

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Drawing Program - A Drawing Class

PDF generated at 14:22 on Tuesday 15th August, 2023

```
1  using System;
2  using SplashKitSDK;
3
4  namespace DrawingProgram
5  {
6      public class Program
7      {
8          public static void Main()
9          {
10              Drawing drawing = new Drawing();
11              Window window = new Window("Shape Drawing 3.3P", 800, 600);
12
13              while(!window.CloseRequested)
14              {
15                  SplashKit.ProcessEvents();
16                  SplashKit.ClearScreen();
17                  //Draw Shape when click Left Button
18                  if (SplashKit.MouseClicked(MouseButton.LeftButton))
19                  {
20                      int xPosition = (int) SplashKit.MouseX();
21                      int yPosition = (int) SplashKit.MouseY();
22                      drawing.AddShape(new Shape(xPosition, yPosition, Color.Black));
23                  }
24                  //Draw Outline (Selected shape) when click Right Button
25                  if(SplashKit.MouseClicked(MouseButton.RightButton))
26                  {
27                      drawing.SelectShapesAt(SplashKit.MousePosition());
28                  }
29                  //Randomly change color when click ESC
30                  if(SplashKit.KeyDown(KeyCode.EscapeKey))
31                  {
32                      foreach(Shape s in drawing.SelectedShapes)
33                      {
34                          s.color = SplashKit.RandomRGBColor(255);
35                      }
36                  }
37                  //Change background color when click Space
38                  if(SplashKit.KeyDown(KeyCode.SpaceKey))
39                  {
40                      drawing.Background = SplashKit.RandomRGBColor(255);
41                  }
42                  //Remove shape if press DEL or Backspace
43                  if (SplashKit.KeyDown(KeyCode.DeleteKey) ||
44 ↪ SplashKit.KeyDown(KeyCode.BackspaceKey))
45                  {
46                      foreach(Shape s in drawing.SelectedShapes)
47                      {
48                          drawing.RemoveShape(s);
49                      }
50                  }
51                  drawing.Draw();
52                  SplashKit.RefreshScreen();
53              }
54          }
55      }
56  }
```

```
53         }  
54     }  
55 }
```

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

```
54
55     public void Draw()
56     {
57         SplashKit.ClearScreen(_background);
58         foreach(Shape s in _shapes)
59         {
60             s.Draw();
61         }
62     }
63
64     public void SelectShapesAt(Point2D point)
65     {
66         foreach(Shape s in _shapes)
67         {
68             if (s.IsAt(point))
69                 s.Selected = true;
70             else
71                 s.Selected = false;
72         }
73     }
74     public void AddShape(Shape s)
75     {
76         _shapes.Add(s);
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 DrawingProgram
5  {
6      public class Shape
7      {
8          private Color _color;
9          private float _x, _y;
10         private bool _selected;
11         private int _width, _height;
12
13         public Shape(int x, int y, Color color)
14         {
15             _color = color;
16             _x = x;
17             _selected = false;
18             _y = y;
19             _width = 100;
20             _height = 100;
21
22         }
23
24         public Shape() : this(0, 0, SplashKit.RGBColor(255, 0, 255)) { }
25
26         public Color color
27         {
28             get
29             {
30                 return _color;
31             }
32             set
33             {
34                 _color = value;
35             }
36         }
37
38         public float X
39         {
40             get
41             {
42                 return _x;
43             }
44             set
45             {
46                 _x = value;
47             }
48         }
49
50         public float Y
51         {
52             get
53             {
```

```
54         return _y;
55     }
56     set
57     {
58         _y = value;
59     }
60 }
61
62 public int Width
63 {
64     get
65     {
66         return _width;
67     }
68     set
69     {
70         _width = value;
71     }
72 }
73
74 public int Height
75 {
76     get
77     {
78         return _height;
79     }
80     set
81     {
82         _height = value;
83     }
84 }
85
86 public bool Selected
87 {
88     get
89     {
90         return _selected;
91     }
92     set
93     {
94         _selected = value;
95     }
96 }
97
98 public void Draw()
99 {
100     if (Selected)
101         DrawOutline();
102     SplashKit.FillRectangle(_color, _x, _y, _width, _height);
103 }
104
105 public bool IsAt(Point2D pt)
106 {
```

```
107         return SplashKit.PointInRectangle(pt, SplashKit.RectangleFrom(_x, _y,
↪      _width, _height));
108     }
109
110     public void DrawOutline()
111     {
112         SplashKit.DrawRectangle(Color.Black, _x - 2, _y - 2, _width + 4, _height
↪      + 4);
113     }
114 }
115 }
116
```


