

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

```
49         if (Selected)
50         {
51             DrawOutline();
52         }
53         SplashKit.DrawLine(Color, X, Y, _endX, _endY);
54     }
55
56     public override void DrawOutline()
57     {
58         SplashKit.FillCircle(Color.Black, X, Y, 2);
59         SplashKit.FillCircle(Color.Black, _endX, _endY, 2);
60     }
61
62     public override bool IsAt(Point2D pt)
63     {
64         if (SplashKit.PointOnLine(pt, SplashKit.LineFrom(X, Y, _endX,
65             _endY)))
66         {
67             return true;
68         }
69         else return false;
70     }
71
72     public override void SaveTo(StreamWriter writer)
73     {
74         writer.WriteLine("Line");
75         base.SaveTo(writer);
76         writer.WriteLine(_endX);
77         writer.WriteLine(_endY);
78     }
79
80     public override void LoadFrom(StreamReader reader)
81     {
82         base.LoadFrom(reader);
83         _endX = reader.ReadInteger();
84         _endY = reader.ReadInteger();
85     }
86 }
87
```