

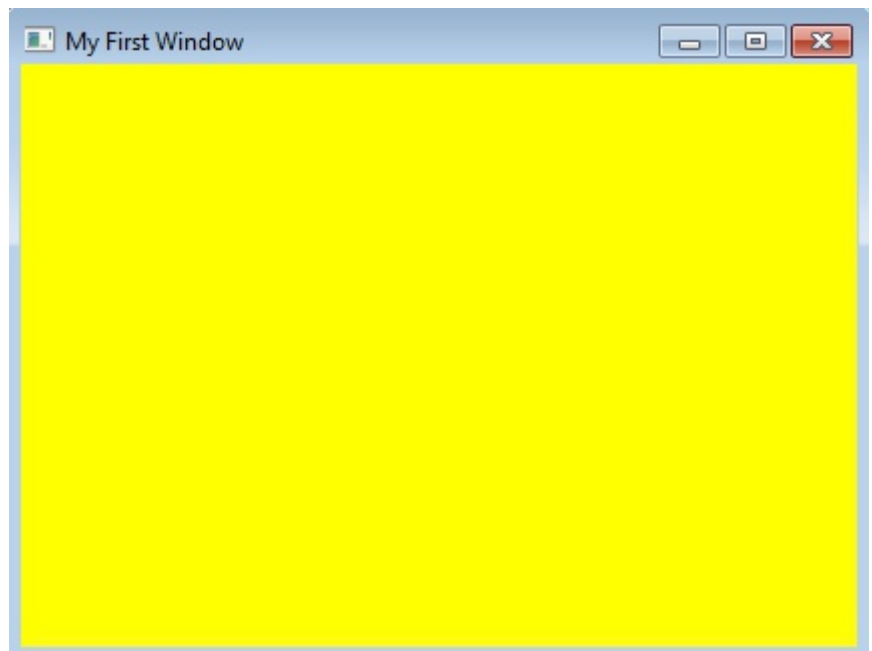
## Using graphics

So far all of our examples, except the turtle graphics, have used the 'TextWindow'. To make our programs more interesting and colourful, Small Basic includes a 'GraphicsWindow' object. This object provides a window on which we can draw. It can be used to develop a wide range of applications from painting programs to games.

Creating a 'GraphicsWindow' is as easy as creating a 'TextWindow', and we can easily set its size, colour and title.

```
1 'Using the graphics window'  
2 GraphicsWindow.Width = 600  
3 GraphicsWindow.Height = 400  
4 GraphicsWindow.Title = "My First Window"  
5 GraphicsWindow.BackgroundColor = "yellow"
```

The above program creates a GraphicsWindow 600 pixels wide, 400 pixels high, its title is set to 'My First Window'; and its background colour is yellow.

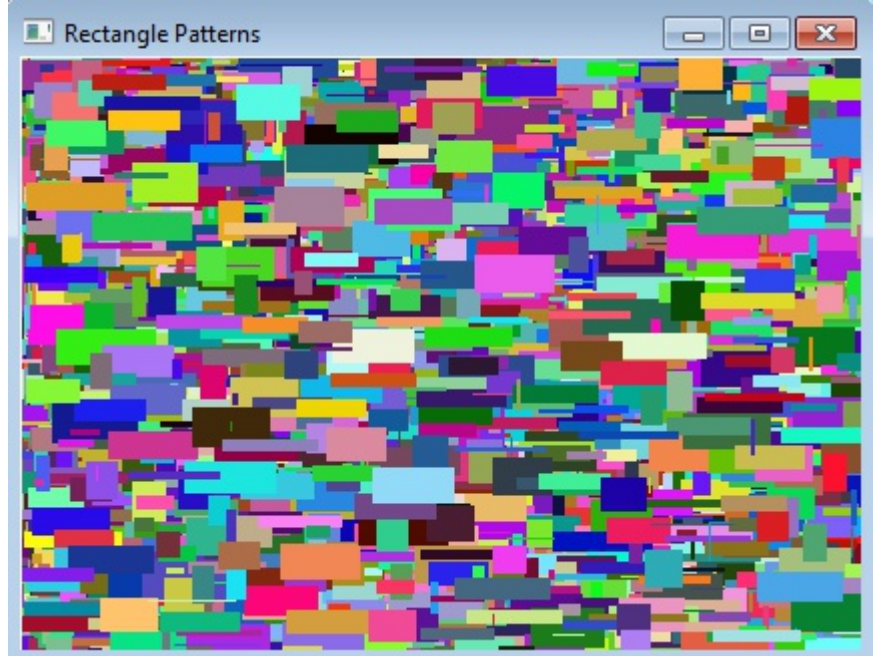


## Drawing shapes

The GraphicsWindow provides lots of interesting methods and properties. We will use the FillRectangle(x, y, width, height) and GetRandomColor() methods to fill the window with colourful rectangles. To use the FillRectangle method we have to pass the position variables x and y which are the coordinates of the rectangle's top left vertex; and we also have to pass the width and height of the rectangle. The GetRandomColor method does not require any parameters.

The followind will draw an infinite number of rectangles of varying sizes and colours all over the graphics window.

```
1 'Using the graphics window
2 GraphicsWindow.Width = 600
3 GraphicsWindow.Height = 400
4 GraphicsWindow.Title = "Rectangle Patterns"
5
6 'Create an infinite loop that will draw rectangles
7 'of various sizes and colours.
8 While "True"
9
10 'Get random dimensions of rectangle.
11 width = Math.GetRandomNumber(50)
12 height = Math.GetRandomNumber(20)
13
14 'Get random position of rectangle.
15 x = Math.GetRandomNumber(600)
16 y = Math.GetRandomNumber(400)
17
18 'Get a random brush colour.
19 color = GraphicsWindow.GetRandomColor()
20 GraphicsWindow.BrushColor = color
21
22 'Draw a random rectangle.
23 GraphicsWindow.FillRectangle(x, y, width, height)
24
25 EndWhile
```



The GraphicsWindow object has three methods for drawing solid shapes they are:

```
GraphicsWindow.FillEllipse(x, y, width, height)
```

```
GraphicsWindow.FillTriangle(x1, y1, x2, y2, z1, z2)
```

```
GraphicsWindow.FillRectangle(x, y, width, height)
```

FillEllipse works in just the same way as the FillRectangle method. However, the FillTriangle needs three coordinates for each of its vertices - that is a total of 6 parameters.

You should now be able to modify the previous program so that it draws either ellipses or triangles.

## Capturing the mouse position

The GraphicsWindow has two properties that allows us to track the position of the mouse, namely MouseX and MouseY. If we assign the position of the mouse in the GraphicsWindow to two variables, say x and y, then we can plot a circle at the position of the mouse. If we continually update the variables x and y by using an infinite loop we can create a very simple drawing program.

```
1 'Capturing the mouse position
2 GraphicsWindow.Width = 600
3 GraphicsWindow.Height = 400
4 GraphicsWindow.Title = "Silly Painting"
5
6 'Create an infinite loop that will update
7 'the new mouse position.
8 While "True"
9
10 'Get the position of the mouse.
11 x = GraphicsWindow.MouseX
12 y = GraphicsWindow.MouseY
13
14 'Get a random colour
15 color = GraphicsWindow.GetRandomColor()
16 GraphicsWindow.BrushColor = color
17
18 'Draw an circle at the current mouse position.
19 GraphicsWindow.FillEllipse(x, y, 10, 10)
20
21 EndWhile
```

To draw a circle we just draw an ellipse with its height set to the same size as its width, in this case 10 pixels.

Each time the while loop is executed the x and y variables are updated with the new mouse coordinates.

Now we can create some silly drawings.

