Swinburne University of Technology

COS20007 Object Oriented Programming

Drawing Program - A Drawing Class

PDF generated at 13:09 on Friday 25th August, 2023

```
using System;
using SplashKitSDK;

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

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

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

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

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

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

                drawing.Draw();

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

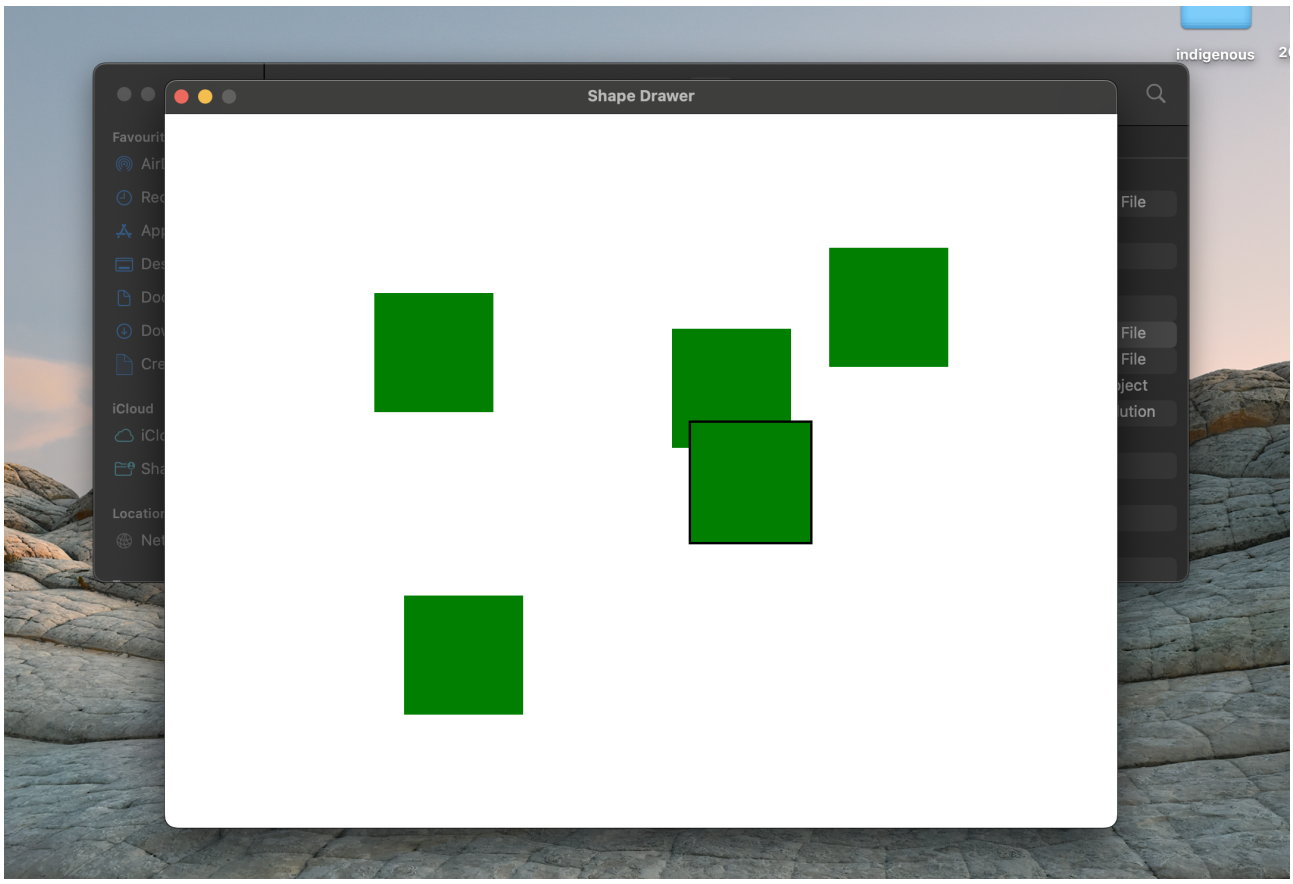
```
using System;
using System.Collections.Generic;
using SplashKitSDK;

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)
        {

        }
        public List<Shape> SelectedShapes
        {
            get
            {
                List<Shape> _selectedShapes = new List<Shape>();
                foreach (Shape s in _shapes)
                {
                    if (s.Selected)
                    {
                        _selectedShapes.Add(s);
                    }
                }
                return _selectedShapes;
            }
        }

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

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

        public Color Background
        {
            get
```

```
54              {
55                  return _background;
56              }
57              set
58              {
59                  _background = value;
60              }
61          }
62
63          public void Draw()
64          {
65              SplashKit.ClearScreen(Background);
66              foreach (Shape shape in _shapes)
67              {
68                  shape.Draw();
69              }
70          }
71          public void SelectShapesAt(Point2D pt)
72          {
73              foreach (Shape s in _shapes)
74              {
75                  if (s.IsAt(pt))
76                  {
77                      s.Selected = true;
78                  }
79                  else s.Selected = false;
80              }
81
82          }
83
84      }
85  }
86
```

```
using System;
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.Green;
            _x = 0;
            _y = 0;
            _width = 100;
            _height = 100;
            _selected = false;
        }

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

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

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

```
53
54          public float Y
55          {
56              get
57              {
58                  return _y;
59              }
60              set
61              {
62                  _y = value;
63              }
64          }
65
66          public int Width
67          {
68              get
69              {
70                  return _width;
71              }
72              set
73              {
74                  _width = value;
75              }
76          }
77
78          public int Height
79          {
80              get
81              {
82                  return _height;
83              }
84              set
85              {
86                  _height = value;
87              }
88          }
89
90          public Color Color
91          {
92              get
93              {
94                  return _color;
95              }
96              set
97              {
98                  _color = value;
99              }
100         }
101
102         public void Draw()
103         {
104             if (Selected)
105             {
```

```
106                 DrawOutline();
107             }
108             SplashKit.FillRectangle(_color, _x, _y, _width, _height);
109         }
110     public bool IsAt(Point2D pt)
111         {
112             return (pt.X >= _x && pt.X <= (_x + _width)) && (pt.Y >= _y && pt.Y <=
    ↪  (_y + _height));
113         }
114     }
115 }
116
```


Shape Drawer
indigenous