Swinburne University of Technology

COS20007 Object Oriented Programming

Drawing Program - A Drawing Class

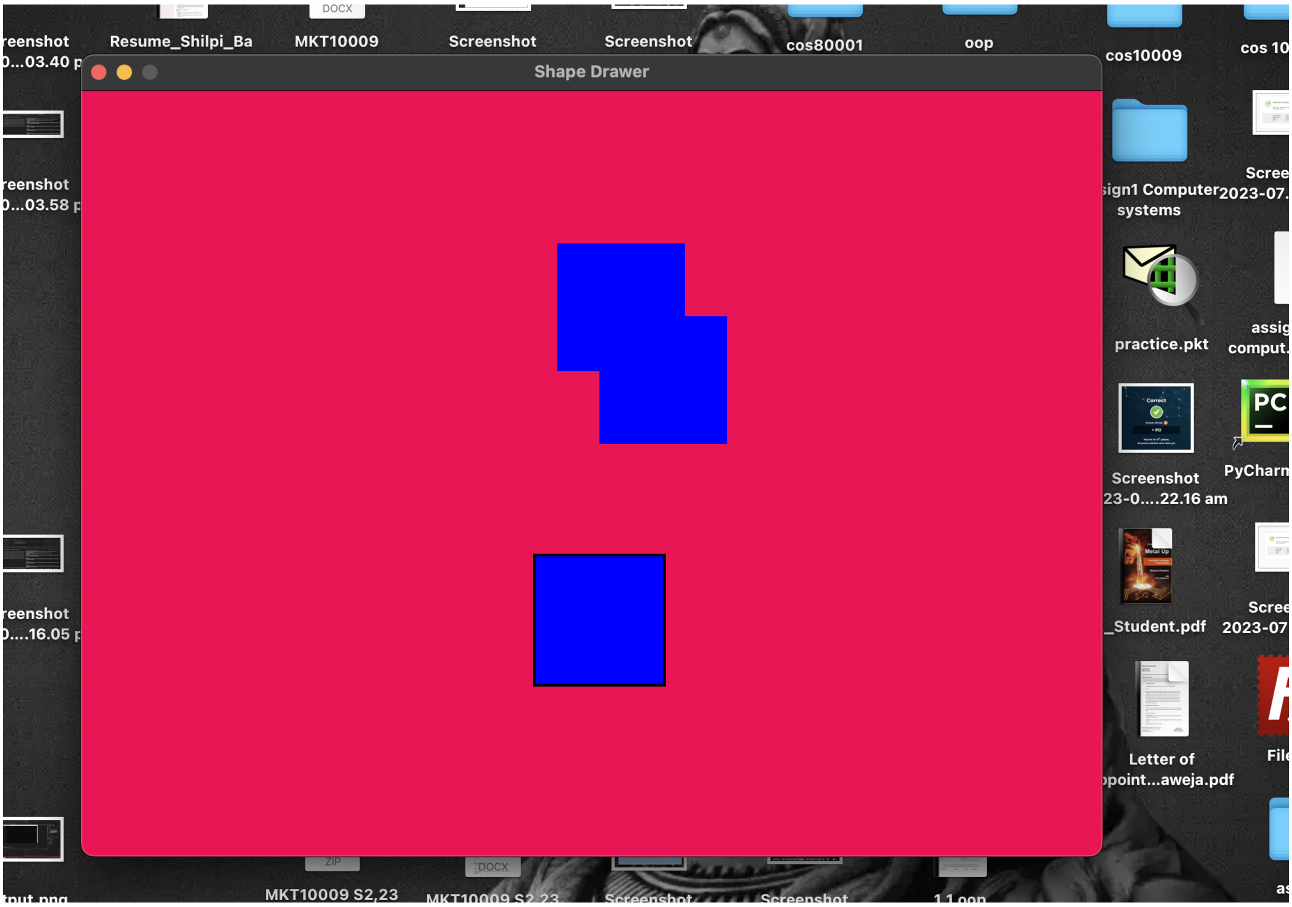
PDF generated at 12:01 on Monday 4th September, 2023

Produced by OOP Submission Formatter

```
using System;
using SplashKitSDK;
namespace ShapeDrawer
{
    public class Program
    {

        public static void Main()
        {
            Window window = new Window("Shape Drawer", 800, 600);

            Drawing dr = new Drawing();
            do
            {
                SplashKit.ProcessEvents();
                SplashKit.ClearScreen();



                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape sh = new Shape();

                    sh.X = SplashKit.MouseX();
                    sh.Y = SplashKit.MouseY();
                    dr.AddShape(sh);

                }


                if (SplashKit.MouseClicked(MouseButton.RightButton))
                {

                    Point2D mp = SplashKit.MousePosition();
                    dr.SelectShapesAt(mp);
                }

                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
                {

                    dr.Background1 = SplashKit.RandomRGBColor(255);
                }

                if (SplashKit.KeyTyped(KeyCode.DeleteKey))
                {
                    foreach (Shape s in dr.Selectedshapes)
                    {
                        dr.DeleteShape(s);
                    }
                }

                if (SplashKit.KeyTyped(KeyCode.BackspaceKey))
                {
                    foreach (Shape s in dr.Selectedshapes)
                    {
                        dr.DeleteShape(s);
                    }
                }


                dr.Draw();
                SplashKit.RefreshScreen();
            } while (!window.CloseRequested);
        }
    }

}
```

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

{
    public class Drawing
    {
        private readonly List<Shape> _shapes;
        private Color _background;
        public Drawing(Color bg)
        {
            _shapes = new List<Shape>();
            _background = bg;

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

        }
        public int ShapeCount
        {
            get
            {
                return _shapes.Count;
            }
        }
        public void AddShape(Shape s)
        {
            _shapes.Add(s);
        }
        public void Draw()
        {
            SplashKit.ClearScreen(Background1);
            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> _result = new List<Shape>();


                foreach (Shape s in _shapes)
                {
                    if (s.Selected == true)
                    {
                        _result.Add(s);
                    }
                }
                return _result;
            }
        }
        public Color Background1
        {
            get
            {
                return _background;
            }
            set
            {
                _background = value;
            }
        }

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

```
using System;
using ShapeDrawer;
using SplashKitSDK;

namespace ShapeDrawer
{
    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 = (float)0;
            _y = (float)0;
            _width = (int)100;
            _height = (int)100;
            _selected = (bool)false;
        }
        public void Draw()
        {
            if (Selected == true)
            {
                Outline();
            }
            SplashKit.FillRectangle(_color, _x, _y, _width, _height);

        }
        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 bool IsAt(Point2D pt)
        {
            if (pt.X >= _x && pt.X < (_x + _width) && pt.Y >= _y && pt.Y <= (_y + _height))
            {

                return true;
            }
            else
            {
                return false;
            }
        }
        public void Outline()
        {
            SplashKit.FillRectangle(Color.Black, _x - 2, _y - 2, _width + 4, _height + 4);
        }

    }
}



```


Shape Drawer
Resume_Shilpi_Ba
MKT10009
Screenshot
Screenshot
cos80001
oop
cos10009
practice.pkt
PyCharm
_Student.pdf
Letter of
MKT10009 S2,23
Screenshot
Screenshot