Program.cs

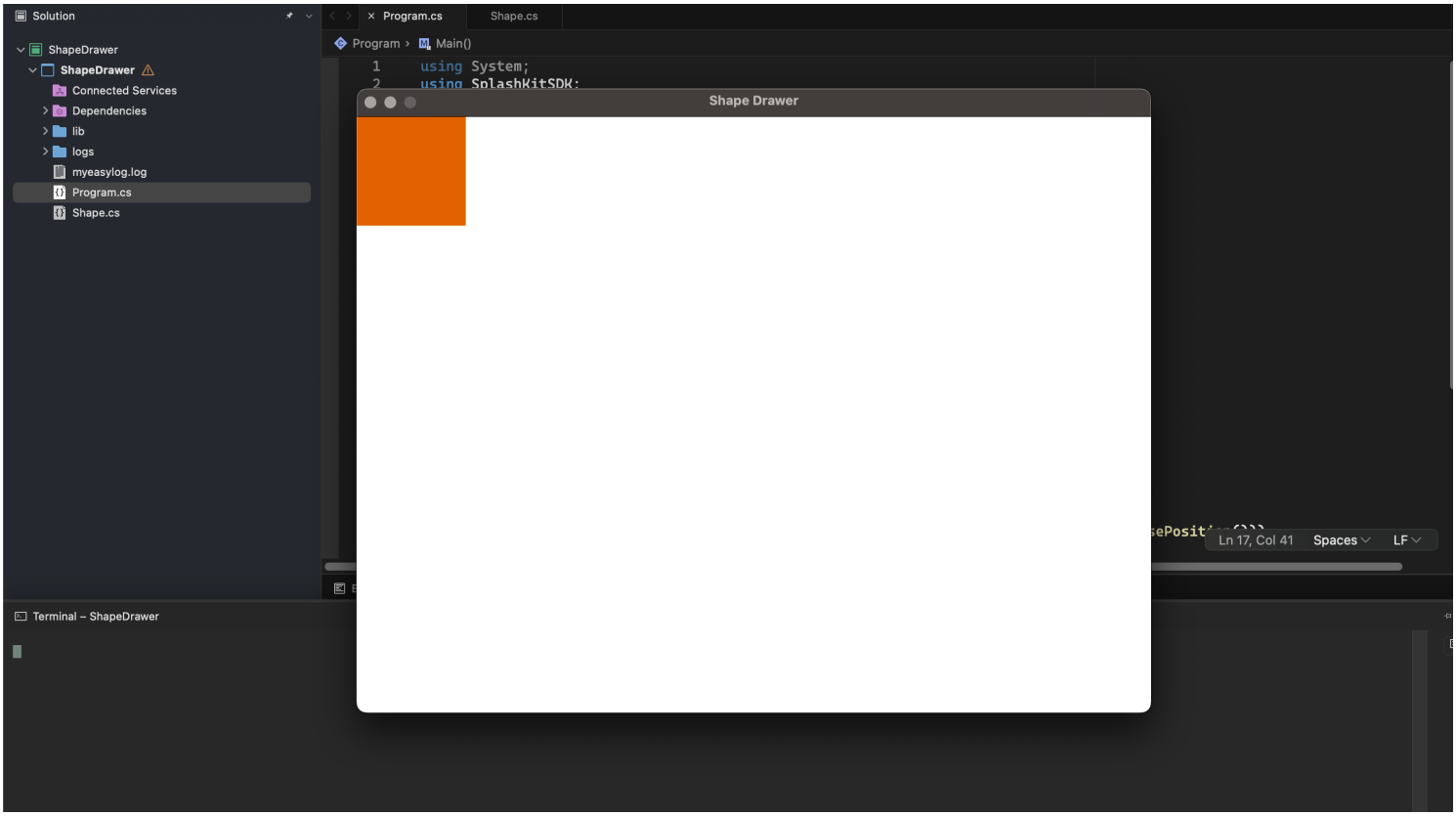
```
using System;
using SplashKitSDK;

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

            Shape myShape = new Shape(110);

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

                myShape.Draw();

                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    myShape.X = SplashKit.MouseX();
                    myShape.Y = SplashKit.MouseY();
                }

                if (SplashKit.KeyTyped(KeyCode.SpaceKey) &&
myShape.IsAt(SplashKit.MousePosition()))
                {
                    myShape.Color = SplashKit.RandomRGBColor(255);
                }

                myShape.Draw();

                SplashKit.RefreshScreen();
            }

            while (!window.CloseRequested);

            myShape.Draw();
        }
    }
}

```

Shape.cs

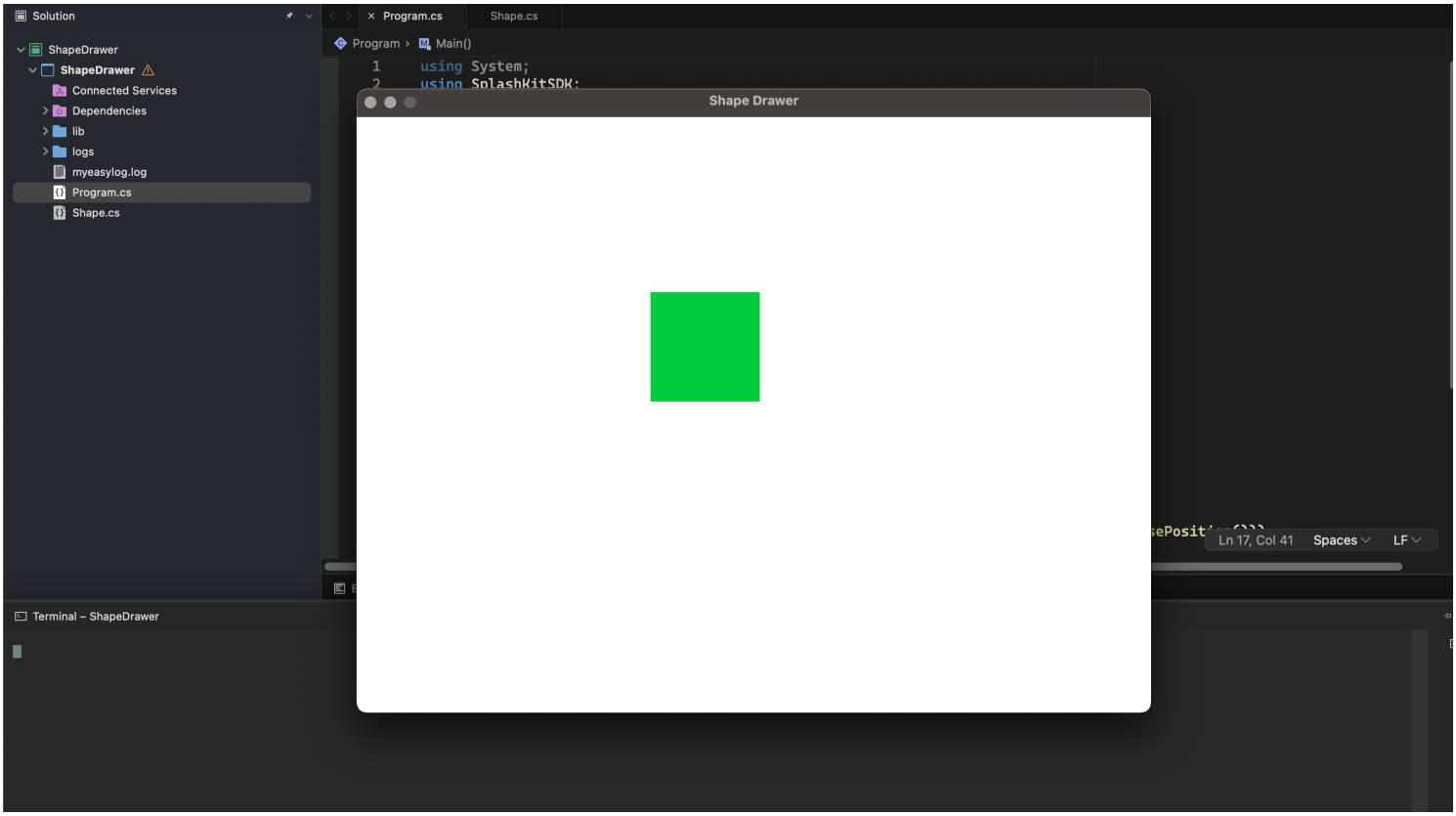
```
using SplashKitSDK;

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

        public Shape(int param)
        {
            _color = Color.Chocolate;
            _x = 0.0f;
            _y = 0.0f;
            _width = param;
            _height = param;
        }

        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);
        }

        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; }
        }
```

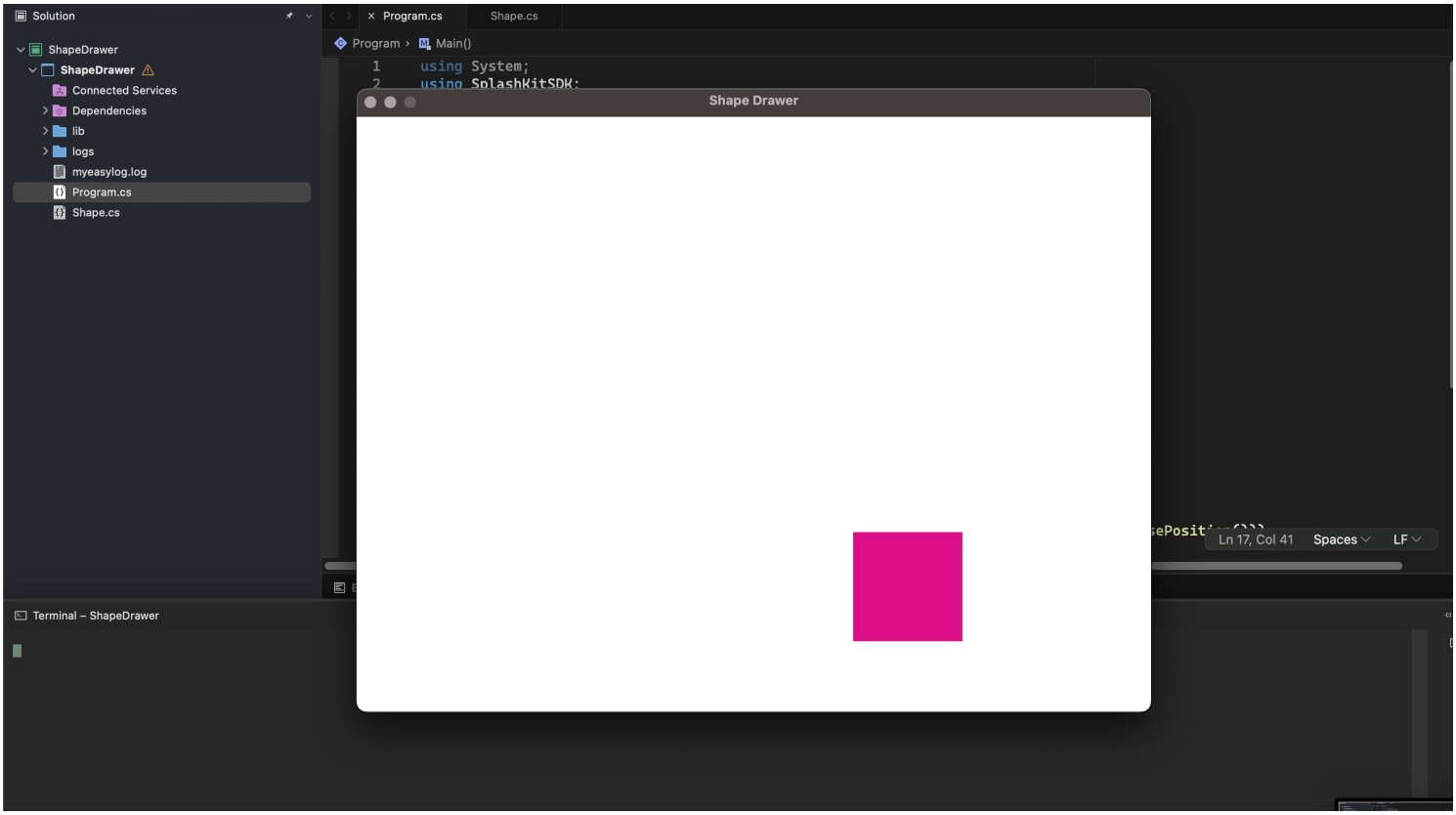
```
54          }
55      }
56
57
```

Ouput of the program:



Different positions and random colours:




Solution
ShapeDrawer
ShapeDrawer
Connected Services
Dependencies
lib
logs
myeasylog.log
Program.cs
Shape.cs
Program.cs
Shape.cs
Program › Main()
1 using System;
2 using SplashKitSDK;
Shape Drawer
Ln 17, Col 41
Spaces
LF
Terminal – ShapeDrawer