

Selected files

3 printable files

Week 3\3.2\Program.cs

Week 3\3.2\Shape.cs

Week 3\3.2\Drawing.cs

Week 3\3.2\Program.cs

```
1  using System;
2  using SplashKitSDK;
3
4  namespace ShapeDrawer
5  {
6      public class Program
7      {
8          public static void Main()
9          {
10              Window window = new Window("Shape Drawer", 800, 600);
11              Drawing myDrawing = new Drawing();
12
13              do
14              {
15                  SplashKit.ProcessEvents();
16                  SplashKit.ClearScreen();
17
18                  if (SplashKit.MouseClicked(MouseButton.LeftButton))
19                  {
20                      Shape myShape = new Shape(50);
21                      myShape.X = SplashKit.MouseX();
22                      myShape.Y = SplashKit.MouseY();
23                      myDrawing.AddShape(myShape);
24                  }
25
26                  if (SplashKit.KeyTyped(KeyCode.SpaceKey))
27                  {
28                      myDrawing.Background = SplashKit.RandomRGBColor(255);
29                  }
30
31                  if (SplashKit.MouseClicked(MouseButton.RightButton))
32                  {
33                      Point2D pt = new Point2D();
34                      pt.X = SplashKit.MouseX();
35                      pt.Y = SplashKit.MouseY();
36
37                      myDrawing.SelectShapesAt(pt);
38                  }
39
40                  if (SplashKit.KeyTyped(KeyCode.DeleteKey) ||
41                      SplashKit.KeyTyped(KeyCode.BackspaceKey))
```

```

41         {
42             myDrawing.RemoveSelectedShapes();
43         }
44
45         myDrawing.Draw();
46         SplashKit.RefreshScreen();
47     } while (!window.CloseRequested);
48 }
49 }
50 }
51

```

Week 3\3.2\Shape.cs

```

1  using SplashKitSDK;
2
3  namespace ShapeDrawer
4  {
5      public class Shape
6      {
7          // Fields
8          private Color _color;
9          private float _x, _y;
10         private int _width, _height;
11         private bool _selected;
12
13         // Constructor
14         public Shape(int param)
15         {
16             _color = Color.Chocolate;
17             _x = 0.0f;
18             _y = 0.0f;
19             _width = param;
20             _height = param;
21             _selected = false;
22         }
23
24         // Methods
25         public void Draw()
26         {
27             SplashKit.FillRectangle(_color, _x, _y, _width, _height);
28
29             if (_selected)
30             {
31                 DrawOutline();
32             }
33         }
34
35         public bool IsAt(Point2D pt)
36         {
37             return (pt.X >= _x) && (pt.X <= (_x + _width))

```

```

38         && (pt.Y >= _y) && (pt.Y <= (_y + _height));
39     }
40
41     public void DrawOutline()
42     {
43         SplashKit.DrawRectangle(Color.Black, _x - 9, _y - 9, _width + 18, _height + 18);
44     }
45
46     // Properties
47     public float X
48     {
49         get { return _x; }
50         set { _x = value; }
51     }
52
53     public float Y
54     {
55         get { return _y; }
56         set { _y = value; }
57     }
58
59     public Color Color
60     {
61         get { return _color; }
62         set { _color = value; }
63     }
64
65     public int Width
66     {
67         get { return _width; }
68         set { _width = value; }
69     }
70
71     public int Height
72     {
73         get { return _height; }
74         set { _height = value; }
75     }
76
77     public bool Selected
78     {
79         get { return _selected; }
80         set { _selected = value; }
81     }
82 }
83 }

```

Week 3\3.2\Drawing.cs

```

1 using System;
2 using System.Collections.Generic;

```

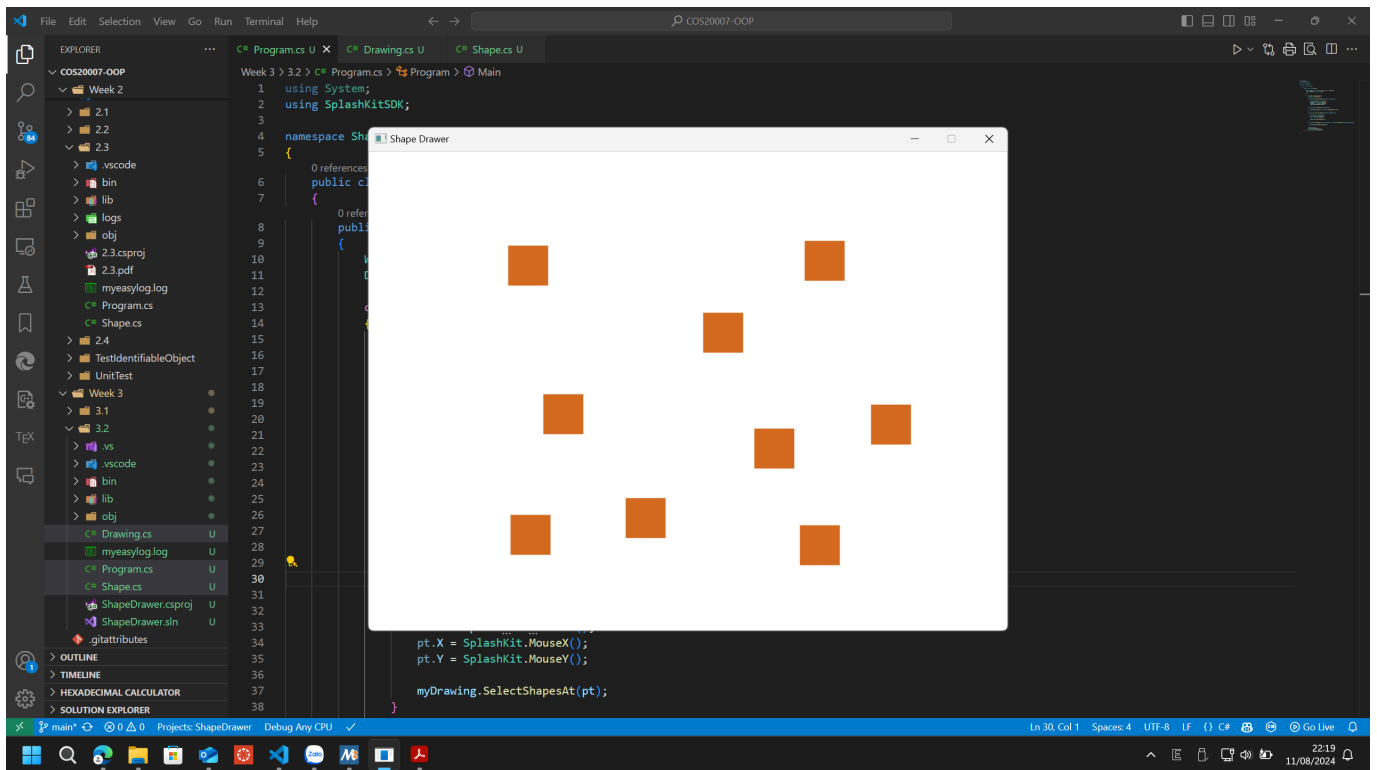
```
3 using SplashKitSDK;
4 using System.Linq;
5 using System.Threading.Tasks;
6
7 namespace ShapeDrawer
8 {
9     public class Drawing
10    {
11        private readonly List<Shape> _shapes;
12        private Color _background;
13
14        // Constructor
15        public Drawing(Color background)
16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

```

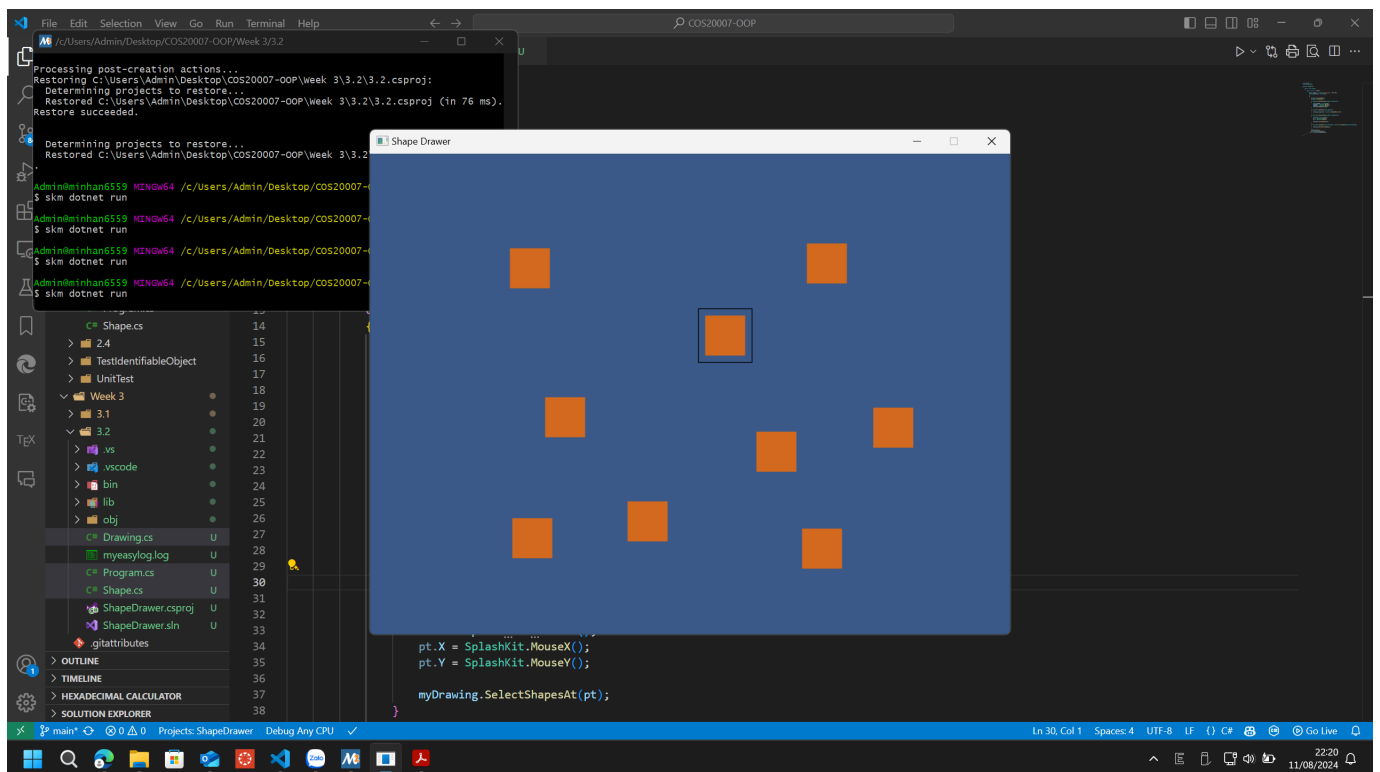
53 // Methods
54 public void AddShape(Shape s)
55 {
56     _shapes.Add(s);
57 }
58
59 public void RemoveShape(Shape s)
60 {
61     _ = _shapes.Remove(s);
62 }
63
64 public void Draw()
65 {
66     SplashKit.ClearScreen(_background);
67     foreach (Shape s in _shapes)
68     {
69         s.Draw();
70     }
71 }
72
73 public void SelectShapesAt(Point2D pt)
74 {
75     foreach (Shape s in _shapes)
76     {
77         s.Selected = s.IsAt(pt);
78     }
79 }
80
81 public void RemoveSelectedShapes()
82 {
83     foreach (Shape s in SelectedShapes)
84     {
85         RemoveShape(s);
86     }
87 }
88 }
89 }

```

Screenshot of adding shapes:



Screenshot of selecting shapes and changing the background:



Screenshot of removing shape:

