

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Drawing Program - A Drawing Class

PDF generated at 13:48 on Tuesday 29th 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 myDrawing = new Drawing();
11
12              new Window("Drawing Shape", 800, 600);
13              do
14              {
15                  SplashKit.ProcessEvents();
16                  SplashKit.ClearScreen();
17
18                  if (SplashKit.MouseClicked(MouseButton.LeftButton))
19                  {
20                      myDrawing.AddShape(new Shape());
21                  }
22
23                  if (SplashKit.MouseClicked(MouseButton.RightButton))
24                  {
25                      myDrawing.SelectedShapeAt(SplashKit.MousePosition());
26                  }
27
28                  if (SplashKit.KeyTyped(KeyCode.BackspaceKey) ||
↪      SplashKit.KeyTyped(KeyCode.DeleteKey))
29                  {
30                      myDrawing.RemoveShape();
31                  }
32
33                  if (SplashKit.KeyTyped(KeyCode.SpaceKey))
34                  {
35                      myDrawing.Background = SplashKit.RandomRGBColor(255);
36                  }
37
38                  myDrawing.Draw();
39
40                  SplashKit.RefreshScreen();
41
42              } while (!SplashKit.WindowCloseRequested("Drawing Shape"));
43          }
44      }
45  }
```

```
1  using System;
2  using System.Linq;
3  using System.Collections.Generic;
4  using SplashKitSDK;
5
6  namespace ShapeDrawer
7  {
8      public class Drawing
9      {
10         private readonly List<Shape> _shapes;
11         private Color _background;
12
13         public Drawing(Color background)
14         {
15             _shapes = new List<Shape>();
16             _background = background;
17         }
18
19         public Drawing() : this(Color.White)
20         {
21         }
22
23         public List<Shape> SelectedShapes()
24         {
25             List<Shape> _selectedShapes = new List<Shape>();
26             foreach (Shape s in _selectedShapes)
27             {
28                 if (s.Selected)
29                 {
30                     _selectedShapes.Add(s);
31                 }
32             }
33             return _selectedShapes;
34         }
35
36         public int ShapeCount
37         {
38             get
39             {
40                 return _shapes.Count;
41             }
42         }
43
44         public Color Background
45         {
46             get
47             {
48                 return _background;
49             }
50             set
51             {
52                 _background = value;
53             }
54         }
55     }
```

```
54     }
55
56     public void Draw()
57     {
58         SplashKit.ClearScreen(_background);
59
60         foreach (Shape s in _shapes)
61         {
62             s.Draw();
63         }
64     }
65
66     public void SelectedShapeAt(Point2D pt)
67     {
68         foreach (Shape s in _shapes)
69         {
70             if (s.IsAt(pt))
71             {
72                 s.Selected = true;
73             }
74             else
75             {
76                 s.Selected = false;
77             }
78         }
79     }
80
81     public void AddShape(Shape s)
82     {
83         _shapes.Add(s);
84     }
85
86     public void RemoveShape()
87     {
88         foreach (Shape s in _shapes.ToList())
89         {
90             if (s.Selected)
91             {
92                 _shapes.Remove(s);
93             }
94         }
95     }
96 }
97 }
```

```
1  using System;
2  using SplashKitSDK;
3
4  namespace ShapeDrawer
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 Shape()
14         {
15             _color = Color.Black;
16             _x = SplashKit.MouseX();
17             _y = SplashKit.MouseY();
18             _width = _height = 100;
19         }
20
21         public Color Color
22         {
23             get
24             {
25                 return _color;
26             }
27             set
28             {
29                 _color = value;
30             }
31         }
32
33         public float X
34         {
35             get
36             {
37                 return _x;
38             }
39             set
40             {
41                 _x = value;
42             }
43         }
44
45         public float Y
46         {
47             get
48             {
49                 return _y;
50             }
51             set
52             {
53                 _y = value;
```

```
54         }
55     }
56
57     public void Draw()
58     {
59         if (Selected)
60         {
61             DrawOutline();
62         }
63         SplashKit.FillRectangle(Color, X, Y, _width, _height);
64     }
65
66     public bool IsAt(Point2D p)
67     {
68         return SplashKit.PointInRectangle(p, SplashKit.RectangleFrom(X, Y,
↵ _width, _height));
69     }
70
71     public bool Selected
72     {
73         get
74         {
75             return _selected;
76         }
77         set
78         {
79             _selected = value;
80         }
81     }
82
83     public void DrawOutline()
84     {
85         SplashKit.FillRectangle(Color, X - 2, Y - 2, _width + 4, _height + 4);
86     }
87 }
88 }
```

