500;
void timer (int) {
   glutTimerFunc (frequency, timer, 100);
   glutPostRedisplay();
}

```

```

void reshape (int width, int height) {
   cout << "reshape(width=" << width << ", height=" << height << endl;
   window.width = width;
   window.height = height;
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D (-1, +1, -1, +1);
   glMatrixMode (GL_MODELVIEW);
   glHint (GL_POINT_SMOOTH_HINT, GL_NICEST);
   glHint (GL_LINE_SMOOTH_HINT, GL_NICEST);
   glHint (GL_POLYGON_SMOOTH_HINT, GL_NICEST);
   radius = 0.9;
   glViewport (0, 0, window.width, window.height);
   float gray = 0x2Fp0 / 0xFFp0;
   glClearColor (gray, gray, gray, 1.0);
}

int main (int argc, char** argv) {
   program_name = basename (argv[0]);
   glutInit (&argc, argv);
   glutInitDisplayMode (GLUT_RGBA | GLUT_DOUBLE);
   glutInitWindowSize (window.width, window.height);
   glutCreateWindow (program_name.c_str());
   glutDisplayFunc (display);
   glutReshapeFunc (reshape);
   glutTimerFunc (frequency, timer, 100);
   glutMainLoop();
   return 0;
}
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting glclock.cpp
glclock.cpp:
     $Id: glclock.cpp,v 1.15 2016-07-20 14:20:55-07 - - $
g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++14 glclock.cpp
        -o glclock -lglut -lGLU -lGL -lX11 -lrt -lm
rm -f glclock.o
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished glclock.cpp
```