

```
1 using System;
2 using SplashKitSDK;
3 using System.Collections.Generic;
4 using System.Drawing;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8
9
10 namespace Drawing_Class_Task
11 {
12     public class Program
13     {
14         public static void Main()
15         {
16             Window window = new Window("Shape Drawer Task 3.3", 800, 600);
17             Drawing myDraw = new Drawing();
18             do
19             {
20                 SplashKit.ProcessEvents();
21                 SplashKit.ClearScreen();
22                 myDraw.Draw();
23
24                 if (SplashKit.MouseClicked(MouseButton.LeftButton))
25                 {
26                     Shape myShape = new Shape();
27
28                     Point2D mouseposition = SplashKit.MousePosition();
29                     myShape.X = (float)SplashKit.MouseX();
30                     myShape.Y = (float)SplashKit.MouseY();
31                     myDraw.AddShape(myShape);
32                 }
33
34                 if (SplashKit.KeyTyped(KeyCode.SpaceKey))
35                 {
36                     myDraw.background = SplashKit.RandomRGBColor(225);
37                 }
38
39                 if (SplashKit.KeyDown(KeyCode.DeleteKey) || SplashKit.KeyDown ↵
40                     (KeyCode.BackspaceKey))
41                 {
42                     foreach (Shape shape in myDraw.SelectedShapes)
43                     {
44                         myDraw.RemoveShape(shape);
45                     }
46                 }
47
48                 if (SplashKit.MouseClicked(MouseButton.RightButton))
49                 {
50                     myDraw.Clear();
51                 }
52             } while (true);
53         }
54     }
55 }
```

```
49         Point2D selected = SplashKit.MousePosition();
50         myDraw.SelectShapesAt(selected);
51     }
52
53     SplashKit.RefreshScreen();
54
55
56     } while (!SplashKit.WindowCloseRequested(window));
57 }
58 }
59 }
60
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Drawing;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using System.Xml.Serialization;
8 using SplashKitSDK;
9
10 namespace Drawing_Class_Task
11 {
12     public class Drawing
13     {
14         private readonly List<Shape> _shapes;
15         private SplashKitSDK.Color _background;
16         internal SplashKitSDK.Color Color;
17
18         public Drawing(SplashKitSDK.Color background)
19         {
20             _background = background;
21             _shapes = new List<Shape>();
22         }
23         public SplashKitSDK.Color background
24         {
25             get
26             {
27                 return _background;
28             }
29             set
30             {
31                 _background = value;
32             }
33         }
34         public Drawing():this(SplashKitSDK.Color.White)
35         {
36
37         }
38         public List<Shape> SelectedShapes
39         {
40             get
41             {
42                 List<Shape> result = new List<Shape>(); ;
43                 foreach (Shape shape in _shapes)
44                 {
45                     if (shape.Selected)
46                     {
47                         result.Add(shape);
48                     }
49                 }
49             }
49         }
49     }
49 }
```

```
50         return result;
51     }
52 }
53 public int ShapeCount
54 {
55     get { return _shapes.Count; }
56 }
57 public void AddShape(Shape shape)
58 {
59     _shapes.Add(shape);
60 }
61 public void RemoveShape(Shape shape)
62 {
63     _shapes.Remove(shape);
64 }
65 public void Draw()
66 {
67     SplashKit.ClearScreen(_background);
68     foreach (Shape shape in _shapes)
69     {
70         shape.Draw();
71     }
72 }
73
74 public void SelectShapesAt(Point2D pt)
75 {
76     foreach (Shape shape in _shapes)
77     {
78         if (shape.IsAt(pt))
79         {
80             shape.Selected = true;
81         }
82         else { shape.Selected = false; }
83     }
84 }
85
86 }
87 }
88
```

```
1 using SplashKitSDK;
2 using System;
3 using System.Collections.Generic;
4 using System.Drawing;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace Drawing_Class_Task
10 {
11     public class Shape
12     {
13         private SplashKitSDK.Color _color;
14         private float _x;
15         private float _y;
16         private int _width;
17         private int _height;
18         private bool _selected;
19         public Shape()
20         {
21             _color = SplashKitSDK.Color.Green;
22             _x = 0;
23             _y = 0;
24             _width = 100;
25             _height = 100;
26
27         }
28
29         public SplashKitSDK.Color Color
30         {
31             get { return _color; }
32             set { _color = value; }
33         }
34
35         public float X
36         {
37             get { return _x; }
38             set { _x = value; }
39         }
40         public float Y
41         {
42             get { return _y; }
43             set { _y = value; }
44         }
45         public bool Selected
46         {
47             get
48             {
49                 return _selected;
```

```
50     }
51     set { _selected = value; }
52 }
53 public void Draw()
54 {
55     SplashKit.FillRectangle(_color, _x, _y, _width, _height);
56     if (Selected)
57     {
58         DrawOutline();
59     }
60 }
61 public bool IsAt(Point2D p)
62 {
63     return SplashKit.PointInRectangle(p, SplashKit.RectangleFrom(X, ↗
        Y, _width, _height));
64 }
65 public void DrawOutline()
66 {
67     SplashKit.DrawRectangle(SplashKit.ColorBlack(), _x-2, _y -2, ↗
        _width +4 , _height +4);
68 }
69 }
70 }
```



