**CSE 4200 Lab 5 – Spencer Wallace**

**Summary:**

For this lab I was able to successfully display the four Archimedean spirals in their respective quadrants. I was also able to make a program which toggled between a green and red square using culling and detecting the user’s mouse click. I would give myself full credit plus the 10 points extra credit, for a score of **30/20**.

**Archimedean spirals (code on next page)**

Shape, background pattern

Description automatically generated

**Archimedean Spiral Code**

[007463307@csusb.edu@csevnc lab5]$ cat arch\_spiral.cpp

#include "canvas.h"

Canvas cvs ( 600, 600, (char\*)"Arch Spirals - Spencer Wallace" );

void spiral(float x1, float y1, int maxPoints)

{

cvs.setWindow( -600, 600, -600, 600 );

float angle = 0.0f;

// Space between the spirals

float a = 5;

float x = 0.0, y = 0.0;

float f = a\*angle;

cvs.moveTo(x1,y1);

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

{

angle = 0.1 \* i;

f = a\*angle;

x = ( f \* cos(angle) );

y = ( f \* sin(angle) );

cvs.lineTo( x1+x, y1+y );

}

}

void display(void)

{

cvs.clearScreen();

cvs.moveTo(0.0, 0.0); //starts at center

cvs.turnTo ( 0.0 ); //points horizontally

glLineWidth ( 1 );

spiral( 300, 300, 300 );

cvs.setColor(1,0,0);

spiral( 300, -300, 300 );

cvs.setColor(0,1,0);

spiral( -300, 300, 300 );

cvs.setColor(0,0,1);

spiral( -300, -300, 300 );

}

**Clicky Square**

A green rectangle with a black background

Description automatically generated with low confidenceShape, square

Description automatically generated

**Clicky Square Code**

[007463307@csusb.edu@csevnc lab5]$ cat clicky\_square.cpp

#include <GL/glut.h>

#include <stdlib.h>

#include <cstdio>

bool FACE = true;

struct square{

int x1;

int x2;

int y1;

int y2;

square();

square(int x1, int x2, int y1, int y2){

this->x1 = x1; this->x2 = x2; this->y1 = y1; this->y2 = y2;

}

void drawSquare(bool CCW)

{

glBegin( GL\_POLYGON );

glVertex2i( x1, y1 );

(CCW) ? glVertex2i( x2, y1 ) : glVertex2i( x1, y2 );

glVertex2i( x2, y2 );

(CCW) ? glVertex2i( x1, y2 ) : glVertex2i( x2, y1 );

glEnd();

}

bool checkClick(int x, int y)

{

if ( (x > x1 && x < x2 ) && ( y < y2 && y > y1 ) )

return true;

return false;

}

};

int SQUARE\_x1 = 100;

int SQUARE\_x2 = 500;

int SQUARE\_y1 = 100;

int SQUARE\_y2 = 500;

square\* SQUARE = new square(SQUARE\_x1, SQUARE\_x2, SQUARE\_y1, SQUARE\_y2);

void init(void)

{

glClearColor (0.0, 0.0, 0.0, 0.0);

glShadeModel (GL\_FLAT);

}

void display(void)

{

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glColor3f (0.0, 1.0, 0.0);

square\* GreenSquare = new square(SQUARE\_x1, SQUARE\_x2, SQUARE\_y1, SQUARE\_y2);

square\* RedSquare = new square(SQUARE\_x1, SQUARE\_x2, SQUARE\_y1, SQUARE\_y2);

glPolygonMode(GL\_BACK, GL\_FILL);

glColor3f (1.0, 0, 0);

RedSquare->drawSquare(false);

glPolygonMode(GL\_FRONT, GL\_FILL);

glColor3f (0, 1.0, 0);

GreenSquare->drawSquare(true);

glFlush ();

}

void mouseClick(int button, int state, int x, int y)

{

if(button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

{

if(SQUARE->checkClick(x, y))

{

glEnable(GL\_CULL\_FACE);

printf("mouseClick | x is : %d and y is : %d\n", x, y);

(FACE) ? glCullFace(GL\_FRONT) : glCullFace(GL\_BACK);

FACE = !FACE;

glutPostRedisplay();

}

}

}

void reshape (int w, int h)

{

glViewport (0, 0, (GLsizei) w, (GLsizei) h);

glMatrixMode (GL\_PROJECTION);

glLoadIdentity ();

gluOrtho2D (0.0, (GLdouble) w, 0.0, (GLdouble) h);

}

void keyboard(unsigned char key, int x, int y)

{

switch (key) {

case 27:

exit(0);

break;

}

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (600, 600);

glutInitWindowPosition (100, 100);

glutCreateWindow ( (char\*)"Clicky Square - Spencer Wallace");

init ();

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutKeyboardFunc(keyboard);

glutMouseFunc(mouseClick);

glutMainLoop();

return 0;

}