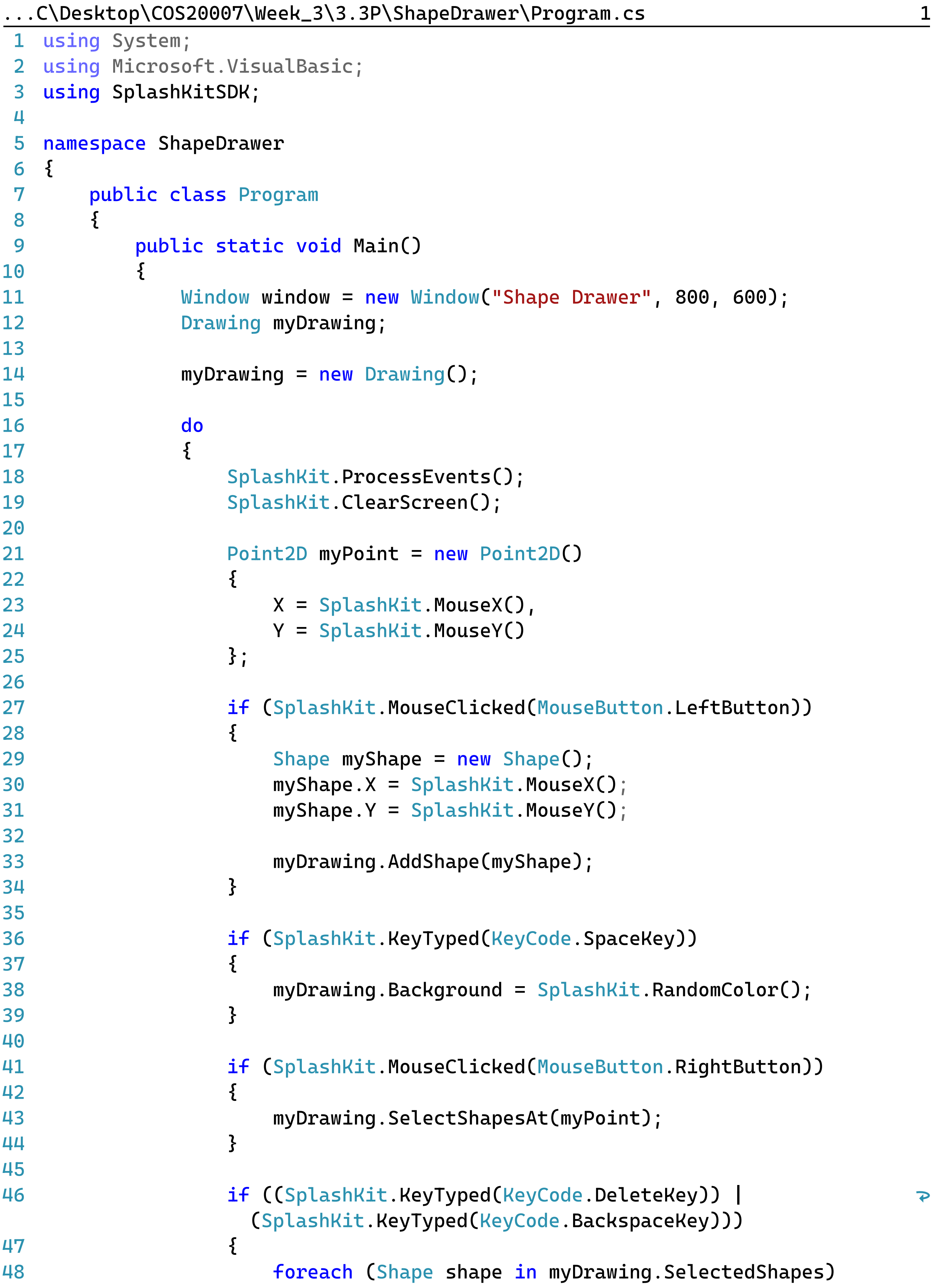
Drawing.cs

```
using SplashKitSDK;
using System.ComponentModel;
using System.Security.Cryptography.X509Certificates;

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

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

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

        public Drawing() : this (Color.White)
        {
            _shapes = new List<Shape>();
            _background  = Color.White;
        }

        public int ShapeCount
        {
            get => _shapes.Count;
        }

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

        public void RemoveShape(Shape shape)
        {
            _= _shapes?.Remove(shape);
        }

        public void Draw()
        {
            SplashKit.ClearScreen(_background);
            for (int i = 0; i < _shapes.Count; i++)
            {
                if (_shapes[i].Selected)
                {
                    _shapes[i].DrawOutline();
                }
                _shapes[i].Draw();
            }
        }

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

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

```

Shape.cs

```
using SplashKitSDK;

namespace ShapeDrawer
{
    internal 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 = _y = 0.0f;
            _width = _height = 100;
        }

        public bool Selected
        {
            get => _selected;
            set => _selected = value;
        }

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

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

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

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

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

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

        public bool IsAt(Point2D point)
        {
            if (point.X >= _x && point.X <= _x + _width)
            {
                if (point.Y >= _y && point.Y <= _y + _height)
                {
                    return true;
                }
            }
            return false;
        }

        public void DrawOutline()
        {
            SplashKit.FillRectangle(Color.Black, _x - 2, _y - 2, _width + 4, _height + 4);
        }
    }
}

```

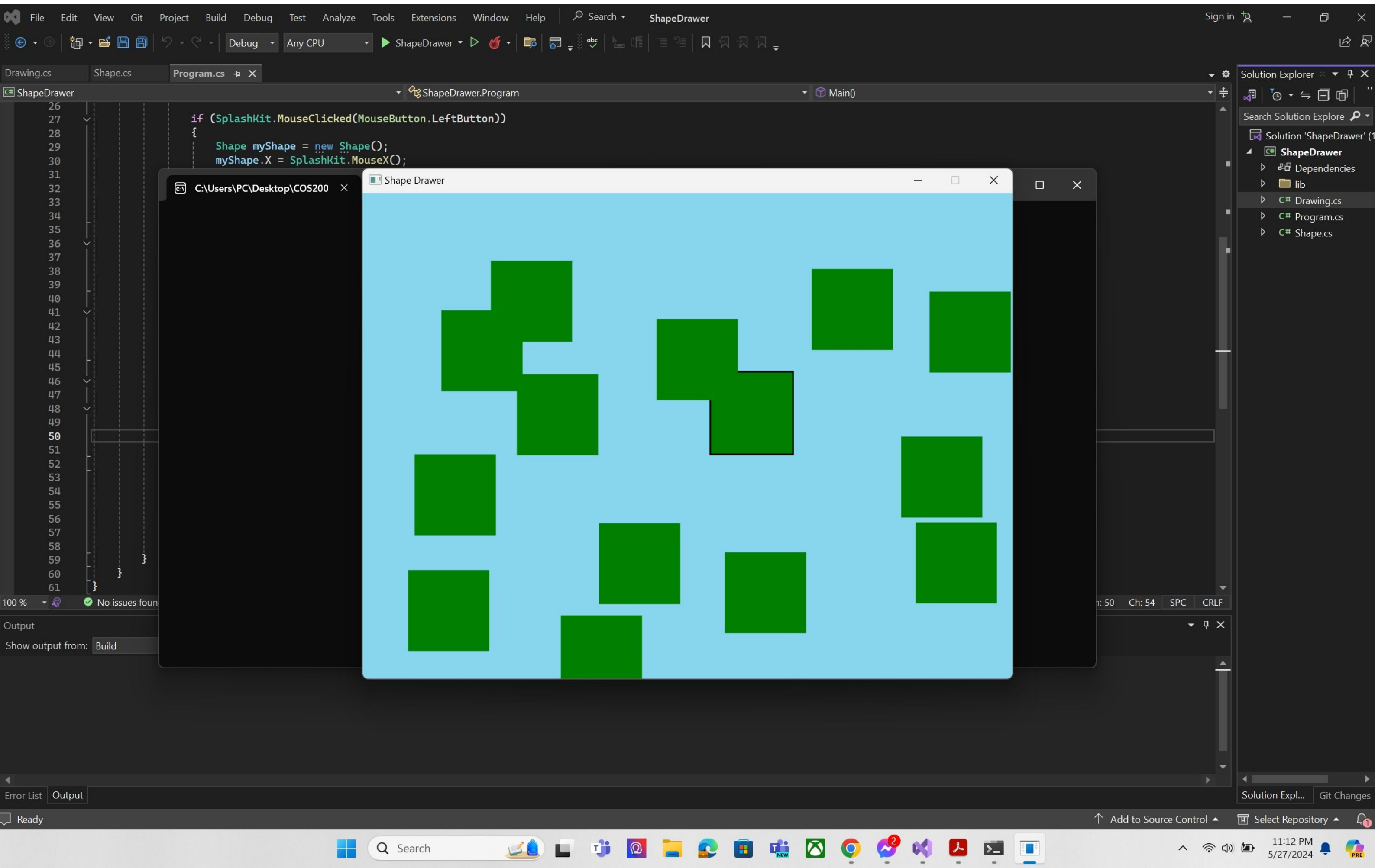
Program.cs



...C\Desktop\COS20007\Week_3\3.3P\ShapeDrawer\Program.cs 1

```csharp
using System;
using Microsoft.VisualBasic;
using SplashKitSDK;

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

            myDrawing = new Drawing();

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

                Point2D myPoint = new Point2D()
                {
                    X = SplashKit.MouseX(),
                    Y = SplashKit.MouseY()
                };

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

                    myDrawing.AddShape(myShape);
                }

                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
                {
                    myDrawing.Background = SplashKit.RandomColor();
                }

                if (SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    myDrawing.SelectShapesAt(myPoint);
                }

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

                myDrawing.Draw();

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

```

Program Output



```
            if (SplashKit.MouseClicked(MouseButton.LeftButton))
            {
                Shape myShape = new Shape();
                myShape.X = SplashKit.MouseX();
            }
        }
}
```