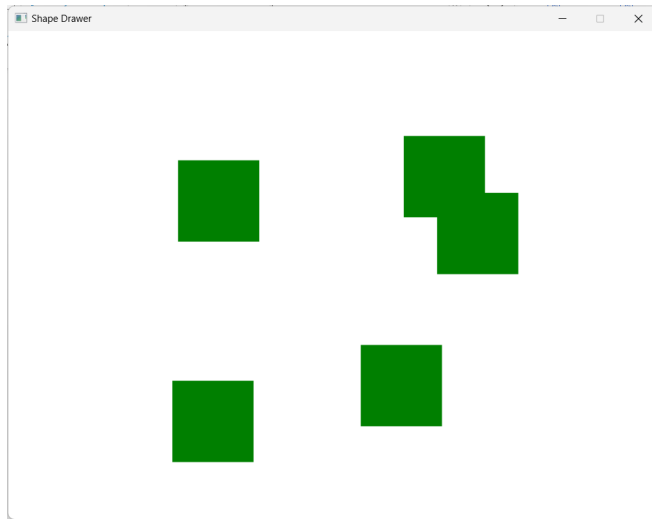


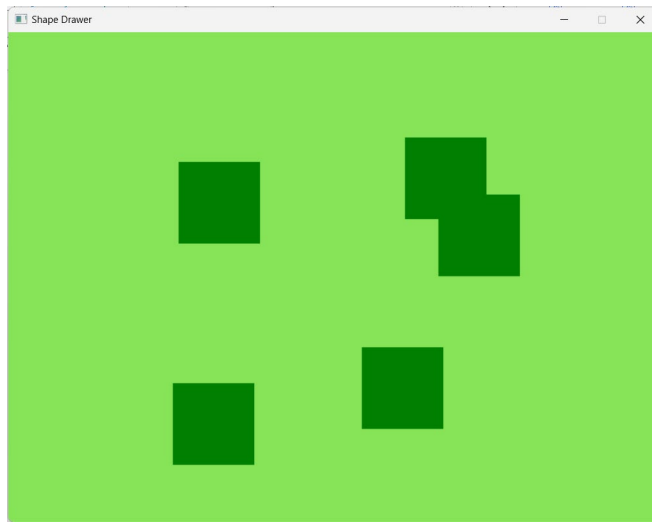
### 3.3P - Drawing Program - A Drawing Class

Jayden Kong, 104547242

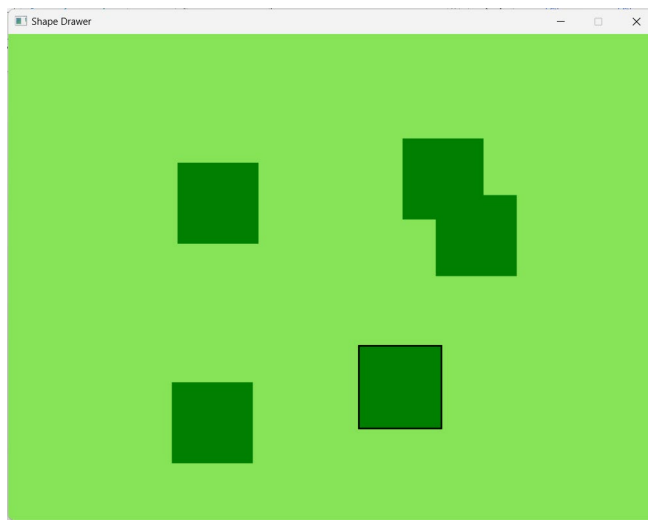
Pressing left mouse button at different positions:



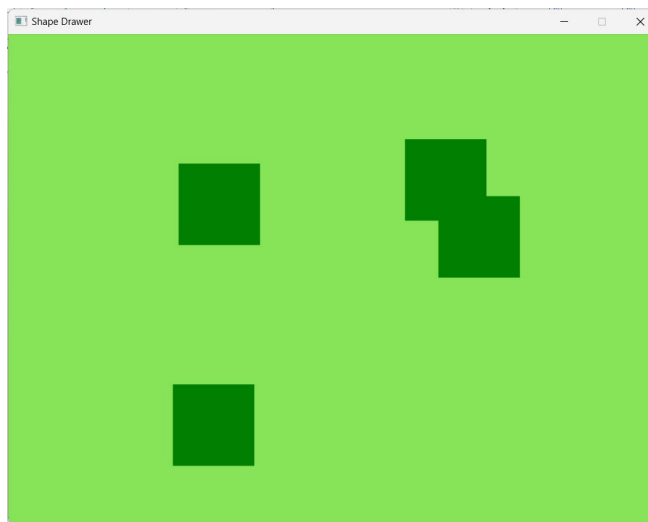
Pressing spacebar:



**Pressing right mouse button while cursor is open a shape:**



**Pressing backspace/delete while shape is selected:**



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using SplashKitSDK;
7
8 namespace ShapeDrawer
9 {
10     public class Shape
11     {
12         private Color _color;
13         private float _x;
14         private float _y;
15         private int _width;
16         private int _height;
17         private bool _selected;
18
19         public Color Color
20         {
21             get
22             {
23                 return _color;
24             }
25             set
26             {
27                 _color = value;
28             }
29         }
30
31         public float X
32         {
33             get
34             {
35                 return _x;
36             }
37             set
38             {
39                 _x = value;
40             }
41         }
42
43         public float Y
44         {
45             get
46             {
47                 return _y;
48             }
49             set
```

```
50         {
51             _y = value;
52         }
53     }
54
55     public int Width
56     {
57         get
58         {
59             return _width;
60         }
61         set
62         {
63             _width = value;
64         }
65     }
66
67     public int Height
68     {
69         get
70         {
71             return _height;
72         }
73         set
74         {
75             _height = value;
76         }
77     }
78
79     public bool Selected
80     {
81         get
82         {
83             return _selected;
84         }
85         set
86         {
87             _selected = value;
88         }
89     }
90
91     public Shape()
92     {
93         _color = Color.Green;
94         _x = 0.0f;
95         _y = 0.0f;
96         _width = 100;
97         _height = 100;
98     }
```

```
99
100     public void Draw()
101     {
102         if (_selected)
103         {
104             DrawOutline();
105         }
106
107         SplashKit.FillRectangle(_color, _x, _y, _width, _height);
108     }
109
110     public bool IsAt(Point2D pt)
111     {
112         if ((pt.X >= X) && (pt.X <= _x + _width) && (pt.Y >= _y) &&    ↗
113             (pt.Y <= _y + _height))
114         {
115             return true;
116         }
117         else
118         {
119             return false;
120         }
121     }
122
123     public void DrawOutline()
124     {
125         SplashKit.FillRectangle(Color.Black, _x - 2, _y - 2, _width +    ↗
126             4, _height + 4);
127     }
128 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using SplashKitSDK;
7
8 namespace ShapeDrawer
9 {
10     public class Drawing
11     {
12         private readonly List<Shape> _shapes;
13         private Color _background;
14
15         public Color Background
16         {
17             get
18             {
19                 return _background;
20             }
21             set
22             {
23                 _background = value;
24             }
25         }
26
27         public int ShapeCount
28         {
29             get
30             {
31                 return _shapes.Count;
32             }
33         }
34
35         public List<Shape> SelectedShapes
36         {
37             get
38             {
39                 List<Shape> result = new List<Shape>();
40                 foreach (Shape s in _shapes)
41                 {
42                     if (s.Selected)
43                     {
44                         result.Add(s);
45                     }
46                 }
47                 return result;
48             }
49         }
50     }
51 }
```

```
50
51
52     public Drawing(Color background)
53     {
54         List<Shape> shapes = new List<Shape>();
55         _shapes = shapes;
56         _background = background;
57     }
58
59     public Drawing() : this (Color.White) { }
60
61     public void AddShape(Shape s)
62     {
63         _shapes.Add(s);
64     }
65
66     public void RemoveShape(Shape s)
67     {
68         _shapes.Remove(s);
69     }
70
71     public void Draw()
72     {
73         SplashKit.ClearScreen(_background);
74         foreach (Shape s in _shapes)
75         {
76             s.Draw();
77         }
78     }
79
80     public void SelectShapesAt(Point2D pt)
81     {
82         foreach (Shape s in _shapes)
83         {
84             s.Selected = s.IsAt(pt);
85         }
86     }
87
88 }
89 }
90
```

```
1 using System;
2 using SplashKitSDK;
3 namespace ShapeDrawer
4 {
5     public class Program
6     {
7         public static void Main()
8         {
9             Window window = new Window("Shape Drawer", 800, 600);
10            Drawing myDrawing = new Drawing();
11
12            do
13            {
14                SplashKit.ProcessEvents();
15                SplashKit.ClearScreen();
16
17                if (SplashKit.MouseClicked(MouseButton.LeftButton))
18                {
19                    Shape shape = new Shape();
20                    shape.X = SplashKit.MouseX();
21                    shape.Y = SplashKit.MouseY();
22                    myDrawing.AddShape(shape);
23                }
24
25                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
26                {
27                    myDrawing.Background = SplashKit.RandomColor();
28                }
29
30                if (SplashKit.MouseClicked(MouseButton.RightButton))
31                {
32                    myDrawing.SelectShapesAt(SplashKit.MousePosition());
33                }
34
35                if (SplashKit.KeyTyped(KeyCode.DeleteKey) ||
36                    SplashKit.KeyTyped(KeyCode.BackspaceKey))
37                {
38                    foreach(Shape s in myDrawing.SelectedShapes)
39                    {
40                        myDrawing.RemoveShape(s);
41                    }
42
43                    myDrawing.Draw();
44
45                    SplashKit.RefreshScreen();
46                } while (!window.CloseRequested);
47            }
48        }
```



49 }

50