Swinburne University of Technology

COS20007 Object Oriented Programming

3.3P - Drawing Program - A Drawing Class

PDF generated at 13:20 on Wednesday 8th March, 2023

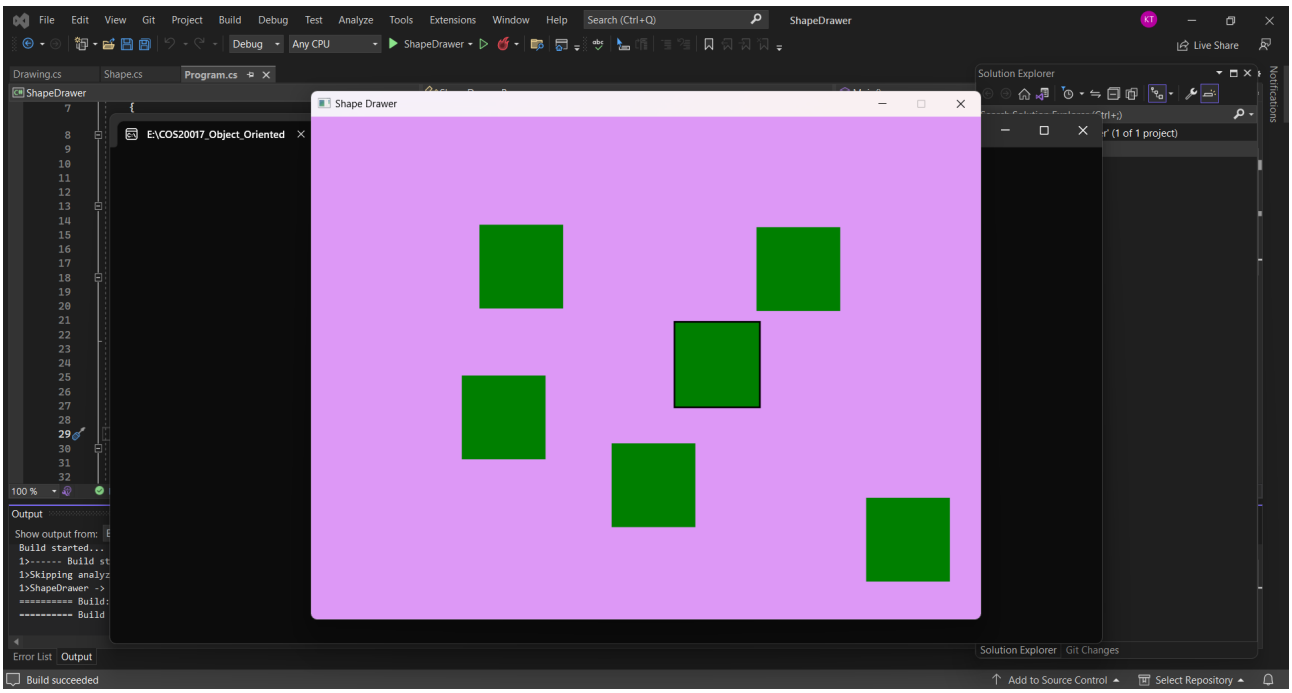
```
using System;
using SplashKitSDK;

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

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

                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape newShape = new Shape(Color.Green, SplashKit.MouseX(),
↪ SplashKit.MouseY(), 100, 100);
                    myDrawing.AddShape(newShape);
                }

                if (SplashKit.KeyDown(KeyCode.SpaceKey))
                    myDrawing.Background = SplashKit.RandomRGBColor(255);

                if (SplashKit.MouseClicked(MouseButton.RightButton))
                    myDrawing.SelectedShapesAt(SplashKit.MousePosition());

                if ((SplashKit.KeyTyped(KeyCode.BackspaceKey)) ||
↪ (SplashKit.KeyTyped(KeyCode.DeleteKey)))
                {
                    foreach (Shape shape in myDrawing.SelectedShapes)
                        myDrawing.RemoveShape(shape);
                }
                myDrawing.Draw();

                SplashKit.RefreshScreen();

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

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

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 shape in _shapes)
                {
                    if (shape.Selected)
                        selectedShapes.Add(shape);
                }
                return selectedShapes;
            }
        }
        public 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)
                shape.Draw();
        }
        public void SelectedShapesAt(Point2D point)
        {
            foreach (Shape shape in _shapes)
```

```
54              {
55                  if (shape.IsAt(point))
56                      shape.Selected = true;
57                  else
58                      shape.Selected = false;
59              }
60          }
61
62          public void AddShape(Shape shape)
63          {
64              _shapes.Add(shape);
65          }
66          public void RemoveShape(Shape shape)
67          {
68              _shapes.Remove(shape);
69          }
70      }
71  }
```

```
using SplashKitSDK;
using System;
using System.Security.Cryptography.X509Certificates;
using System.Threading.Tasks.Dataflow;

namespace ShapeDrawer
{
    public class Shape
    {
        private Color _color;

        private float _x;
        private float _y;

        private float _width;
        private float _height;

        private bool _selected;
        public Shape(Color color, float x, float y, float width, float height)
        {
            _color = color;

            _x = x;
            _y = y;

            _width = width;
            _height = 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 float Width
        {
            get { return _width; }
            set { _width = value; }
        }
        public float Height
        {
            get { return _height; }
            set { _height = value; }
```

```
54          }
55
56          public void Draw()
57          {
58              if (_selected)
59                  DrawOutline();
60
61              SplashKit.FillRectangle(_color, _x, _y, _width, _height);
62          }
63
64          public bool IsAt(Point2D point2D)
65          {
66              if (point2D.X >= _x && point2D.X < _x + _width && point2D.Y >= _y &&
    ↪  point2D.Y < _y + _height)
67                  return true;
68              else
69                  return false;
70          }
71
72          public bool Selected
73          {
74              get { return _selected; }
75              set { _selected = value; }
76          }
77
78          public void DrawOutline()
79          {
80              SplashKit.FillRectangle(Color.Black, _x - 2, _y - 2, _width + 4, _height
    ↪  + 4);
81          }
82      }
83  }
```


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
ShapeDrawer
Debug
Any CPU
ShapeDrawer
Live Share
Drawing.cs
Shape.cs
Program.cs
ShapeDrawer
Shape Drawer
E:\COS20017_Object_Oriented
Solution Explorer
Output
Show output from:
Build started...
1>------ Build st
1>Skipping analyz
1>ShapeDrawer ->
========== Build:
---------- Build
Error List
Output
Solution Explorer
Git Changes
Build succeeded
Add to Source Control
Select Repository