Swinburne University of Technology

COS20007 Object Oriented Programming

Drawing Program - A Drawing Class

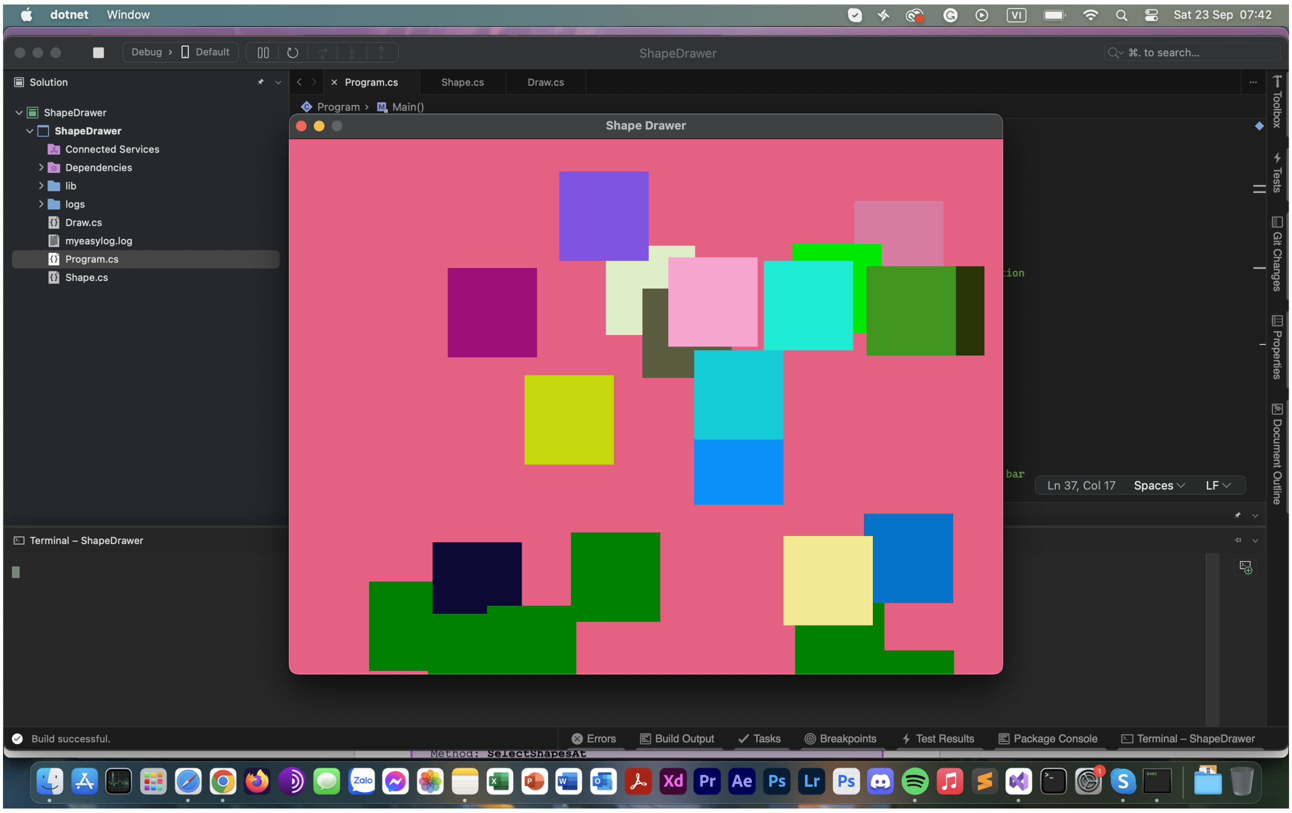
PDF generated at 10:45 on Saturday 23rd September, 2023

Produced by OOP Submission Formatter

```
using System;
using SplashKitSDK;

namespace ShapeDrawer
{
    public class Program
    {
        public static void Main()
        {
            Window window = new Window("Shape Drawer", 800, 600);
            Drawing drawing = new Drawing();

            do
            {
                SplashKit.ProcessEvents();
                SplashKit.ClearScreen();

                drawing.Draw();

                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape newShape = new Shape();
                    newShape.X = SplashKit.MouseX();
                    newShape.Y = SplashKit.MouseY();
                    drawing.AddShape(newShape);
                }

                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
                {
                    foreach (Shape shape in drawing.SelectedShapes)
                    {
                        shape.Color = SplashKit.RandomRGBColor(255);
                    }
                }

                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
                {
                    drawing.Background = SplashKit.RandomRGBColor(255);
                }

                if (SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    drawing.SelectShapesAt(SplashKit.MousePosition());
                }

                if (SplashKit.KeyTyped(KeyCode.DeleteKey) || SplashKit.KeyTyped(KeyCode.BackspaceKey))
                {
                    foreach (Shape shape in drawing.SelectedShapes)
                    {
                        drawing.RemoveShape(shape);
                    }
                }

                SplashKit.RefreshScreen();
            } while (!window.CloseRequested);
        }
    }
}
```

```
using System.Collections.Generic;
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 Color Background
        {
            get { return _background; }
            set { _background = value; }
        }

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

        public void AddShape(Shape shape)
        {
            _shapes.Add(shape);
        }

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

        public void SelectShapesAt(Point2D pt)
        {
            foreach (Shape shape in _shapes)
            {
                shape.Selected = shape.IsAt(pt);
            }
        }

        internal void RemoveShape(Shape shape)
        {
            throw new NotImplementedException();
        }

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

```
using System;
using SplashKitSDK;

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

        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; }
        }

        public void Draw()
        {
            SplashKit.FillRectangle(_color, _x, _y, _width, _height);
        }

        public bool IsAt(Point2D pt)
        {
            return (pt.X >= _x && pt.X <= _x + _width && pt.Y >= _y && pt.Y <= _y + _height);
        }
    }
}
```



Build successful.