

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using SplashKitSDK;
7
8 namespace MultipleShapeKinds
9 {
10     public class MyLine:Shape
11     {
12         private float _endX;
13         private float _endY;
14
15         public MyLine() : this(Color.Red, 0.0f, 0.0f, 10, 10)
16         {
17
18         }
19
20         public MyLine(Color color, float startX, float startY, float endX, ↗
21             float endY) : base(color)
22         {
23             X = startX;
24             Y = startY;
25         }
26
27         public float EndX
28         {
29             get
30             {
31                 return _endX;
32             }
33             set
34             {
35                 _endX = value;
36             }
37         }
38
39         public float EndY
40         {
41             get
42             {
43                 return _endY;
44             }
45             set
46             {
47                 _endY = value;
48             }
49         }
50     }
51 }
```

```
49
50     public override void Draw()
51     {
52         if (Selected)
53         {
54             DrawOutline();
55         }
56         SplashKit.DrawLine(Color, X, Y, _endX, _endY);
57     }
58
59     public override void DrawOutline()
60     {
61         SplashKit.FillCircle(Color.Black, X, Y, 2);
62         SplashKit.FillCircle(Color.Black, _endX, _endY, 2);
63     }
64
65     public override bool IsAt(Point2D pt)
66     {
67         if (SplashKit.PointOnLine(pt, SplashKit.LineFrom(X, Y, _endX,
68             _endY)))
69         {
70             return true;
71         }
72         else return false;
73     }
74 }
75
```