COS 20007
Task 3.3

Duc Thuan Tran

104330455

I. Code

1. Drawing.cs

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

namespace DrawingShape
{
   public class Drawing
   {
      private readonly List<Shape> _shapes;
      private Color _background;

      public Drawing(Color background)
      {
         _background = background;
         _shapes = new List<Shape>();
      }
      public Drawing() : this(Color.White)
      {

      }
      public Color Background
      {
         get { return _background; }
         set { _background = value; }
      }

      public void Draw()
      {
         SplashKit.ClearScreen(_background);
         foreach (Shape s in _shapes)
         {
            s.Draw();
         }

      }

      public void SelectShapesAt(Point2D pt)
      {
            foreach (Shape s in _shapes)
            {
                if (s.IsAt(pt))
                    s.Selected = true;
                else
                    s.Selected = false;
            }
        }

        public List<Shape> SelectedShapes
        {
            get
            {
                List<Shape> _selectedShapes = new List<Shape>();
                foreach(Shape s in _shapes)
                {
                    if (s.Selected)
                    _selectedShapes.Add(s);
                }
                return _selectedShapes;
            }
        }

        public int ShapeCount
        {
            get { return _shapes.Count; }
        }

        public void AddShape(Shape s)
        {
            _shapes.Add(s);
        }

        public void RemoveShape(Shape s)
        {
            _shapes.Remove(s);
        }

    }
}
```

2. Shape.cs

```
using System;
using SplashKitSDK;

namespace DrawingShape
{
    public class Shape
    {
              private Color _color;
              private float _x;
              private float _y;
              private int _width;
              private int _height;
private bool _selected;

              public Shape()
              {
                        _color = Color.Blue;
                        _x = 0;
                        _y = 0;
                        _width = 100;
                        _height = 100;
   _selected = false;
              }

public Color Color
{
   get { return _color; }
   set { _color = value; }
}

public float X
{
   get { return _x; }
   set { _x = value; }
}

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; }
```

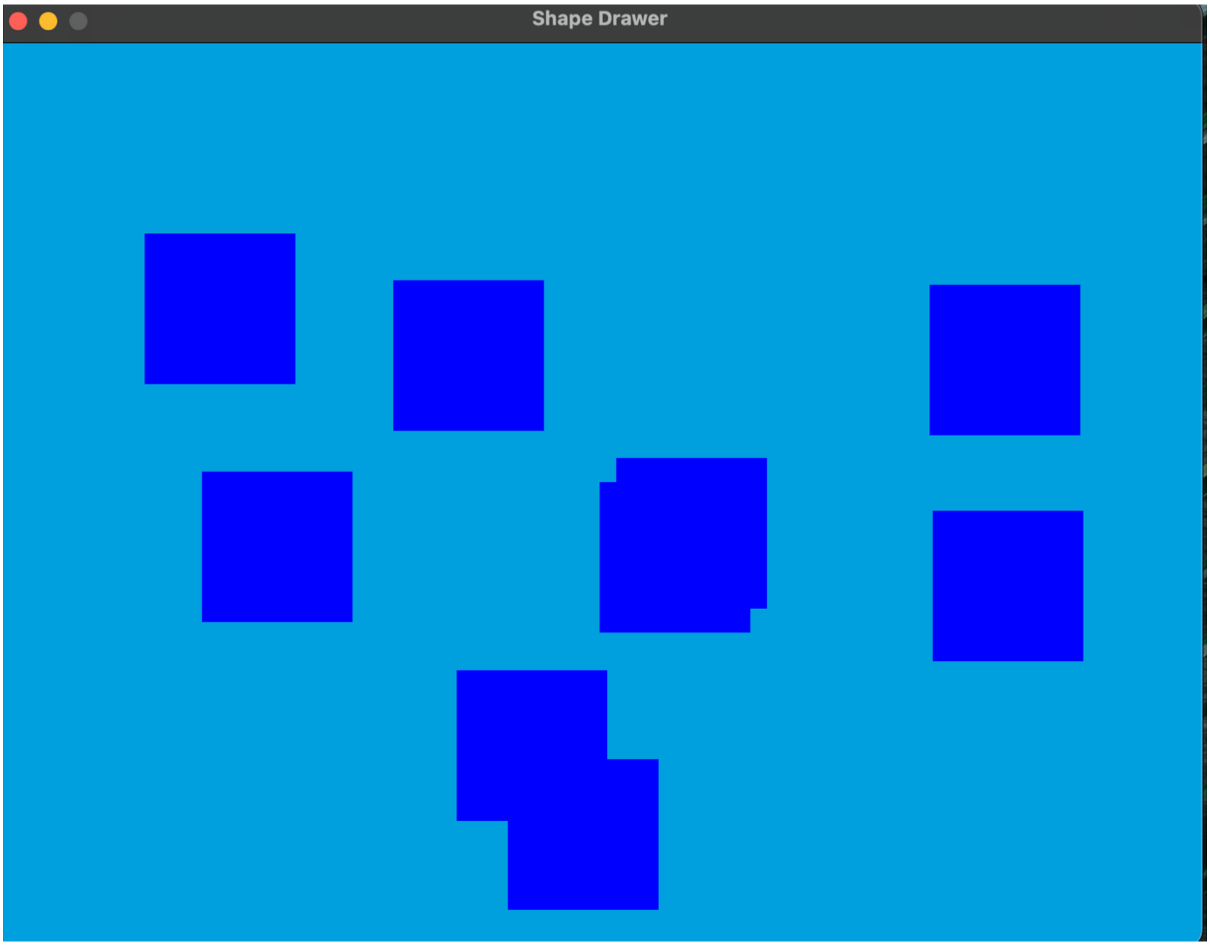
```
        }

        public bool Selected
        {
            get { return _selected; }
            set { _selected = value; }
        }

        public void Draw()
                             {
            if (Selected)
                DrawOutine();

            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);
        }

        public void DrawOutine()
        {
            SplashKit.DrawRectangle(Color.Black, _x - 2, _y - 2, _width + 4, _height + 4);
        }
    }
}
```

3. Program.cs

```
using System;
using SplashKitSDK;

namespace DrawingShape
{
    public class Program
    {
        public static void Main()
        {
            Window window = new Window("Drawing Shape", 800, 600);
            Drawing myDrawing = new Drawing();
            do
            {
                SplashKit.ProcessEvents();
                SplashKit.ClearScreen();

                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape s = new Shape();
                    s.X = SplashKit.MouseX();
                    s.Y = SplashKit.MouseY();
                    myDrawing.AddShape(s);
                }

                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
                {
                    myDrawing.Background = SplashKit.RandomRGBColor(255);
                }


                if (SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    myDrawing.SelectShapesAt(SplashKit.MousePosition());
                }

                if
(SplashKit.KeyDown(KeyCode.DeleteKey)||SplashKit.KeyDown(KeyCode.BackspaceKey))
                {
                    foreach(Shape s in myDrawing.SelectedShapes)
                    {
                        myDrawing.RemoveShape(s);
                    }
                }
                    myDrawing.Draw();

                SplashKit.RefreshScreen();
            } while (!window.CloseRequested);
        }
    }
}
```

II. Image

1. Program's output


Shape Drawer