Make fun with Lines

Here in this mini project I am making a windows form application in which user can provide some inputs like- no_of_lines, angle, increment, length.

When user is clicking on Go button a pattern will be drawn inside the windows form. I created 2 panels inside the form in panel 1 all the inputs and button is there and background colour is black and in the panel 2(canvas) the pattern/shape will be drawn.

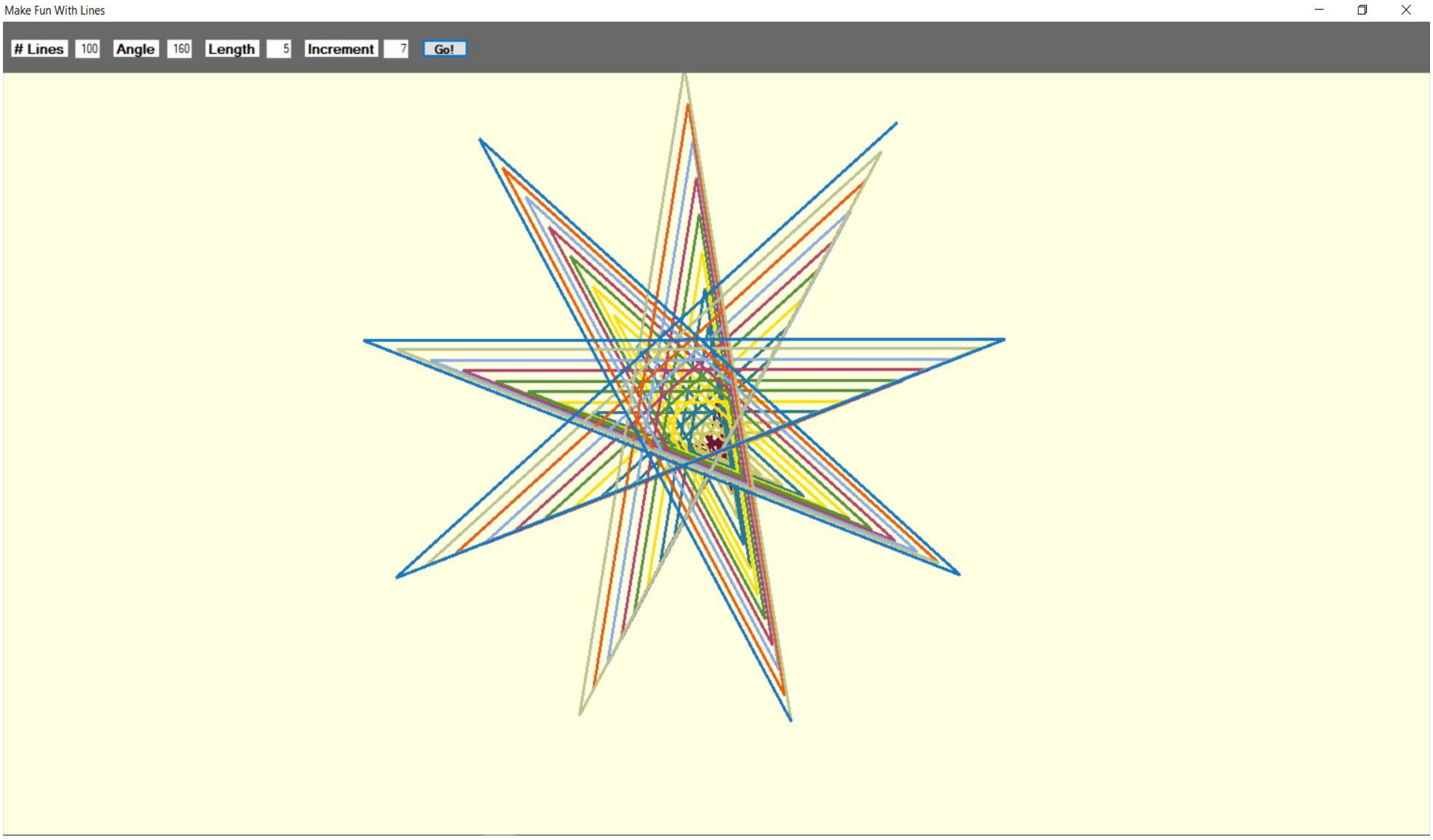
I used random function to get different colour of each line that has to be drawn.

The basic idea behind this application is every time we calculate start_x and start_y and by this, calculate end_x, end_y. After each line that has been drawn we change the value of start_x and start_y with end_x and end_y respectively.

So, after varying some angles and no_of_lines, we can get some cool shapes.

After changing the input from the user, the following shapes can be get:

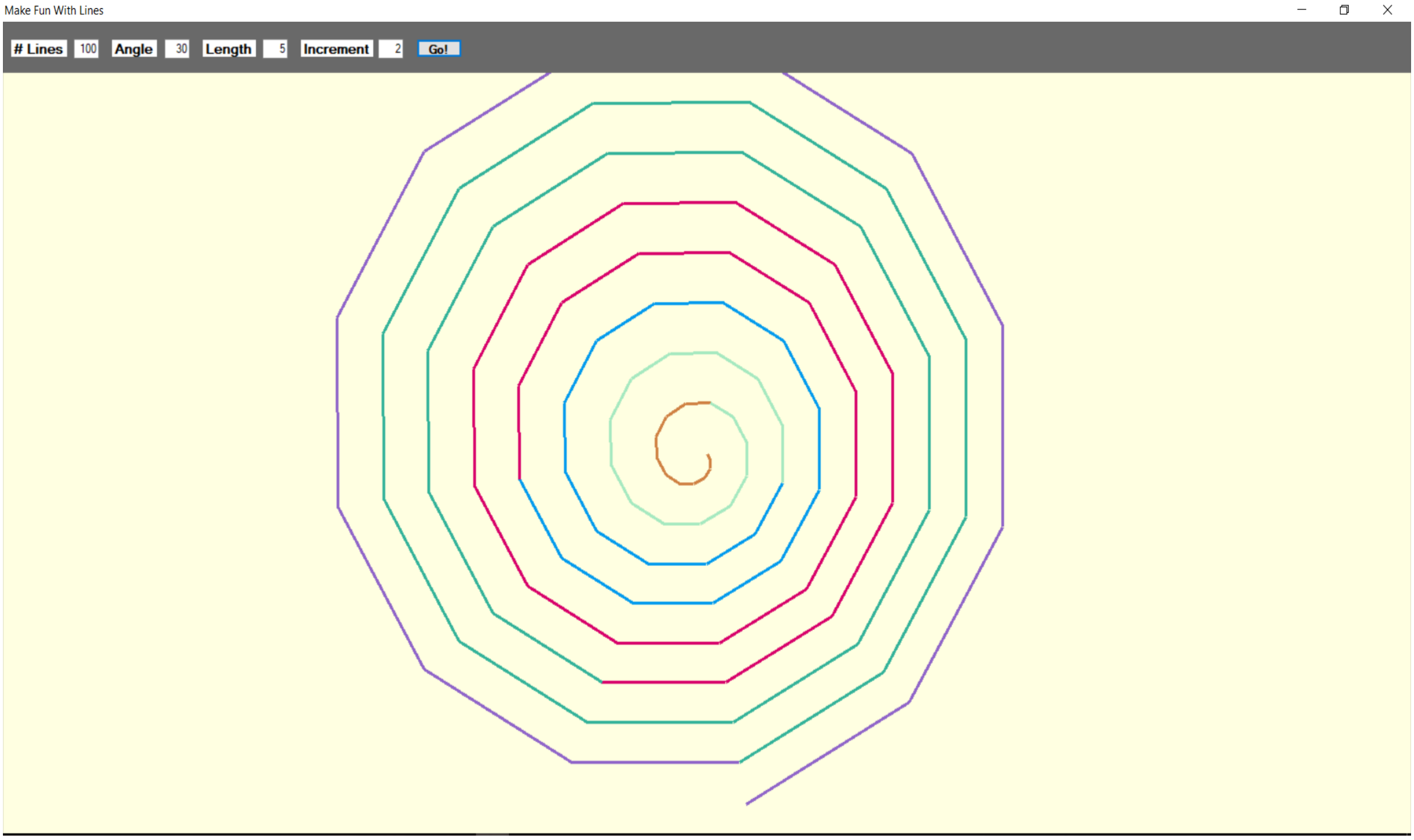
If Lines = 100, angle = 160, Length = 5, Increment = 7



Make Fun With Lines

Lines 100 Angle 160 Length 5 Increment 7

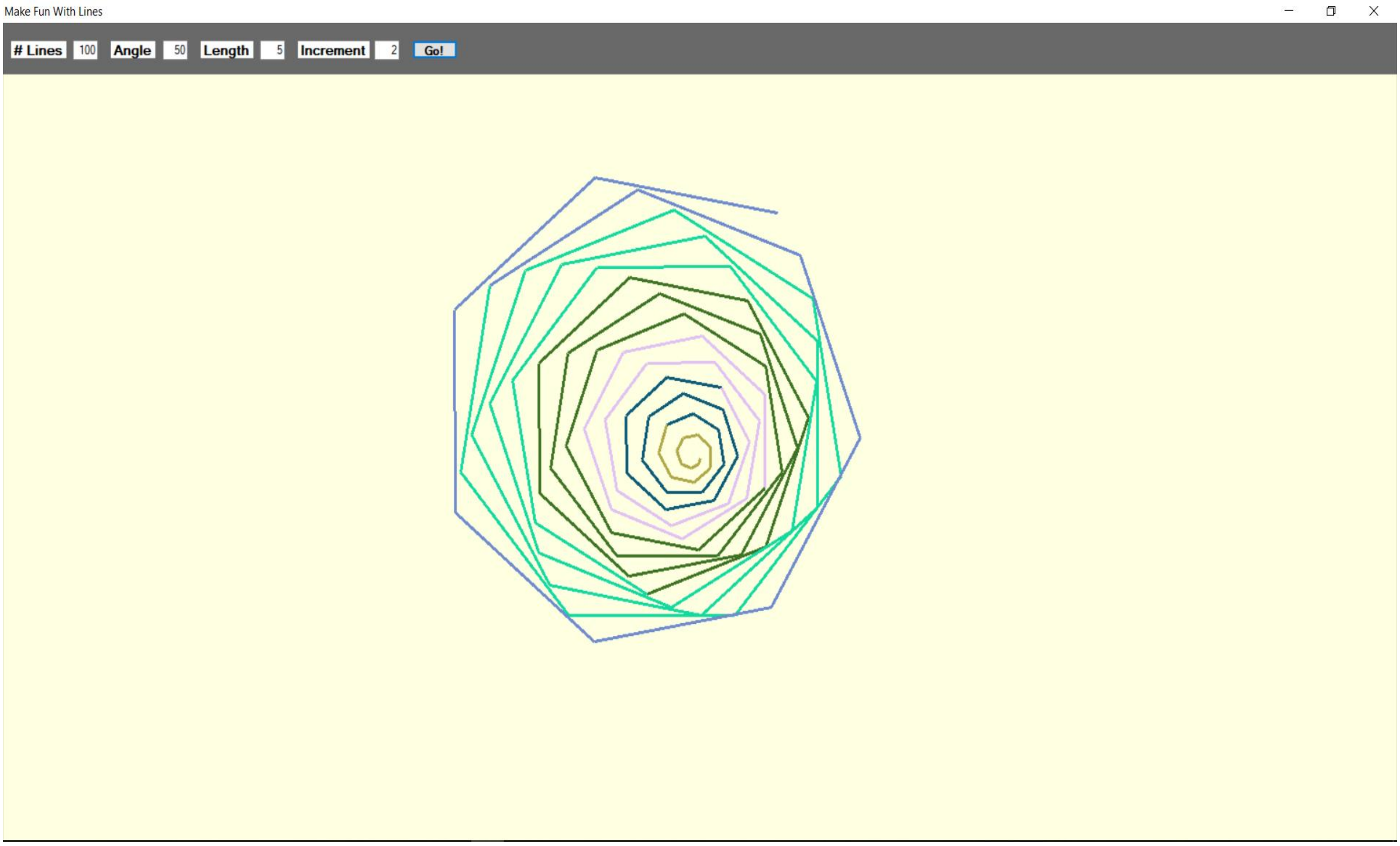
If Lines = 100, angle = 30, Length = 5, Increment = 2



Make Fun With Lines

Lines 100 Angle 30 Length 5 Increment 2

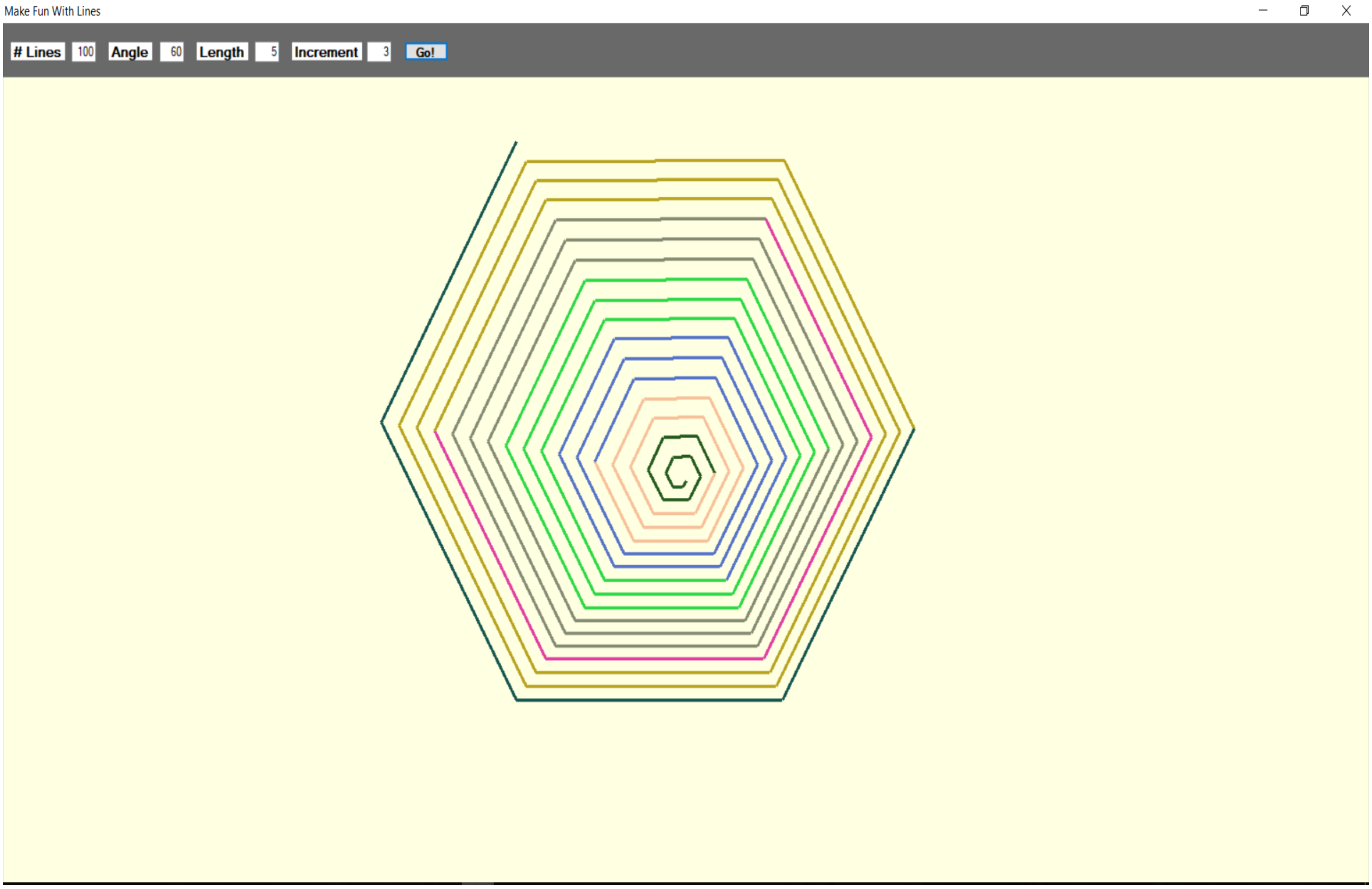
If Lines = 100, angle = 50, Length = 5, Increment = 2



Make Fun With Lines

Lines 100 Angle 50 Length 5 Increment 2

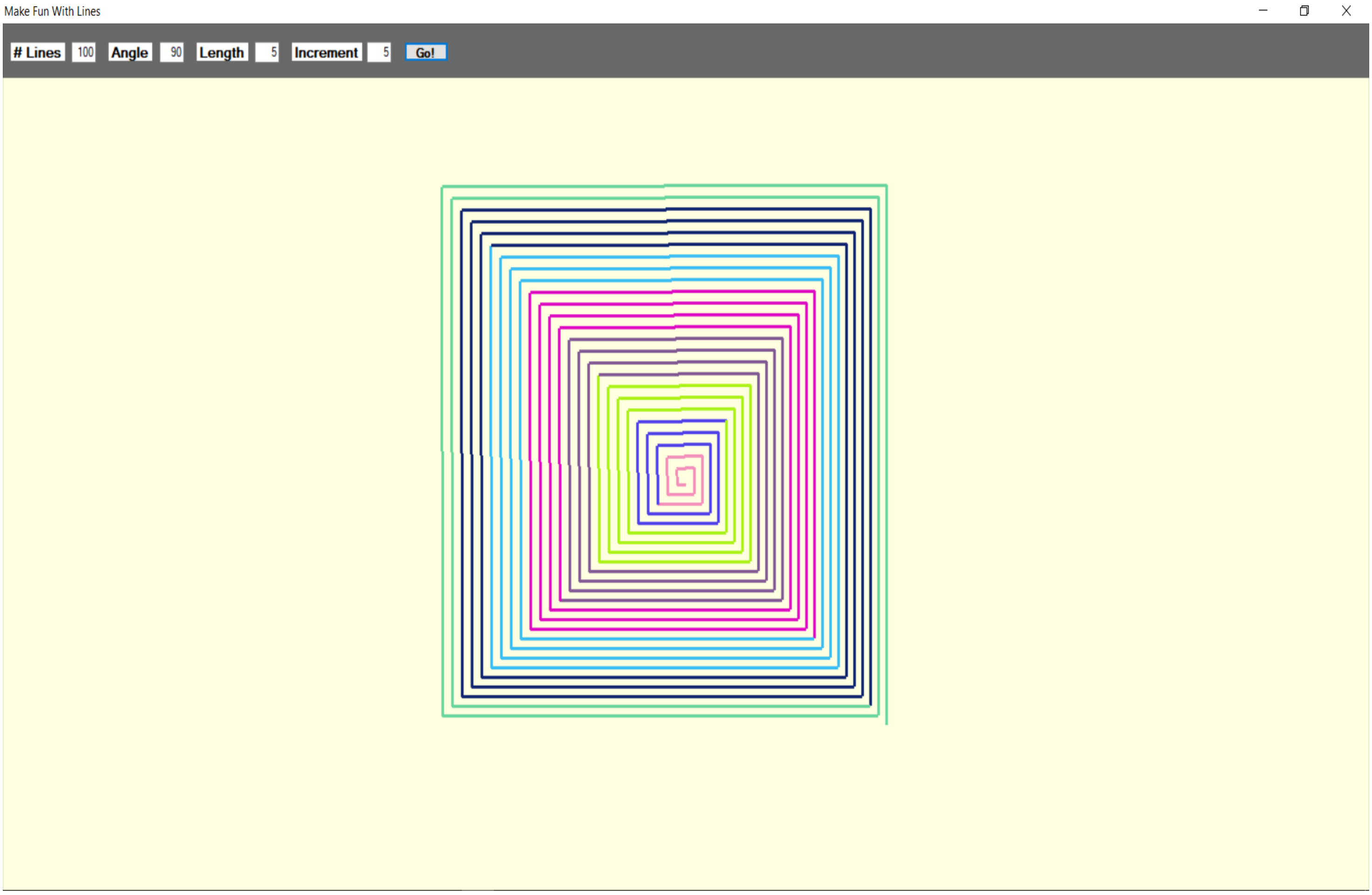
If Lines = 100, angle = 60, Length = 5, Increment = 3



Make Fun With Lines

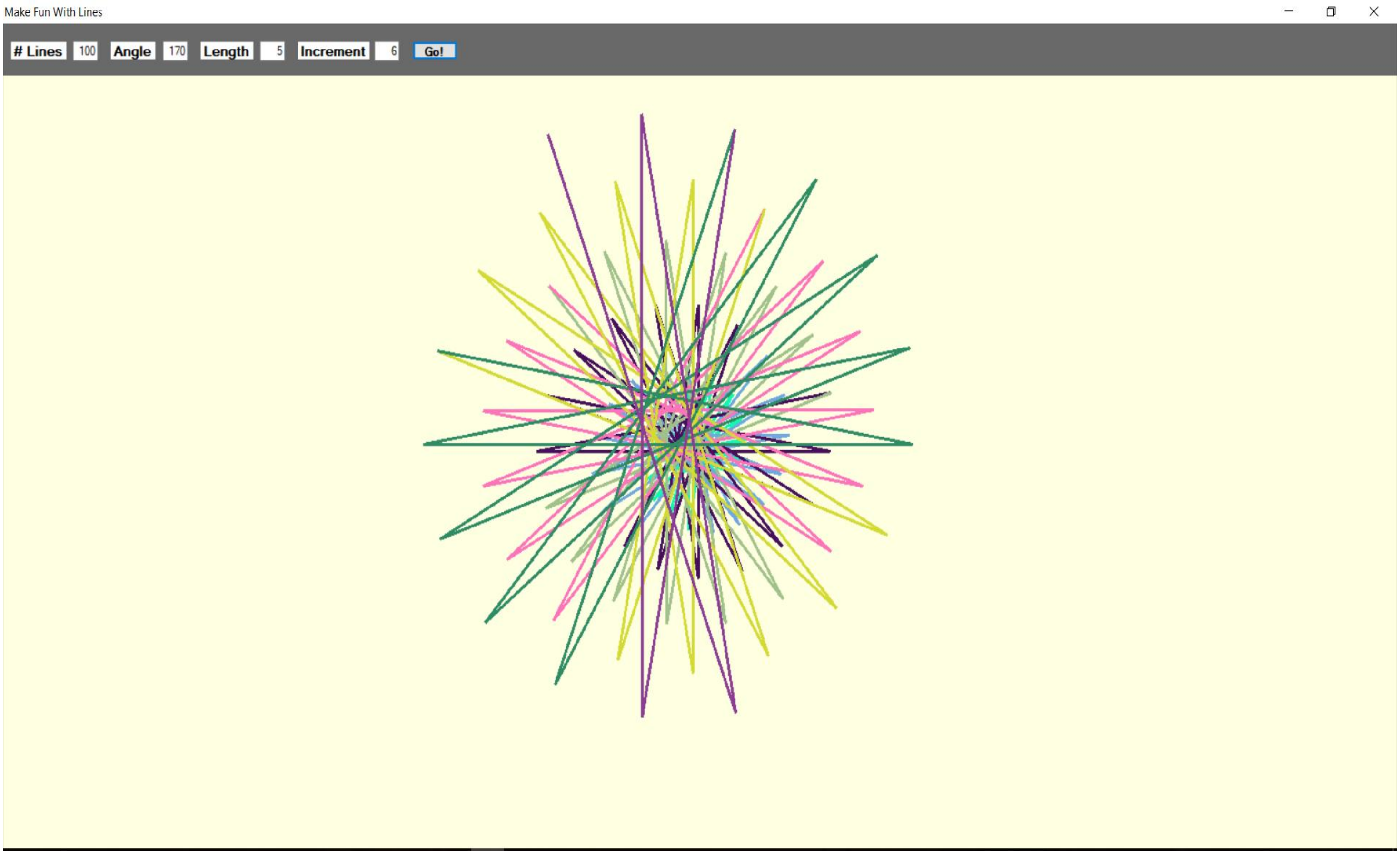
Lines 100 Angle 60 Length 5 Increment 3

If Lines = 100, angle = 90, Length = 5, Increment = 5



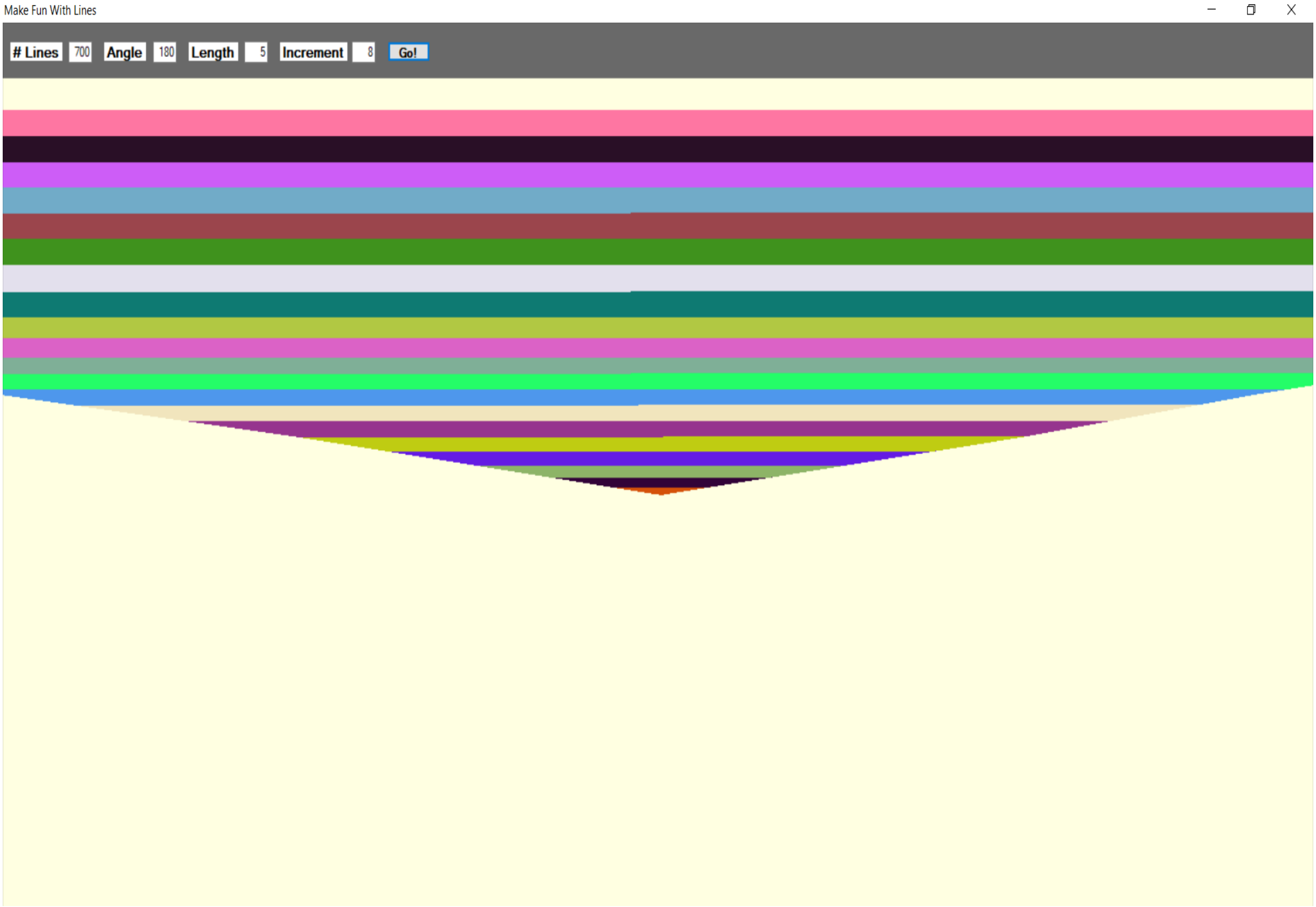
Lines 100 Angle 90 Length 5 Increment 5

If Lines = 100, angle = 170, Length = 5, Increment = 6



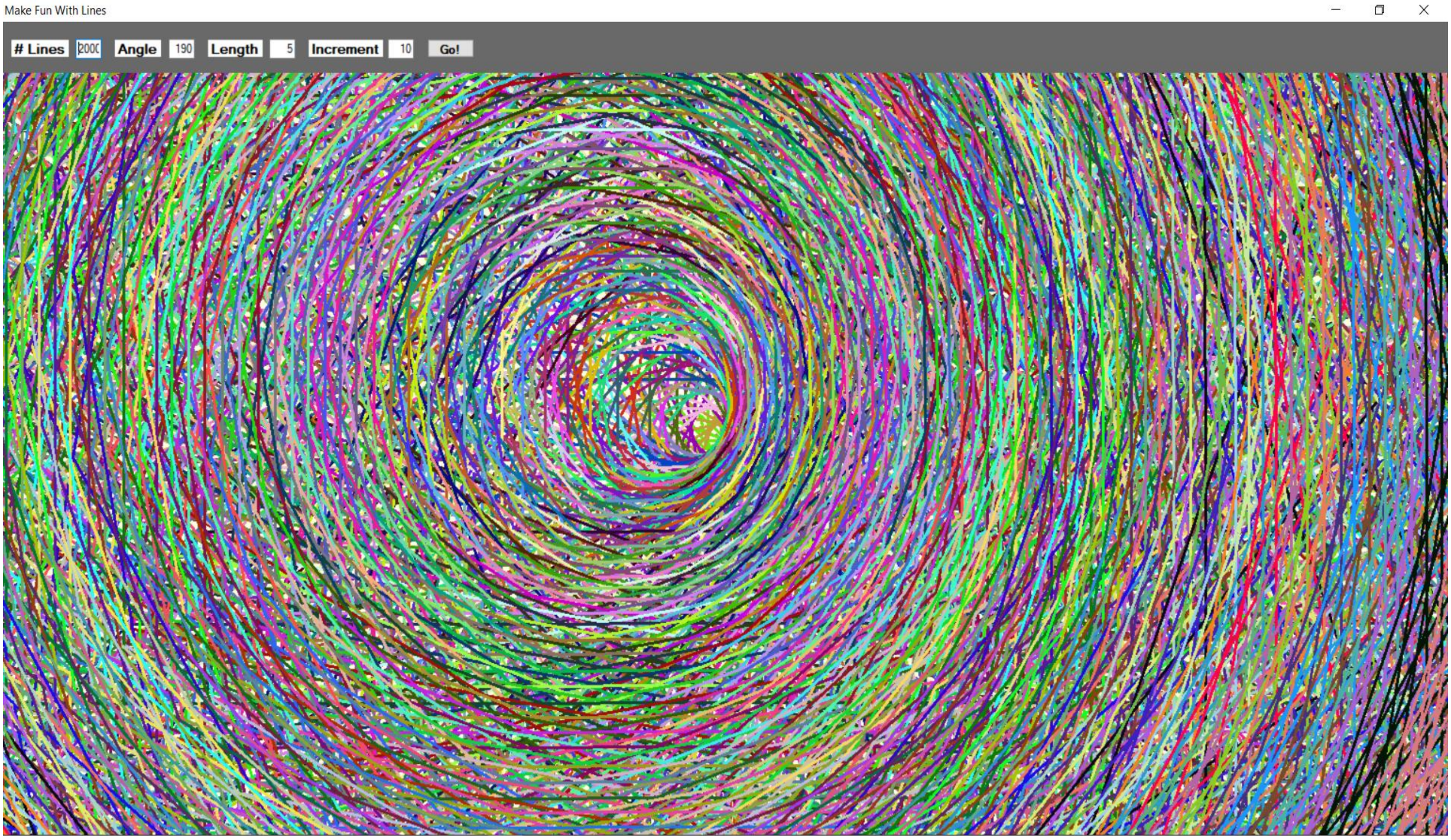
Lines 100 Angle 170 Length 5 Increment 6

If Lines = 700, angle = 180, Length = 5, Increment = 8



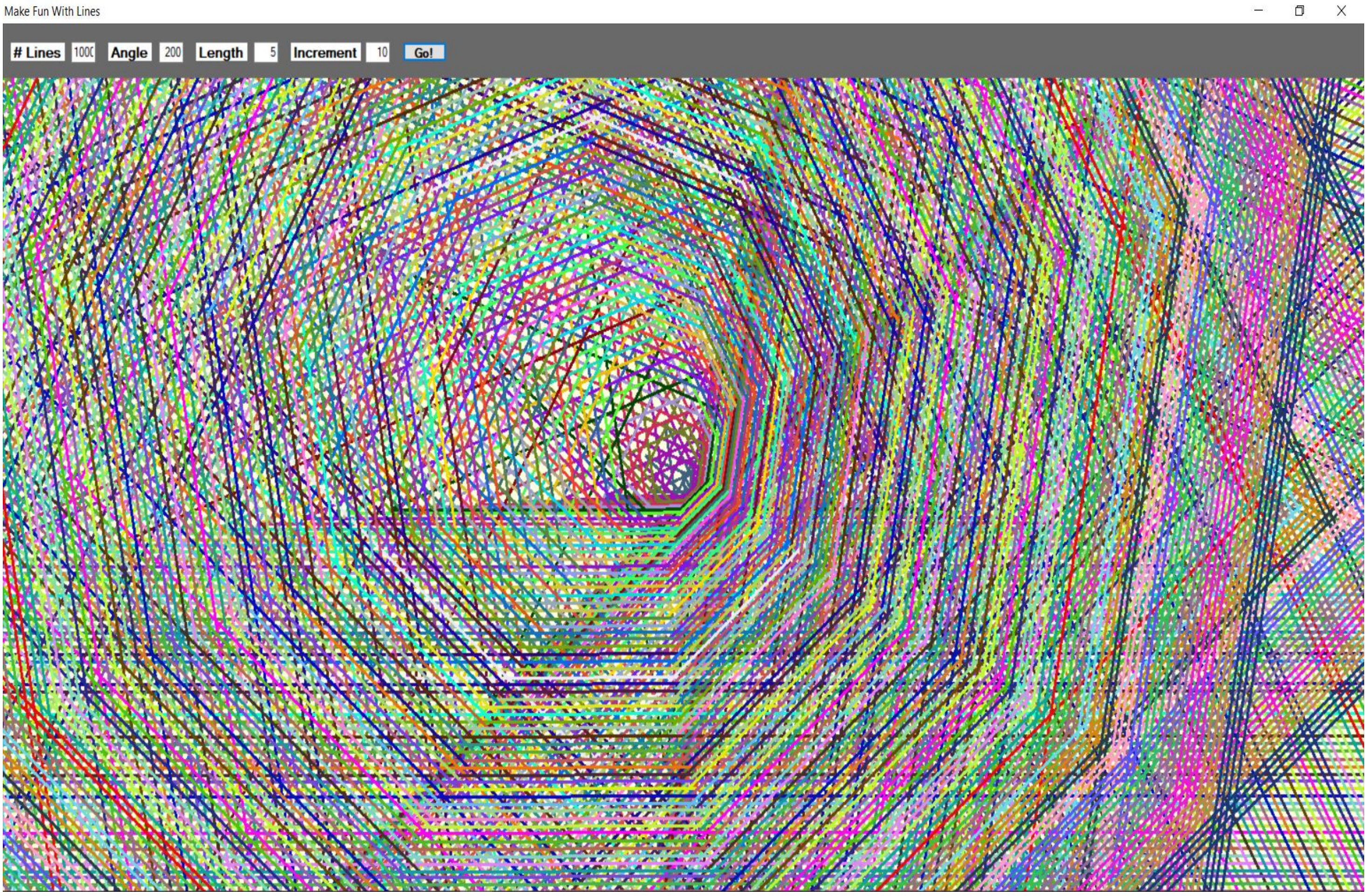
Lines 700 Angle 180 Length 5 Increment 8

If Lines = 20000, angle = 190, Length = 5, Increment = 10

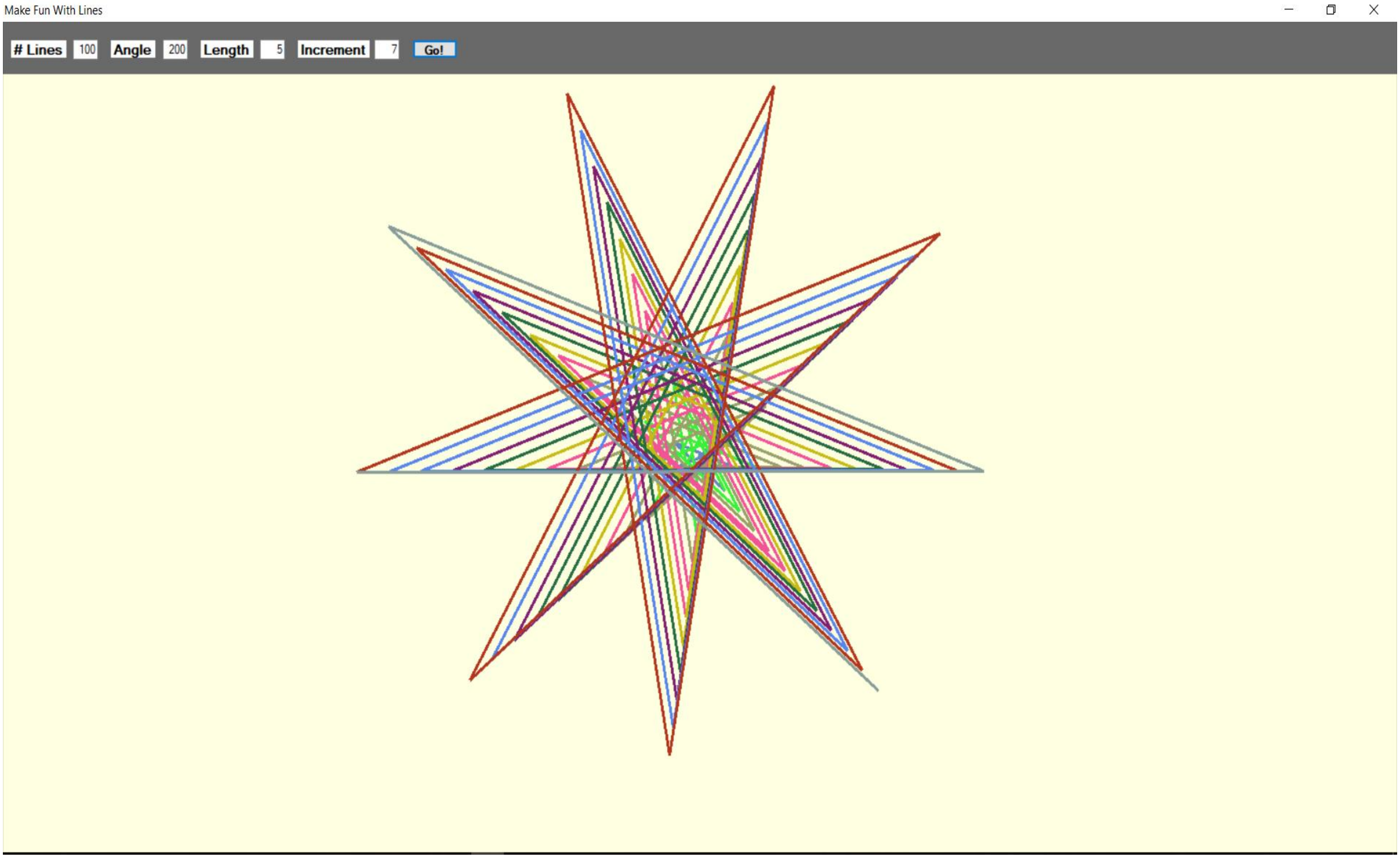


Lines 2000 Angle 190 Length 5 Increment 10

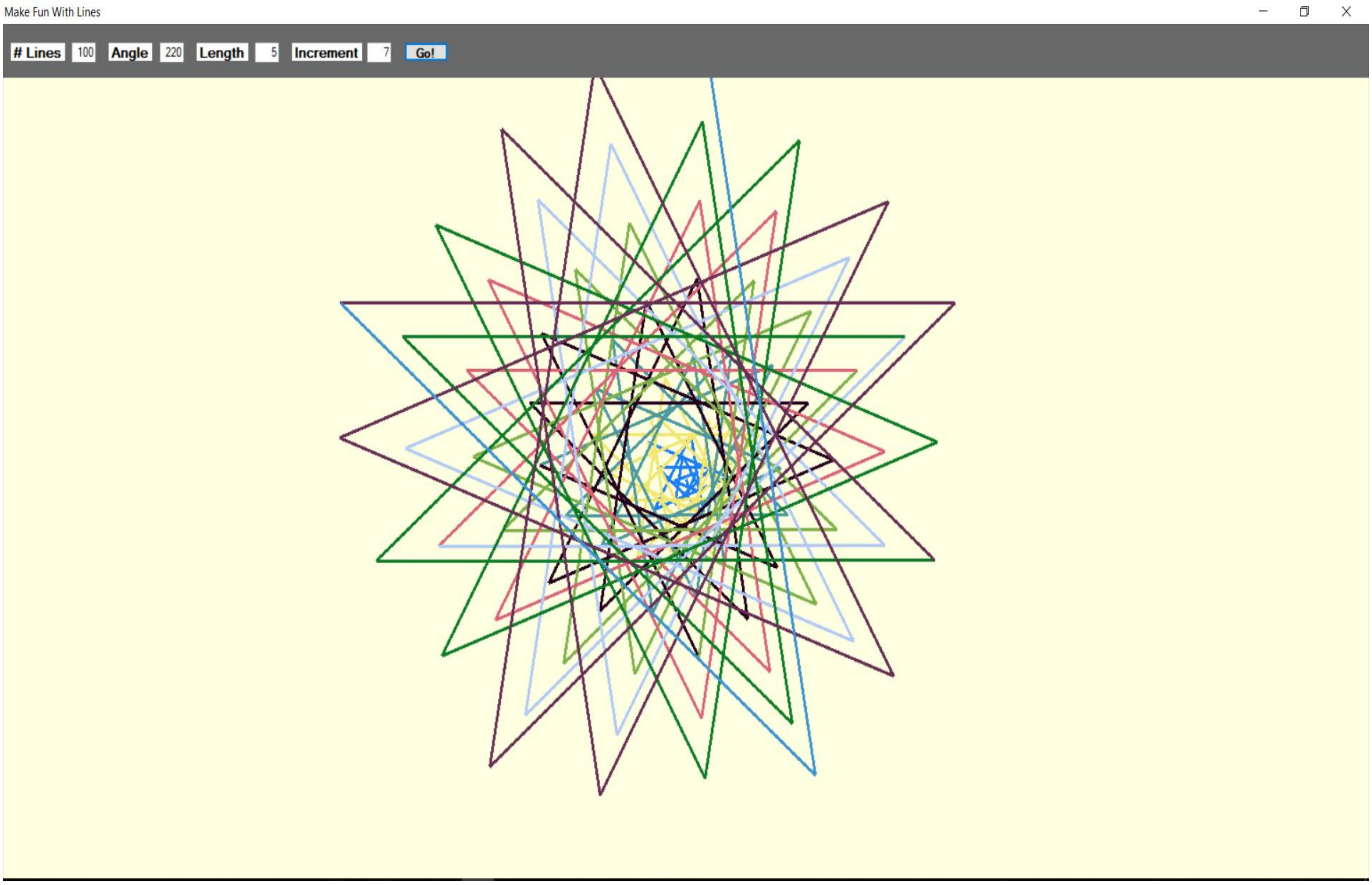
If Lines = 1000, angle = 200, Length = 5, Increment = 10



If Lines = 100, angle = 200, Length = 5, Increment = 7

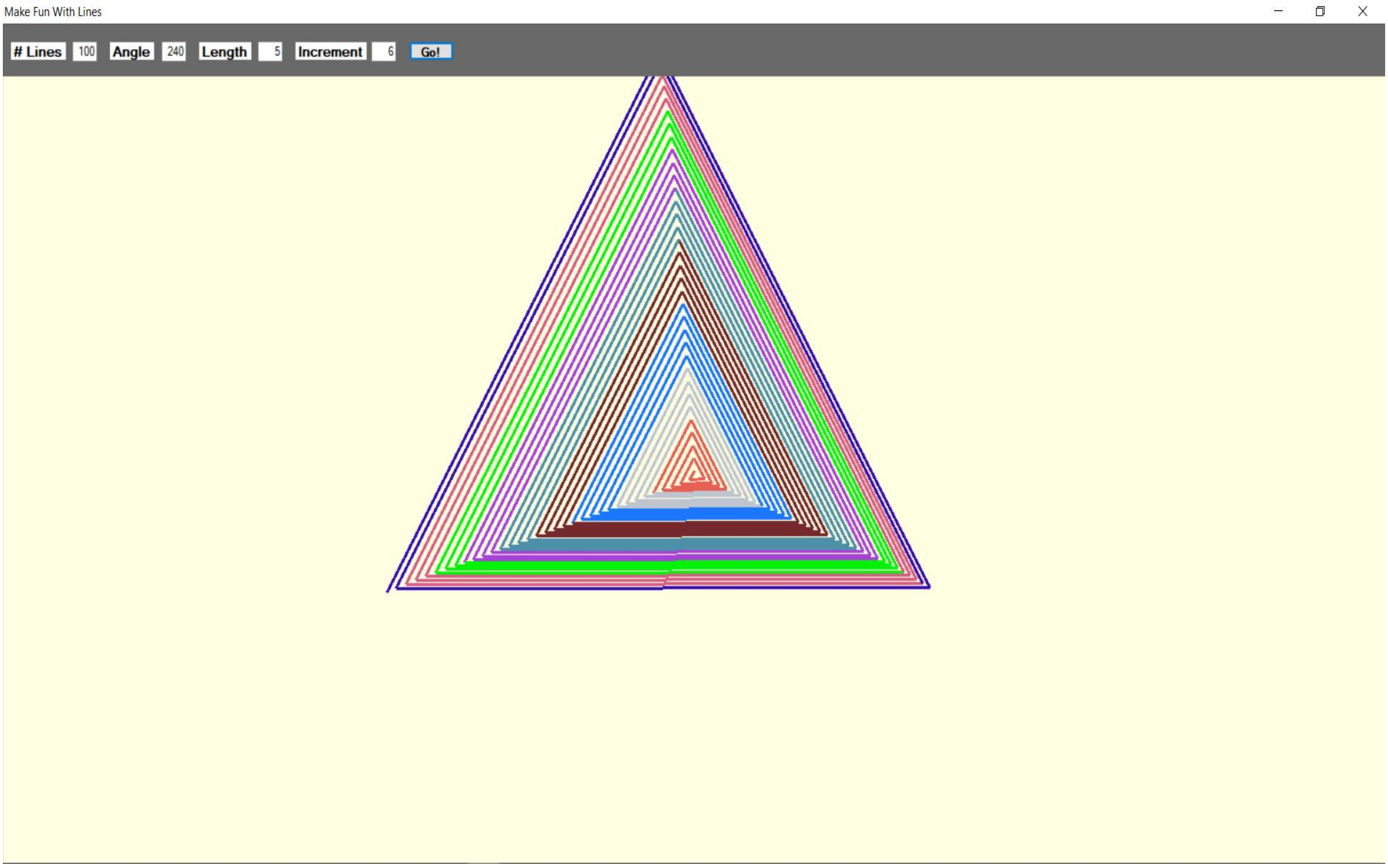


If Lines = 100, angle = 220, Length = 5, Increment = 7



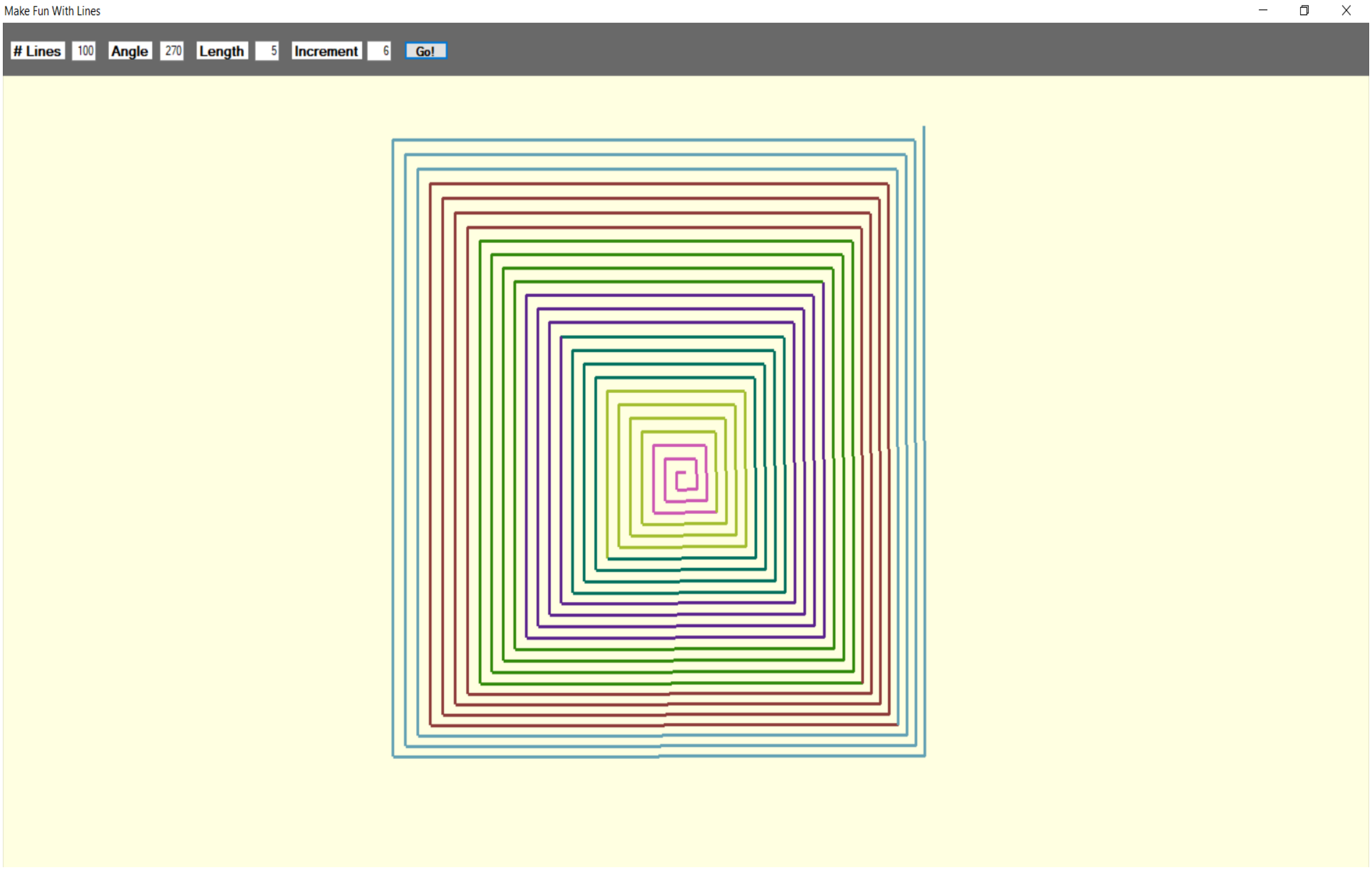
Lines 100 Angle 220 Length 5 Increment 7

If Lines = 100, angle = 240, Length = 5, Increment = 6



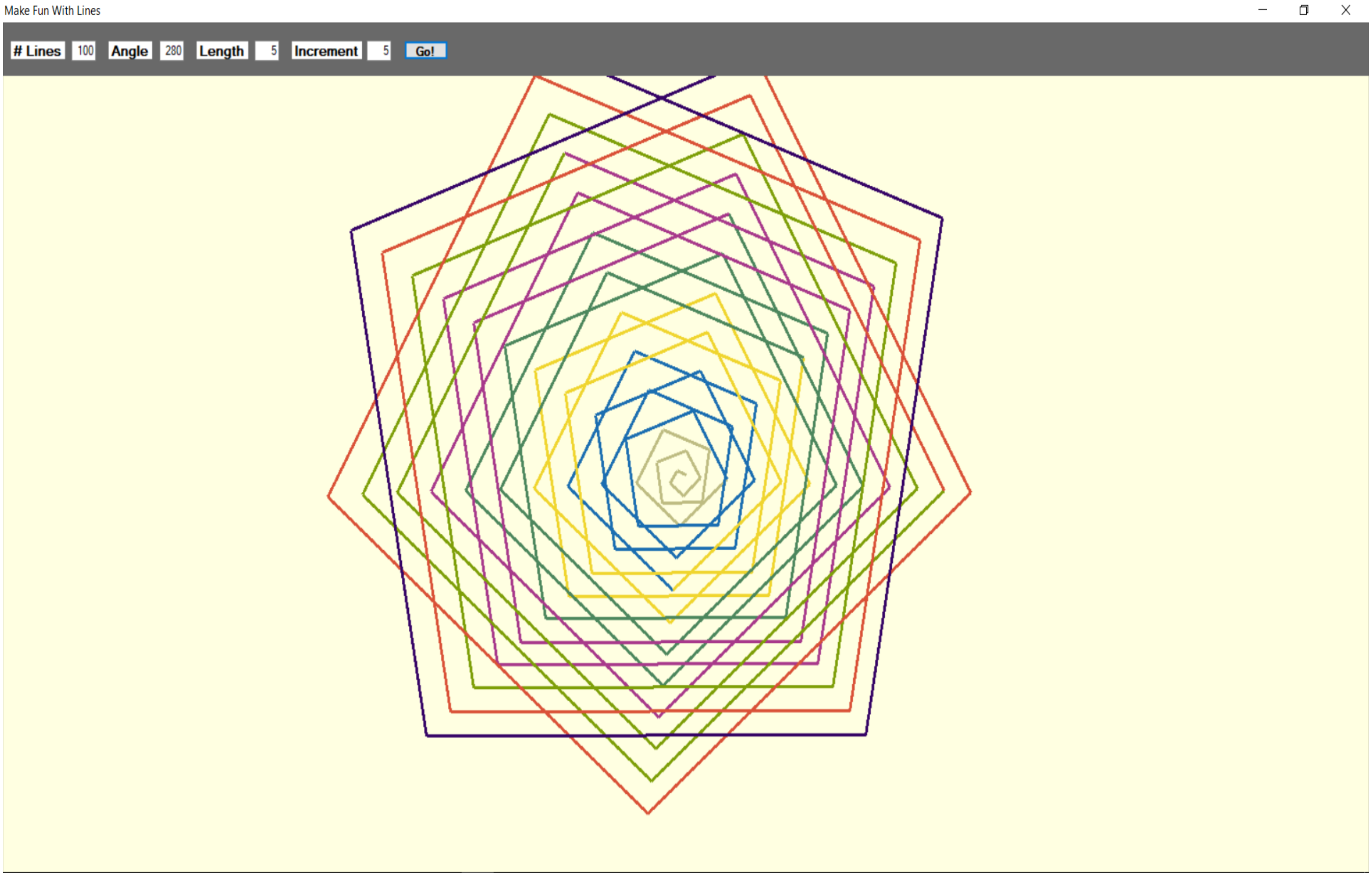
Lines 100 Angle 240 Length 5 Increment 6

If Lines = 100, angle = 270, Length = 5, Increment = 6



Lines 100 Angle 270 Length 5 Increment 6

If Lines = 100, angle = 280, Length = 5, Increment = 5

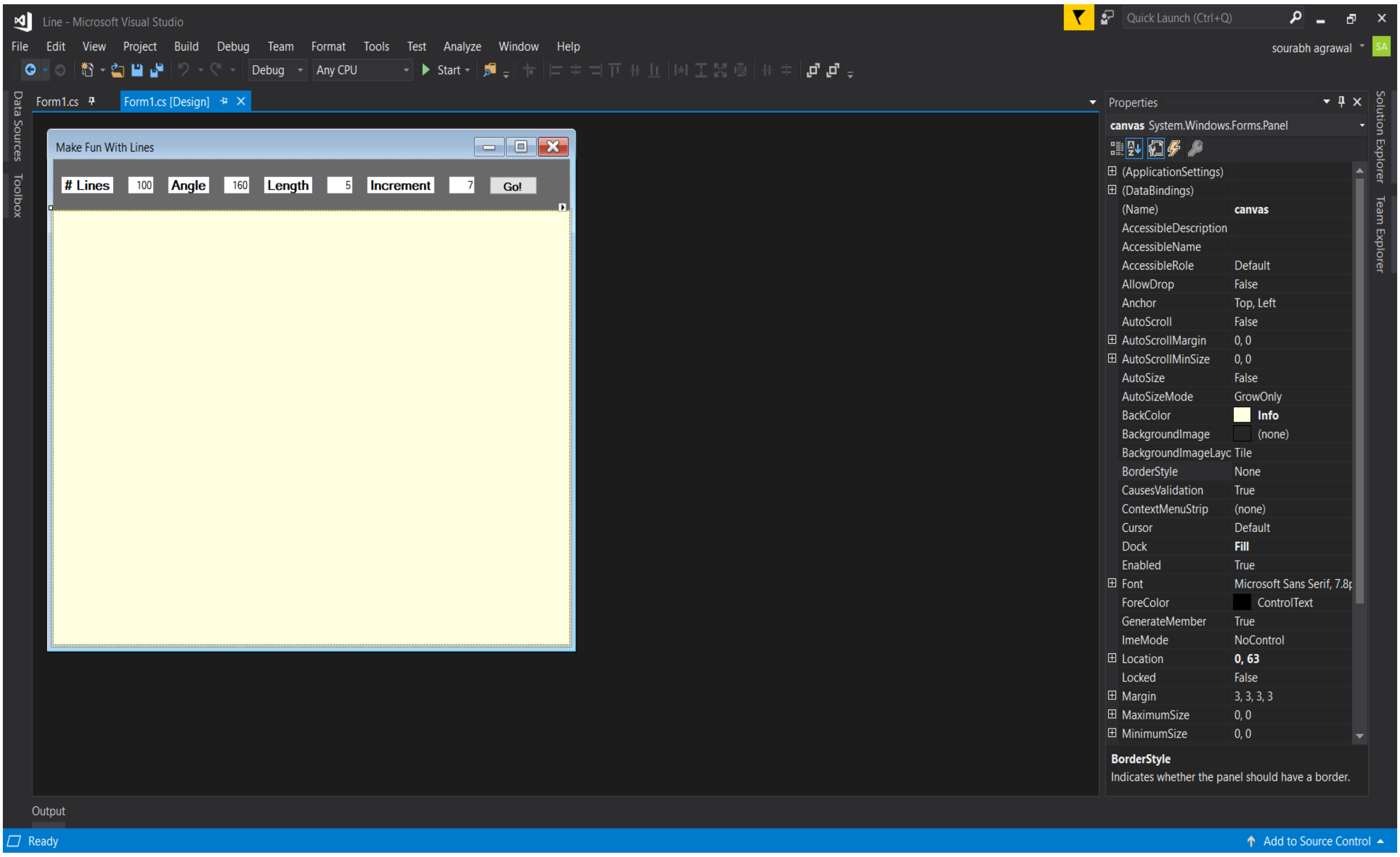


Lines 100 Angle 280 Length 5 Increment 5

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Line
{
    public partial class Form1 : Form
    {
        Pen p = new Pen(Color.Green);
        Graphics g = null;

        static int start_x, start_y;
        static int end_x, end_y;

        static int my_length = 0;
        static int my_increment = 0;
        static int my_angle = 0;
        static int no_lines = 0;

        public Form1()
        {
            InitializeComponent();
            start_x = canvas.Width / 2;
            start_y = canvas.Height / 2;
        }

        private void canvas_Paint(object sender, PaintEventArgs e)
        {
            no_lines = Int32.Parse(number_of_lines.Text);
            p.Width = 3;
            my_length = Int32.Parse(length.Text);
            g = canvas.CreateGraphics();
            for(int i = 0; i < no_lines; i++)
            {
                drawLine();
            }

        }

        private void drawLine() {
            Random randomGen = new Random();
            p.Color = Color.FromArgb(randomGen.Next(255), randomGen.Next(255),
randomGen.Next(255));

            my_angle += Int32.Parse(angle.Text);
            my_length += Int32.Parse(increment.Text);
            end_x = (int)(start_x + Math.Cos(my_angle * 0.017453292519) * my_length);
            end_y = (int)(start_y + Math.Sin(my_angle * 0.017453292519) * my_length);

            Point[] points = {
                new Point(start_x , start_y),
                new Point(end_x , end_y)
            };
            g.DrawLines(p, points);

            start_x = end_x;
            start_y = end_y;


        }
        private void button1_Click(object sender, EventArgs e)
        {
            my_length = Int32.Parse(length.Text);
            my_increment = Int32.Parse(increment.Text);
            my_angle = Int32.Parse(angle.Text);
            start_x = canvas.Width / 2;
            start_y = canvas.Height / 2;
            no_lines = Int32.Parse(number_of_lines.Text);
            canvas.Refresh();
        }
    }
}
```

Form design:



canvas System.Windows.Forms.Panel

⊞ (ApplicationSettings)	
⊞ (DataBindings)	
(Name)	**canvas**
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
AutoScroll	False
⊞ AutoScrollMargin	0, 0
⊞ AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
BackColor	**Info**
BackgroundImage	(none)
BackgroundImageLayc	Tile
BorderStyle	None
CausesValidation	True
ContextMenuStrip	(none)
Cursor	Default
Dock	**Fill**
Enabled	True
⊞ Font	Microsoft Sans Serif, 7.8p
ForeColor	ControlText
GenerateMember	True
ImeMode	NoControl
⊞ Location	**0, 63**
Locked	False
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0
⊞ MinimumSize	0, 0

BorderStyle
Indicates whether the panel should have a border.