Swinburne University of Technology

COS20007 Object Oriented Programming

3.3P - Drawing Program - A Drawing Class

PDF generated at 13:21 on Sunday 26th March, 2023

```
using System;
using System.ComponentModel.Design;
using System.Globalization;
using System.Reflection.Metadata.Ecma335;
using SplashKitSDK;

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

            do
            {
                SplashKit.ProcessEvents();
                SplashKit.ClearScreen();

                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape NewShape = new Shape();
                    NewShape.X = SplashKit.MouseX();
                    NewShape.Y = SplashKit.MouseY();

                    NewDrawing.AddShape(NewShape);
                }

                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
                {
                    NewDrawing.Background = Color.RandomRGB(255);
                }

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

                if (SplashKit.KeyTyped(KeyCode.BackspaceKey) ||
↪   SplashKit.KeyTyped(KeyCode.DeleteKey))
                {
                    foreach (Shape s in NewDrawing.SelectedShape())
                    {
                        NewDrawing.RemoveShape(s);
                    }
                }

                NewDrawing.Draw();
                SplashKit.RefreshScreen();

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

```
53
54              }
55
56          }
57
58
59  }
```

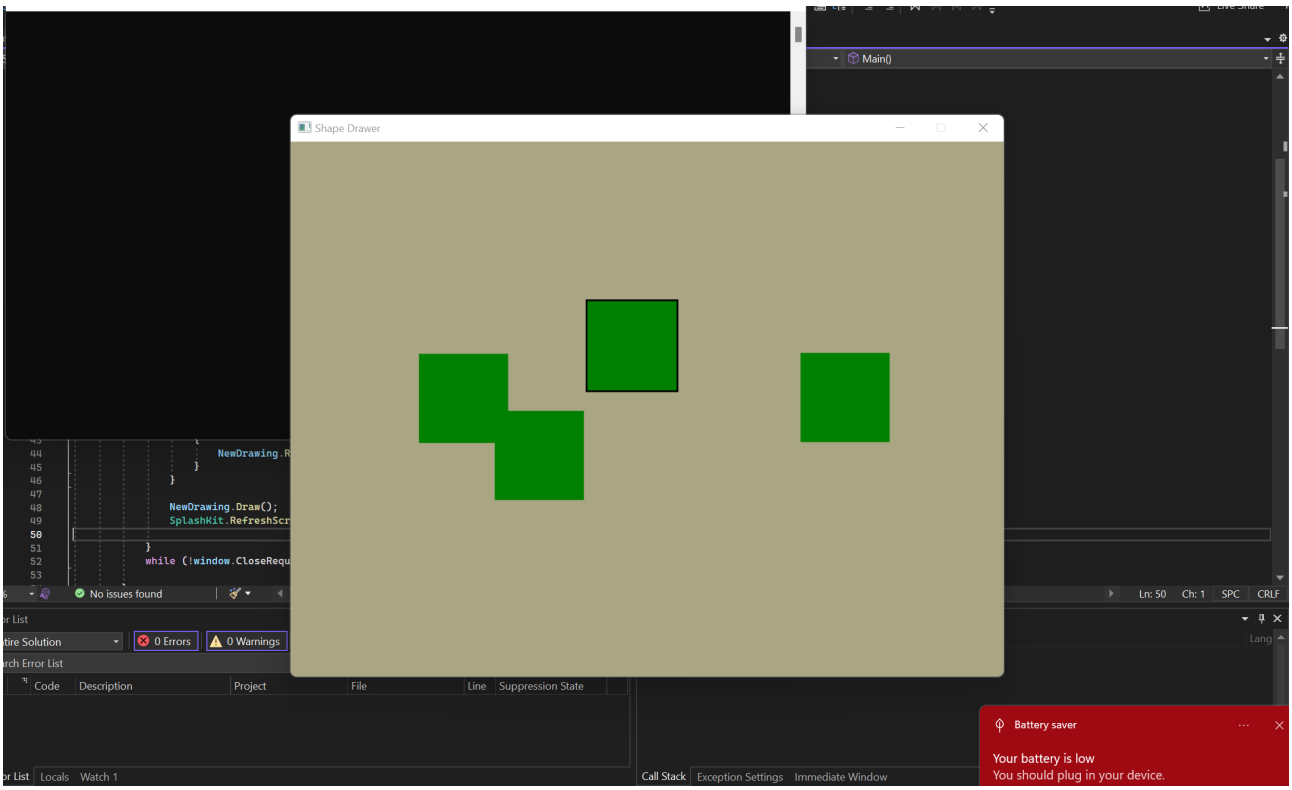
```
using SplashKitSDK;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;



namespace ShapeDrawer
{
    internal 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) { }

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

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


        public void Draw()
        {
            SplashKit.ClearScreen(_background);
            foreach (Shape s in _shapes)
            {
                s.Draw();
            }
        }

        public Color Background
        {
            get
            {
                return _background;
```

```
54              }
55              set
56              {
57                  _background = value;
58              }
59          }
60
61          public void SelectShapeAt(Point2D pt)
62          {
63              foreach (Shape s in _shapes)
64              {
65                  if (s.IsAt(pt))
66                  {
67                      s.Selected = true;
68                  }
69                  else
70                  {
71                      s.Selected = false;
72                  }
73              }
74
75          }
76
77          public List<Shape> SelectedShape()
78          {
79              List<Shape> result = new List<Shape>();
80              foreach (Shape s in _shapes)
81              {
82                  if (s.Selected)
83                  {
84                      result.Add(s);
85                  }
86              }
87              return result;
88          }
89
90          public void RemoveShape(Shape s)
91          {
92              _shapes.Remove(s);
93
94          }
95
96      }
97
98  }
```

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

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.Green;
            _x = 0;
            _y = 0;
            _width = 100;
            _height = 100;

        }
        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 +_width)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        public float X
        {
            get
            {
                return _x;
```

```
54              }
55              set
56              {
57                  _x = value;
58              }
59
60          }
61          public float Y
62          {
63              get
64              {
65                  return _y;
66              }
67              set
68              {
69                  _y = value;
70              }
71          }
72          public int Width
73          {
74              get
75              {
76                  return _width;
77              }
78              set
79              {
80                  _width = value;
81              }
82          }
83          public int Height
84          {
85              get
86              {
87                  return _height;
88              }
89              set
90              {
91                  _height = value;
92              }
93          }
94          public Color Color
95          {
96              get
97              {
98                  return _color;
99              }
100             set
101             {
102                 _color = value;
103             }
104         }
105         public bool Selected
106         {
```

```
107             get
108             {
109                 return _selected;
110             }
111             set
112             {
113                 _selected = value;
114             }
115         }
116 
117         public void DrawOutline()
118         {
119             SplashKit.FillRectangle(Color.Black, X - 2, Y - 2, Width + 4, Height +
    ↪ 4);
120         }
121     }
122 }
```


Shape Drawer
Main()
NewDrawing.Draw();
SplashKit.RefreshScr
while (!window.CloseRequ
No issues found
Ln: 50
Ch: 1
SPC
CRLF
0 Errors
0 Warnings
Code
Description
Project
File
Line
Suppression State
Locals
Watch 1
Call Stack
Exception Settings
Immediate Window
Battery saver
Your battery is low
You should plug in your device.