Swinburne University of Technology

COS20007 Object Oriented Programming

Drawing Program - A Drawing Class

PDF generated at 14:59 on Thursday 31st August, 2023

```
using System;
using SplashKitSDK;

namespace ShapeDrawer
{
    public class Program
    {
        public static void Main()
        {
            // Create a window for the shape drawer
            Window window = new Window("Shape Drawer", 800, 600);
            Drawing myDrawing = new Drawing(Color.White);
            do
            {
                // Processes events from user input
                SplashKit.ProcessEvents();
                SplashKit.ClearScreen();

                Point2D pt = SplashKit.MousePosition();

                // Set x to be mouseX and y to be MouseY if mouse clicked is true
                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape newShape = new Shape();
                    newShape.X = SplashKit.MouseX();
                    newShape.Y = SplashKit.MouseY();
                    myDrawing.AddShape(newShape);
                }

                // Select shapes at the mouse position when right mouse button is
    ↪  clicked
                if (SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    myDrawing.SelectShapesAt(pt);
                }

                // Change the background color to a random color when Space key is
    ↪  typed
                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
                {
                    myDrawing.Background = Color.RandomRGB(255);
                }

                // Remove selected shapes when Backspace key is typed
                if (SplashKit.KeyTyped(KeyCode.BackspaceKey))
                {
                    List<Shape> selectedShapes = myDrawing.SelectedShapes;
                    foreach (Shape shapeToRemove in selectedShapes)
                    {
                        myDrawing.RemoveShape(shapeToRemove);
                    }
                }
                myDrawing.Draw();
```

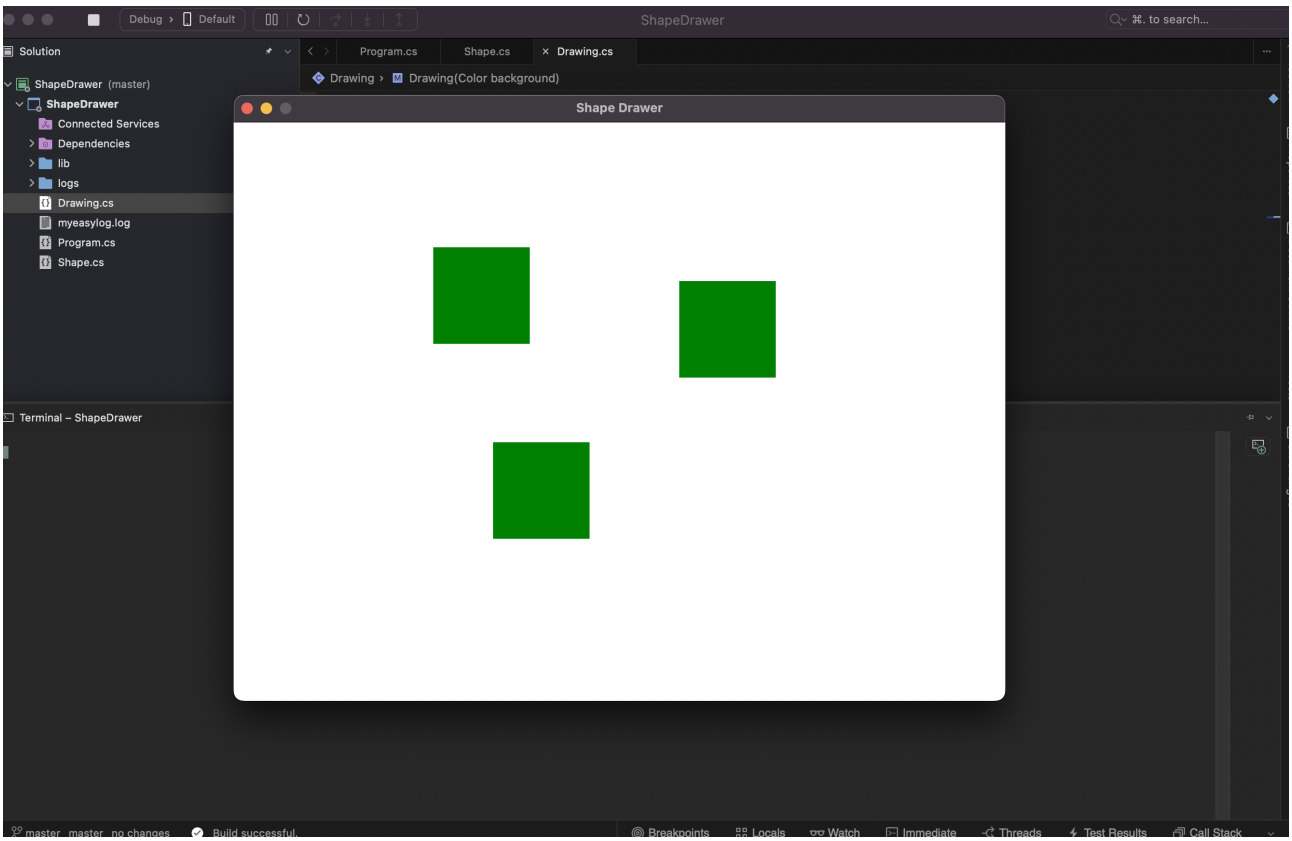
```
52                  SplashKit.RefreshScreen();
53              } while (!window.CloseRequested);
54          }
55      }
56  }
57
```

```
using System;
using System.Collections.Generic;
using ShapeDrawer;
using SplashKitSDK;

namespace ShapeDrawer
{
    public class Drawing
    {
        private readonly List<Shape> _shapes;
        private Color _background;

        public Drawing(Color background)
        {
            _shapes = new List<Shape>();
            _background = background;
        }

        public Drawing() : this(Color.White)
        {
        }

        public int ShapeCount
        {
            get { return _shapes.Count; }
        }

        public Color Background
        {
            get { return _background; }
            set { _background = value; }
        }

        public List<Shape> SelectedShapes
        {
            get
            {
                List<Shape> selectedShapes = new List<Shape>();
                foreach (Shape shape in _shapes)
                {
                    if (shape.Selected)
                    {
                        selectedShapes.Add(shape);
                    }
                }
                return selectedShapes;
            }
        }

        public void Draw()
        {
            SplashKit.ClearScreen(_background);
            foreach (Shape shape in _shapes)
```

```
54              {
55                  shape.Draw();
56              }
57          }
58
59          public void SelectShapesAt(Point2D pt)
60          {
61              foreach (Shape shape in _shapes)
62              {
63                  shape.Selected = shape.IsAt(pt);
64              }
65          }
66
67          public void AddShape(Shape shape)
68          {
69              _shapes.Add(shape);
70          }
71
72          public void RemoveShape(Shape shape)
73          {
74              _shapes.Remove(shape);
75          }
76      }
77  }
78
```

```
using System;
using SplashKitSDK;

namespace ShapeDrawer
{
    public class Shape
    {
        private Color _color;
        private float _x;
        private float _y;
        private int _width;
        private int _height;
        private bool _selected;

        public Shape()
        {
            _color = Color.Green;
            _x = 0;
            _y = 0;
            _width = 100;
            _height = 100;
        }

        public Color Color
        {
            get { return _color; }
            set { _color = value; }
        }

        public float X
        {
            get { return _x; }
            set { _x = value; }
        }

        public float Y
        {
            get { return _y; }
            set { _y = value; }
        }

        public int Width
        {
            get { return _width; }
            set { _width = value; }
        }

        public int Height
        {
            get { return _height; }
            set { _height = value; }
        }

```

```
54          public void Draw()
55          {
56              SplashKit.FillRectangle(_color, _x, _y, _width, _height);
57              DrawOutline();
58          }
59
60          public bool IsAt(Point2D pt)
61          {
62              if (_x < pt.X && pt.X < (_x + _width) && _y < pt.Y && pt.Y < (_y +
    ↪  _height))
63              {
64                  return true;
65              }
66              else
67              {
68                  return false;
69              }
70          }
71
72          public bool Selected
73          {
74              get { return _selected; }
75              set { _selected = value; }
76          }
77
78          public void DrawOutline()
79          {
80              if (Selected)
81              {
82                  SplashKit.DrawRectangle(Color.Black, _x - 2, _y - 2, _width + 4,
    ↪  _height + 4);
83              }
84          }
85      }
86  }
87
88
89
90
```


Debug
Default
ShapeDrawer
Solution
Program.cs
Shape.cs
Drawing.cs
Drawing
Drawing(Color background)
ShapeDrawer (master)
ShapeDrawer
Connected Services
Dependencies
lib
logs
Drawing.cs
myeasylog.log
Program.cs
Shape.cs
Shape Drawer
Terminal – ShapeDrawer
Breakpoints
Locals
Watch
Immediate
Threads
Test Results
Call Stack