Swinburne University of Technology

COS20007 Object Oriented Programming

Drawing Program - A Drawing Class

PDF generated at 20:51 on Thursday 9th November, 2023

Produced by OOP Submission Formatter

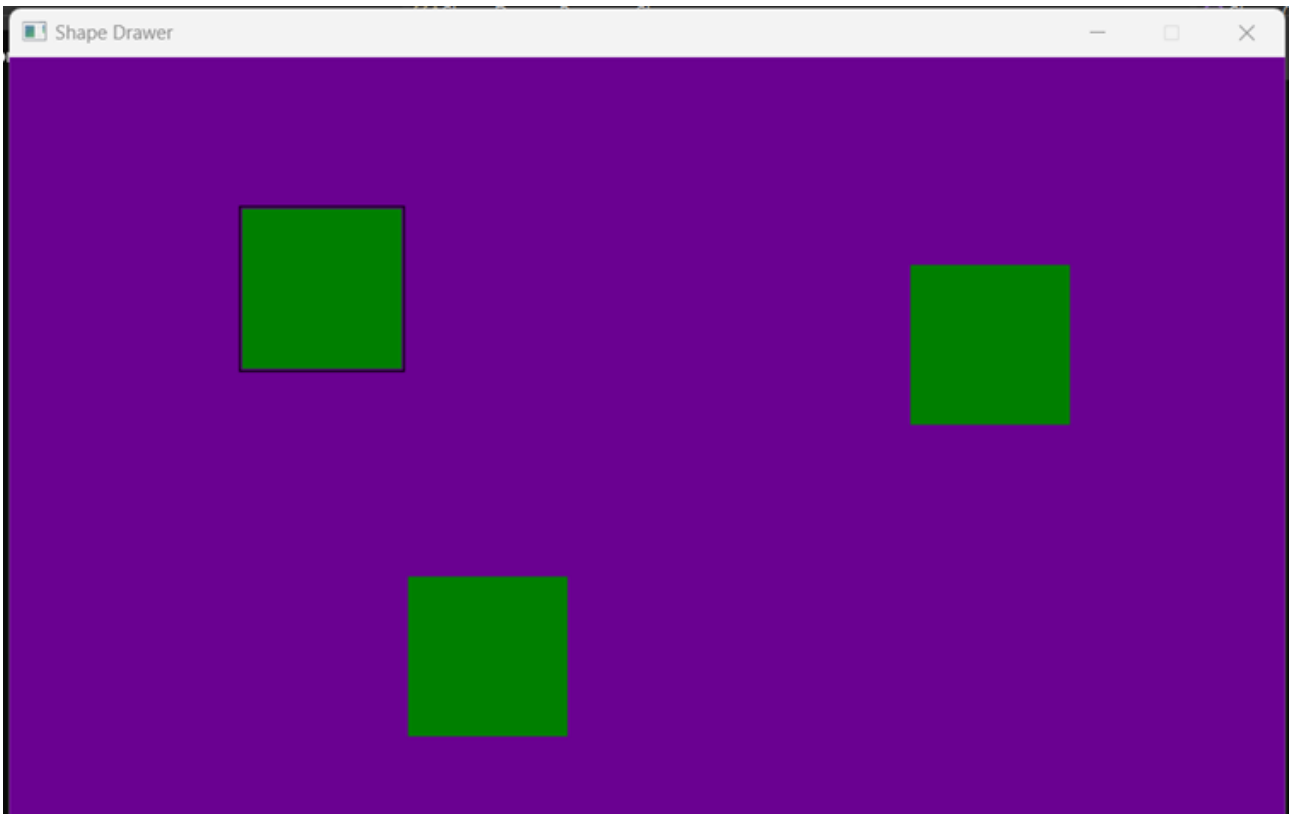
```
using System;
using SplashKitSDK;
using System.Collections.Generic;

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

            Drawing draw = new Drawing();
            do
            {
                SplashKit.ProcessEvents();

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

                if (SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    draw.SelectShapeAt(SplashKit.MousePosition());
                }

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

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

                draw.Draw();

                SplashKit.RefreshScreen();

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

```
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)
        {
            _shapes = new List<Shape>();
        }

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

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

        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 void AddShape(Shape newShape)
```

```
54          {
55              _shapes.Add(newShape);
56          }
57
58          public void Draw()
59          {
60              SplashKit.ClearScreen(_background);
61              foreach (Shape shape in _shapes)
62              {
63                  shape.Draw();
64              }
65          }
66
67          public void SelectShapeAt(Point2D pt)
68          {
69              foreach (Shape s in _shapes)
70              {
71                  if (s.IsAt(pt))
72                  {
73                      s.Selected = true;
74                  }
75              }
76          }
77
78          public void RemoveShape(Shape shape)
79          {
80              _shapes.Remove(shape);
81          }
82      }
83  }
```

```
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
        {
            set { _color = value; }
            get { return _color; }
        }

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

        public float Y
        {
            set { _y = value; }
            get { return _y; }
        }

        public int Width
        {
            set { _width = value; }
            get { return _width; }
        }

        public int Height
        {
            set { _height = value; }
```

```
54              get { return _height; }
55          }
56
57          public bool Selected
58          {
59              set { _selected = value; }
60              get { return _selected; }
61          }
62
63          public void Draw()
64          {
65              SplashKit.FillRectangle(_color, _x, _y, _width, _height);
66              if (Selected == true)
67              {
68                  DrawOutline();
69              }
70          }
71
72          public bool IsAt(Point2D pt)
73          {
74
75              if ((X < pt.X) && (X + _width > pt.X) && (Y < pt.Y) && (Y + _height >
    ↪  pt.Y))
76              {
77                  return true;
78              }
79              else
80              {
81                  return false;
82              }
83          }
84
85          public void DrawOutline()
86          {
87              SplashKit.DrawRectangle(Color.Black, (X - 2), (Y - 2), (Width + 4),
    ↪  (Height + 4));
88          }
89      }
90  }
```


Shape Drawer