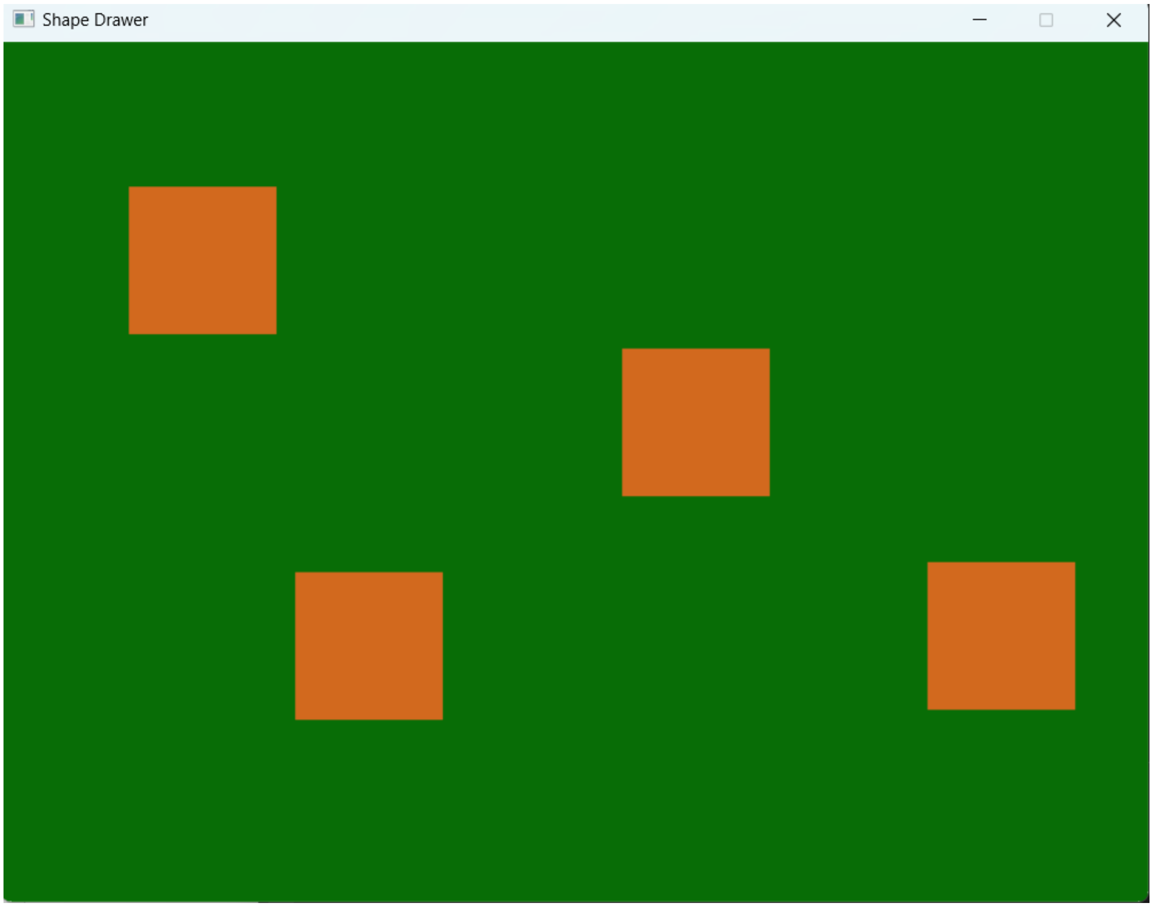
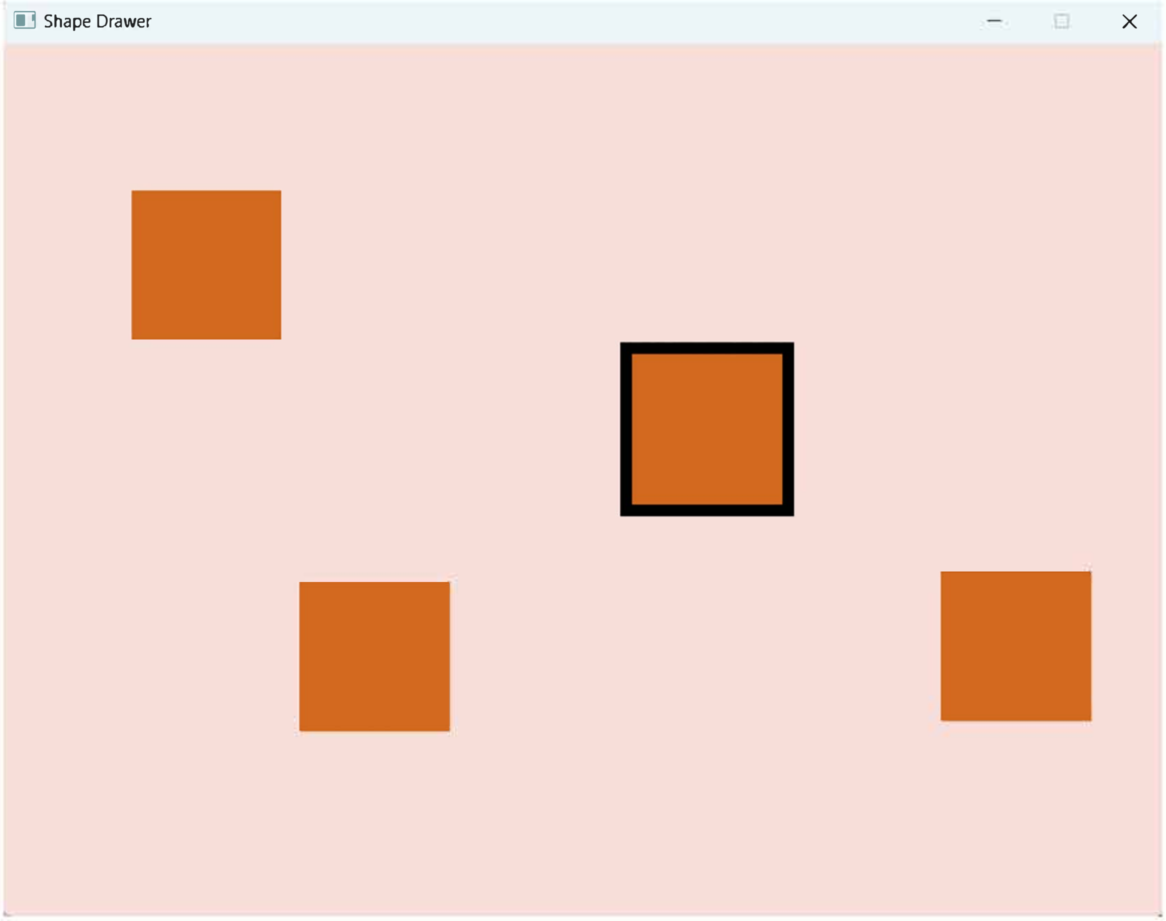
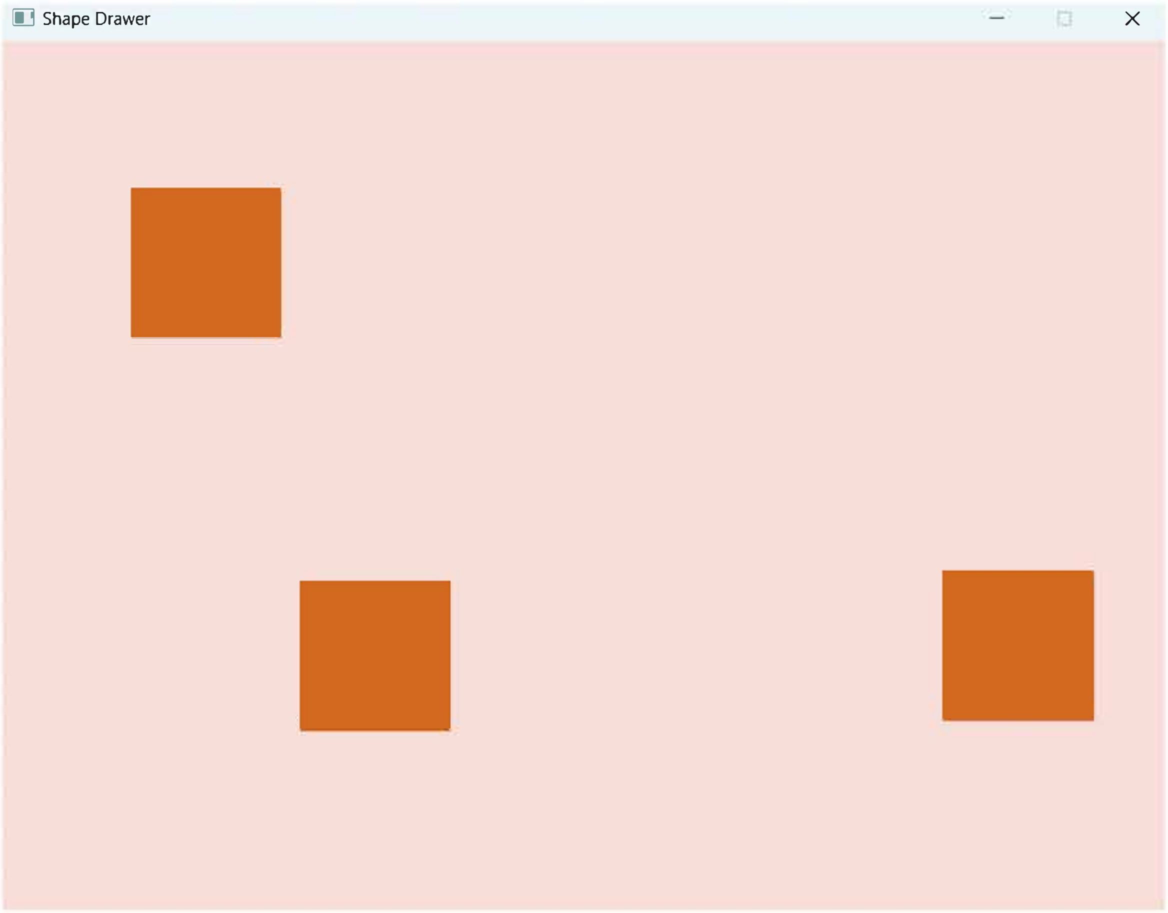
```
using SplashKitSDK;
using System.ComponentModel;

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

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

        }

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

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

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

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

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

        public void SelectShapeAt(Point2D pt)
        {
            foreach (Shape s in _shapes)
            {
                if(s.IsAt(pt))
                {
                    s.Selected = true;
                }
                else
                {
                    s.Selected = false;
                }
            }
        }

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

        public void RemoveShape(Shape s)
        {
            _ = _shapes.Remove(s);
        }
    }
}
```

```
using SplashKitSDK;

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

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

        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()
        {
            if(Selected)
            {
                DrawOutline();
            }
            SplashKit.FillRectangle(_color, _x, _y, _width, _height);
        }

        public Boolean IsAt(Point2D pt)
        {
            if (pt.X > _x && pt.X < _x + _width && pt.Y > _y && pt.Y < _y + _height)
            {
                return true;
            }
            else
            {
                return false;
            }
        }

        private bool _selected;
        public bool Selected
        {
            get
            {
                return _selected;
            }
            set
            {
                _selected = value;
            }
        }

        public void DrawOutline()
        {
            SplashKit.FillRectangle(Color.Black, _x-(5+3), _y-(5+3), _width+2*(5+3), _height+2*(5+3));
        }
    }
}

```

```
using System;
using SplashKitSDK;

namespace DrawingClass
{
    public class Program
    {
        public static void Main()
        {


            Window window = new Window("Shape Drawer", 800, 600);

            Drawing myDrawing = new Drawing();

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

                Shape newShape = new Shape(103);
                if(SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    newShape.X = SplashKit.MouseX();
                    newShape.Y = SplashKit.MouseY();
                    myDrawing.AddShape(newShape);

                }

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

                if (SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    myDrawing.SelectShapeAt(SplashKit.MousePosition());
                }

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

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

    }
}

```




Shape Drawer


Shape Drawer