

[KingSchlock](#) / [UploadedCodeToPrintAgain](#) Private[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)[main](#) ▾

...

[UploadedCodeToPrintAgain](#) / [Program.cs](#) / <> Jump to ▾

KingSchlock Add files via upload



1 contributor

107 lines (89 sloc) | 3.53 KB

...

```
1  using SplashKitSDK;
2  using System;
3
4  namespace _4._2P
5  {
6      public class Program
7      {
8          private enum ShapeKind
9          {
10              Rectangle,
11              Circle,
12              Line
13          }
14
15          public static void Main()
16          {
17              Window window = new("4.2P: Thomas Horsley - 103071494", 800, 600);
18              Drawing drawing = new();
19
20              ShapeKind kindToAdd = ShapeKind.Circle;
21
22              do
23              {
24                  SplashKit.ProcessEvents();
25                  SplashKit.ClearScreen();
26
27                  Point2D mouseLocation = SplashKit.MousePosition();
28
29                  //? Draws a shape at mouse position and adds it to a shapes array
30                  if (SplashKit.MouseClicked(MouseButton.LeftButton))
31                  {
32                      if (kindToAdd == ShapeKind.Rectangle)
```

```
33     {
34         Shape rectangleShape = new MyRectangle(Color.Green, 0, 0, false, 100, 100);
35         {
36             X = (float)mouseLocation.X,
37             Y = (float)mouseLocation.Y
38         };
39
40         drawing.AddShape(rectangleShape);
41     }
42     if (kindToAdd == ShapeKind.Circle)
43     {
44         Shape circleShape = new MyCircle(Color.Red, 0, 0, false, 50)
45         {
46             X = (float)mouseLocation.X,
47             Y = (float)mouseLocation.Y
48         };
49
50         drawing.AddShape(circleShape);
51     }
52     if (kindToAdd == ShapeKind.Line)
53     {
54         Shape lineShape = new MyLine(Color.GreenYellow, 0, 0, false, 50)
55         {
56             X = (float)mouseLocation.X,
57             Y = (float)mouseLocation.Y,
58         };
59
60         drawing.AddShape(lineShape);
61     }
62 }
63
64 //? Relates keystrokes to shape kinds
65 if (SplashKit.KeyReleased(KeyCode.RKey))
66 {
67     kindToAdd = ShapeKind.Rectangle;
68 }
69 else if (SplashKit.KeyReleased(KeyCode.CKey))
70 {
71     kindToAdd = ShapeKind.Circle;
72 }
73 else if (SplashKit.KeyReleased(KeyCode.LKey))
74 {
75     kindToAdd = ShapeKind.Line;
76 } //TODO <--- can i use cases instead?
77
78
79 //? Checks if shape is selected
80 if (SplashKit.MouseClicked(MouseButton.RightButton))
81 {
82     drawing.SelectShapesAt(mouseLocation);
83 }
84
```

```
85
86     //? Changes background color when user presses space
87     if (SplashKit.KeyReleased(KeyCode.SpaceKey))
88     {
89         drawing.Background = SplashKit.RandomRGBColor(255);
90     }
91
92     if (SplashKit.KeyReleased(KeyCode.DeleteKey) || SplashKit.KeyReleased(KeyCode.I
93     {
94         foreach (Shape genericShape in drawing.SelectedShapes)
95         {
96             drawing.RemoveShape(genericShape);
97         }
98     }
99
100    drawing.Draw();
101    SplashKit.RefreshScreen(60);
102    } while (!window.CloseRequested);
103    }
104    }
105    }
106
107
```