

Drawing.cs

...C\Desktop\COS20007\Week_3\3.3P\ShapeDrawer\Drawing.cs

1

```
1  using SplashKitSDK;
2  using System.ComponentModel;
3  using System.Security.Cryptography.X509Certificates;
4
5  namespace ShapeDrawer
6  {
7      internal class Drawing
8      {
9          private readonly List<Shape> _shapes;
10         private Color _background;
11
12         public Color Background
13         {
14             get => _background;
15             set => _background = value;
16         }
17
18         public Drawing(Color background)
19         {
20             _shapes = new List<Shape>();
21             _background = background;
22         }
23
24         public Drawing() : this (Color.White)
25         {
26             _shapes = new List<Shape>();
27             _background = Color.White;
28         }
29
30         public int ShapeCount
31         {
32             get => _shapes.Count;
33         }
34
35         public void AddShape(Shape shape)
36         {
37             _shapes.Add(shape);
38         }
39
40         public void RemoveShape(Shape shape)
41         {
42             _= _shapes?.Remove(shape);
43         }
44
45         public void Draw()
46         {
47             SplashKit.ClearScreen(_background);
48             for (int i = 0; i < _shapes.Count; i++)
49             {
```

```
50             if (_shapes[i].Selected)
51             {
52                 _shapes[i].DrawOutline();
53             }
54             _shapes[i].Draw();
55         }
56     }
57
58     public void SelectShapesAt(Point2D point)
59     {
60         foreach (Shape s in _shapes)
61         {
62             s.Selected = s.IsAt(point);
63         }
64     }
65
66     public List<Shape> SelectedShapes
67     {
68         get
69         {
70             List<Shape> result = new List<Shape>();
71             foreach (Shape s in _shapes)
72             {
73                 if (s.Selected)
74                 {
75                     result.Add(s);
76                 }
77             }
78             return result;
79         }
80     }
81 }
82 }
83 }
```

Shape.cs

..\PC\Desktop\COS20007\Week_3\3.3P\ShapeDrawer\Shape.cs

1

```
1  using SplashKitSDK;
2
3  namespace ShapeDrawer
4  {
5      internal class Shape
6      {
7          private Color _color;
8          private float _x;
9          private float _y;
10         private int _width;
11         private int _height;
12         private bool _selected;
13
14         public Shape()
15         {
16             _color = Color.Green;
17             _x = _y = 0.0f;
18             _width = _height = 100;
19         }
20
21         public bool Selected
22         {
23             get => _selected;
24             set => _selected = value;
25         }
26
27         public Color Color
28         {
29             get => _color;
30             set => _color = value;
31         }
32
33         public float X
34         {
35             get => _x;
36             set => _x = value;
37         }
38
39         public float Y
40         {
41             get => _y;
42             set => _y = value;
43         }
44
45         public int Width
46         {
47             get => _width;
48             set => _width = value;
49         }
```

```
50
51     public int Height
52     {
53         get => _height;
54         set => _height = value;
55     }
56
57     public void Draw()
58     {
59         SplashKit.FillRectangle(_color, _x, _y, _width, _height);
60     }
61
62     public bool IsAt(Point2D point)
63     {
64         if (point.X >= _x && point.X <= _x + _width)
65         {
66             if (point.Y >= _y && point.Y <= _y + _height)
67             {
68                 return true;
69             }
70         }
71         return false;
72     }
73
74     public void DrawOutline()
75     {
76         SplashKit.FillRectangle(Color.Black, _x - 2, _y - 2, _width + 4, _height + 4);
77     }
78 }
79 }
80 }
```

Program.cs

..C\Desktop\COS20007\Week_3\3.3P\ShapeDrawer\Program.cs 1

```
1  using System;
2  using Microsoft.VisualBasic;
3  using SplashKitSDK;
4
5  namespace ShapeDrawer
6  {
7      public class Program
8      {
9          public static void Main()
10         {
11             Window window = new Window("Shape Drawer", 800, 600);
12             Drawing myDrawing;
13
14             myDrawing = new Drawing();
15
16             do
17             {
18                 SplashKit.ProcessEvents();
19                 SplashKit.ClearScreen();
20
21                 Point2D myPoint = new Point2D()
22                 {
23                     X = SplashKit.MouseX(),
24                     Y = SplashKit.MouseY()
25                 };
26
27                 if (SplashKit.MouseClicked(MouseButton.LeftButton))
28                 {
29                     Shape myShape = new Shape();
30                     myShape.X = SplashKit.MouseX();
31                     myShape.Y = SplashKit.MouseY();
32
33                     myDrawing.AddShape(myShape);
34                 }
35
36                 if (SplashKit.KeyTyped(KeyCode.SpaceKey))
37                 {
38                     myDrawing.Background = SplashKit.RandomColor();
39                 }
40
41                 if (SplashKit.MouseClicked(MouseButton.RightButton))
42                 {
43                     myDrawing.SelectShapesAt(myPoint);
44                 }
45
46                 if ((SplashKit.KeyTyped(KeyCode.DeleteKey)) ||
47                     (SplashKit.KeyTyped(KeyCode.BackspaceKey)))
48                 {
49                     foreach (Shape shape in myDrawing.SelectedShapes)
```

```
49             {
50                 myDrawing.RemoveShape(shape);
51             }
52         }
53
54         myDrawing.Draw();
55
56         SplashKit.RefreshScreen();
57     }
58     while (!window.CloseRequested);
59 }
60 }
61 }
62 }
```

Program Output

The screenshot shows a Windows desktop environment with a Visual Studio IDE open. The IDE displays a C# application titled "Shape Drawer". The application window shows a light blue background with a fractal pattern composed of green squares of varying sizes. The Visual Studio interface includes:

- Solution Explorer:** Shows the project structure with files: ShapeDrawer.csproj, Dependencies, lib, Drawing.cs, Program.cs, and Shape.cs.
- Toolbars:** Standard Visual Studio toolbars for File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Code Editor:** Shows the source code for Program.cs, which contains logic for handling mouse clicks to draw shapes.
- Status Bar:** Displays file number (50), character count (Ch: 54), and line endings (SPC CRLF).
- Taskbar:** Shows the Windows Start button, Search bar, and pinned icons for various applications like File Explorer, Edge, and File History.
- System Tray:** Shows the date and time (11:12 PM, 5/27/2024) and system icons.

```
26
27     if (SplashKit.MouseClicked(MouseButton.LeftButton))
28     {
29         Shape myShape = new Shape();
30         myShape.X = SplashKit.MouseX();
31
32         myShape.Y = SplashKit.MouseY();
33
34         myShape.Width = 10;
35
36         myShape.Height = 10;
37
38         myShape.FillColor = Colors.Green;
39
40         myShape.StrokeColor = Colors.Black;
41
42         myShape.StrokeWidth = 1;
43
44         myShape.CenterX = SplashKit.MouseX();
45
46         myShape.CenterY = SplashKit.MouseY();
47
48         myShape.CenterX = SplashKit.MouseX();
49
50         myShape.CenterY = SplashKit.MouseY();
51
52         myShape.CenterX = SplashKit.MouseX();
53
54         myShape.CenterY = SplashKit.MouseY();
55
56         myShape.CenterX = SplashKit.MouseX();
57
58         myShape.CenterY = SplashKit.MouseY();
59
60         myShape.CenterX = SplashKit.MouseX();
61
62     }
63 }
```