

```
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     struct Size
12     {
13         public int height { get; private set; }
14
15         public int width { get; private set; }
16
17         public Size(int width, int height) : this()
18         {
19             this.height = height;
20             this.width = width;
21         }
22     }
23
24 }
25 /// <summary>
26 /// Enum used to determine the current GameState
27 /// </summary>
28 public enum GameState { mainMenu, howTo, inGame, enterName,
29                         viewLeaderboards,over }
30
31 class MainMenu
32 {
33     private List<List<MenuOption>> menus;
34
35     public MainMenu()
36     {
37         //Adds lists of GUI elements to the menu list
38         menus = new List<List<MenuOption>>
39         { //MainMenu
40             (new List<MenuOption>
41             {
42                 new MenuOption("playButton",2),
43                 new MenuOption("scoresButton",2),
44                 new MenuOption("quitButton",2)
45             }),
46             //Enter name menu
47             (new List<MenuOption>
48             {
49                 new MenuOption("doneButton",2),
50             })
51         };
52         //Sets the OnClick event on all menu items
53         for (int i = 0; i < menus.Count; i++)
54         {
55
```

```
56         foreach (MenuOption button in menus[i])
57         {
58             button.ClickEvent += OnClick;
59         }
60     }
61
62 }
63
64 public void LoadContent(ContentManager content, Size windowSize)
65 {
66
67     //Loads the content of all other GUI elements
68     for (int i = 0; i < menus.Count; i++)
69     {
70         foreach (MenuOption button in menus[i])
71         {
72             button.LoadContent(content);
73             button.CenterElement(windowSize);
74         }
75     }
76     //Sets offsets of the buttons
77     menus[0].Find(x => x.ElementName == "playButton").MoveElement  ↗
78     (-170, -350);
79     menus[0].Find(x => x.ElementName == "scoresButton").MoveElement  ↗
80     (-170, -100);
81     menus[0].Find(x => x.ElementName == "quitButton").MoveElement  ↗
82     (-170, 150);
83     menus[1].Find(x => x.ElementName == "doneButton").MoveElement  ↗
84     (-150, 300);
85 }
86
87 /// <summary>
88 /// Updates all our GUI element(Mainly checks if any button is  ↗
89 pressed)
90 /// </summary>
91 public void Update()
92 {
93     //Update each element according to the current gameState
94     switch (SpriteManager.gameState)
95     {
96     }
97
98     //If you are the mainMenu screen handle the current button  ↗
99     input
100     case GameState.mainMenu:
101         var gesture = default(GestureSample);
102         while (TouchPanel.IsGestureAvailable)
103         {
104             gesture = TouchPanel.ReadGesture();
105             foreach (MenuOption button in menus[0]) //MainMenu
106             {
107                 button.Update(gesture);
108             }
109         }
110         break;
```

```
106         //if you are at the leaderboards screen handle the current button input
107         case GameState.viewLeaderboards:
108             gesture = default(GestureSample);
109             while (TouchPanel.IsGestureAvailable)
110             {
111                 gesture = TouchPanel.ReadGesture();
112                 foreach (MenuOption button in menus[1]) //MainMenu
113                 {
114
115                     button.Update(gesture);
116                 }
117             }
118             break;
119         case GameState.over:
120             break;
121     }
122 }
123
124 public void Draw(SpriteBatch spriteBatch)
125 {
126     //Draws each element according to the current gameState
127     switch (SpriteManager.gameState)
128     {
129     case GameState.mainMenu:
130         foreach (MenuOption button in menus[0]) //MainMenu
131         {
132             button.Draw(spriteBatch);
133         }
134         break;
135     case GameState.viewLeaderboards:
136         foreach (MenuOption button in menus[1]) //leaderboards menu
137         {
138             button.Draw(spriteBatch);
139         }
140         break;
141     case GameState.inGame: //Ingame GUI
142         break;
143     }
144 }
145
146 }
147
148 /// <summary>
149 /// Method called every time a GUI element is clicked
150 /// </summary>
151 /// <param name="element"></param>
152 public void OnClick(string element)
153 {
154     if (element == "playButton") //PlayButton
155     {
156         SpriteManager.gameState = GameState.inGame;
157     }
158     if (element == "scoresButton") //scores button
159     {
160         SpriteManager.gameState = GameState.viewLeaderboards;
```

```
161         }
162         if (element == "quitButton")//Quit button
163         {
164             SpriteManager.gameState = GameState.over;
165         }
166         if( element== "doneButton")
167         {
168             SpriteManager.gameState = GameState.mainMenu;
169         }
170     }
171 }
172 }
173 }
174 }
```