## SWINBURNE UNIVERSITY OF TECHNOLOGY

## COS20007 OBJECT ORIENTED PROGRAMMING

## Drawing Program - Saving and Loading

PDF generated at 03:01 on Saturday  $30^{\rm th}$  September, 2023

File 1 of 8 Program class

```
using System;
   using SplashKitSDK;
   namespace Task5._3
        public class Program
5
6
            private enum ShapeKind
                 Rectangle, Circle, Line
            }
10
11
            public static void Main()
12
13
                 Window window = new Window("Shape Drawer", 800, 600);
15
                 Drawing dr = new Drawing();
                 ShapeKind kindToAdd = ShapeKind.Circle;
17
                 do
18
                 {
19
                     SplashKit.ProcessEvents();
20
                     SplashKit.ClearScreen();
22
                     if (SplashKit.MouseClicked(MouseButton.LeftButton))
23
24
                         Shape newShape;
25
                         if (kindToAdd == ShapeKind.Circle)
26
27
                              newShape = new MyCircle();
                         }
29
                         else if (kindToAdd == ShapeKind.Rectangle)
30
31
                              newShape = new MyRectangle();
32
                         }
                         else
34
                         {
35
                              newShape = new MyLine();
36
                         }
37
38
                         newShape.X = SplashKit.MouseX();
39
40
                         newShape.Y = SplashKit.MouseY();
41
42
                         dr.AddShape(newShape);
43
                     }
44
45
                     if
                         (SplashKit.KeyTyped(KeyCode.RKey))
46
                     {
47
                         kindToAdd = ShapeKind.Rectangle;
48
                     }
49
50
                     if (SplashKit.KeyTyped(KeyCode.CKey))
51
                     {
52
                         kindToAdd = ShapeKind.Circle;
53
```

File 1 of 8 Program class

```
}
54
                      if
                          (SplashKit.KeyTyped(KeyCode.LKey))
55
                      {
56
                           kindToAdd = ShapeKind.Line;
                      }
58
                          (SplashKit.MouseClicked(MouseButton.RightButton))
                      if
59
                      {
60
61
                           dr.SelectShapesAt(SplashKit.MousePosition());
62
                      }
63
64
                          (SplashKit.KeyTyped(KeyCode.SpaceKey))
65
                      {
66
67
                           dr.Background1 = SplashKit.RandomRGBColor(255);
68
                      }
70
                      if (SplashKit.KeyTyped(KeyCode.DeleteKey) ||
         SplashKit.KeyTyped(KeyCode.BackspaceKey))
72
73
                           foreach (Shape s in dr. Selectedshapes)
74
                           {
75
                               dr.DeleteShape(s);
76
                           }
77
                      }
78
                      if (SplashKit.KeyTyped(KeyCode.SKey))
79
                      {
                           dr.Save("Desktop\\TestDrawing.txt");
81
                      }
82
                          (SplashKit.KeyTyped(KeyCode.OKey))
83
                      {
84
                           try
                           {
86
                               dr.Load("Desktop\\TestDrawing.txt");
87
                           }
88
                           catch(Exception e)
89
                           {
90
                               Console.Error.WriteLine("Error Loadinhg file: {0}",
91
         e.Message);
                           }
92
                      }
93
94
95
                      dr.Draw();
97
                      SplashKit.RefreshScreen();
98
                  } while (!window.CloseRequested);
99
             }
100
         }
102
    }
103
104
```

File 1 of 8 Program class

File 2 of 8 ExtensionMethods class

```
using System;
   using SplashKitSDK;
   namespace Task5._3
       public static class ExtensionMethods
5
6
            public static int ReadInteger(this StreamReader reader)
            {
                return Convert.ToInt32(reader.ReadLine());
            public static float ReadSingle(this StreamReader reader)
12
                return Convert.ToSingle(reader.ReadLine());
13
            public static Color ReadColor(this StreamReader reader)
15
                return Color.RGBColor(reader.ReadSingle(), reader.ReadSingle(),
17
                reader.ReadSingle());
18
19
            public static void WriteColor(this StreamWriter writer, Color clr)
20
                writer.WriteLine("\{0\}\n\{1\}\n\{2\}", clr.R, clr.G, clr.B);
22
            }
23
        }
24
   }
25
26
```

File 3 of 8 Drawing class

```
using System;
   using SplashKitSDK;
   using System.Collections.Generic;
   using System.Reflection.PortableExecutable;
   namespace Task5._3
6
   {
        public class Drawing
        {
10
            private readonly List<Shape> _shapes;
11
            private Color _background;
12
            public Drawing(Color bg)
13
                 _shapes = new List<Shape>();
15
                 _background = bg;
17
18
            public Drawing() : this(Color.White)
19
            {
20
            }
22
            public int ShapeCount
23
24
                get
25
                 {
26
                     return _shapes.Count;
27
29
            public void AddShape(Shape s)
30
31
                 _shapes.Add(s);
32
            public void Draw()
34
            {
35
                 SplashKit.ClearScreen(Background1);
36
                 foreach (Shape s in _shapes)
37
                     s.Draw();
39
                 }
40
41
            }
42
43
            public void SelectShapesAt(Point2D pt)
                 foreach (Shape s in _shapes)
46
47
                     if (s.IsAt(pt))
48
                     {
49
                          s.Selected = true;
50
                     }
51
                     else
52
                     {
53
```

File 3 of 8 Drawing class

```
s.Selected = false;
54
                      }
55
                  }
56
             }
             public List<Shape> Selectedshapes
58
59
                  get
60
                  {
61
                      List<Shape> _result = new List<Shape>();
62
63
64
                       foreach (Shape s in _shapes)
65
66
                           if (s.Selected == true)
67
                           {
68
                                _result.Add(s);
                           }
70
                       }
                      return _result;
72
                  }
73
             }
             public Color Background1
             {
76
                  get
77
                  {
78
                       return _background;
79
                  }
                  set
                  {
82
                       _background = value;
83
                  }
84
             }
85
             public void DeleteShape(Shape s)
87
             {
88
                  _shapes.Remove(s);
89
             }
90
             public void Save(string filename)
92
             {
93
                  StreamWriter writer = new StreamWriter(filename);
94
                  //Shape s;
95
96
                  writer.WriteColor(Background1);
                  writer.WriteLine(ShapeCount);
99
                  foreach(Shape s in _shapes)
100
101
                       s.SaveTo(writer);
102
                  }
103
                  writer.Close();
104
105
             public void Load(string filename)
106
```

File 3 of 8 Drawing class

```
{
107
                  StreamReader reader = new StreamReader(filename);
108
                  try {
109
                  int count;
111
                  Shape s;
112
                  string kind;
113
                  Background1 = reader.ReadColor();
114
                  count = reader.ReadInteger();
116
                  _shapes.Clear();
117
118
                       for (int i = 0; i < count; i += 1)
119
120
                           kind = reader.ReadLine();
121
                           switch (kind)
122
123
                                case "Rectangle":
124
                                     s = new MyRectangle();
125
                                     break;
126
                                case "Circle":
128
                                     s = new MyCircle();
129
                                     break;
130
                                case "Line":
131
                                     s = new MyLine();
132
                                     break;
133
                                default:
134
                                     throw new InvalidDataException("Unknown Shape Kind: " +
135
        kind);
                           }
136
137
                           s.LoadFrom(reader);
                           AddShape(s);
139
                       }
140
                  }
141
                  finally
142
                       reader.Close();
144
145
146
             }
147
         }
148
149
    }
150
151
152
153
```

File 4 of 8 Shape class

```
using System;
    using SplashKitSDK;
   namespace Task5._3
5
6
        public abstract class Shape
             protected Color _color;
             private float _x;
10
             private float _y;
11
12
             private bool _selected;
13
14
             public Shape(Color c)
15
16
                 _color = c;
17
                 _x = (float)0;
18
                 _y = (float)0;
19
20
             public Shape():this(Color.Yellow)
22
             {
23
24
             }
25
26
             public abstract void Draw();
27
28
             public Color Color
29
             {
30
                 get
31
                  {
32
                      return _color;
34
                 set
35
36
                      _color = value;
37
                  }
38
             }
39
40
             public float X
41
42
                 get
43
                  {
44
                      return _x;
                 }
46
                 set
47
48
                      _x = value;
49
                 }
50
             }
51
             public float Y
52
             {
53
```

File 4 of 8 Shape class

```
get
54
                  {
55
                      return _y;
56
                  }
                  set
58
                  {
59
                      _y = value;
60
                 }
61
62
             }
63
64
             public bool Selected
65
66
                 get
67
                  {
68
                      return _selected;
                 }
70
                 set
71
72
                      _selected = value;
73
                  }
             }
75
             public abstract bool IsAt(Point2D pt);
76
77
             public abstract void Outline();
78
79
             public virtual void SaveTo(StreamWriter writer)
             {
                 writer.WriteColor(Color);
82
                 writer.WriteLine(X);
83
                 writer.WriteLine(Y);
84
85
             }
             public virtual void LoadFrom(StreamReader reader)
87
             {
88
                 Color = reader.ReadColor();
89
                 X = reader.ReadInteger();
90
                 Y = reader.ReadInteger();
             }
92
        }
93
   }
94
95
96
97
```

File 5 of 8 MyRectangle class

```
using System;
   using SplashKitSDK;
   namespace Task5._3
   {
5
        public class MyRectangle:Shape
6
            private int _width;
            private int _height;
            public MyRectangle():this(Color.Green, 0, 0, 100, 100)
            {
12
            }
13
            public MyRectangle(Color clr, float x, float y, int width, int
        height):base(clr)
            {
15
16
                 Width = width;
17
                 Height = height;
18
            }
19
            public int Width
21
                 get
22
                 {
23
                     return _width;
24
                 }
25
                 set
26
                      _width = value;
28
                 }
29
            }
30
31
            public int Height
33
                 get
34
                 {
35
                     return _height;
36
38
                 set
                 {
39
                      _height = value;
40
                 }
41
42
43
            public override void Draw()
45
                 if (Selected == true)
46
47
                      Outline();
48
49
                 SplashKit.FillRectangle(Color, X, Y, _width, _height);
50
51
            }
52
```

File 5 of 8 MyRectangle class

```
public override void Outline()
53
            {
54
                 SplashKit.FillRectangle(Color.Black, X - 2, Y - 2, _width + 4, _height +
55
        4);
            }
56
57
            public override bool IsAt(Point2D pt)
58
            {
59
60
                      if (pt.X \ge X \&\& pt.X < (X + \_width) \&\& pt.Y \ge Y \&\& pt.Y <= (Y + \_width))
61
        _height))
62
63
                      return true;
64
                 }
65
                 else
                 {
67
                     return false;
68
69
            }
70
            public override void SaveTo(StreamWriter writer)
72
                 writer.WriteLine("Rectangle");
                 base.SaveTo(writer);
74
75
                 writer.WriteLine(Width);
76
                 writer.WriteLine(Height);
79
            public override void LoadFrom(StreamReader reader)
80
81
                 base.LoadFrom(reader);
82
                 Width = reader.ReadInteger();
                 Height = reader.ReadInteger();
84
            }
85
        }
86
   }
87
```

File 6 of 8 MyCircle class

```
using System;
   using SplashKitSDK;
   namespace Task5._3
        public class MyCircle : Shape
5
6
            private int _radius;
            public MyCircle():this(Color.Blue, 0, 0, 50)
            {
10
11
            }
12
            public MyCircle(Color clor,int x, int Y, int radius):base(clor)
13
                 _color = clor;
15
                 _radius = radius;
            }
17
18
            public int Radius
19
            {
20
                 get
                 {
22
                     return _radius;
23
24
                 set
25
26
                      _radius = value;
27
                 }
28
29
            }
30
            public override void Draw()
31
32
                 if (Selected)
                     Outline();
34
                 SplashKit.FillCircle(Color, X, Y, _radius);
35
36
            public override void Outline()
37
            {
38
39
                 SplashKit.FillCircle(Color.Black, X , Y , Radius+2);
40
41
42
43
            public override bool IsAt(Point2D pt)
46
                 double point1 = (pt.X - X) * (pt.X - X);
47
48
                 double point2 = (pt.Y - Y) * (pt.Y - Y);
49
50
                 if(Math.Sqrt(point1+point2)<_radius)</pre>
51
52
                     return true;
53
```

File 6 of 8 MyCircle class

```
}
54
                 else
55
                 {
56
                     return false;
                 }
58
59
            }
60
            public override void SaveTo(StreamWriter writer)
61
62
                 writer.WriteLine("Circle");
63
                 base.SaveTo(writer);
65
                 writer.WriteLine(Radius);
66
67
            public override void LoadFrom(StreamReader reader)
68
                 base.LoadFrom(reader);
70
                 Radius = reader.ReadInteger();
71
            }
72
        }
73
   }
75
```

File 7 of 8 MyLine class

```
using System;
   using System.Numerics;
   using SplashKitSDK;
   namespace Task5._3
    {
5
        public class MyLine : Shape
6
            private float _endX;
            private float _endY;
10
11
            public MyLine() : this(Color.Orange, 0, 0, 200, 300)
12
            {
13
14
15
            public MyLine(Color clr, float startX, float startY, float endX, float endY)
16
        : base(clr)
             {
17
18
                 EndX = endX;
19
                 EndY = endY;
21
            }
22
23
            public float EndX
24
25
                 get
26
                 {
                      return _endX;
28
                 }
29
                 set
30
                 {
31
                      _endX = value;
                 }
33
            }
34
35
            public float EndY
36
             {
38
                 get
                 {
39
                      return _endY;
40
                 }
41
                 set
42
                 {
43
                      _endY = value;
                 }
45
            }
46
47
            public override void Draw()
48
            {
49
                 if (Selected)
50
                      Outline();
51
                 SplashKit.DrawLine(Color, X, Y, EndX, EndY,
52
        SplashKit.OptionLineWidth(5));
```

File 7 of 8 MyLine class

```
53
            }
54
            public override void Outline()
55
57
58
                 SplashKit.FillCircle(Color.Black, X, Y, 4);
59
                 SplashKit.FillCircle(Color.Black, EndX, EndY, 4);
60
            }
            public override bool IsAt(Point2D pt)
63
64
65
66
                 Line 1 = SplashKit.LineFrom(X, Y, EndX, EndY);
67
69
                 return SplashKit.PointOnLine(pt, 1, 10);
70
71
            public override void SaveTo(StreamWriter writer)
72
                 writer.WriteLine("Line");
                 base.SaveTo(writer);
76
                 writer.WriteLine(EndX);
                 writer.WriteLine(EndY);
79
            }
            public override void LoadFrom(StreamReader reader)
81
            {
82
                 base.LoadFrom(reader);
83
                 EndX = reader.ReadInteger();
84
                 EndY = reader.ReadInteger();
            }
86
        }
87
   }
88
89
91
92
93
94
95
96
97
98
```

