```csharp
 1  /*-------------------------------------------------*\
 2   * Author     : Salvi Cyril
 3   * Date       : 8th juny 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 System.Linq;
17  using System.Threading.Tasks;
18  using Windows.UI.Xaml;
19
20  namespace RaspiHomeTabletWindows.Modules.Home.Location.OtherRoom
21  {
22      public class RoomModel
23      {
24          #region Fields
25          #region Constants
26          #endregion
27
28          #region Varaibles
29          private RoomView _view = null;
30
31          private string _messageReaded = "";
32
33          DispatcherTimer _dTimer = null;
34
35          Windows.Storage.ApplicationDataContainer localSettings =
                Windows.Storage.ApplicationData.Current.LocalSettings;
36          #endregion
37          #endregion
38
39          #region Properties
40          public RoomView View
41          {
42              get
43              {
44                  return _view;
45              }
46
47              set
48              {
49                  _view = value;
50              }
51          }
52
53          public string MessageReceive
54          {
55              get
```

```csharp
56                {
57                    return _messageReaded;
58                }
59
60                set
61                {
62                    _messageReaded = value;
63                }
64            }
65        #endregion
66
67        #region Constructor
68        /// <summary>
69        /// Constructor: Initializer
70        /// </summary>
71        public RoomModel(RoomView paramView)
72        {
73            this.View = paramView;
74
75            this._dTimer = new DispatcherTimer();
76            this._dTimer.Interval = new TimeSpan(200);
77            this._dTimer.Tick += _dTimer_Tick; ;
78
79            this._dTimer.Start();
80
81            this.InitializeState();
82        }
83
84        private void _dTimer_Tick(object sender, object e)
85        {
86            if (localSettings.Values["ReceiveMessageFromServer"] != null)
87            {
88                var messageToSend = localSettings.Values
                     ["ReceiveMessageFromServer"];
89                this.MessageReceive = messageToSend.ToString();
90                UpDateView();
91                localSettings.Values.Remove("ReceiveMessageFromServer");
92
93                this._dTimer.Stop();
94            }
95        }
96        #endregion
97
98        #region Events
99        /// <summary>
100       /// Initialize at the start (check if the sense hat exist)
101       /// </summary>
102       private void InitializeState()
103       {
104           // Update state room
105           var locationName = localSettings.Values["NameButtonClicked"];
106
107           if (locationName != null)
108               localSettings.Values["SendMessageToServer"] = "etat " +
                     locationName;
109           else
```

```csharp
110                    localSettings.Values["SendMessageToServer"] = "etat salon";
111            }
112        #endregion
113
114        #region Methods
115        /// <summary>
116        /// Save value to be send
117        /// </summary>
118        /// <param name="action"></param>
119        /// <param name="component"></param>
120        public void SendMessage(string action, string component)
121        {
122            var locationName = localSettings.Values["NameButtonClicked"];
123
124            if (locationName != null)
125                localSettings.Values["SendMessageToServer"] = action + " " +   ⏎
                    component + " " + locationName;
126        }
127
128        /// <summary>
129        /// Update state values on the view
130        /// </summary>
131        private async void UpDateView()
132        {
133            await Task.Delay(TimeSpan.FromMilliseconds(200));
134
135            this.View.EnableDisplayState();
136
137            var informations = this.MessageReceive.Split(';');
138            foreach (var info in informations)
139            {
140                switch (info.Split('=').First())
141                {
142                    case "TEMP":
143                        this.View.StrTemp = info.Split('=').Last();
144                        break;
145                    case "HUMI":
146                        this.View.StrHumi = info.Split('=').Last();
147                        break;
148                    case "PRES":
149                        this.View.StrPres = info.Split('=').Last();
150                        break;
151                }
152            }
153        }
154        #endregion
155    }
156 }
157
```