

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

---

## Drawing Program - Saving and Loading

---

PDF generated at 03:01 on Saturday 30<sup>th</sup> September, 2023

```
1  using System;
2  using SplashKitSDK;
3  namespace Task5._3
4  {
5      public class Program
6      {
7          private enum ShapeKind
8          {
9              Rectangle, Circle, Line
10         }
11
12         public static void Main()
13         {
14             Window window = new Window("Shape Drawer", 800, 600);
15
16             Drawing dr = new Drawing();
17             ShapeKind kindToAdd = ShapeKind.Circle;
18             do
19             {
20                 SplashKit.ProcessEvents();
21                 SplashKit.ClearScreen();
22
23                 if (SplashKit.MouseClicked(MouseButton.LeftButton))
24                 {
25                     Shape newShape;
26                     if (kindToAdd == ShapeKind.Circle)
27                     {
28                         newShape = new MyCircle();
29                     }
30                     else if (kindToAdd == ShapeKind.Rectangle)
31                     {
32                         newShape = new MyRectangle();
33                     }
34                     else
35                     {
36                         newShape = new MyLine();
37                     }
38
39                     newShape.X = SplashKit.MouseX();
40
41                     newShape.Y = SplashKit.MouseY();
42
43                     dr.AddShape(newShape);
44                 }
45
46                 if (SplashKit.KeyTyped(KeyCode.RKey))
47                 {
48                     kindToAdd = ShapeKind.Rectangle;
49                 }
50
51                 if (SplashKit.KeyTyped(KeyCode.CKey))
52                 {
53                     kindToAdd = ShapeKind.Circle;
```

```

54         }
55         if (SplashKit.KeyTyped(KeyCode.LKey))
56         {
57             kindToAdd = ShapeKind.Line;
58         }
59         if (SplashKit.MouseClicked(MouseButton.RightButton))
60         {
61
62             dr.SelectShapesAt(SplashKit.MousePosition());
63         }
64
65         if (SplashKit.KeyTyped(KeyCode.SpaceKey))
66         {
67
68             dr.Background1 = SplashKit.RandomRGBColor(255);
69         }
70
71         if (SplashKit.KeyTyped(KeyCode.DeleteKey) ||
↪ SplashKit.KeyTyped(KeyCode.BackspaceKey))
72         {
73
74             foreach (Shape s in dr.Selectedshapes)
75             {
76                 dr.DeleteShape(s);
77             }
78         }
79         if (SplashKit.KeyTyped(KeyCode.SKey))
80         {
81             dr.Save("Desktop\\TestDrawing.txt");
82         }
83         if (SplashKit.KeyTyped(KeyCode.OKey))
84         {
85             try
86             {
87                 dr.Load("Desktop\\TestDrawing.txt");
88             }
89             catch(Exception e)
90             {
91                 Console.Error.WriteLine("Error Loadinhg file: {0}",
↪ e.Message);
92             }
93         }
94
95
96
97         dr.Draw();
98         SplashKit.RefreshScreen();
99     } while (!window.CloseRequested);
100 }
101 }
102
103 }
104

```

105  
106  
107  
108

```
1 using System;
2 using SplashKitSDK;
3 namespace Task5._3
4 {
5     public static class ExtensionMethods
6     {
7         public static int ReadInteger(this StreamReader reader)
8         {
9             return Convert.ToInt32(reader.ReadLine());
10        }
11        public static float ReadSingle(this StreamReader reader)
12        {
13            return Convert.ToSingle(reader.ReadLine());
14        }
15        public static Color ReadColor(this StreamReader reader)
16        {
17            return Color.RGBColor(reader.ReadSingle(), reader.ReadSingle(),
18            reader.ReadSingle());
19        }
20        public static void WriteColor(this StreamWriter writer, Color clr)
21        {
22            writer.WriteLine("{0}\n{1}\n{2}", clr.R, clr.G, clr.B);
23        }
24    }
25 }
26
```

```
1  using System;
2  using SplashKitSDK;
3  using System.Collections.Generic;
4  using System.Reflection.PortableExecutable;
5
6  namespace Task5._3
7  {
8  {
9      public class Drawing
10     {
11         private readonly List<Shape> _shapes;
12         private Color _background;
13         public Drawing(Color bg)
14         {
15             _shapes = new List<Shape>();
16             _background = bg;
17
18         }
19         public Drawing() : this(Color.White)
20         {
21
22         }
23         public int ShapeCount
24         {
25             get
26             {
27                 return _shapes.Count;
28             }
29         }
30         public void AddShape(Shape s)
31         {
32             _shapes.Add(s);
33         }
34         public void Draw()
35         {
36             SplashKit.ClearScreen(Background1);
37             foreach (Shape s in _shapes)
38             {
39                 s.Draw();
40             }
41
42         }
43
44         public void SelectShapesAt(Point2D pt)
45         {
46             foreach (Shape s in _shapes)
47             {
48                 if (s.IsAt(pt))
49                 {
50                     s.Selected = true;
51                 }
52                 else
53                 {
```

```
54         s.Selected = false;
55     }
56 }
57
58 public List<Shape> Selectedshapes
59 {
60     get
61     {
62         List<Shape> _result = new List<Shape>();
63
64
65         foreach (Shape s in _shapes)
66         {
67             if (s.Selected == true)
68             {
69                 _result.Add(s);
70             }
71         }
72         return _result;
73     }
74 }
75 public Color Background1
76 {
77     get
78     {
79         return _background;
80     }
81     set
82     {
83         _background = value;
84     }
85 }
86
87 public void DeleteShape(Shape s)
88 {
89     _shapes.Remove(s);
90 }
91
92 public void Save(string filename)
93 {
94     StreamWriter writer = new StreamWriter(filename);
95     //Shape s;
96
97     writer.WriteColor(Background1);
98     writer.WriteLine(ShapeCount);
99
100    foreach(Shape s in _shapes)
101    {
102        s.SaveTo(writer);
103    }
104    writer.Close();
105 }
106 public void Load(string filename)
```

```
107     {
108         StreamReader reader = new StreamReader(filename);
109         try {
110
111             int count;
112             Shape s;
113             string kind;
114             Background1 = reader.ReadColor();
115             count = reader.ReadInteger();
116
117             _shapes.Clear();
118
119             for (int i = 0; i < count; i += 1)
120             {
121                 kind = reader.ReadLine();
122                 switch (kind)
123                 {
124                     case "Rectangle":
125                         s = new MyRectangle();
126                         break;
127
128                     case "Circle":
129                         s = new MyCircle();
130                         break;
131                     case "Line":
132                         s = new MyLine();
133                         break;
134                     default:
135                         throw new InvalidDataException("Unknown Shape Kind: " +
↪ kind);
136                 }
137
138                 s.LoadFrom(reader);
139                 AddShape(s);
140             }
141         }
142         finally
143         {
144             reader.Close();
145         }
146     }
147 }
148 }
149 }
150
151
152
153
```



```
1  using System;
2
3  using SplashKitSDK;
4
5  namespace Task5._3
6  {
7      public abstract class Shape
8      {
9          protected Color _color;
10         private float _x;
11         private float _y;
12
13         private bool _selected;
14
15         public Shape(Color c)
16         {
17             _color = c;
18             _x = (float)0;
19             _y = (float)0;
20
21         }
22         public Shape():this(Color.Yellow)
23         {
24
25         }
26
27         public abstract void Draw();
28
29         public Color Color
30         {
31             get
32             {
33                 return _color;
34             }
35             set
36             {
37                 _color = value;
38             }
39         }
40
41         public float X
42         {
43             get
44             {
45                 return _x;
46             }
47             set
48             {
49                 _x = value;
50             }
51         }
52         public float Y
53         {
```

```
54         get
55         {
56             return _y;
57         }
58         set
59         {
60             _y = value;
61         }
62     }
63
64
65     public bool Selected
66     {
67         get
68         {
69             return _selected;
70         }
71         set
72         {
73             _selected = value;
74         }
75     }
76     public abstract bool IsAt(Point2D pt);
77
78     public abstract void Outline();
79
80     public virtual void SaveTo(StreamWriter writer)
81     {
82         writer.WriteColor(Color);
83         writer.WriteLine(X);
84         writer.WriteLine(Y);
85     }
86
87     public virtual void LoadFrom(StreamReader reader)
88     {
89         Color = reader.ReadColor();
90         X = reader.ReadInteger();
91         Y = reader.ReadInteger();
92     }
93 }
94
95
96
97
```

```
1  using System;
2  using SplashKitSDK;
3
4  namespace Task5._3
5  {
6      public class MyRectangle:Shape
7      {
8          private int _width;
9          private int _height;
10         public MyRectangle():this(Color.Green, 0, 0, 100, 100)
11         {
12
13         }
14         public MyRectangle(Color clr, float x, float y, int width, int
↵ height):base(clr)
15         {
16
17             Width = width;
18             Height = height;
19         }
20         public int Width
21         {
22             get
23             {
24                 return _width;
25             }
26             set
27             {
28                 _width = value;
29             }
30         }
31
32         public int Height
33         {
34             get
35             {
36                 return _height;
37             }
38             set
39             {
40                 _height = value;
41             }
42         }
43
44         public override void Draw()
45         {
46             if (Selected == true)
47             {
48                 Outline();
49             }
50             SplashKit.FillRectangle(Color, X, Y, _width, _height);
51
52         }
```

```
53     public override void Outline()
54     {
55         SplashKit.FillRectangle(Color.Black, X - 2, Y - 2, _width + 4, _height +
↵ 4);
56     }
57
58     public override bool IsAt(Point2D pt)
59     {
60
61         if (pt.X >= X && pt.X < (X + _width) && pt.Y >= Y && pt.Y <= (Y +
↵ _height))
62         {
63
64             return true;
65         }
66         else
67         {
68             return false;
69         }
70     }
71     public override void SaveTo(StreamWriter writer)
72     {
73         writer.WriteLine("Rectangle");
74         base.SaveTo(writer);
75
76         writer.WriteLine(Width);
77
78         writer.WriteLine(Height);
79     }
80     public override void LoadFrom(StreamReader reader)
81     {
82         base.LoadFrom(reader);
83         Width = reader.ReadInteger();
84         Height = reader.ReadInteger();
85     }
86 }
87 }
88
```

```
1  using System;
2  using SplashKitSDK;
3  namespace Task5._3
4  {
5      public class MyCircle : Shape
6      {
7          private int _radius;
8
9          public MyCircle():this(Color.Blue, 0, 0, 50)
10         {
11
12         }
13         public MyCircle(Color clor,int x, int Y, int radius):base(clor)
14         {
15             _color = clor;
16             _radius = radius;
17         }
18
19         public int Radius
20         {
21             get
22             {
23                 return _radius;
24             }
25             set
26             {
27                 _radius = value;
28             }
29         }
30
31         public override void Draw()
32         {
33             if (Selected)
34                 Outline();
35             SplashKit.FillCircle(Color, X, Y, _radius);
36         }
37         public override void Outline()
38         {
39
40             SplashKit.FillCircle(Color.Black, X , Y , Radius+2);
41
42         }
43
44         public override bool IsAt(Point2D pt)
45         {
46
47             double point1 = (pt.X - X) * (pt.X - X);
48
49             double point2 = (pt.Y - Y) * (pt.Y - Y);
50
51             if(Math.Sqrt(point1+point2)<_radius)
52             {
53                 return true;
```

```
54         }
55         else
56         {
57             return false;
58         }
59     }
60 }
61 public override void SaveTo(StreamWriter writer)
62 {
63     writer.WriteLine("Circle");
64
65     base.SaveTo(writer);
66     writer.WriteLine(Radius);
67 }
68 public override void LoadFrom(StreamReader reader)
69 {
70     base.LoadFrom(reader);
71     Radius = reader.ReadInteger();
72 }
73 }
74 }
75
```

```
1  using System;
2  using System.Numerics;
3  using SplashKitSDK;
4  namespace Task5._3
5  {
6      public class MyLine : Shape
7      {
8          private float _endX;
9          private float _endY;
10
11
12         public MyLine() : this(Color.Orange, 0, 0, 200, 300)
13         {
14
15         }
16         public MyLine(Color clr, float startX, float startY, float endX, float endY)
↪      : base(clr)
17         {
18
19             EndX = endX;
20             EndY = endY;
21
22         }
23
24         public float EndX
25         {
26             get
27             {
28                 return _endX;
29             }
30             set
31             {
32                 _endX = value;
33             }
34         }
35
36         public float EndY
37         {
38             get
39             {
40                 return _endY;
41             }
42             set
43             {
44                 _endY = value;
45             }
46         }
47
48         public override void Draw()
49         {
50             if (Selected)
51                 Outline();
52             SplashKit.DrawLine(Color, X, Y, EndX, EndY,
↪      SplashKit.OptionLineWidth(5));
```

```
53
54     }
55     public override void Outline()
56     {
57
58         SplashKit.FillCircle(Color.Black, X, Y, 4);
59         SplashKit.FillCircle(Color.Black, EndX, EndY, 4);
60
61     }
62     public override bool IsAt(Point2D pt)
63     {
64
65         Line l = SplashKit.LineFrom(X, Y, EndX, EndY);
66
67         return SplashKit.PointOnLine(pt, l, 10);
68     }
69     public override void SaveTo(StreamWriter writer)
70     {
71         writer.WriteLine("Line");
72         base.SaveTo(writer);
73
74         writer.WriteLine(EndX);
75
76         writer.WriteLine(EndY);
77     }
78     public override void LoadFrom(StreamReader reader)
79     {
80         base.LoadFrom(reader);
81         EndX = reader.ReadInteger();
82         EndY = reader.ReadInteger();
83     }
84 }
85
86
87
88
89
90
91
92
93
94
95
96
97
98
```



