```
using System;
using System.Collections.Generic;
using SplashKitSDK;

namespace DrawingShape
{
    public class Program
    {
        private enum ShapeKind
        {
            Rectangle,
            Circle,
            Line
        }

        public static void Main()
        {
            ShapeKind kindToAdd = ShapeKind.Circle;
            Window window = new Window("Shape Drawer", 800, 600);
            Drawing myDrawing = new Drawing();
            do
            {
                SplashKit.ProcessEvents();
                SplashKit.ClearScreen();
                if (SplashKit.KeyTyped(KeyCode.RKey))
                {
                    kindToAdd = ShapeKind.Rectangle;
                }
                if (SplashKit.KeyTyped(KeyCode.CKey))
                {
                    kindToAdd = ShapeKind.Circle;
                }
                if (SplashKit.KeyTyped(KeyCode.LKey))
                {
                    kindToAdd = ShapeKind.Line;
                }
                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape newShape;
                    switch (kindToAdd)
                    {
                        case ShapeKind.Circle:
                            newShape = new MyCircle();
                            break;

                        case ShapeKind.Line:
                            newShape = new MyLine();
                            break;

                        default:
                            newShape = new MyRectangle();
                            break;
                    }
                    newShape.X = SplashKit.MouseX();
                    newShape.Y = SplashKit.MouseY();
                    myDrawing.AddShape(newShape);
                }
                Point2D pt = SplashKit.MousePosition();
                if(SplashKit.KeyTyped(KeyCode.SpaceKey))
                {
                    myDrawing.Background = SplashKit.RandomRGBColor(255);
                }
                if(SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    foreach(Shape s in myDrawing.Shapes)
                    {
                        if(s.IsAt(pt))
                        {
                            s.Selected = !s.Selected;
                        }
                    }
                }
                if(SplashKit.KeyTyped(KeyCode.DeleteKey)||
                  SplashKit.KeyTyped(KeyCode.BackspaceKey))
                {
                    for (int i = myDrawing.ShapeCount - 1; i >= 0; i--)
                    {
                        if (myDrawing.Shapes[i].Selected)
                        {
                            myDrawing.Shapes.RemoveAt(i);
                        }
                    }
                }
                myDrawing.Draw();
                SplashKit.RefreshScreen();
            } while (!window.CloseRequested);
        }
    }
}
```