



MOD006127

**COMPUTER GRAPHICS
PROGRAMMING**

**PORTFOLIO OF
IN-CLASS EXERCISES**

Prepared for: Ashim Chakraborty

Prepared by: SID_19-03-999

Due Date: 10th of June 2022

Word Count: 400 words

Table of Contents

INSTRUCTIONS	2
WEEK 1 – CIRCLE IN SQUARE	2
WEEK 2 – TRIANGLES.....	2
WEEK 3 – XOR.....	3
WEEK 4 – DOUBLE BUFFERING.....	4
WEEK 5 – LINE AND POINTS	4
CONCLUSION	5

Instructions

Open folder (SID1903999/Portfolio – Weekly Exercises/) where you will be presented with a choice of 7 Folders as demonstrated in with Fig6, below.

Name	Date modified	Type	Size
1. CircleInSquare	30/05/2022 18:56	File folder	
2. Triangles	30/05/2022 18:52	File folder	
3. XOR	30/05/2022 17:30	File folder	
4. DoubleBuffering	30/05/2022 19:14	File folder	
5. LineandPoints	16/05/2022 23:59	File folder	
6. MatrixMultiplier	16/05/2022 23:59	File folder	
7. Rotation	16/05/2022 23:59	File folder	

Fig6: Weekly Exercises

Once opened each folder will have the source code (relevant .cs files) and the executable file (a single .exe file) for each weeks in-class exercise.

Week 1 – Circle In Square

The Program was modified, to draw a circle inside a square **as demonstrated in Fig3, below**. The circle was drawn with the help of drawing an Ellipse, which inherited the same coordinates used earlier for the rectangle. You'll find the new line of code commented as “// Draw a circle within a square” stored inside the “SimpleDrawing.cs” file.

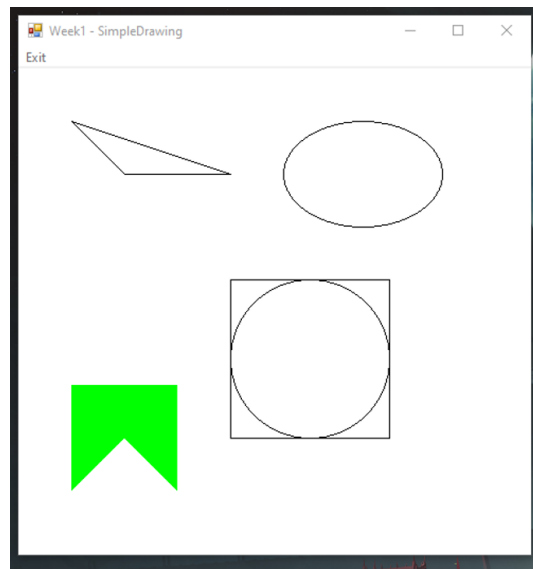


Fig3 – Week 1 Portfolio exercise “SimpleDrawing”

Week 2 – Triangles

Existent source code was downloaded from Canvas and added to the project. The source code was modified, to draw the triangle shape demonstrated in Fig4, below. Starting co-ordinates for the largest triangle were (100,100), (500,100), (300,446). A

designated for loop was created to draw up to 3 triangles, each new triangle is created from the midpoint of the sides of the previous one. This is achieved, with the help of midPoints() method which calculates new co-ordinates with the help of math.

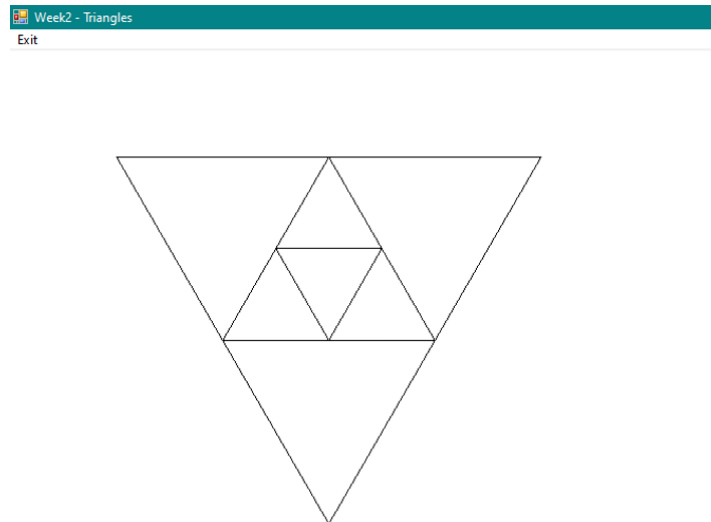


Fig4: Week 2 Portfolio exercise “Triangles”

Week 3 – XOR

The existent source code from canvas was expanded to show a small moving square making a transition from top to bottom of the form displaying a fill colour which is as a result of continuously filling the small square with red in XOR mode.

A screengrab of the program in action is demonstrated in Fig5, below.

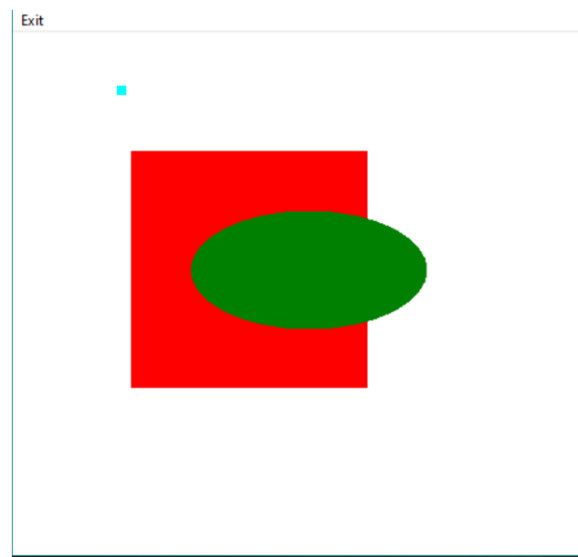


Fig5: Week 3 Portfolio exercise “XORDrawing”

Week 4 – Double Buffering

The existent source code from canvas was expanded to infinitely loop with the square bouncing off the edges of the window at right angles. The drawings are to a back buffer screen where at the end of each loop iteration are drawn to the front buffer screen. A screengrab of the program in action is demonstrated in Fig7, below.

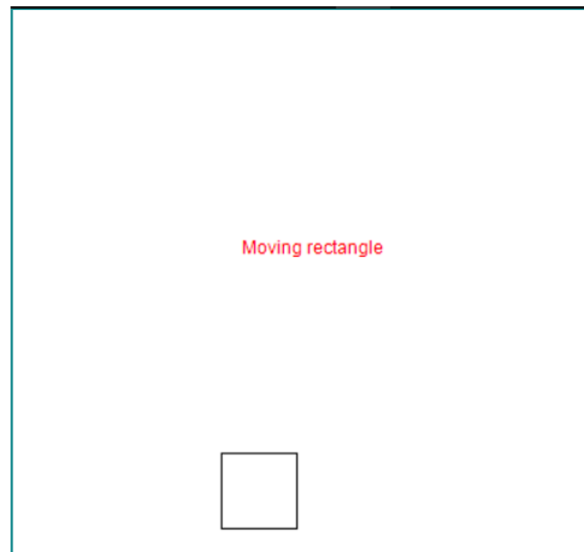


Fig7: Week 4 Portfolio exercise "DoubleBuffering"

Week 5 – Line and Points

The Program was created from scratch. Some of the code was inherited from week1 CircleInSquare. The form was created on a 700w by 500h. Inside the OnPaint() method 2x tables were declared and initialised with 10 elements each. The whole drawing is then drawn with the use of a 1x Drawline() statement found inside a for loop, which is inside a for loop. A screengrab of the program in action is demonstrated in Fig8, below.

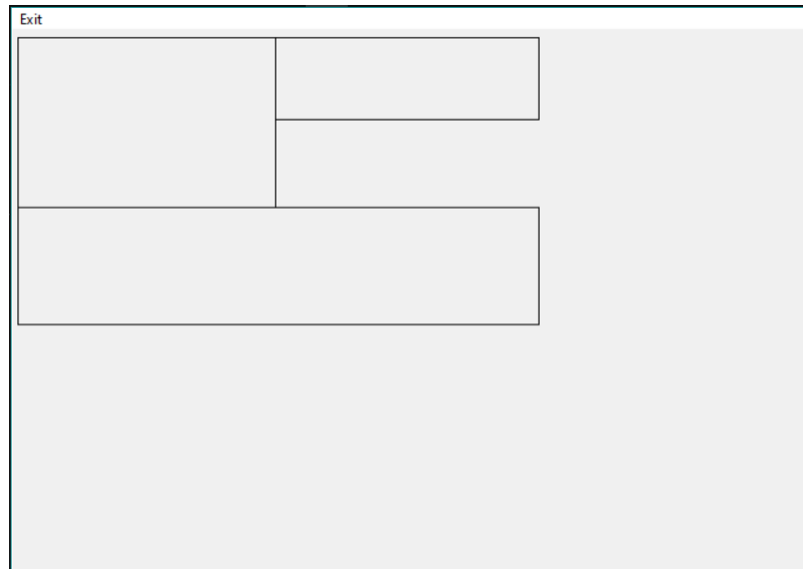


Fig8: Week 5 Portfolio Exercise "LineAndPoints"

Conclusion

For a better UX, where possible (i.e., due to animations) the portfolio applications have had an 'Exit' button added to them, as demonstrated in Fig1 below. Once pressed, this will cleanly terminate the program.

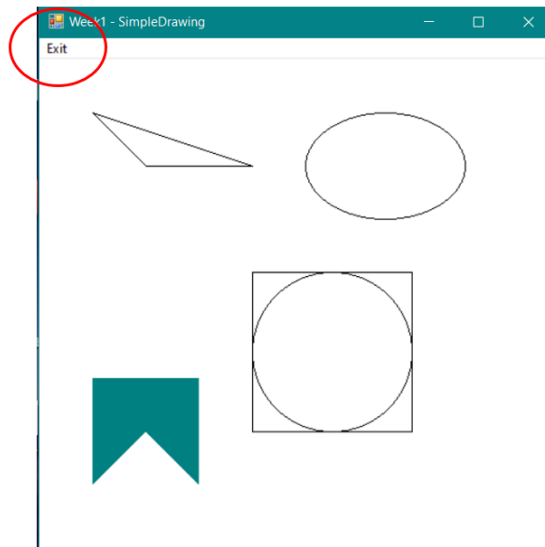


Fig1: Exit Button

Lastly, every program was modified to run without the "program.cs" file as method 'Main()' was embedded inside the class, as demonstrated in Fig2 below.

```
// Removed the 'Program.cs' file for ease of use ( method Main() was embedded inside the class )  
public static void Main()  
{  
    Application.Run(new SimpleDrawing());  
}
```

Fig2: method Main()