

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

```
50     }
51
52     public void Draw()
53     {
54         SplashKit.FillRectangle(_color, _x, _y, _width, _height);
55     }
56
57     public bool IsAt(Point2D pt)
58     {
59         return pt.X >= _x && pt.Y >= _y && pt.X <= _x + _width && pt.Y
           <= _y + _height;
60     }
61 }
62 }
63
```

```
1 using SplashKitSDK;
2 using System.Security.Cryptography.X509Certificates;
3 namespace ShapeDrawer
4 {
5     public class Program
6     {
7         public static void Main()
8         {
9             Shape myshape = new Shape(184);
10            Window window = new Window("Shape Drawer", 800, 600);
11            do
12            {
13                SplashKit.ProcessEvents();
14                SplashKit.ClearScreen();
15                myshape.Draw();
16                if (SplashKit.MouseClicked(MouseButton.LeftButton))
17                {
18                    myshape.X = SplashKit.MouseX();
19                    myshape.Y = SplashKit.MouseY();
20                }
21                if (SplashKit.KeyTyped(KeyCode.SpaceKey) && myshape.IsAt
22                (SplashKit.MousePosition()))
23                {
24                    myshape.Color = Color.Random();
25                }
26                SplashKit.RefreshScreen();
27            } while (!window.CloseRequested);
28        }
29    }
30 }
31
32
```

