

Modified Code:

#include <stdio.h>

#include <time.h>

#include <string.h>

#include <gl/glut.h>

#include <math.h>

void init()

{

glClearColor(0.2, 0.2, 0.2, 1.0);

glColor3f(0.0f, 0.0f, 0.0f);

glPointSize(2.0f);

//gluOrtho2D(0.0f, 640.0f, 0.0f, 480.0f);

glEnable(GL\_POINT\_SMOOTH);

glEnable(GL\_POLYGON\_SMOOTH);

glEnable(GL\_LINE\_SMOOTH);

}

double seconds = 0;

double minutes = 0;

double hours = 0;

void Timer(int value)

{

double tickInterval = 10; // Chnage this value to set the step size.

double tickDegree = (360 / 60) \* (tickInterval / 1000);

glutTimerFunc(tickInterval, Timer, 0);

glutPostRedisplay();

seconds += tickDegree;

minutes += tickDegree / 60;

hours += (tickDegree / 60) / 12;

if (seconds > 360)

seconds -= 360;

if (minutes > 360)

minutes -= 360;

if (hours > 360)

hours -= 360;

printf("%s %f\n", "Seconds = ", seconds);

}

double ticks = 0;

// Your first Point The hello world application

void drawpoint()

{

glPushMatrix();

glRotated(ticks += 1, 0, 0, 1);

glTranslated(0.95, 0, 0);

glColor3f(1, 1, 1);

glBegin(GL\_POINTS);

glVertex2d(0, 0);

glEnd();

glPopMatrix();

}

double mark = 0;

double space = 360 / 12;

void drawTimeMarker()

{

glPushMatrix();

glRotated(mark += space, 0, 0, 1);

glColor3f(1, 1, 1);

glBegin(GL\_POLYGON);

glVertex2d(-0.003, 0.8);

glVertex2d(0.003, 0.8);

glVertex2d(-0.003, 0.9);

glVertex2d(0.003, 0.9);

glEnd();

glPopMatrix();

}

void drawSecondsHand()

{

glPushMatrix(); //Tells OpenGL to store the current state that we are in.

glRotated(-seconds, 0, 0, 1);

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_POLYGON);

glVertex2d(-0.01, 0);

glVertex2d(0.01, 0);

glVertex2d(0.01, 0.6);

glVertex2d(0, 0.9);

glVertex2d(-0.01, 0.6);

glEnd();

glPopMatrix(); //Then when we want to go back to our previous state, we call glPopMatrix().

}

void drawMinuteHand()

{

glPushMatrix();

glRotated(-minutes, 0, 0, 1);

glColor3f(0.0, 1.0, 0.0);

glBegin(GL\_POLYGON);

glVertex2d(-0.01, 0);

glVertex2d(0.01, 0);

glVertex2d(0.01, 0.5);

glVertex2d(0, 0.8);

glVertex2d(-0.01, 0.5);

glEnd();

glPopMatrix();

}

void drawHoursHand()

{

glPushMatrix();

glRotated(-hours, 0, 0, 1);

glColor3f(0.0, 0.0, 1.0);

glBegin(GL\_POLYGON);

glVertex2d(-0.02, 0);

glVertex2d(0.02, 0);

glVertex2d(0.02, 0.4);

glVertex2d(0, 0.7);

glVertex2d(-0.02, 0.4);

glEnd();

glPopMatrix();

}

void drawcircle()

{

// glLoadIdentity(); //Reset the drawing perspective

for (int i = 0; i < 360; i++)

{

drawpoint();

}

for (int i = 0; i < 12; i++)

{

drawTimeMarker();

}

/\*

OpenGL keeps a stack of matrices to quickly apply and remove transformations. glPushMatrix copies the top matrix and pushes it onto the stack, while glPopMatrix pops the top matrix off the

stack. All transformation functions (glScaled, etc.) function on the top matrix, and the top matrix is what all rendering commands use to transform their data.

By pushing and popping matrices, you can control what transformations apply to which objects, as well as apply transformations to groups of objects, and easily reverse the transformations

so that they don't affect other objects.

\*/

// hours hand

drawHoursHand();

// Minutes hand

drawMinuteHand();

// Seconds hand

drawSecondsHand();

}

void display(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glLineWidth(2.5);

glColor3f(1.0, 0.0, 0.0);

drawcircle();

glFlush();

}

int main(int argc, char \*\*argv) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(200, 200);

glutInitWindowSize(480, 480);

glutCreateWindow("Square");

glutDisplayFunc(display);

init();

Timer(0);

glutMainLoop();

return 0;

}