```
/*-------------------------------------------------*\
 * Author    : Salvi Cyril
 * Date      : 7th juny 2017
 * Diploma   : RaspiHome
 * Classroom : T.IS-E2B
 *
 * Description:
 *      RaspiHomePiFaceDigital2 is a program who use
 *   a PiFace Digital 2, it's an electronic card who
 *   can be use to plug electronic component. This
 *   program use the PiFace Digital 2 to activate
 *   light and store.
\*-------------------------------------------------*/

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Globalization;
using System.Linq;
using System.Reflection;
using System.Text;

namespace RaspiHomePiFaceDigital2
{
    public class ModelPiFaceDigital2
    {
        #region Fields
        #region Constants
        #endregion

        #region Variables
        private ViewPiFaceDigital2 _vPiFace;

        private List<Component> _components;
        private CommunicationWithServer _comWithServer;

        // Command to know
        private List<string> _raspiHomeComponentKnown = new List<string>()
        {
            "lumiere","lumieres",
            "store","stores",
            "television","televisions",
            "porte","portes",
            "fenetre","fenetres",
        };

        private List<string> _raspiHomeActionKnown = new List<string>()
        {
            "allumer","allume",
            "eteindre","eteins",
            "monter","monte",
            "descendre","descends",
            "stopper","stop",
            "ouvrir","ouvre",
            "fermer","ferme",
            "stopper","stop",
```

```
57          };
58
59          // Word translation
60          private Dictionary<string, string> _raspiLanguageTranslation = new
              Dictionary<string, string>()
61          {
62              { "lumiere","Light"}, { "lumieres","Light"},
63              { "store","Store"}, { "stores","Store"},
64          };
65
66          // KEY=[ACTION NAME], VALUE[KEY=[PROPERTY NAME], VALUE=[VALUE TO SET
              THEPROPERTY]]
67          private Dictionary<string, Dictionary<string, bool>>
              _raspiBooleanCommandTranslation = new Dictionary<string,
              Dictionary<string, bool>>()
68          {
69              { "allume", new Dictionary<string, bool> { { "IsOn", true } } },
                { "allumer", new Dictionary<string, bool> { { "IsOn",
                true } } },
70              { "eteins", new Dictionary<string, bool> { { "IsOn", false } } },
                { "eteindre", new Dictionary<string, bool> { { "IsOn",
                false } } },
71              { "monte", new Dictionary<string, bool> { { "IsUp", true } } },
                { "monter", new Dictionary<string, bool> { { "IsUp", true } } },
72              { "descends", new Dictionary<string, bool> { { "IsDown",
                true } } }, { "descendre", new Dictionary<string, bool>
                { { "IsDown", true } } },
73              { "stop",new Dictionary<string, bool> { {"IsStop",true } } },
                {"stopper",new Dictionary<string, bool> { {"IsStop",true } } },
74          };
75          #endregion
76          #endregion
77
78          #region Properties
79          public ViewPiFaceDigital2 VPiFace
80          {
81              get
82              {
83                  return _vPiFace;
84              }
85
86              set
87              {
88                  _vPiFace = value;
89              }
90          }
91
92          public List<Component> Components
93          {
94              get
95              {
96                  return _components;
97              }
98
99              set
100             {
```

```
101                 _components = value;
102             }
103         }
104
105         public CommunicationWithServer ComWithServer
106         {
107             get
108             {
109                 return _comWithServer;
110             }
111
112             set
113             {
114                 _comWithServer = value;
115             }
116         }
117         #endregion
118
119         #region Constructors
120         /// <summary>
121         /// Constructor: Initializer
122         /// </summary>
123         /// <param name="paramView"></param>
124         public ModelPiFaceDigital2(ViewPiFaceDigital2 paramView)
125         {
126             // Communication like Model-View
127             this.VPiFace = paramView;
128
129             // Initialize the components and add the components linked with
                  the Raspberry
130             this.Components = new List<Component>();
131             this.Components.Add(new Light());
132             this.Components.Add(new Store());
133
134             // Initilize the PiFace Digital 2
135             InitializePiFace();
136
137             // Initialize the server communication
138             this.ComWithServer = new CommunicationWithServer(this);
139         }
140         #endregion
141
142         #region Methods
143         /// <summary>
144         /// Initialize the PiFace Digital 2
145         /// </summary>
146         private async void InitializePiFace()
147         {
148             try
149             {
150                 await MCP23S17.InitilizeSPI();
151
152                 MCP23S17.InitializeMCP23S17();
153                 MCP23S17.SetPinMode(0x00FF); // 0x0000 = all outputs,
                      0xffff=all inputs, 0x00FF is PIFace Default
154                 MCP23S17.PullupMode(0x00FF); // 0x0000 = no pullups,
```

```
                    0xffff=all pullups, 0x00FF is PIFace Default
155                 MCP23S17.WriteWord(0x0000); // 0x0000 = no pullups, 0xffff=all ↵
                       pullups, 0x00FF is PIFace Default
156             }
157             catch (Exception ex)
158             {
159                 Debug.WriteLine(ex.Message);
160             }
161         }
162
163         /// <summary>
164         /// Set the value to be writed on the PiFace
165         /// </summary>
166         /// <param name="messageRead"> message read from the server </param>
167         public void SetValue(string messageRead)
168         {
169             // Initialize the message value
170             string sentence = this.RemoveDiacritics(messageRead);
171             string action = this.GetActionFromSentence(sentence);
172             string actionValue = this.ReadValueOfSelectedComponent(action);
173             string component = this.GetComponentFromSentence(sentence);
174             Type componentType = this.GetComponentType(component);
175
176             foreach (Component itemType in this.Components)
177             {
178                 if (itemType.GetType() == componentType)
179                 {
180                     this.WriteValue(itemType, action, itemType.GetType        ↵
                        ().GetProperty(actionValue));
181                 }
182             }
183         }
184
185         /// <summary>
186         /// Find location exist
187         /// </summary>
188         /// <param name="sentence"> sentence order</param>
189         /// <returns> return the action linked to the action word </returns>
190         private string GetActionFromSentence(string sentence)
191         {
192             string result = "";
193             string[] words = sentence.ToLower().Split(' ');
194
195             foreach (var word in words)
196             {
197                 if (this._raspiHomeActionKnown.Contains(word))
198                 {
199                     result = word;
200                     break;
201                 }
202             }
203
204             return result;
205         }
206
207         /// <summary>
```

```
208         /// Get the componnent called
209         /// </summary>
210         /// <param name="sentence"> sentence order </param>
211         /// <returns> return the component linked to the component word </
              returns>
212         private string GetComponentFromSentence(string sentence)
213         {
214             string result = "";
215             string[] words = sentence.ToLower().Split(' ');
216 
217             foreach (var word in words)
218             {
219                 if (this._raspiHomeComponentKnown.Contains(word))
220                 {
221                     result = word;
222                     break;
223                 }
224             }
225 
226             return result;
227         }
228 
229         /// <summary>
230         /// Find all client who have the object in the sentence
231         /// </summary>
232         /// <param name="componentName"></param>
233         /// <returns>the object type</returns>
234         private Type GetComponentType(string componentName)
235         {
236             Type result = null;
237             Type[] types = typeof(Component).GetTypeInfo().Assembly.GetTypes
                  ();
238 
239             foreach (var typeOfComonent in types)
240             {
241                 if (typeOfComonent.Name == this._raspiLanguageTranslation
                      [componentName])
242                 {
243                     result = typeOfComonent;
244                     break;
245                 }
246             }
247 
248             return result;
249         }
250 
251         /// <summary>
252         /// Read properties value of classes
253         /// </summary>
254         /// <param name="actionName"> name used to change the good property </
              param>
255         /// <returns> return the name of the property to change the value </
              returns>
256         private string ReadValueOfSelectedComponent(string actionName)
257         {
258             string result = "";
```

```
259
260             foreach (var actionKeys in this._raspiBooleanCommandTranslation.Keys)
261                 if (actionKeys == actionName)
262                 {
263                     // Find the Value of the dictionary trough the inner dictionary to get the first value
264                     result = this._raspiBooleanCommandTranslation[actionName].First().Key;
265                     break;
266                 }
267
268             return result;
269         }
270
271         /// <summary>
272         /// Search the val to change
273         /// </summary>
274         /// <param name="component"> the component to write value </param>
275         /// <param name="action"> the action (ON/OFF) </param>
276         /// <param name="typeVariable"> the property to change value </param>
277         private void WriteValue(Component component, string action, PropertyInfo typeVariable)
278         {
279             switch (typeVariable.PropertyType.Name)
280             {
281                 case "Boolean":
282                     // Set the new value dynamicaly with value registered in an boolean dictionary
283                     typeVariable.SetValue(component, this._raspiBooleanCommandTranslation[action][typeVariable.Name]);
284                     break;
285                 case "Double":
286                     break;
287                 case "Int16":
288                 case "Int32":
289                 case "Int64":
290                     break;
291             }
292         }
293
294         /// <summary>
295         /// Stack Overflow solution to delete accents in strings
296         /// http://stackoverflow.com/questions/249087/how-do-i-remove-diacritics-accents-from-a-string-in-net
297         /// </summary>
298         /// <param name="sentence"> sentence with diacritics to remove </param>
299         /// <returns> same sentence without diacritics </returns>
300         private string RemoveDiacritics(string sentence)
301         {
302             var normalizedString = sentence.Normalize(NormalizationForm.FormD);
303             var stringBuilder = new StringBuilder();
304
```

```
305             foreach (var c in normalizedString)
306             {
307                 var unicodeCategory = CharUnicodeInfo.GetUnicodeCategory(c);
308                 if (unicodeCategory != UnicodeCategory.NonSpacingMark)
309                 {
310                     stringBuilder.Append(c);
311                 }
312             }
313 
314             return stringBuilder.ToString().Normalize
                  (NormalizationForm.FormC);
315         }
316         #endregion
317     }
318 }
319 
```