Swinburne University of Technology

COS20007 OBJECT ORIENTED PROGRAMMING

2.3P Shape Drawing

`Program Class

```csharp
using System;
using SplashKitSDK;

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

        if (SplashKit.MouseClicked(MouseButton.LeftButton))
        {
          myShape.X = SplashKit.MouseX();
          myShape.Y = SplashKit.MouseY();
        }
        if (SplashKit.KeyTyped(KeyCode.SpaceKey))
        {
          if (myShape.IsAt(SplashKit.MousePosition()))
          {
            myShape.Color = SplashKit.RandomColor();
          }
        }
        myShape.Draw();
        SplashKit.RefreshScreen();
      } while (!window.CloseRequested);
    }
  }
}
```

Shape class

```csharp
using SplashKitSDK;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

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

        public Shape()
        {
            _color = Color.Green;
            _x = 0.0f;
            _y = 0.0f;
            _width = 100;
            _height = 100;
        }

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

        public double X
        {
            get { return _x; }
            set => _x = value;
        }

        public double 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