GM LAB TEST - Monday Batch - 11 Oct 2021

- S Vishakan
CSE - C
18 5001 196

Question: **2. Generate a triangle wave and apply appropriate 2D transformations to convert it into a sawtooth wave.**

Code:

```
/*
TRIANGLE WAVE TO SAWTOOTH WAVE USING OPENGL --- Vishakan Subramanian,
CSE-C, 18 5001 196
*/

#include <GL/glew.h>
#include <GL/freeglut.h>
#include <stdio.h>
#include <iostream>
#include <cstring>

const int WINDOW_WIDTH = 800;
const int WINDOW_HEIGHT = 800;
const int FPS = 60;

void initializeDisplay();
void dummyFunction();
void mainLoop(int val);
void drawTriangleWave();
void drawSawtoothWave();

using namespace std;

class Point{    //Wrapper class for 2d point
private:
        float x, y, h;

public:
        Point(){
                h = 1;
        }
```

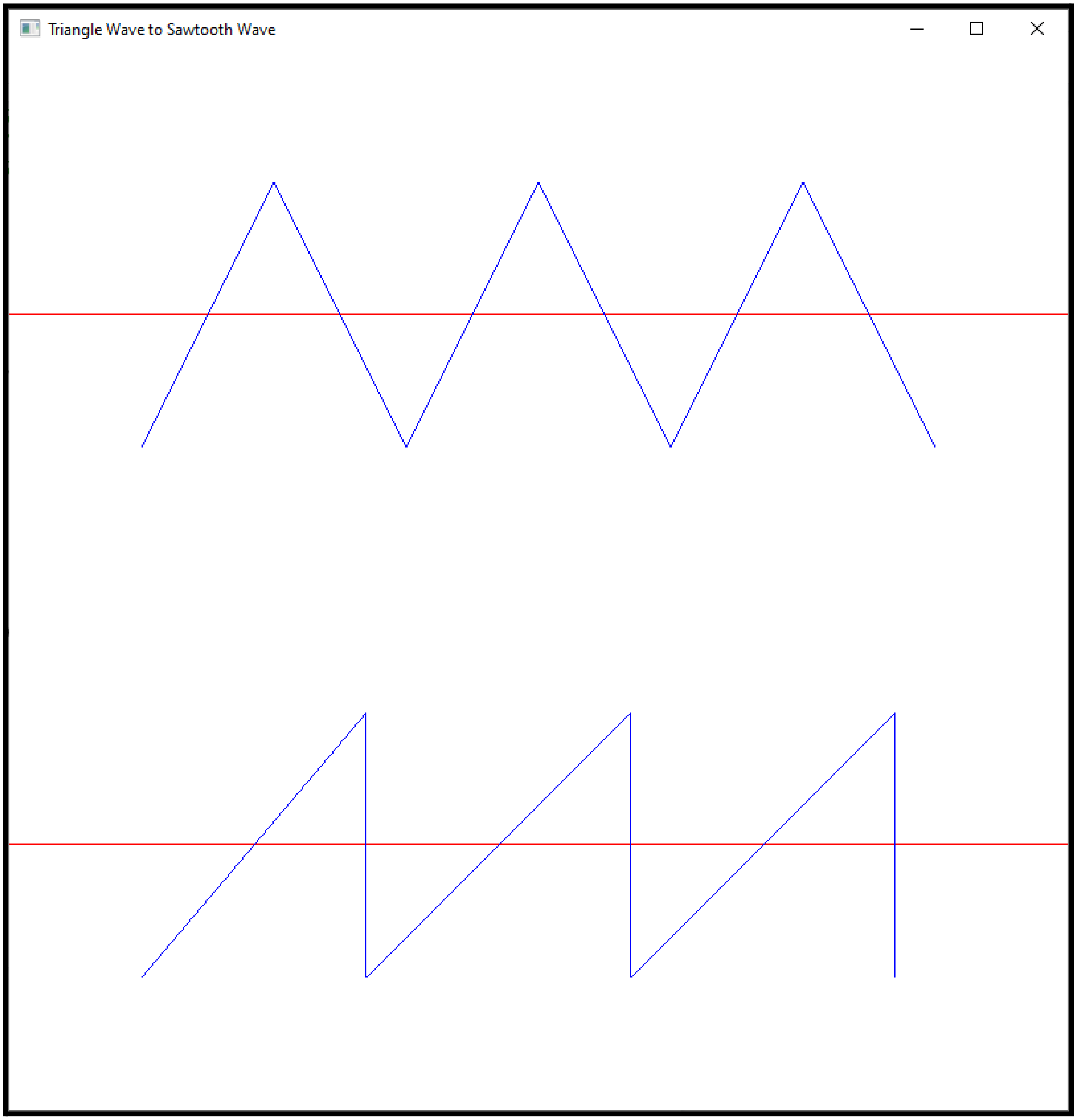
```
    Point(float X, float Y, float H){
        x = X;
        y = Y;
        h = H;
    }

    float getX(){
        return x;
    }

    float getY(){
        return y;
    }

    float getH(){
        return h;
    }

    void setX(float X){
        x = X;
    }

    void setY(float Y){
        y = Y;
    }
};

class Wave{             //Wrapper class for 2d wave
private:
    Point *points;
    int num_points;

public:
    Wave(int no_points){
        num_points = no_points;
        points = new Point[num_points];
    }

    void setPoint(int i, Point p){
        points[i] = p;
    }

    void setPoint(int i, float x, float y){
        points[i].setX(x);
        points[i].setY(y);
}

Point getPoint(int i){
        return points[i];
}

void drawWave(){
        //To plot a given wave

        glColor3d(0, 0, 1);
        glBegin(GL_LINE_STRIP);

        for(int i = 0; i < num_points; i++){
                glVertex2f(points[i].getX(), points[i].getY());
        }

        glEnd();
        glFlush();
}

Wave toSawtooth(){
        //To convert a given triangle wave to sawtooth wave

        float shear = 0.10;

        float shearMatrix[3][3] =
        {{1, shear, 0},
        {0, 1, 0},
        {0, 0, 1}};

        Wave sawtoothWave(num_points);

        float prev = 0;

        cout << "\nSawtooth Wave coordinates: " << endl;

        for(int i = 0; i < num_points; i++){
                //Use 2D transformation to convert the points

                Point p = points[i];

                float values[3] = {p.getX(), p.getY(), p.getH()};
```

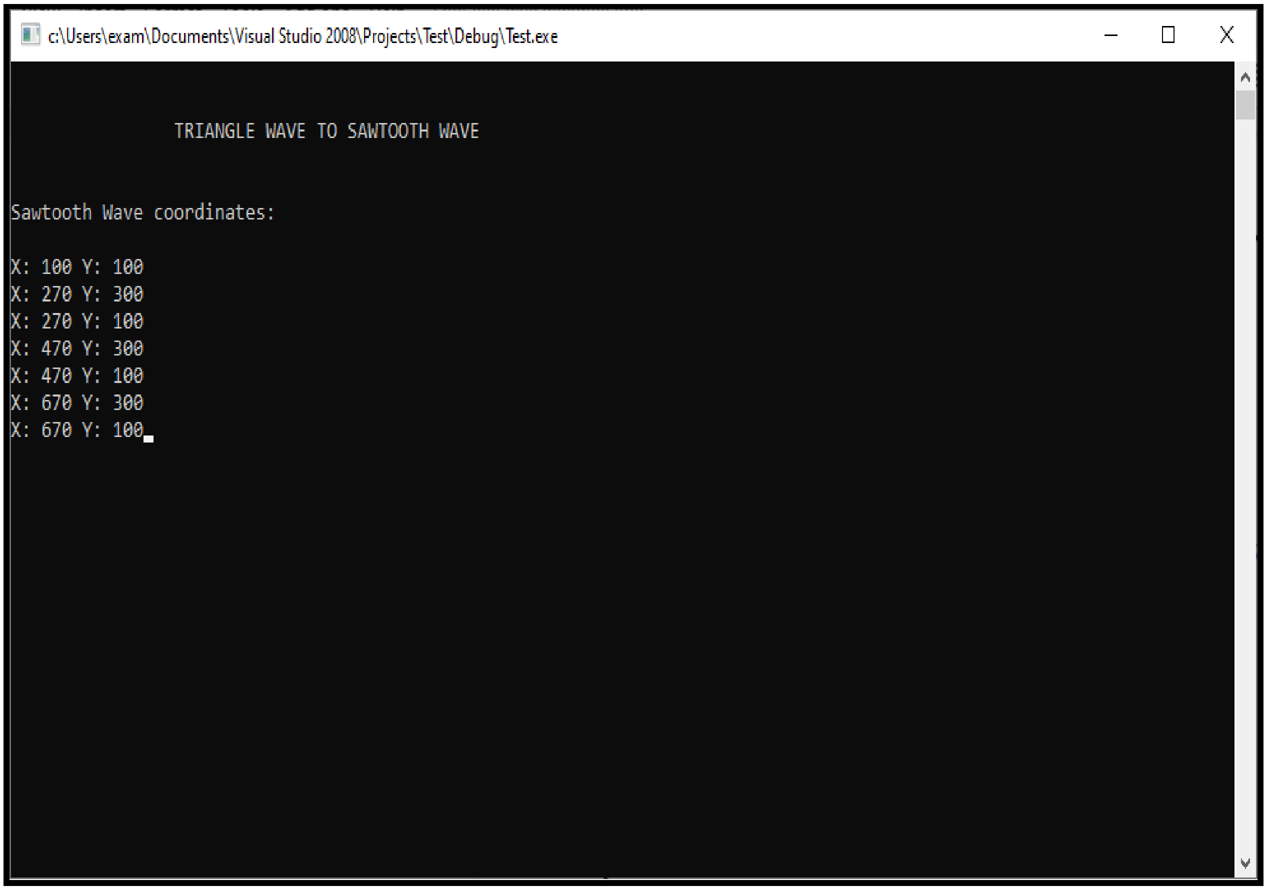
```
            for(int j = 0; j < 3; j++){
                if(i == 0){
                    //Don't shear the first point (lows)
                    break;
                }

                else if(i % 2 == 1){
                    //Shear the odd points (highs)
                    values[j] = shearMatrix[j][0] * p.getX() +
                            shearMatrix[j][1] * p.getY() +
                            shearMatrix[j][2] * p.getH();
                }

                else{
                    //Maintain the same X of the odd shear points (highs) alone
                    //Do not shear the even points (lows)
                    values[0] = prev;
                    break;
                }
            }

            if(i % 2 == 1){
            //Note the previous high's X coordinate for the next low's X coordinate
                prev = values[0];
            }

//Translate down the Y coordinate for all points to fit in the axis drawn from (0, 200) to (800, 200)
            values[1] -= 400;       // 600 - 200 = 400


            cout << "\nX: " << values[0] << " Y: " << values[1];

            Point pDash(values[0]/p.getH(), values[1]/p.getH(), p.getH());
            sawtoothWave.setPoint(i, pDash);
        }

        return sawtoothWave;
    }

};

//Objects for triangle wave and sawtooth wave
```

```
Wave triangle(7);
Wave sawtooth(7);

int main(int argc, char **argv){
	glutInit(&argc, argv);
	glutInitWindowPosition(100, 100);
	glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT);
	glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
	glutCreateWindow("Triangle Wave to Sawtooth Wave");

	cout << "\n\n\t\tTRIANGLE WAVE TO SAWTOOTH WAVE\n\n";

	initializeDisplay();
	glutDisplayFunc(dummyFunction);
	glutTimerFunc(1000/FPS, mainLoop, 0);
	glutMainLoop();

	return 1;
}


void initializeDisplay(){
	//Initialize display parameters

	glClearColor(1, 1, 1, 0);
	gluOrtho2D(0, WINDOW_WIDTH, 0, WINDOW_HEIGHT);
	glMatrixMode(GL_PROJECTION);

	glClear(GL_COLOR_BUFFER_BIT);

	glEnable(GL_BLEND);
	glDepthMask(GL_FALSE);
	glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
}

void dummyFunction(){
	//Placeholder function
}

void mainLoop(int val){
	//Program Driver Function
	drawTriangleWave();
	drawSawtoothWave();
}
```

```
void drawTriangleWave(){
	//To draw a triangle wave

	//Axis
	glColor3d(1, 0, 0);
	glBegin(GL_LINES);
	glVertex2d(0, 600);
	glVertex2d(800, 600);
	glEnd();

	//Triangle Wave
	triangle.setPoint(0, 100, 500);
	triangle.setPoint(1, 200, 700);
	triangle.setPoint(2, 300, 500);
	triangle.setPoint(3, 400, 700);
	triangle.setPoint(4, 500, 500);
	triangle.setPoint(5, 600, 700);
	triangle.setPoint(6, 700, 500);

	triangle.drawWave();

	glFlush();
}

void drawSawtoothWave(){
	//To draw a sawtooth wave from a given triangle wave

	//Axis
	glColor3d(1, 0, 0);
	glBegin(GL_LINES);
	glVertex2d(0, 200);
	glVertex2d(800, 200);
	glEnd();

	//Sawtooth Wave
	sawtooth = triangle.toSawtooth();
	sawtooth.drawWave();

	glFlush();
}
```

Output Snippets:





c:\Users\exam\Documents\Visual Studio 2008\Projects\Test\Debug\Test.exe

```
                TRIANGLE WAVE TO SAWTOOTH WAVE


Sawtooth Wave coordinates:

X: 100 Y: 100
X: 270 Y: 300
X: 270 Y: 100
X: 470 Y: 300
X: 470 Y: 100
X: 670 Y: 300
X: 670 Y: 100
```