Swinburne University of Technology

COS20007 Object Oriented Programming

Drawing Program - A Drawing Class

PDF generated at 00:16 on Wednesday 2nd August, 2023

```
using System;
using SplashKitSDK;

namespace DrawingProgram
{
    public class Program
    {
        private const string _title = "Drawing Program";
        private const int _width = 800;
        private const int _height = 600;
        public static void ClearScreen()
        {
            SplashKit.ProcessEvents();
            SplashKit.ClearScreen();
        }

        public static void Main()
        {
            Drawing drawing = new Drawing();
            Window window = new Window(_title, _width, _height);

            while(!window.CloseRequested)
            {
                ClearScreen();
                //Draw Shape when click Left Button
                if(SplashKit.MouseClicked(MouseButton.LeftButton))
                {
                    int xPosition = (int) SplashKit.MouseX();
                    int yPosition = (int) SplashKit.MouseY();
                    drawing.AddShape(new Shape(xPosition, yPosition));
                }
                //Draw Outline (Selected shape) when click Right Button
                if(SplashKit.MouseClicked(MouseButton.RightButton))
                {
                    drawing.SelectShapesAt(SplashKit.MousePosition());
                }
                //Randomly change color when click ESC
                if(SplashKit.KeyDown(KeyCode.EscapeKey))
                {
                    foreach(Shape s in drawing.SelectedShapes)
                    {
                        s.color = SplashKit.RandomRGBColor(255);
                    }
                }
                //Change background color when click Space
                if(SplashKit.KeyDown(KeyCode.SpaceKey))
                {
                    drawing.Background = SplashKit.RandomRGBColor(255);
                }
                //Remove shape if press DEL or Backspace
                if (SplashKit.KeyDown(KeyCode.DeleteKey) ||
↪   SplashKit.KeyDown(KeyCode.BackspaceKey))
                {
```

```
53                      foreach(Shape s in drawing.SelectedShapes)
54                      {
55                          drawing.RemoveShape(s);
56                      }
57                  }
58                  drawing.Draw();
59                  SplashKit.RefreshScreen();
60              }
61          }
62      }
63  }
```

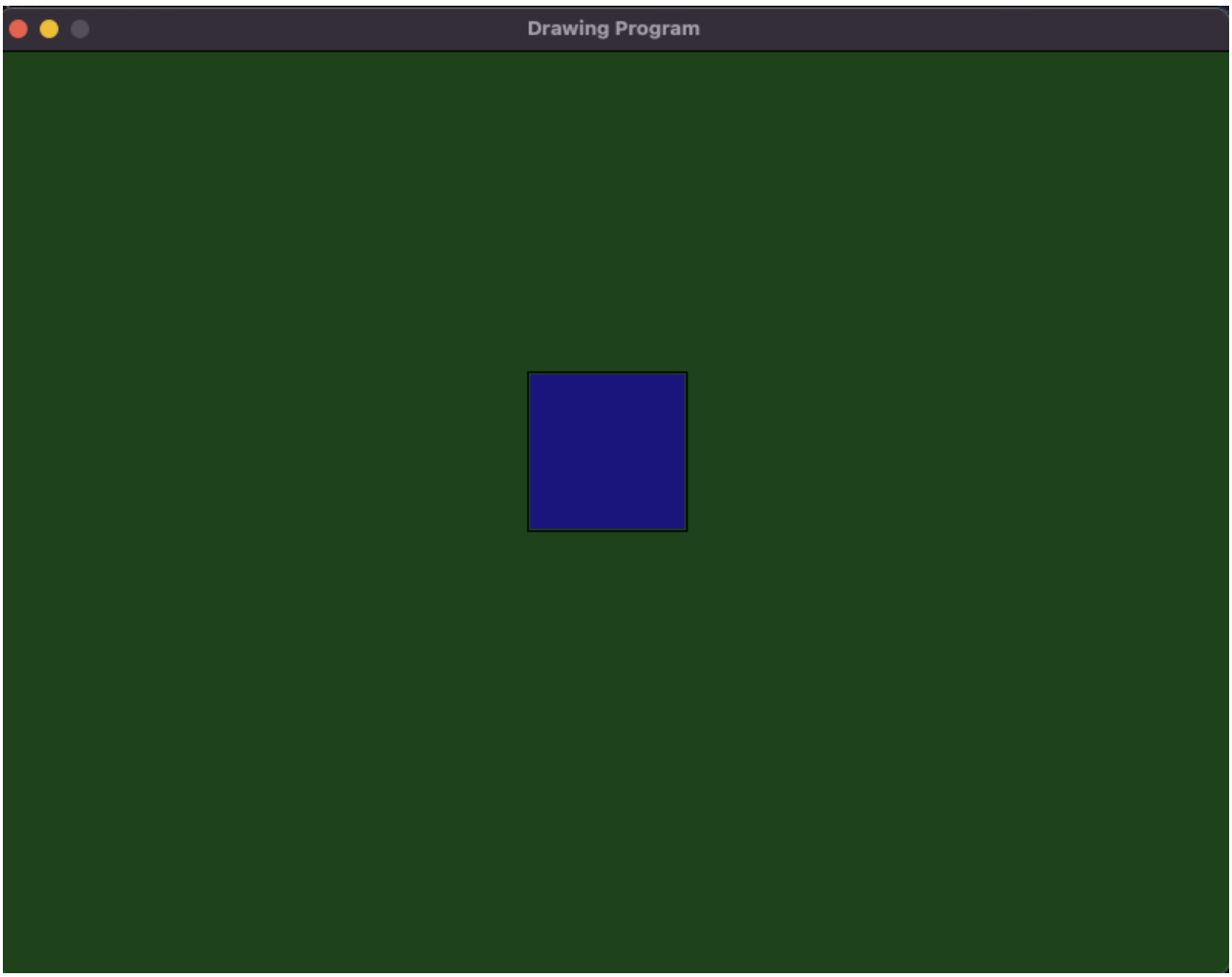
```
using System;
using SplashKitSDK;
using System.Collections.Generic;
namespace DrawingProgram
{
    public class Drawing
    {
        private readonly List<Shape> _shapes;
        private Color _background;

        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 Color Background
        {
            get
            {
                return _background;
            }

            set
            {
                _background = value;
            }
        }
```

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

```
using System;
using SplashKitSDK;

namespace DrawingProgram
{
    public class Shape
    {
        private Color _color;
        private float _x, _y;
        private bool _selected;
        private int _width, _height;

        public Shape(int x, int y)
        {
            _color = SplashKit.RGBColor(255, 0, 255);
            _x = x;
            _y = y;
            _width = 100;
            _height = 100;

        }

        public Shape() : this(0, 0) { }

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

        public int Height
        {
            get
            {
                return _height;
            }
            set
            {
                _height = value;
            }
        }

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

        public void Draw()
        {
            if (Selected)
                DrawOutline();
            SplashKit.FillRectangle(_color, _x, _y, _width, _height);
        }

        public bool IsAt(Point2D pt)
        {
            return SplashKit.PointInRectangle(pt, SplashKit.RectangleFrom(_x, _y,
↪  _width, _height)) ? true : false;
```

```
107         }
108
109         public void DrawOutline()
110         {
111             SplashKit.DrawRectangle(Color.Black, _x - 2, _y - 2, _width + 4, _height
    ↪  + 4);
112         }
113     }
114 }
115
```


Drawing Program