```
1  /*
2   * Author  : Dubas Loïc
3   * Class   : I.FA-P3B
4   * School  : CFPT-I
5   * Date    : June 2018
6   * Descr.  : Store hand data
7   * Version : 1.0
8   * Ext. dll: LeapCSharp.NET4.5
9   */
10
11 using System;
12 using System.Collections.Generic;
13 using System.Linq;
14 using System.Text;
15 using System.Threading.Tasks;
16 // References to add
17 using Leap;
18
19 namespace fingers_cloner
20 {
21     public class MyHand
22     {
23         #region Initialization
24         // get
25         private string _name;
26         private string _description;
27         private Vector _palmNormPos;
28         private List<Vector> _fingersNormPos;
29         private string _image;
30         // set
31         // name
32         public string Name { get => _name; set => _name = value; }
33         // description
34         public string Description { get => _description; set => _description =
             value; }
35         // normalized position of the palm
36         public Vector PalmNormPos { get => _palmNormPos; set => _palmNormPos =
             value; }
37         // normalized positions of the fingers
38         public List<Vector> FingersNormPos { get => _fingersNormPos; set =>
             _fingersNormPos = value; }
39         // image of the position as a string
40         public string Image { get => _image; set => _image = value; }
41         #endregion
42
43         /// <summary>
44         /// default constructor
45         /// </summary>
46         public MyHand() { }
47
48         /// <summary>
49         /// MyHand constructor
50         /// </summary>
51         /// <param name="palmPosNorm">Normalized position of the palm</param>
52         /// <param name="fingersPosNorm">Normalized positions of the fingers</
             param>
```

```
53          public MyHand(Vector palmPosNorm, List<Vector> fingersPosNorm)
54          {
55              this.PalmNormPos = palmPosNorm;
56              this.FingersNormPos = fingersPosNorm;
57          }
58
59          /// <summary>
60          /// MyHand constructor
61          /// </summary>
62          /// <param name="name">Name of the position</param>
63          /// <param name="description">Description of the position</param>
64          /// <param name="palmPosNorm">Normalized position of the palm</param>
65          /// <param name="fingersPosNorm">Normalized positions of the fingers</
              param>
66          public MyHand(string name, string description, Vector palmPosNorm,
              List<Vector> fingersPosNorm)
67          {
68              this.Name = name;
69              this.Description = description;
70              this.PalmNormPos = palmPosNorm;
71              this.FingersNormPos = fingersPosNorm;
72          }
73
74          /// <summary>
75          /// MyHand constructor
76          /// </summary>
77          /// <param name="name">Name of the position</param>
78          /// <param name="description">Description of the position</param>
79          /// <param name="palmPosNorm">Normalized position of the palm</param>
80          /// <param name="fingersPosNorm">Normalized positions of the fingers</
              param>
81          /// <param name="image">Image of the position as a string</param>
82          public MyHand(string name, string description, Vector palmPosNorm,
              List<Vector> fingersPosNorm, string image)
83          {
84              this.Name = name;
85              this.Description = description;
86              this.PalmNormPos = palmPosNorm;
87              this.FingersNormPos = fingersPosNorm;
88              this.Image = image;
89          }
90      }
91  }
92
```