Swinburne University of Technology

COS20007 Object Oriented Programming

Drawing Program - A Drawing Class

PDF generated at 02:08 on Monday 25th September, 2023

```
using System;
using SplashKitSDK;

namespace DrawingProgram
{
    public class Program
    {
        public static void Main()
        {
            Drawing myDraw = new();
            Window window = new("Shape Drawer 2", 800, 600); //draw window
            do
            {
                SplashKit.ProcessEvents();
                SplashKit.ClearScreen(Color.White);
                if (SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    Shape myShape = new()
                    {
                        X = SplashKit.MouseX(), //change the myShape's X value
                        Y = SplashKit.MouseY() //change the myShape's Y value
                    };
                    myDraw.AddShape(myShape);
                }
                if (SplashKit.KeyTyped(KeyCode.SpaceKey))
                {
                    myDraw.Background = SplashKit.RandomColor();
                }
                if (SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    myDraw.SelectShapesAt(SplashKit.MousePosition());
                }
                List<Shape> select = new();
                select = myDraw.SelectedShapes;
                if (SplashKit.KeyTyped(KeyCode.DeleteKey) ||
    ↪   SplashKit.KeyTyped(KeyCode.BackspaceKey))
                {
                    foreach (Shape s in select)
                    {
                        myDraw.RemoveShape(s);
                    }
                }
                myDraw.Draw();
                SplashKit.RefreshScreen();
            } while (!SplashKit.WindowCloseRequested("Shape Drawer 2")) ;
        }

    }
}
```

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

namespace DrawingProgram
{
    public class Drawing
    {
        public readonly List<Shape> oshapes;
        private Color obackground;
        public Color Background
        {
            get
            {
                return obackground;
            }
            set
            {
                obackground = value;
            }
        }
        public Drawing(Color background)
        {
            oshapes = new List<Shape>();
            obackground = background;
        }
        public Drawing() : this(Color.White) { }
        public List<Shape> SelectedShapes
        {
            get
            {
                List<Shape> result = new();
                foreach (Shape s in oshapes)
                {
                    if (s.Selected == true)
                     {
                        result.Add(s);
                    }
                }
                return result;
            }
        }
        public int ShapeCount
        {
            get
            {
                return oshapes.Count;
            }
        }
        public void AddShape(Shape shape)
```

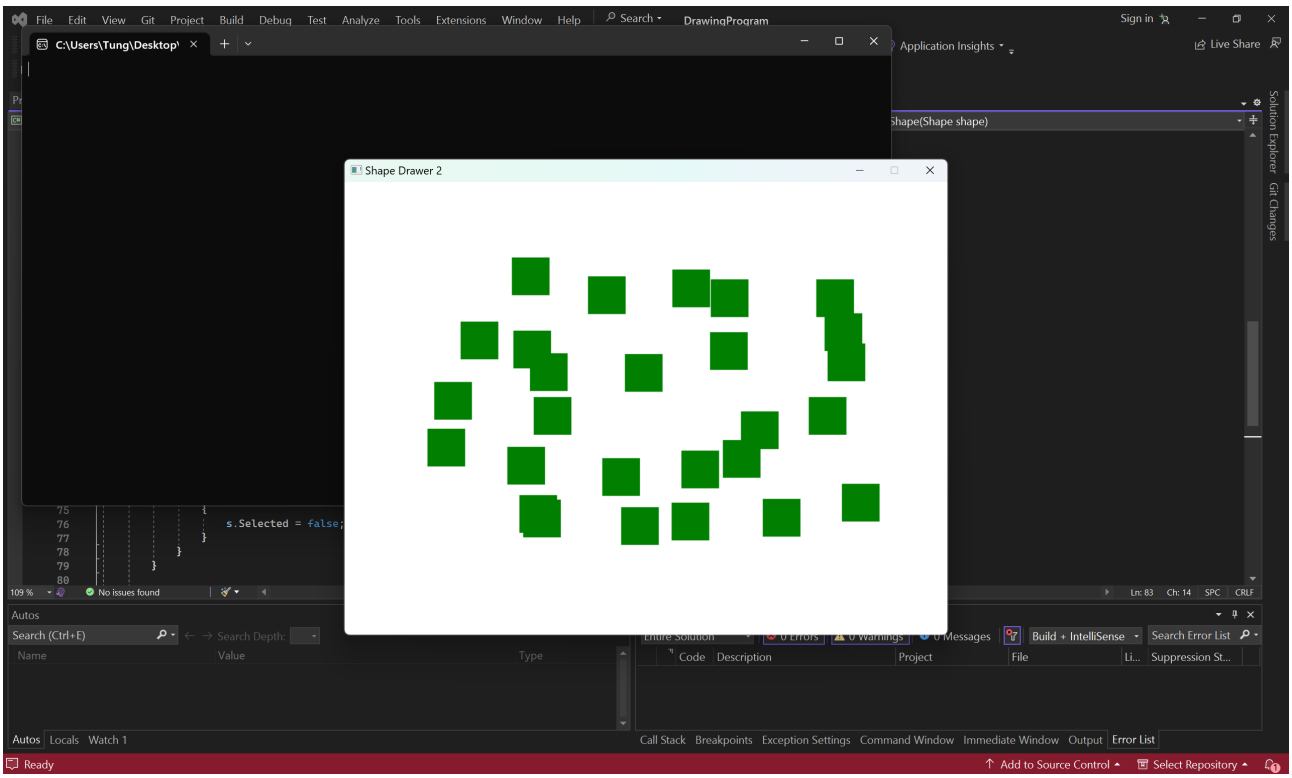
```
54          {
55              oshapes.Add(shape);
56          }
57
58          public void Draw()
59          {
60              SplashKit.ClearScreen(obackground);
61              foreach (Shape shape in oshapes)
62              {
63                  shape.Draw();
64              }
65          }
66          public void SelectShapesAt(Point2D pt)
67          {
68              foreach (Shape s in oshapes)
69              {
70                  if (s.IsAt(pt))
71                  {
72                      s.Selected = true;
73                  }
74                  else
75                  {
76                      s.Selected = false;
77                  }
78              }
79          }
80
81          public void RemoveShape(Shape shape)
82          {
83              oshapes.Remove(shape);
84          }
85
86      }
87  }
```

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

namespace DrawingProgram
{
    public class Shape
    {
        private bool oselected;
        private float ox, oy;
        private int owidth, oheight;
        private Color ocolor;
        public float X
        {
            get
            {
                return ox;
            }
            set
            {
                ox = value;
            }
        }
        public float Y
        {
            get
            {
                return oy;
            }
            set
            {
                oy = value;
            }
        }
        public int Width
        {
            get
            {
                return owidth;
            }
            set
            {
                owidth = value;
            }
        }
        public int Height
        {
            get
            {
                return oheight;
```

```
54              }
55              set
56              {
57                  oheight = value;
58              }
59          }
60          public bool Selected
61          {
62              get
63              {
64                  return oselected;
65              }
66              set
67              {
68                  oselected = value;
69              }
70          }
71          public Color Color
72          {
73              get
74              {
75                  return ocolor;
76              }
77              set
78              {
79                  ocolor = value;
80              }
81          }
82          public Shape()
83          {
84              ox = 0;
85              oy = 0;
86              owidth = 50;
87              oheight = 50;
88              ocolor = SplashKit.ColorGreen();
89          }
90          public Shape(Color color, float x, float y, int width, int height)
91          {
92              ocolor = color;
93              ox = x;
94              oy = y;
95              owidth = width;
96              oheight = height;
97          }
98          public void Draw()
99          {
100             if (oselected == true)
101             {
102                 DrawOutline();
103             }
104             SplashKit.FillRectangle(ocolor, ox, oy, owidth, oheight); //draw shape
105         }
106
```

```
107         public bool IsAt(Point2D pt) //the result return bool so need to set bool
    ↪ here, pt is param
108         {
109             if ((((pt.X >= ox) && (pt.X <= (ox + owidth))) && (pt.Y >= oy) && (pt.Y
    ↪ <= (oy + oheight))))
110             // mouse x-coor >= shape x-coor && mouse x-coor <= shape x-coor + shape
    ↪ width
111             // mouse y-coor >= shape y-coor && mouse y-coor <= shape y-coor + height
112             {
113                 return true;
114             }
115             else
116             {
117                 return false;
118             }
119         }
120 
121         public void DrawOutline()
122         {
123             SplashKit.FillRectangle(SplashKit.ColorBlack(), ox - 2, oy - 2, owidth +
    ↪ 4, oheight + 4); //draw shape
124         }
125     }
126 }
```


Shape Drawer 2
C:\Users\Tung\Desktop\
DrawingProgram
Sign in
Live Share
Application Insights
Shape(Shape shape)
s.Selected = false;
No issues found
Ln: 83 Ch: 14 SPC CRLF
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
Code
Description
Project
File
Suppression St...
Build + IntelliSense
Search Error List
Autos
Locals
Watch 1
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Error List
Ready
Add to Source Control
Select Repository