

```
1  /*-----*\
2  * Author    : Salvi Cyril
3  * Date      : 8th june 2017
4  * Diploma  : RaspiHome
5  * Classroom : T.IS-E2B
6  *
7  * Description:
8  *     RaspiHomeTabletWindows is a program
9  *     compatible with the Windows tablet. It's a
10 *     program that can be use as tactil graphic
11 *     interface to order the component linked with
12 *     the other Raspberry Pi.
13 \*-----*/
14
15 using System;
16 using Windows.UI;
17 using Windows.UI.Xaml;
18 using Windows.UI.Xaml.Controls;
19 using Windows.UI.Xaml.Media;
20 using Windows.UI.Xaml.Media.Imaging;
21
22 // The User Control item template is documented at http://go.microsoft.com/ ↗
23     fwlink/?LinkId=234236
24 namespace RaspiHomeTabletWindows.Menu.MenuToolbar
25 {
26     public sealed partial class ToolbarButtonView : UserControl
27     {
28         #region Fields
29         #region Constants
30         #endregion
31
32         #region Variables
33         private ToolbarButtonModel _model = null;
34
35         public event EventHandler _click;
36
37         private bool _isSelected = false;
38         private bool _isPressed = false;
39
40         private string _whoseButtonClicked = "";
41         #endregion
42         #endregion
43
44         #region Properties
45         public ToolbarButtonModel Model
46         {
47             get
48             {
49                 return _model;
50             }
51
52             set
53             {
54                 _model = value;
55             }
56         }
57     }
58 }
```

```
56     }
57
58     public bool IsSelected
59     {
60         get
61         {
62             return _isSelected;
63         }
64         set
65         {
66             _isSelected = value;
67
68             btnToolbar.IsTabStop = _isSelected;
69
70             if (value)
71             {
72                 btnToolbar.BorderThickness = new Thickness(6.5);
73                 btnToolbar.BorderBrush = new SolidColorBrush      ↗
74                     (Color.FromArgb(255, 73, 130, 5));
75             }
76             else
77             {
78                 btnToolbar.BorderThickness = new Thickness(5);
79                 btnToolbar.BorderBrush = new SolidColorBrush      ↗
80                     (Color.FromArgb(255, 76, 74, 75));
81             }
82         }
83     }
84     public bool IsPressed
85     {
86         get
87         {
88             return _isPressed;
89         }
90         set
91         {
92             _isPressed = value;
93         }
94     }
95
96     public string WhoseButtonClicked
97     {
98         get
99         {
100             return _whoseButtonClicked;
101         }
102         set
103         {
104             _whoseButtonClicked = value;
105         }
106     }
107 }
108
109 #endregion
```

```
110
111     #region Constructor
112     /// <summary>
113     /// Constructor: Initializer
114     /// </summary>
115     public ToolbarButtonView(string frameChoose, string description, ↗
116         string iconLink)
117     {
118         InitializeComponent();
119
120         this.Model = new ToolbarButtonModel(this);
121
122         this.WhoseButtonClicked = frameChoose;
123
124         SetInformation(description, iconLink);
125     }
126     #endregion
127
128     #region Events
129     /// <summary>
130     /// Event on the button
131     /// </summary>
132     /// <param name="sender"></param>
133     /// <param name="e"></param>
134     private void btnToolbar_Click(object sender, RoutedEventArgs e)
135     {
136         if (this._click != null)
137             this._click(this, null);
138         this.IsPressed = true;
139     }
140     #endregion
141
142     #region Methods
143     /// <summary>
144     /// Set information of the button on the toolbar
145     /// </summary>
146     /// <param name="description"></param>
147     /// <param name="iconLink"></param>
148     private void SetInformation(string description, string iconLink)
149     {
150         this.Model.SetInformation(description, iconLink);
151     }
152
153     /// <summary>
154     /// Change the Icon of the button on the toolbar
155     /// </summary>
156     /// <param name="iconPath"></param>
157     public void ChangeIcon(string iconPath)
158     {
159         if (iconPath != "")
160             this.imgMenuToolbarButton.Source = new BitmapImage(new Uri ↗
161                 (iconPath));
162     }
163 }
164 #endregion
165 }
```