Drawing.cs


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

```
50                  {
51                      _shapes[i].DrawOutline();
52                  }
53                  _shapes[i].Draw();
54              }
55          }
56
57          public void SelectShapesAt(Point2D point)
58          {
59              foreach (Shape s in _shapes)
60              {
61                  s.Selected = s.IsAt(point);
62              }
63          }
64
65          public List<Shape> SelectedShapes
66          {
67              get
68              {
69                  List<Shape> result = new List<Shape>();
70                  foreach (Shape s in _shapes)
71                  {
72                      if (s.Selected)
73                      {
74                          result.Add(s);
75                      }
76                  }
77                  return result;
78              }
79          }
80
81          public void Save(string filename)
82          {
83              StreamWriter writer;
84
85              writer = new StreamWriter(filename);
86              try
87              {
88                  writer.WriteColor(Background);
89                  writer.WriteLine(ShapeCount);
90
91                  foreach (Shape s in _shapes)
92                  {
93                      s.SaveTo(writer);
94                  }
95              }
96              finally
97              {
98                  writer.Close();
```

```
 99              }
100          }
101
102          public void Load(string filename)
103          {
104              StreamReader reader = new StreamReader(filename);
105              Shape s;
106              try
107              {
108                  Background = reader.ReadColor();
109                  int count = reader.ReadInteger();
110                  _shapes.Clear();
111
112                  for (int i = 0; i < count; i++)
113                  {
114                      string kind = reader.ReadLine();
115
116                      switch (kind)
117                      {
118                          case "Rectangle":
119                              s = new MyRectangle();
120                              break;
121
122                          case "Circle":
123                              s = new MyCircle();
124                              break;
125
126                          case "Line":
127                              s = new MyLine();
128                              break;
129
130                          default:
131                              throw new InvalidDataException("Unknown shape
                        kind: " + kind);
132                      }
133
134                      s.LoadFrom(reader);
135                      _shapes.Add(s);
136                  }
137              }
138              finally
139              {
140                  reader.Close();
141              }
142          }
143      }
144  }
145
```

ExtensionMethods.cs


```
1  using System;
2  using System.IO;
3  using SplashKitSDK;
4
5  namespace ShapeDrawer
6  {
7      public static class ExtensionMethods
8      {
9          public static int ReadInteger(this StreamReader reader)
10         {
11             return Convert.ToInt32(reader.ReadLine());
12         }
13
14         public static float ReadSingle(this StreamReader reader)
15         {
16             return Convert.ToSingle(reader.ReadLine());
17         }
18
19         public static Color ReadColor(this StreamReader reader)
20         {
21             return Color.RGBColor(reader.ReadSingle(), reader.ReadSingle(), ↵
                   reader.ReadSingle());
22         }
23
24         public static void WriteColor(this StreamWriter writer, Color clr)
25         {
26             writer.WriteLine("{0}\n{1}\n{2}", clr.R, clr.G, clr.B);
27         }
28     }
29 }
30
```

MyCircle.cs


```
1  using SplashKitSDK;
2
3  namespace ShapeDrawer
4  {
5      public class MyCircle : Shape
6      {
7          private int _radius;
8          public int Radius
9          {
10             get => _radius;
11             set => _radius = value;
12         }
13
14         public MyCircle(Color color, int radius) : base(color)
15         {
16             _radius = radius;
17         }
18
19         public MyCircle() : this (Color.Blue, 50)
20         {
21             _radius = 50;
22             Color = Color.Blue;
23         }
24
25         public override void DrawOutline()
26         {
27             SplashKit.FillCircle(Color.Black, X, Y, _radius + 2);
28         }
29
30         public override void Draw()
31         {
32             if (Selected)
33             {
34                 DrawOutline();
35             }
36
37             SplashKit.FillCircle(Color, X, Y, _radius);
38         }
39
40         public override bool IsAt(Point2D point)
41         {
42             if (point.X > X - _radius && point.X < X + _radius)
43             {
44                 if (point.Y > Y - _radius && point.Y < Y + _radius)
45                 {
46                     return true;
47                 }
48             }
49             return false;
```

```
50          }
51
52          public override void SaveTo(StreamWriter writer)
53          {
54              writer.WriteLine("Circle");
55              base.SaveTo(writer);
56              writer.WriteLine(Radius);
57          }
58
59          public override void LoadFrom(StreamReader reader)
60          {
61              base.LoadFrom(reader);
62              Radius = reader.ReadInteger();
63          }
64      }
65  }
66
```

MyLine.cs


```
1  using SplashKitSDK;
2
3  namespace ShapeDrawer
4  {
5      public class MyLine : Shape
6      {
7          private float _endX;
8          private float _endY;
9
10         public MyLine(Color color, float x, float y, float endX, float
             endY) : base(color)
11         {
12             X = x;
13             Y = y;
14             _endX = endX;
15             _endY = endY;
16         }
17
18         public MyLine() : this (Color.Red, 0.0f, 0.0f, 100.0f, 0.0f)
19         {
20             Color = Color.Red;
21             X = 0.0f;
22             Y = 0.0f;
23             _endX = 100.0f;
24             _endY = 0.0f;
25         }
26
27         public float EndX
28         {
29             get => _endX;
30             set => _endX = value;
31         }
32
33         public float EndY
34         {
35             get => _endY;
36             set => _endY = value;
37         }
38
39         public override bool IsAt(Point2D point)
40         {
41             if (point.X >= X && point.X <= X + EndX)
42             {
43                 if (point.Y >= Y - 2 && point.Y <= Y + 2)
44                 {
45                     return true;
46                 }
47             }
48             return false;
```

```
49          }
50
51          public override void Draw()
52          {
53              if (Selected)
54              {
55                  DrawOutline();
56              }
57              SplashKit.DrawLine(Color, X, Y, X + EndX, Y + EndY);
58          }
59
60          public override void DrawOutline()
61          {
62              SplashKit.FillCircle(Color.Black, X, Y, 4);
63              SplashKit.FillCircle(Color.Black, X + EndX, Y + EndY, 4);
64          }
65
66          public override void SaveTo(StreamWriter writer)
67          {
68              writer.WriteLine("Line");
69              base.SaveTo(writer);
70          }
71
72          public override void LoadFrom(StreamReader reader)
73          {
74              base.LoadFrom(reader);
75          }
76      }
77  }
78
```

MyRectangle.cs


```
1  using SplashKitSDK;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Runtime.InteropServices;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace ShapeDrawer
10 {
11     public class MyRectangle : Shape
12     {
13         private int _width;
14         private int _height;
15
16         public MyRectangle(Color color, float x, float y, int width, int
             height) : base(color)
17         {
18             X = x;
19             Y = y;
20             _width = width;
21             _height = height;
22         }
23
24         public MyRectangle() : this (Color.Green, 0.0f, 0.0f, 100, 100)
25         {
26             Color = Color.Green;
27             X = 0.0f;
28             Y = 0.0f;
29             _width = 100;
30             _height = 100;
31         }
32         public int Width
33         {
34             get => _width;
35             set => _width = value;
36         }
37
38         public int Height
39         {
40             get => _height;
41             set => _height = value;
42         }
43
44         public override void DrawOutline()
45         {
46             SplashKit.FillRectangle(Color.Black, X - 2, Y - 2, _width + 4,
                 _height + 4);
47         }
```

```
48
49          public override void Draw()
50          {
51              if (Selected)
52              {
53                  DrawOutline();
54              }
55              SplashKit.FillRectangle(Color, X, Y, _width, _height);
56          }
57
58          public override bool IsAt(Point2D point)
59          {
60              if (point.X >= X && point.X <= X + _width)
61              {
62                  if (point.Y >= Y && point.Y <= Y + _height)
63                  {
64                      return true;
65                  }
66              }
67              return false;
68          }
69
70          public override void SaveTo(StreamWriter writer)
71          {
72              writer.WriteLine("Rectangle");
73              base.SaveTo(writer);
74              writer.WriteLine(Width);
75              writer.WriteLine(Height);
76          }
77
78          public override void LoadFrom(StreamReader reader)
79          {
80              base.LoadFrom(reader);
81              Width = reader.ReadInteger();
82              Height = reader.ReadInteger();
83          }
84      }
85  }
86
```

Shape.cs


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

```
50          public abstract bool IsAt(Point2D point);
51
52          public abstract void DrawOutline();
53
54          public virtual void SaveTo(StreamWriter writer)
55          {
56              writer.WriteColor(Color);
57              writer.WriteLine(X);
58              writer.WriteLine(Y);
59          }
60
61          public virtual void LoadFrom(StreamReader reader)
62          {
63              Color = reader.ReadColor();
64              X = reader.ReadInteger();
65              Y = reader.ReadInteger();
66          }
67      }
68  }
69
```

Program.cs


```
 1  using System;
 2  using Microsoft.VisualBasic;
 3  using SplashKitSDK;
 4
 5  namespace ShapeDrawer
 6  {
 7      public class Program
 8      {
 9          private enum ShapeKind
10          {
11              Rectangle,
12              Circle,
13              Line
14          }
15          public static void Main()
16          {
17              ShapeKind kindToAdd = ShapeKind.Circle;
18              Window window = new Window("Shape Drawer", 800, 600);
19              Drawing myDrawing;
20
21              myDrawing = new Drawing();
22
23              do
24              {
25                  SplashKit.ProcessEvents();
26                  SplashKit.ClearScreen();
27
28                  if (SplashKit.KeyTyped(KeyCode.RKey))
29                  {
30                      kindToAdd = ShapeKind.Rectangle;
31                  }
32
33                  if (SplashKit.KeyTyped(KeyCode.CKey))
34                  {
35                      kindToAdd = ShapeKind.Circle;
36                  }
37
38                  if (SplashKit.KeyTyped(KeyCode.LKey))
39                  {
40                      kindToAdd = ShapeKind.Line;
41                  }
42
43                  if (SplashKit.MouseClicked(MouseButton.LeftButton))
44                  {
45                      Shape myShape;
46                      switch(kindToAdd)
47                      {
48                          case ShapeKind.Circle:
49                              myShape = new MyCircle();
```

```
50                              break;
51
52                          case ShapeKind.Line:
53                              myShape = new MyLine();
54                              break;
55
56                          default:
57                              myShape = new MyRectangle();
58                              break;
59                      }
60                      myShape.X = SplashKit.MouseX();
61                      myShape.Y = SplashKit.MouseY();
62
63                      myDrawing.AddShape(myShape);
64                  }
65
66                  Point2D myPoint = new Point2D()
67                  {
68                      X = SplashKit.MouseX()
69                      , Y = SplashKit.MouseY()
70                  };
71
72                  if (SplashKit.KeyTyped(KeyCode.SpaceKey))
73                  {
74                      myDrawing.Background = SplashKit.RandomColor();
75                  }
76
77                  if (SplashKit.MouseClicked(MouseButton.RightButton))
78                  {
79                      myDrawing.SelectShapesAt(myPoint);
80                  }
81
82                  if ((SplashKit.KeyTyped(KeyCode.DeleteKey)) |
                      (SplashKit.KeyTyped(KeyCode.BackspaceKey)))
83                  {
84                      foreach (Shape shape in myDrawing.SelectedShapes)
85                      {
86                          myDrawing.RemoveShape(shape);
87                      }
88                  }
89
90                  if ((SplashKit.KeyTyped(KeyCode.SKey)))
91                  {
92                      myDrawing.Save("C:/Users/PC/Desktop/COS20007/
                       Week_5/5.3C/ShapeDrawer/TextDrawing.txt");
93                  }
94
95                  if ((SplashKit.KeyTyped(KeyCode.OKey)))
96                  {
```

```
 97                      try
 98                      {
 99                          myDrawing.Load("C:/Users/PC/Desktop/COS20007/
                        Week_5/5.3C/ShapeDrawer/TextDrawing.txt");
100                      }
101                      catch (Exception ex)
102                      {
103                          Console.Error.WriteLine("Error loading file: {0}",
                         ex.Message);
104                      }
105                  }
106
107                  myDrawing.Draw();
108
109                  SplashKit.RefreshScreen();
110              }
111              while (!window.CloseRequested);
112          }
113      }
114  }
115
```

Program Execution