

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

---

## Drawing Program - A Drawing Class

---

PDF generated at 05:17 on Monday 21<sup>st</sup> August, 2023

```
1  using System;
2  using SplashKitSDK;
3
4  namespace ShapeDrawer
5  {
6      public class Program
7      {
8          public static void Main()
9          {
10              Drawing myDraw = new Drawing();
11              Window window = new Window("Shape Drawer", 800, 600);
12
13              do
14              {
15                  SplashKit.ProcessEvents();
16                  SplashKit.ClearScreen();
17                  myDraw.Draw();
18
19                  if (SplashKit.MouseClicked(MouseButton.LeftButton))
20                  {
21                      Shape myShape = new Shape();
22                      myShape.X = SplashKit.MouseX();
23                      myShape.Y = SplashKit.MouseY();
24                      myDraw.AddShape(myShape);
25                  }
26
27                  if (SplashKit.MouseClicked(MouseButton.RightButton))
28                  {
29                      myDraw.SelectedShapesAt(SplashKit.MousePosition());
30                  }
31
32                  if (SplashKit.KeyDown(KeyCode.SpaceKey))
33                  {
34                      myDraw.Background = SplashKit.RandomColor();
35                  }
36
37                  if (SplashKit.KeyDown(KeyCode.DeleteKey) ||
↪      SplashKit.KeyDown(KeyCode.BackspaceKey))
38                  {
39                      foreach (Shape s in myDraw.SelectedShapes)
40                      {
41                          myDraw.RemoveShape(s);
42                      }
43                  }
44
45                  SplashKit.RefreshScreen();
46              } while (!window.CloseRequested);
47          }
48      }
49  }
```

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

```
54         {
55             shape.Draw();
56         }
57     }
58     public void SelectedShapesAt(Point2D pt)
59     {
60         foreach (Shape s in _shapes)
61         {
62             if (s.IsAt(pt))
63             {
64                 s.Selected = true;
65             }
66             else
67             {
68                 s.Selected = false;
69             }
70         }
71     }
72
73     public void AddShape(Shape s)
74     {
75         _shapes.Add(s);
76     }
77
78     public void RemoveShape(Shape s)
79     {
80         _shapes.Remove(s);
81     }
82
83     }
84 }
```

```
1  using System;
2  using SplashKitSDK;
3
4  namespace ShapeDrawer
5  {
6      public class Shape
7      {
8          private Color _color;
9          private bool _selected;
10         private float _x;
11         private float _y;
12         private int _width;
13         private int _height;
14
15         public Shape()
16         {
17             _color = Color.Green;
18             _x = 0;
19             _y = 0;
20             _width = 100;
21             _height = 100;
22             _selected = false;
23         }
24
25         public Color Color
26         {
27             get
28             {
29                 return _color;
30             }
31
32             set
33             {
34                 _color = value;
35             }
36         }
37
38         public bool Selected
39         {
40             get
41             {
42                 return _selected;
43             }
44
45             set
46             {
47                 _selected = value;
48             }
49         }
50
51         public float X
52         {
53             get
```

```
54         {
55             return _x;
56         }
57
58         set
59         {
60             _x = value;
61         }
62     }
63
64     public float Y
65     {
66         get
67         {
68             return _y;
69         }
70
71         set
72         {
73             _y = value;
74         }
75     }
76
77     public int Width
78     {
79         get
80         {
81             return _width;
82         }
83
84         set
85         {
86             _width = value;
87         }
88     }
89
90     public int Height
91     {
92         get
93         {
94             return _height;
95         }
96
97         set
98         {
99             _height = value;
100         }
101     }
102
103     public void Draw()
104     {
105         SplashKit.FillRectangle(_color, _x, _y, _width, _height);
106         if (_selected)
```

```
107         {
108             this.DrawOutline();
109         }
110     }
111
112     public bool IsAt(Point2D point)
113     {
114         if (((point.X >= _x) && (point.X <= (_x + _width))) && (point.Y >= _y)
↪ && (point.Y <= (_y + _height))))
115             return true;
116         else
117             return false;
118     }
119
120     public void DrawOutline()
121     {
122         SplashKit.DrawRectangle(SplashKit.ColorBlack(), _x - 2, _y - 2, _width +
↪ 4, _height + 4);
123     }
124
125
126
127 }
128 }
```

