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

```
50                  return _background;
51              }
52              set
53              {
54                  _background = value;
55              }
56          }
57
58          public void Draw()
59          {
60              SplashKit.ClearScreen(_background);
61              for(int i=0; i<ShapeCount; i++)
62              {
63                  _shapes[i].Draw();
64              }
65          }
66
67          public void SelectShapeAt(Point2D pt)
68          {
69              foreach (Shape s in _shapes)
70              {
71                  if(s.IsAt(pt))
72                  {
73                      s.Selected = true;
74                  }
75                  else
76                  {
77                      s.Selected = false;
78                  }
79              }
80          }
81
82          public void AddShape(Shape s)
83          {
84              _shapes.Add(s);
85          }
86
87          public void RemoveShape(Shape s)
88          {
89              _ = _shapes.Remove(s);
90          }
91      }
92  }
93
```

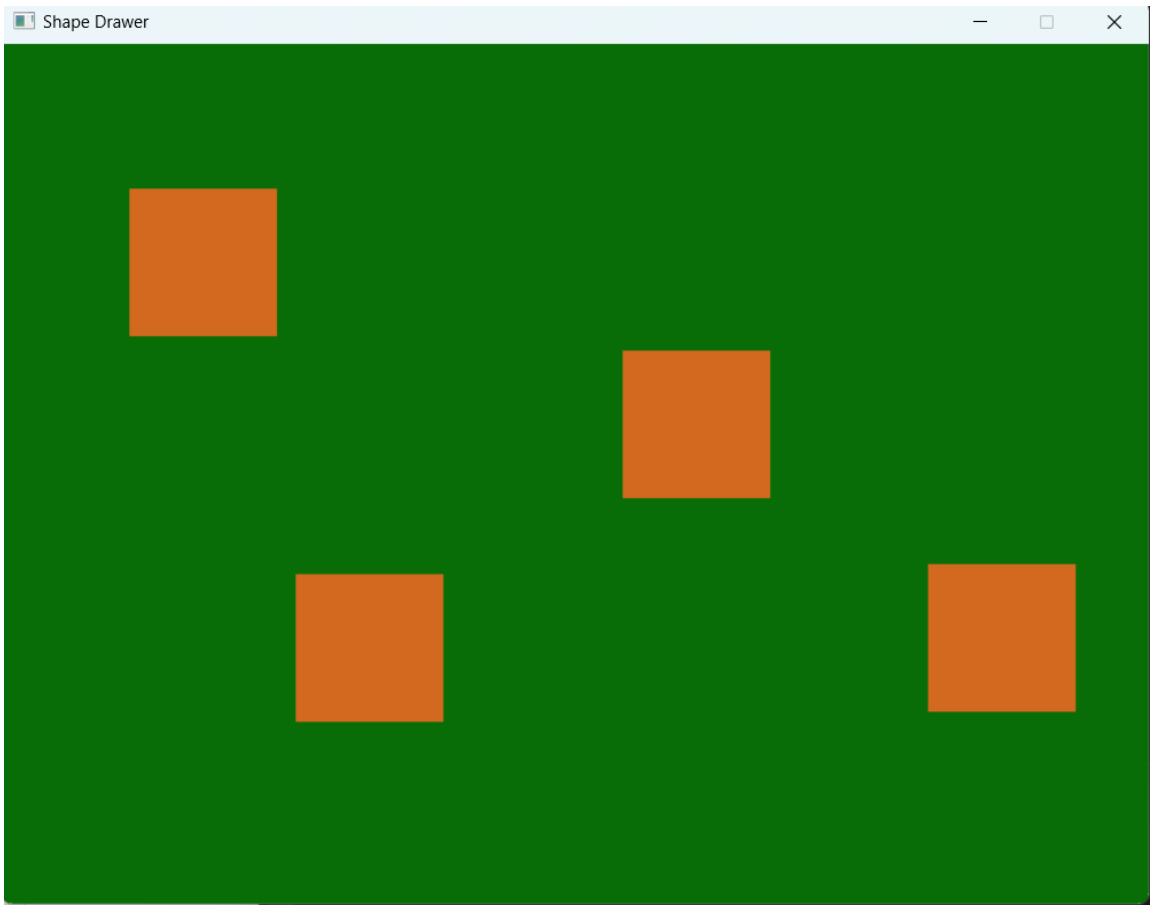
```
 1  using SplashKitSDK;
 2
 3  namespace DrawingClass
 4  {
 5      public class Shape
 6      {
 7          private Color _color;
 8          private float _x;
 9          private float _y;
10          private int _width;
11          private int _height;
12
13          public Shape(int param)
14          {
15              _color = Color.Chocolate;
16              _x = 0.0f;
17              _y = 0.0f;
18              _width = param;
19              _height = param;
20          }
21
22          public Color Color
23          {
24              get
25              {
26                  return _color;
27              }
28              set
29              {
30                  _color = value;
31              }
32          }
33
34          public float X
35          {
36              get
37              {
38                  return _x;
39              }
40              set
41              {
42                  _x = value;
43              }
44          }
45
46          public float Y
47          {
48              get
49              {
```

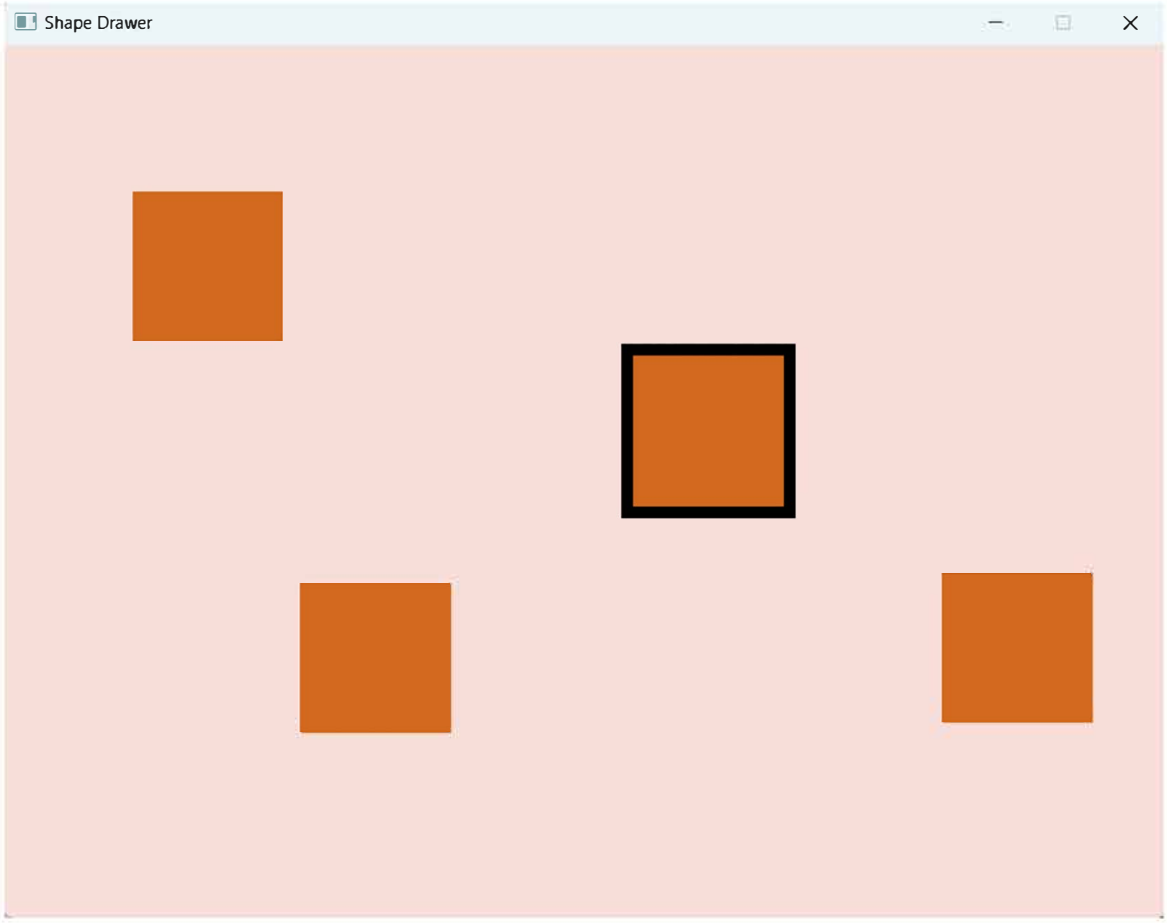
```
50                  return _y;
51              }
52              set
53              {
54                  _y = value;
55              }
56          }
57
58          public int Width
59          {
60              get
61              {
62                  return _width;
63              }
64              set
65              {
66                  _width = value;
67              }
68          }
69
70          public int Height
71          {
72              get
73              {
74                  return _height;
75              }
76              set
77              {
78                  _height = value;
79              }
80          }
81
82          public void Draw()
83          {
84              if(Selected)
85              {
86                  DrawOutline();
87              }
88              SplashKit.FillRectangle(_color, _x, _y, _width, _height);
89          }
90
91          public Boolean IsAt(Point2D pt)
92          {
93              if (pt.X > _x && pt.X < _x + _width && pt.Y > _y && pt.Y < _y
                  + _height)
94              {
95                  return true;
96              }
97              else
```

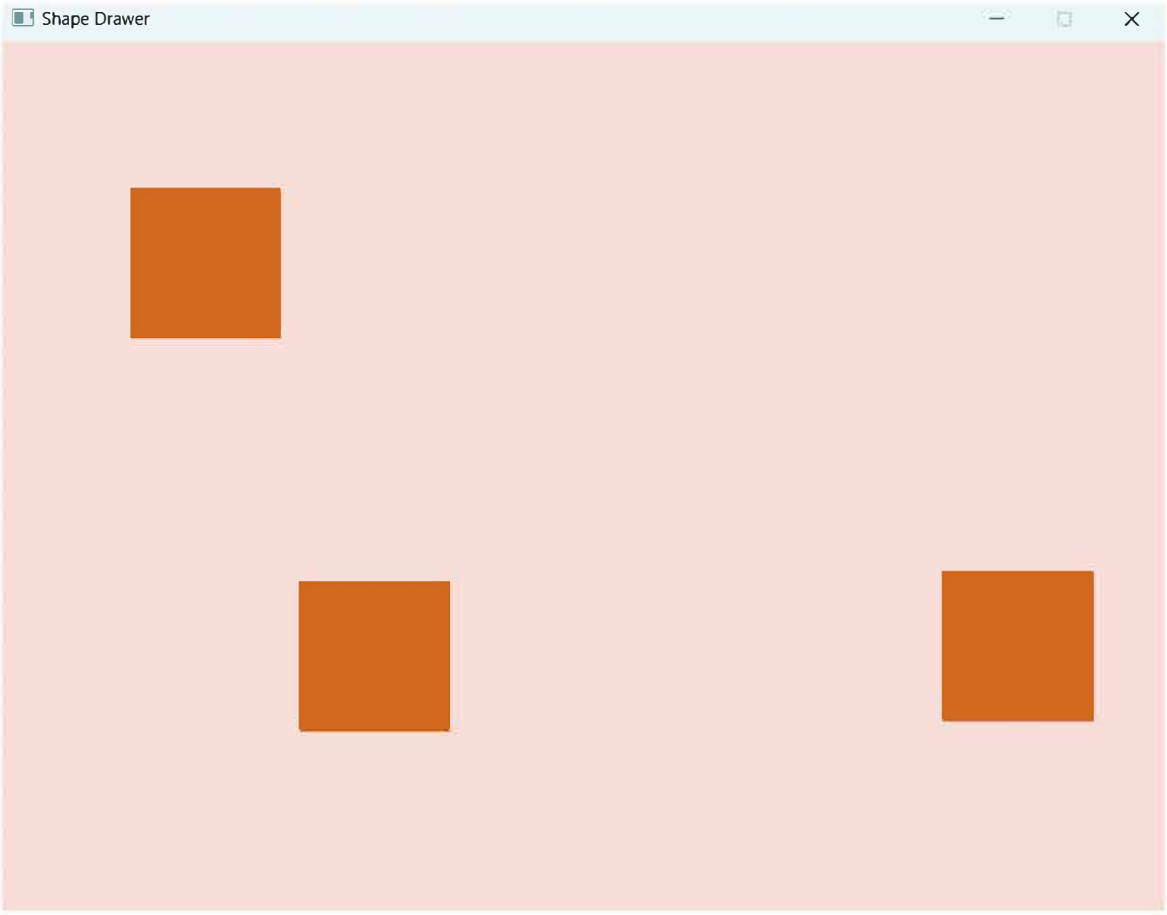
```
 98              {
 99                  return false;
100              }
101          }
102
103          private bool _selected;
104          public bool Selected
105          {
106              get
107              {
108                  return _selected;
109              }
110              set
111              {
112                  _selected = value;
113              }
114          }
115
116          public void DrawOutline()
117          {
118              SplashKit.FillRectangle(Color.Black, _x-(5+3), _y-(5+3),
                   _width+2*(5+3), _height+2*(5+3));
119          }
120      }
121  }
122
```

```
1  using System;
2  using SplashKitSDK;
3
4  namespace DrawingClass
5  {
6      public class Program
7      {
8          public static void Main()
9          {
10
11
12              Window window = new Window("Shape Drawer", 800, 600);
13
14              Drawing myDrawing = new Drawing();
15
16              do
17              {
18                  SplashKit.ProcessEvents();
19                  SplashKit.ClearScreen();
20
21                  Shape newShape = new Shape(103);
22                  if(SplashKit.MouseClicked(MouseButton.LeftButton))
23                  {
24                      newShape.X = SplashKit.MouseX();
25                      newShape.Y = SplashKit.MouseY();
26                      myDrawing.AddShape(newShape);
27
28                  }
29
30                  if(SplashKit.KeyTyped(KeyCode.SpaceKey))
31                  {
32                      myDrawing.BackGround = SplashKit.RandomColor();
33                  }
34
35                  if (SplashKit.MouseClicked(MouseButton.RightButton))
36                  {
37                      myDrawing.SelectShapeAt(SplashKit.MousePosition());
38                  }
39
40                  if(SplashKit.KeyTyped(KeyCode.DeleteKey) ||
                      SplashKit.KeyTyped(KeyCode.BackspaceKey))
41                  {
42                      foreach (Shape s in myDrawing.SelectedShapes)
43                      {
44                          myDrawing.RemoveShape(s);
45                      }
46                  }
47
48                  myDrawing.Draw();
```

```
49                  SplashKit.RefreshScreen();
50              } while (!window.CloseRequested);
51          }
52
53      }
54  }
55
```




Shape Drawer


Shape Drawer