

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

---

## 3.3P - Drawing Program - A Drawing Class

---

PDF generated at 15:30 on Monday 10<sup>th</sup> April, 2023

```
1  using System;
2  using System.Collections.Generic;
3  using SplashKitSDK;
4
5
6  namespace DrawingClass
7  {
8      public class Program
9      {
10         public static void Main()
11         {
12             Window window = new Window("Drawing Class", 800, 600);
13             Drawing _drawing = new Drawing();
14
15             do
16             {
17                 SplashKit.ProcessEvents();
18                 SplashKit.ClearScreen();
19
20
21                 if (SplashKit.MouseClicked(MouseButton.LeftButton))
22                 {
23                     Shape _newshape = new Shape();
24                     _newshape.X = SplashKit.MouseX();
25                     _newshape.Y = SplashKit.MouseY();
26
27                     _drawing.AddShape(_newshape);
28
29                 }
30
31                 if (SplashKit.KeyDown(KeyCode.SpaceKey))
32                 {
33                     _drawing.Background = SplashKit.RandomRGBColor(255);
34                 }
35
36
37                 if (SplashKit.MouseClicked(MouseButton.RightButton))
38                 {
39                     _drawing.SelectShapesAt(SplashKit.MousePosition());
40                 }
41
42
43                 if (SplashKit.KeyDown(KeyCode.DeleteKey) ||
44 ↪ SplashKit.KeyDown(KeyCode.BackspaceKey))
45                 {
46                     foreach (Shape shape in _drawing.SelectedShapes)
47                     {
48                         _drawing.RemoveShape(shape);
49                     }
50
51                 }
52
53                 _drawing.Draw();
54             }
```

```
53         SplashKit.RefreshScreen();
54
55     } while (!window.CloseRequested);
56
57 }
58
59
60 }
61 }
```

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

```
54
55     public void Draw()
56     {
57         SplashKit.ClearScreen(Background);
58
59         foreach (Shape shape in _shapes)
60         {
61             shape.Draw();
62         }
63     }
64
65     public void SelectShapesAt(Point2D pt)
66     {
67         foreach (Shape s in _shapes)
68         {
69             if (s.IsAt(pt))
70             {
71                 s.Selected = true;
72             }
73
74             else
75             {
76                 s.Selected = false;
77             }
78         }
79     }
80
81
82     public void AddShape(Shape shape)
83     {
84         _shapes.Add(shape);
85     }
86
87     public void RemoveShape(Shape shape)
88     {
89         _shapes.Remove(shape);
90     }
91
92 }
93
94 }
95
```

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

```
54     }
55
56
57     public void Draw()
58     {
59         if (Selected == true)
60         {
61             DrawOutline();
62         }
63
64         SplashKit.FillRectangle(_color, _x, _y, _width, _height);
65     }
66
67
68     public void DrawOutline()
69     {
70         SplashKit.FillRectangle(Color.Black, _x - 2, _y - 2, _width + 4, _height
↵ + 4);
71     }
72
73
74     public bool IsAt(Point2D pt)
75     {
76         if (_x < pt.X && pt.X < (_x + _width) && _y < pt.Y && pt.Y < (_y +
↵ _height))
77         {
78             return true;
79         }
80         else
81         {
82             return false;
83         }
84
85     }
86
87 }
88 }
89
```

