

# **Khant Thu Aung/105292912**

## **2.3 Shape Drawer**

### **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(100);

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

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

        }
    }
}
```

### **Shape.cs**

```
using SplashKitSDK;

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

        public Shape(int param)
        {
```

```

        _color = Color.LightGreen;
        _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 X { get; internal set; }

    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()
    {
        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);
    }
}
}

```

## Outputs

