Swinburne University of Technology

COS20007 Object Oriented Programming

Drawing Program - A Drawing Class

PDF generated at 20:38 on Monday 25th September, 2023

```
using SplashKitSDK;
using System.Runtime.Intrinsics.X86;

namespace ShapeDrawer
{
    public class Program
    {
        public static void Main()
        {



            Drawing _drawing = new Drawing();


            Window window = new Window("Shape Drawer", 800, 600);
            do
            {
                SplashKit.ProcessEvents();
                SplashKit.ClearScreen();
                _drawing.Draw();
                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape myShape = new Shape();
                    myShape.X = SplashKit.MouseX();
                    myShape.Y = SplashKit.MouseY();
                    _drawing.AddShape(myShape);


                }

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


                if (SplashKit.KeyTyped(KeyCode.DeleteKey))
                    {
                    foreach (Shape s in _drawing.SelectedShapes)
                    {
                        _drawing.RemoveShape(s);
                    }
                }




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

```

```
54
55
56                  SplashKit.RefreshScreen();
57
58
59              } while (!window.CloseRequested);
60          }
61
62      }
63
64
65  }
```

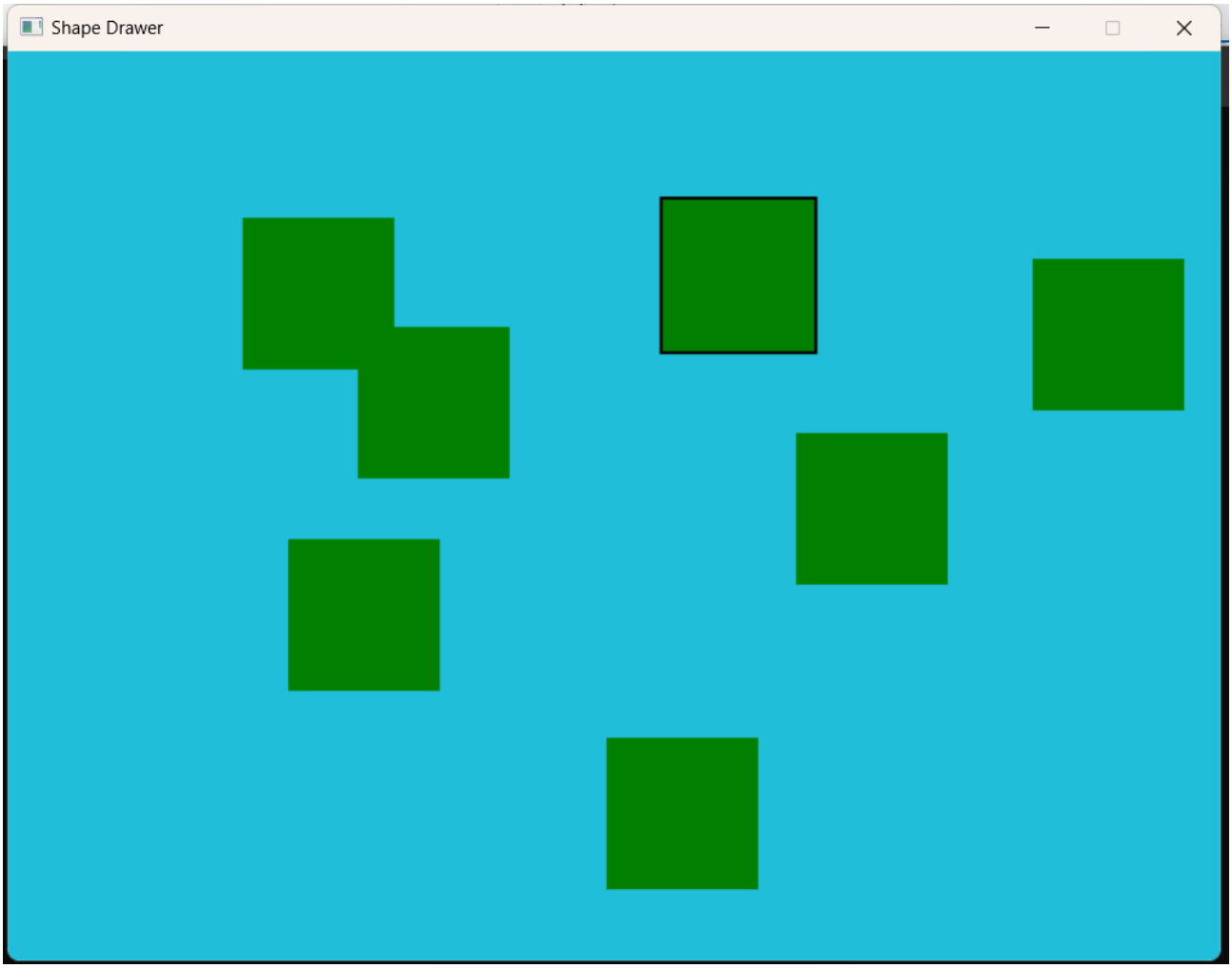
```
using SplashKitSDK;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ShapeDrawer
{
    public class Drawing
    {
        private readonly List<Shape> _shapes;
        private Color _background;


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

        int ShapeCount { get { return _shapes.Count; } }
        public Color Background { get { return _background; } set { _background =
↪  value; } }

        public void Draw()
            {
            SplashKit.ClearScreen(_background);

            foreach (Shape shape in _shapes) { }

            foreach (Shape i in _shapes)
            {
                i.Draw();
            }


            }
        public void SelectShapesAt( Point2D pt)
        {
            foreach (Shape i in _shapes)
            {
               i.Selected = i.IsAt(pt);

            }

        }


        public List<Shape> SelectedShapes
        {


```

```
            get
            {
                List<Shape> result = new List<Shape>();
                foreach (Shape i in _shapes)
                {
                    if (i.Selected == true)
                        result.Add(i);
                }
                return result;
            }
        }
```

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

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

    }
}

/*
 * using SplashKitSDK;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
*/
```

```
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 float _x, _y;
        private int _width, _height;
        private bool _selected;

        public Shape()
        {
            _color = Color.Green;
            _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 { set; get; }
        public int Height { set; get; }

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

        }
```

```
54          public bool IsAt(Point2D pt)
55          {
56
57              if (pt.X < _x + _width && pt.X > _x)
58              {
59                  if (pt.Y < _y + _height && pt.Y > _y)
60                  {
61                      return true;
62                  }
63              }
64              return false;
65          }
66
67          public bool Selected
68          {
69              get { return _selected; }
70              set { _selected = value; }
71          }
72
73          public void DrawOutline()
74          {
75              SplashKit.FillRectangle(Color.Black, _x -2, _y-2, _width+4, _height+4);
76          }
77
78      }
79  }
```


Shape Drawer