Selected files

6 printable files

39

```
Week_4/4.1/MultipleShape/Drawing.cs
Week_4/4.1/MultipleShape/MyCircle.cs
Week_4/4.1/MultipleShape/MyLine.cs
Week_4/4.1/MultipleShape/MyRectangle.cs
Week_4/4.1/MultipleShape/Program.cs
Week_4/4.1/MultipleShape/Program.cs
Week_4/4.1/MultipleShape/Shape.cs

Week_4/4.1/MultipleShape/Drawing.cs

| using System;
| using System.Collections.Generic;
```

```
using SplashKitSDK;
 3 |
   using System.Ling;
   using System.Threading.Tasks;
 5
 6
7
   namespace MultipleShape
8
   {
9
        public class Drawing
10
            private readonly List<Shape> _shapes;
11
            private Color _background;
12
13
14
            // Constructor
            public Drawing(Color background)
15
16
                _shapes = new List<Shape>();
17
                _background = background;
18
19
            }
20
            public Drawing() : this(Color.White)
21
22
            {
            }
23
24
25
            // Properties
26
            public Color Background
27
            {
                get { return _background; }
28
29
                set { _background = value; }
            }
30
31
32
            public int ShapeCount
33
            {
                get { return _shapes.Count; }
34
35
            }
36
37
            public List<Shape> SelectedShapes
38
```

get

```
{
40
41
                     List<Shape> result = new List<Shape>();
                     foreach (Shape s in _shapes)
42
43
                     {
44
                         if (s.Selected)
45
                         {
46
                              result.Add(s);
47
                         }
48
                     }
49
                     return result;
                }
50
            }
51
52
53
            // Methods
            public void AddShape(Shape s)
54
55
            {
56
                _shapes.Add(s);
57
            }
58
            public void RemoveShape(Shape s)
59
60
61
                _ = _shapes.Remove(s);
62
            }
63
            public void Draw()
64
65
                SplashKit.ClearScreen(_background);
66
                foreach (Shape s in _shapes)
67
                 {
68
                     s.Draw();
69
70
                 }
71
            }
72
            public void SelectShapesAt(Point2D pt)
73
74
75
                 foreach (Shape s in _shapes)
76
77
                     s.Selected = s.IsAt(pt);
78
                 }
            }
79
80
            public void RemoveSelectedShapes()
81
82
                 foreach (Shape s in SelectedShapes)
83
84
                     RemoveShape(s);
85
86
                 }
            }
87
88
        }
89 }
```

Week_4/4.1/MultipleShape/MyCircle.cs

```
using System;
   using System.Collections.Generic;
 3 using System.Linq;
   using System.Threading.Tasks;
 4
   using SplashKitSDK;
 6
 7
 8
   namespace MultipleShape
 9
        public class MyCircle : Shape
10
11
12
            private float _radius;
13
            public MyCircle() : this(Color.Blue, 0.0f, 0.0f, 129)
14
15
            {
            }
16
17
            public MyCircle(Color color, float x, float y, float radius) : base(color)
18
19
20
                X = X;
21
                Y = y;
22
                Radius = radius;
23
            }
24
            public float Radius
25
26
            {
27
                get => _radius;
28
                set => _radius = value;
29
            }
30
31
            public override void Draw()
32
            {
33
                SplashKit.FillCircle(Color, X, Y, _radius);
34
35
                if (Selected)
36
                    DrawOutline();
37
38
                }
            }
39
40
41
            public override bool IsAt(Point2D pt)
42
            {
43
                Circle c = SplashKit.CircleAt(X, Y, _radius);
44
                return SplashKit.PointInCircle(pt, c);
45
            }
46
47
            public override void DrawOutline()
48
            {
49
                SplashKit.DrawCircle(Color.Black, X, Y, _radius + 2);
```

```
50 }
51 }
52 }
```

Week_4/4.1/MultipleShape/MyLine.cs

```
using System;
 2
   using System.Collections.Generic;
   using System.Linq;
   using System.Threading.Tasks;
 5
   using SplashKitSDK;
 6
 7
   namespace MultipleShape
 8
   {
 9
        public class MyLine : Shape
        {
10
            private float _endX, _endY;
11
12
            public MyLine() : this(Color.Red, 0.0f, 0.0f, 88, 88)
13
            {
14
            }
15
16
17
            public MyLine(Color color, float x, float y, float endX, float endY) :
   base(color)
            {
18
19
                X = X;
                Y = y;
20
                EndX = endX;
21
                EndY = endY;
22
            }
23
24
            public float EndX
25
26
27
                get => _endX;
28
                set => _endX = value;
29
            }
30
31
            public float EndY
32
            {
33
                get => _endY;
34
                set => _endY = value;
            }
35
36
            public override void Draw()
37
38
            {
                SplashKit.DrawLine(Color, X, Y, EndX, EndY);
39
40
41
                if (Selected)
42
43
                     DrawOutline();
44
                 }
```

```
45
            }
46
47
            public override bool IsAt(Point2D pt)
48
            {
49
                return SplashKit.PointOnLine(pt, SplashKit.LineFrom(X, Y, EndX, EndY));
            }
50
51
            public override void DrawOutline()
52
53
                SplashKit.DrawCircle(Color.Black, X, Y, 5);
54
55
                SplashKit.DrawCircle(Color.Black, EndX, EndY, 5);
56
            }
        }
57
58 }
Week_4/4.1/MultipleShape/MyRectangle.cs
    using System;
    using System.Collections.Generic;
   using SplashKitSDK;
 3
    using System.Linq;
 5
```

```
using System.Threading.Tasks;
 7
   namespace MultipleShape
   {
 8
 9
        public class MyRectangle : Shape
        {
10
            private float _width, _height;
11
12
            public MyRectangle() : this(Color.Green, 0.0f, 0.0f, 194, 194)
13
            {
14
            }
15
16
17
            public MyRectangle(Color color, float x, float y, float width, float height)
    : base(color)
            {
18
19
                X = X;
20
                Y = y;
21
                Width = width;
                Height = height;
22
            }
23
24
25
            public float Width
26
27
                get => _width;
28
                 set => _width = value;
29
            }
30
31
            public float Height
32
            {
33
                get => _height;
34
                 set => _height = value;
```

```
35
            }
36
37
            // Methods
            public override void Draw()
38
39
40
                SplashKit.FillRectangle(Color, X, Y, Width, Height);
41
42
                if (Selected)
43
                     DrawOutline();
44
                 }
45
            }
46
47
48
            public override bool IsAt(Point2D pt)
49
50
                 return (pt.X >= X) && (pt.X <= (X + _width))
                     && (pt.Y >= Y) && (pt.Y <= (Y + _{height}));
51
            }
52
53
54
            public override void DrawOutline()
55
56
                SplashKit.DrawRectangle(Color.Black, X - 9, Y - 9, Width + 18, Height +
    18);
57
            }
58
        }
59 }
Week_4/4.1/MultipleShape/Program.cs
```

```
using System;
   using MultipleShape;
 2
   using SplashKitSDK;
 3
 4
 5
   namespace MultipleShape
 6
   {
 7
        public class Program
        {
 8
 9
            private enum ShapeKind
10
            {
                Rectangle,
11
12
                Circle,
13
                Line
            }
14
15
16
            public static void Main()
17
            {
18
                Window window = new Window("Shape Drawer", 800, 600);
19
                Drawing myDrawing = new Drawing();
20
21
                ShapeKind kindToAdd = ShapeKind.Circle;
22
23
                // My ID: 104844794 => Last digit: 4
```

```
24
                // So I'm only able to draw a maximum of X lines within the timeframe
25
                int maxLines = 4;
26
                do
27
28
                {
29
                     SplashKit.ProcessEvents();
30
                     SplashKit.ClearScreen();
31
32
                     if (SplashKit.KeyTyped(KeyCode.RKey))
33
                     {
34
                         kindToAdd = ShapeKind.Rectangle;
35
                     }
36
37
                     // If the user presses the C key or has run out of lines to draw,
    they will draw circles
38
                     if (SplashKit.KeyTyped(KeyCode.CKey) || maxLines <= 0)</pre>
39
                     {
                         kindToAdd = ShapeKind.Circle;
40
41
                     }
42
                     // If the user presses the L key and has lines left to draw, they
43
   will draw lines
44
                     if (SplashKit.KeyTyped(KeyCode.LKey) && maxLines > 0)
45
                     {
46
                         kindToAdd = ShapeKind.Line;
47
                     }
48
                     if (SplashKit.MouseClicked(MouseButton.LeftButton))
49
50
51
                         Shape myShape;
52
                         switch (kindToAdd)
53
54
55
                             case ShapeKind.Circle:
56
                                 myShape = new MyCircle();
57
                                 break;
58
                             case ShapeKind.Line:
59
                                 myShape = new MyLine();
60
                                 maxLines--;
61
                                 break;
62
                             default:
                                 myShape = new MyRectangle();
63
64
                                 break;
                         }
65
66
67
                         myShape.X = SplashKit.MouseX();
68
                         myShape.Y = SplashKit.MouseY();
69
                         myDrawing.AddShape(myShape);
70
                     }
71
72
                     if (SplashKit.KeyTyped(KeyCode.SpaceKey))
```

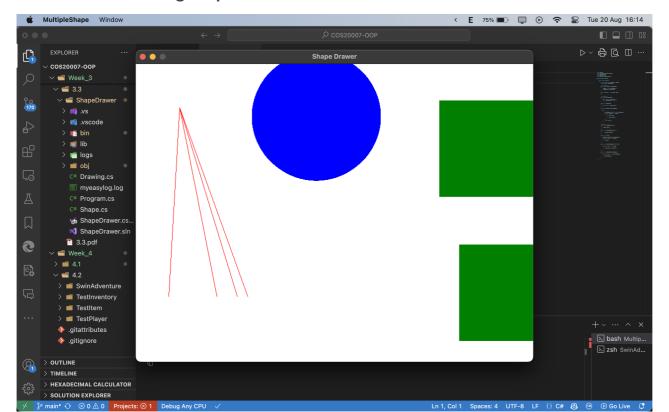
```
73
                     {
74
                         myDrawing.Background = SplashKit.RandomRGBColor(255);
75
                     }
76
77
                     if (SplashKit.MouseClicked(MouseButton.RightButton))
78
79
                         Point2D pt = new Point2D();
80
                         pt.X = SplashKit.MouseX();
81
                         pt.Y = SplashKit.MouseY();
82
                         myDrawing.SelectShapesAt(pt);
83
                     }
84
85
86
                     if (SplashKit.KeyTyped(KeyCode.DeleteKey) ||
    SplashKit.KeyTyped(KeyCode.BackspaceKey))
                     {
87
88
                         myDrawing.RemoveSelectedShapes();
89
                     }
90
91
                     myDrawing.Draw();
                     SplashKit.RefreshScreen();
92
93
                } while (!window.CloseRequested);
            }
94
95
        }
96
   }
97
```

Week_4/4.1/MultipleShape/Shape.cs

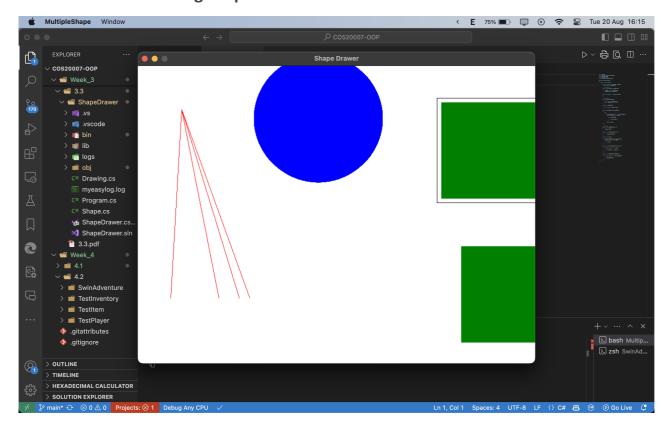
```
1
    using SplashKitSDK;
 2
 3
   namespace MultipleShape
 4
   {
 5
        public abstract class Shape
 6
 7
            // Fields
 8
             private Color _color;
 9
             private float _x, _y;
             private bool _selected;
10
11
12
             // Constructor
13
            public Shape() : this(Color.Yellow)
14
             {
15
             }
16
            public Shape(Color color)
17
18
             {
19
                 _color = color;
20
                 _{x} = 0.0f;
                 _{y} = 0.0f;
21
22
                 _selected = false;
23
             }
```

```
24
25
            // Properties
26
            public float X
27
            {
28
                 get { return _x; }
29
                 set { _x = value; }
30
            }
31
32
            public float Y
33
            {
34
                 get { return _y; }
35
                 set { _y = value; }
            }
36
37
            public Color Color
38
39
            {
                 get { return _color; }
40
                 set { _color = value; }
41
42
            }
43
44
            public bool Selected
45
            {
46
                 get { return _selected; }
                 set { _selected = value; }
47
            }
48
49
50
            // Methods
51
            public abstract void Draw();
            public abstract void DrawOutline();
52
53
            public abstract bool IsAt(Point2D pt);
        }
54
55
    }
```

Screenshot of adding shapes:



Screenshot of selecting shape:



Screenshot of deleting shape:

