  
  
  
  
  
#include <GL/glut.h>

#include <cmath>

#include <iostream>

#ifndef M\_PI

#define M\_PI 3.14159265358979323846

#endif

float centerX = 200.0f;

float centerY = 200.0f;

float radius = 100.0f;

void drawCircle(float cx, float cy, float r) {

const int num\_segments = 100;

const float theta = 2 \* M\_PI / num\_segments;

const float c = cosf(theta);

const float s = sinf(theta);

float x = r;

float y = 0;

glBegin(GL\_POLYGON);

for (int i = 0; i < num\_segments; ++i) {

glVertex2f(cx + x, cy + y);

float tempX = x;

x = c \* x - s \* y;

y = s \* tempX + c \* y;

}

glEnd();

}

void drawTriangle(float cx, float cy, float r) {

const float angle = 2 \* M\_PI / 3.0;

glBegin(GL\_TRIANGLES);

for (int i = 0; i < 3; ++i) {

float x = cx + r \* cosf(i \* angle);

float y = cy + r \* sinf(i \* angle);

glVertex2f(x, y);

}

glEnd();

}

void drawSquare(float cx, float cy, float side) {

float halfSide = (side ) / 2;

glBegin(GL\_POLYGON);

glVertex2f(cx - halfSide - 10, cy - halfSide);

glVertex2f(cx + halfSide - 10, cy - halfSide);

glVertex2f(cx + halfSide - 10, cy + halfSide);

glVertex2f(cx - halfSide - 10, cy + halfSide);

glEnd();

}

void display(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0, 1.0, 1.0);

glLineWidth(3.0f);

drawCircle(centerX, centerY, radius);

float triangleRadius = radius;

float triangleSide = triangleRadius \* sqrt(3.0);

glColor3f(0.0, 1.0, 0.0);

drawTriangle(centerX, centerY, triangleRadius);

float height = triangleSide \* sqrt(3.0) / 3.0;

float squareSide = (height / sqrt(2.0)) + 8;

glColor3f(1.0, 0.0, 0.0);

drawSquare(centerX, centerY, squareSide);

float circleRadius = squareSide / 2.0;

glColor3f(0.0, 0.0, 1.0);

drawCircle(190, centerY, circleRadius);

glFlush();

}

void init(void) {

glClearColor(0.0, 0.0, 0.0, 1.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0, 400, 0, 400);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(400, 400);

glutInitWindowPosition(100, 100);

glutCreateWindow("OST");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}