```
 1  using Microsoft.Xna.Framework;
 2  using Microsoft.Xna.Framework.Content;
 3  using Microsoft.Xna.Framework.Graphics;
 4  using Microsoft.Xna.Framework.Input.Touch;
 5  using System;
 6  using System.Collections.Generic;
 7  using System.Text;
 8
 9  namespace GameDemo.Shared.Menu
10  {
11      class MenuOption
12      {
13          int scale;
14
15          /// <summary>
16          /// The GUIElement's texture
17          /// </summary>
18          private Texture2D guiTexture;
19
20          /// <summary>
21          /// The GUIElement's rectangle, this is used to determine if the
              lement is clicked
22          /// </summary>
23          private Rectangle guiRectangle;
24
25          /// <summary>
26          /// The name of the GUIElement, we need this to select a specific
              element
27          /// </summary>
28          private string elementName;
29
30          /// <summary>
31          /// Proptery to get the element's name
32          /// </summary>
33          public string ElementName
34          {
35              get { return elementName; }
36          }
37
38          /// <summary>
39          /// Delegate used to the Element clicked event
40          /// </summary>
41          /// <param name="element">Name of the clicked element</param>
42          public delegate void ElementClicked(string element);
43
44          /// <summary>
45          /// Event triggered every time an element is clicked
46          /// </summary>
47          public event ElementClicked clickEvent;
48
49          /// <summary>
50          /// The GUIElements constructor
51          /// </summary>
52          /// <param name="name">name of the element(also name of the texture)</
              param>
53          public MenuOption(string name,int scale)
```

```
54          {
55
56              this.scale = scale;
57              elementName = name;
58
59          }
60
61          /// <summary>
62          /// Loads the element's texture
63          /// </summary>
64          /// <param name="content"></param>
65          public virtual void LoadContent(ContentManager content)
66          {
67              guiTexture = content.Load<Texture2D>(elementName);
68          }
69
70          /// <summary>
71          /// The update checks if the GUIElement is clicked
72          /// </summary>
73          public virtual void Update(GestureSample gesture)
74          {
75
76
77                  if (guiRectangle.Contains(new Point((int)gesture.Position.X,
                      (int)gesture.Position.Y)) && gesture.GestureType ==
                      GestureType.Tap)
78                  {
79
80                      //This element was clicked
81                      clickEvent(elementName);
82
83                  }
84          }
85
86
87          /// <summary>
88          /// Draws the GUIElement
89          /// </summary>
90          /// <param name="spriteBatch">SpriteBatch</param>
91          public virtual void Draw(SpriteBatch spriteBatch)
92          {
93              spriteBatch.Draw(guiTexture, guiRectangle, Color.White);
94          }
95
96          /// <summary>
97          /// Centers the GUIElement in the GameWindow
98          /// </summary>
99          /// <param name="windowSize"></param>
100         public void CenterElement(Size windowSize)
101         {
102             guiRectangle = new Rectangle
103                 (
104                     (windowSize.height / 2) - (this.guiTexture.Width / 2),
105                     (windowSize.width / 2) - (this.guiTexture.Height / 2),
106                     guiTexture.Width*scale,
107                     guiTexture.Height*scale
```

```
108                 );
109         }
110
111         /// <summary>
112         /// Move the GUIElement
113         /// </summary>
114         /// <param name="x">X-Offset</param>
115         /// <param name="y">Y-Offset</param>
116         public void MoveElement(int x, int y)
117         {
118             guiRectangle = new Rectangle
119                 (
120                     guiRectangle.X + x,
121                     guiRectangle.Y + y,
122                     guiRectangle.Width,
123                     guiRectangle.Height
124                 );
125         }
126     }
127 }
128
129
```