Swinburne University of Technology

COS20007 Object Oriented Programming

3.3P - Drawing Program - A Drawing Class

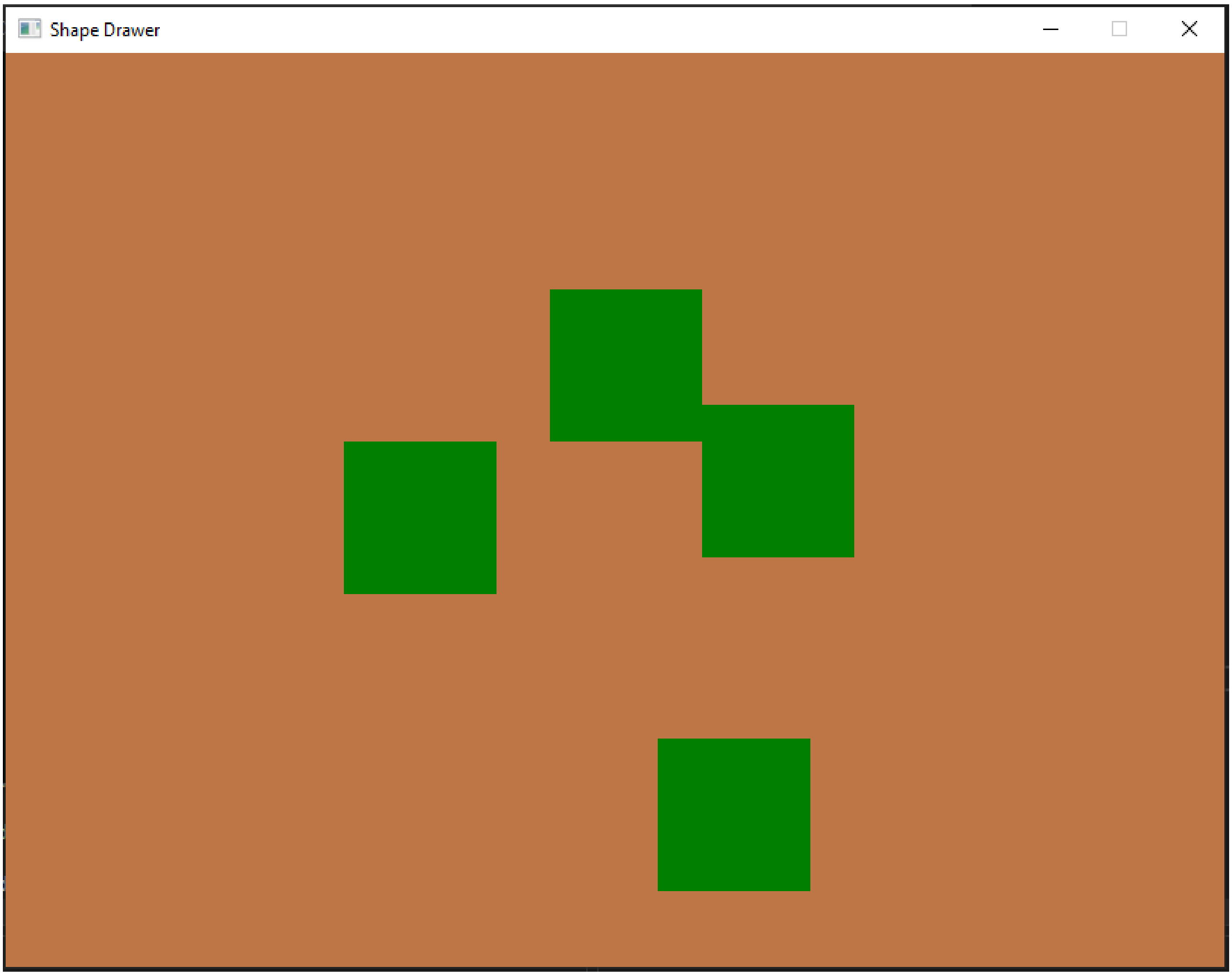
PDF generated at 16:19 on Thursday 20th April, 2023

Produced by OOP Submission Formatter

```
using System;
using SplashKitSDK;

namespace ShapeDrawer
{
    public class Program
    {

        private static Drawing myDrawing = new Drawing();
        public static void Main()
        {

            Window window = new Window("Shape Drawer", 800, 600);

            do {

                SplashKit.ProcessEvents();
                SplashKit.ClearScreen();
                if (SplashKit.MouseClicked(SplashKitSDK.MouseButton.LeftButton))
                {
                    Shape newShape = new Shape();
                    newShape.X = SplashKit.MouseX();
                    newShape.Y = SplashKit.MouseY();

                    myDrawing.AddShape(newShape);
                    myDrawing.Background = SplashKit.RandomRGBColor(255);
                }
                if (SplashKit.KeyTyped(SplashKitSDK.KeyCode.SpaceKey)) {
                    myDrawing.Background = SplashKit.RandomRGBColor(255);
                }
                if (SplashKit.MouseClicked(SplashKitSDK.MouseButton.RightButton))
                {
                    myDrawing.SelectShapesAt(SplashKit.MousePosition());
                }
                if (SplashKit.KeyTyped(SplashKitSDK.KeyCode.BackspaceKey) || SplashKit.KeyTyped(SplashKitSDK.KeyCode.DeleteKey))
                {
                    foreach (Shape s in myDrawing.Selected_Shapes)
                    {
                        myDrawing.RemoveShape(s);
                    }
                }
                myDrawing.Draw();
                SplashKit.RefreshScreen();
            } while (!window.CloseRequested);
        }
    }
}
```

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

namespace ShapeDrawer {

    public class Drawing {

        private readonly List<Shape> _shapes;

        private SplashKitSDK.Color _background;

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

        public Drawing() : this(SplashKitSDK.Color.White) { }

        public List<Shape> Selected_Shapes
        {
            get
            {
                List<Shape> _selectedShapes = new List<Shape>();

                foreach (Shape shape in _shapes)
                {
                    if (shape.Selected)
                    {
                        _selectedShapes.Add(shape);
                    }
                }

                return _selectedShapes;
            }
        }

        public SplashKitSDK.Color Background
        {
            get { return _background; }

            set { _background = value; }

        }

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

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

        public void Draw()
        {
            SplashKit.ClearScreen();
            SplashKit.FillRectangle(_background, 0, 0, 800, 600);
            foreach (Shape shape in _shapes)
            {
                shape.Draw();
            }

        }

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

        public void RemoveShape(Shape shape)
        {

            foreach (Shape s in _shapes) {

                if (shape == s) {

                        _shapes.Remove(shape);
                }
            }

        }
    }
}
```

```
using System;
using SplashKitSDK;

namespace ShapeDrawer
{

        public class Shape {

            private Color _color = Color.Green;

            private float _x = 0;

            private float _y = 0;

            private int _width = 100;

            private int _height = 100;

            private bool _selected;

            public Color color { set{ _color = value;}  }
            public float X { set { _x = value; } }
            public float Y { set { _y = value; } }

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

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

            public void Draw() {
                if (_selected)
                {
                    DrawOutline();
                }
                SplashKit.FillRectangle(_color, _x, _y, _width, _height);

            }

            public bool IsAt(Point2D pt) {

                if (pt.X > _x && pt.X < (_x + _width) && pt.Y > _y && pt.Y < (_y + _height)) {

                    return true;

                } else {

                    return false;

            }
        }
    }
}
```


Shape Drawer