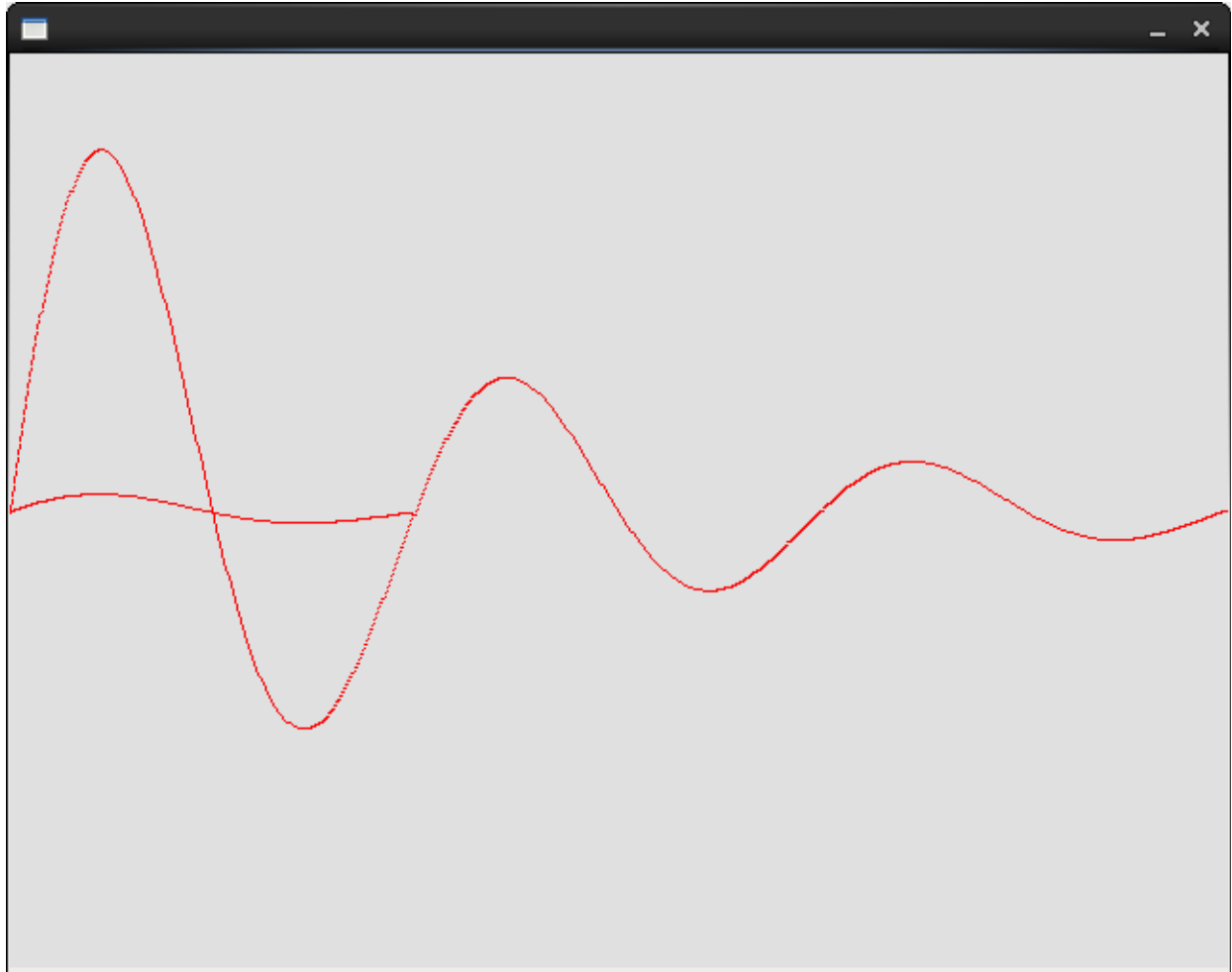


CSE420  
Samuel Marrujo  
Professor Yu  
Lab 03

### Plots

In this part of the lab, it was a modification of the original plots function. After changing the function towards y, the resulting image displayed an entirely different function than what was given. Because of this change, I believe I was able to accomplish this task successfully, since there were no errors and the display window was changed. Here are my results for this program:



```

/*
plots.cpp
Demonstrate the drawing of a line using the surface class developed
by T.L. Yu

@Author: T.L. Yu, April 15, 2006

*/
#include <SDL/SDL.h>
#include <stdlib.h>
#include <stdio.h>

#include "draw.h"
#include "surface.h"

using namespace std;

double f ( double x )
{
    double y = exp ( -fabs ( x ) ) * sin ( (2) * 3.1415926 * x );

    return y;
}

int main()
{

    const int VWIDTH = 640;
    const int VHEIGHT = 480;

    const Point center ( VWIDTH/2, VHEIGHT/2 ); //center of screen
    Surface surf( VWIDTH, VHEIGHT, (char *) "draw_line" );

    surf.clearScreen();                //clear screen
    surf.updateSurface();
    SDL_Delay ( 1000 );                //dealy one second, just for demo

    surf.setBackgroundColor ( 0xe0, 0xe0, 0xe0 ); //set background to grey

    //draw a line
    surf.setColor ( 0xff, 0, 0 );      //using red color

    int sx, sy;
    double x, y, a, b, c, d;
    a = VWIDTH / 3;
    c = 0;
    b = -VHEIGHT / 2.0;
    d = VHEIGHT / 2.0;

```

```

x = 0.0;                //initial position
y = f ( 0.0 );
sx = (int) ( a * x + c );
sy = (int) ( b * y + d );
surf.moveTo ( sx, sy );

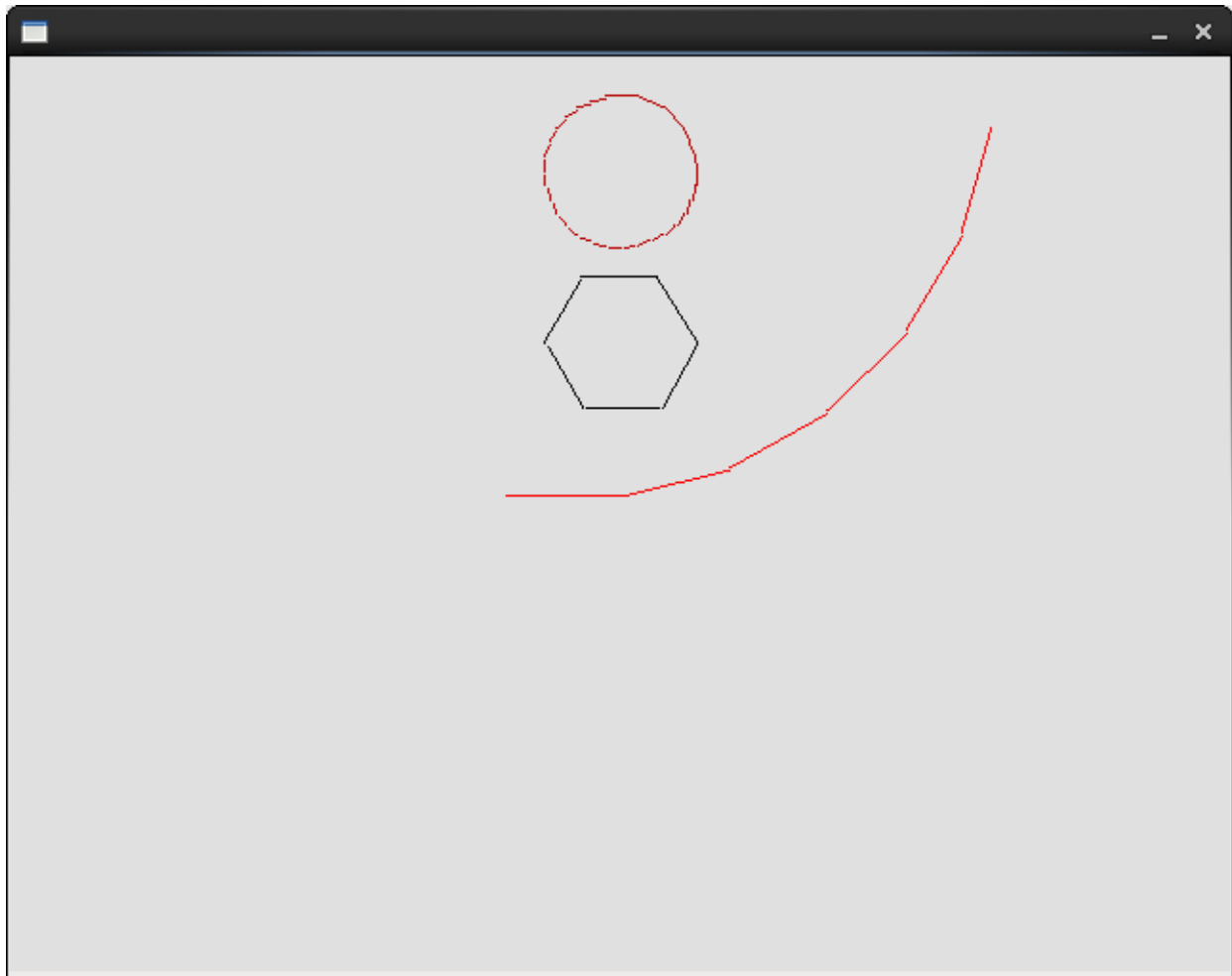
for ( x = 0; x < 4.0; x += 0.005 ) {
    y = f ( x );
    sx = (int) ( a * x + c );
    sy = (int) ( b * y + d );
    surf.lineTo ( sx, sy );
}

surf.updateSurface();
SDL_Delay ( 10000 );    //display 10 seconds before exit
return 1;
}

```

## Hook

Lastly, in the Hook program, we are to use the turtle algorithms discussed in class to form polygons, arcs, and circles. Due to the screen-shots given and the code supplied, I was able to successfully complete the Lab. So, the outcome of the programs led to the following screen shots in the program:



```

/*
draw_demo.cpp
Demonstrate the use of turtle graphics using the surface class developed
by T.L. Yu
edited by Samuel Alex Marrujo

@Author: T.L. Yu, April 15, 2006
Revised on 2013/09/30
*/
#include <SDL/SDL.h>
#include <stdlib.h>
#include <stdio.h>

#include "draw.h"
#include "surface.h"

//draw a hook
void draw_hook ( Surface &surf, int L )
{
    surf.forward ( L, 1 );
    surf.turn( 90 );
    surf.forward ( L/5, 1 );
    surf.turn ( 90 );
    surf.forward ( L/3, 1 );
}

void draw_arc ( Surface &surf, int L ) {

    for(int i = 0; i < 6; ++i) {
        surf.forward(L,1);
        surf.turn(15);
    }
}

//draw a star pattern
void draw_star( Surface &surf, int L )
{
    for ( int i = 0; i < 5; ++i ) {
        surf.forward( L, 1 );
        surf.turn( 144 );
    }
}

//draw an n-sided regular polygon
void draw_polygon ( Surface &surf, int n, int radius, float rotAngle )
{
    if ( n < 3 ) return;                //bad number of sides
    int cx = surf.getCP().x;
    int cy = surf.getCP().y;

```

```

double angle = rotAngle * 3.14159265 / 180;    //initial angle
double angleInc = 2 * 3.14159265 / n;        //angle increment
surf.moveTo ( ( int) (radius * cos( angle ) + cx),
              ( int) ( radius * sin ( angle ) + cy ) );
for ( int k = 0; k < n; k++ ) {                //repeat n times
    angle += angleInc;
    surf.lineTo ( ( int) (radius * cos( angle ) + cx),
                  ( int) ( radius * sin ( angle ) + cy ) );
}
} //draw_polygon

```

```

//draw rosette with N-sided polygon
void rosette (Surface &surf, int N, int radius )
{
    if ( N < 3 ) return;
    Point pt[N+1];

    int cx = surf.getCP().x;
    int cy = surf.getCP().y;
    double angle = 0;                          //initial angle
    double angleInc = 2 * 3.14159265 / N;        //angle increment
    pt[0] = Point ( ( int) (radius * cos( angle ) + cx),
                    ( int) ( radius * sin ( angle ) + cy ) );
    for ( int k = 1; k < N; k++ ) {                //repeat n times
        angle += angleInc;
        pt[k] = Point ( ( int) (radius * cos( angle ) + cx),
                        ( int) ( radius * sin ( angle ) + cy ) );
    }
    for ( int i = 0; i < N - 1; i++ ) {
        for ( int j = i + 1; j < N; j++ ) {
            surf.moveTo ( pt[i] ); //connect all vertices
            surf.lineTo ( pt[j] );
        }
    }
} //rosette

```

```

int main()
{

```

```

#ifdef ARM
    const int VWIDTH = 640;
    const int VHEIGHT = 480;
#else
    const int VWIDTH = 320;
    const int VHEIGHT = 240;
#endif

```

```

    const Point center ( VWIDTH/2, VHEIGHT/2 ); //center of screen
    Surface surf( VWIDTH, VHEIGHT, (char *) "Draw_demo" );

```

```

surf.clearScreen();           //clear screen
surf.updateSurface();
SDL_Delay ( 1000 );          //dealy one second, just for demo

surf.setBackgroundColor ( 0xe0, 0xe0, 0xe0 );    //set background to grey

//draw a hook
//surf.setColor ( 0xff, 0, 0 );          //using red color
//surf.moveTo ( center );                //move to center of screen
//surf.turnTo ( 0 );                     //points horizontally
//draw_hook ( surf, 180 );

//draw a star
//surf.setColor ( 0, 0xff, 0 );          //using green color
//surf.moveTo ( center.x + 90, center.y );
//surf.turnTo ( 0 );                     //points horizontally
//draw_star ( surf, 50 );

//draw an octagon
surf.setColor ( 0x00, 0, 0 );            //using black color
surf.moveTo ( center.x, center.y - 90 );
// draw_polygon ( surf, 8, 40, 0 );      //draw an octagon
draw_polygon ( surf, 6, 40, 0 );          //draw a hexagon

surf.setColor ( 0xBF, 0, 0 );
surf.moveTo (center.x, center.y - 180 );
draw_polygon ( surf, 32, 40, 0 );        //draw a circle

surf.setColor ( 0xFF, 0, 0 );
surf.moveTo (center.x-60, center.y-10);
surf.turnTo(0);
draw_arc(surf, 60);                      //draw an arc

//draw an 8-sided rosette
//surf.setColor ( 0, 0, 0 );              //using black color
//surf.moveTo ( center.x - 90, center.y + 90 );
//rosette ( surf, 8, 50 );

surf.updateSurface();
SDL_Delay ( 5000 );                    //display 5 seconds before exit
return 1;
}

```