```
using SplashKitSDK;
using System.IO;

namespace ShapeDrawer
{
    public 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;
            }
        }

        public void Save(string filename)
        {
            StreamWriter writer;

            writer = new StreamWriter(filename);
            try
            {
                writer.WriteColor(Background);
                writer.WriteLine(ShapeCount);

                foreach (Shape s in _shapes)
                {
                    s.SaveTo(writer);
                }
            }
            finally
            {
                writer.Close();
            }
        }

        public void Load(string filename)
        {
            StreamReader reader = new StreamReader(filename);
            Shape s;
            try
            {
                Background = reader.ReadColor();
                int count = reader.ReadInteger();
                _shapes.Clear();

                for (int i = 0; i < count; i++)
                {
                    string kind = reader.ReadLine();

                    switch (kind)
                    {
                        case "Rectangle":
                            s = new MyRectangle();
                            break;

                        case "Circle":
                            s = new MyCircle();
                            break;

                        case "Line":
                            s = new MyLine();
                            break;

                        default:
                            throw new InvalidDataException("Unknown shape kind: " + kind);
                    }

                    s.LoadFrom(reader);
                    _shapes.Add(s);
                }
            }
            finally
            {
                reader.Close();
            }
        }
    }
}

```