```
1  /*
2   * Author  : Dubas Loïc
3   * Class   : I.FA-P3B
4   * School  : CFPT-I
5   * Date    : June 2018
6   * Descr.  : Detect hand and calculate finger's position
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 using System.Windows.Forms;
17 // References to add
18 using Leap;
19 using System.Xml;
20
21 namespace fingers_cloner
22 {
23     class LeapController : Controller
24     {
25         #region Initialization
26         // set
27         // List of detected hands and the first detected hand
28         private List<Hand> _hands;
29         private Hand _firstHand;
30
31         // Palm raw, normalized and stabilized location
32         private Vector _palmPos;
33         private Vector _palmNormPos;
34
35         // List of all the detected fingers
36         private List<Finger> _fingers;
37
38         // Fingers raw and normalized location
39         private List<Vector> _fingersPos;
40         private List<Vector> _fingersNormPos;
41
42         // User's hand
43         private MyHand _userHand;
44
45         // get
46         public List<Hand> Hands { get => _hands; set => _hands = value; }
47         public Hand FirstHand { get => _firstHand; set => _firstHand = value; }
48         public List<Finger> Fingers { get => _fingers; set => _fingers =
             value; }
49         public List<Vector> FingersStabPos { get => _fingersPos; set =>
             _fingersPos = value; }
50         public List<Vector> FingersNormPos { get => _fingersNormPos; set =>
             _fingersNormPos = value; }
51         public Vector PalmPos { get => _palmPos; set => _palmPos = value; }
52         public Vector PalmNormPos { get => _palmNormPos; set => _palmNormPos =
             value; }
```

```
53          public MyHand UserHand { get => _userHand; set => _userHand = value; }
54          #endregion
55
56          /// <summary>
57          /// Leap Motion's default constructor
58          /// </summary>
59          public LeapController()
60          {
61              EventContext = WindowsFormsSynchronizationContext.Current;
62              FrameReady += newFrameHandler;
63          }
64
65          /// <summary>
66          /// Refresh the fingers info on every frame of the Leap Motion
67          /// </summary>
68          /// <param name="sender"></param>
69          /// <param name="eventArgs"></param>
70          public void newFrameHandler(object sender, FrameEventArgs eventArgs)
71          {
72              Frame frame = eventArgs.frame;
73              InteractionBox iBox = frame.InteractionBox;
74
75              if (frame.Hands.Count > 0)
76              {
77                  Hands = frame.Hands;
78                  FirstHand = Hands[0];
79
80                  PalmPos = FirstHand.PalmPosition;
81                  PalmNormPos = iBox.NormalizePoint(PalmPos);
82
83                  Fingers = FirstHand.Fingers;
84                  FingersStabPos = new List<Vector>();
85                  FingersNormPos = new List<Vector>();
86
87                  for (int i = 0; i < Fingers.Count; i++)
88                  {
89                      FingersStabPos.Add(Fingers[i].StabilizedTipPosition);
90                      FingersNormPos.Add(iBox.NormalizePoint(FingersStabPos[i]));
91                  }
92
93                  UserHand = new MyHand(PalmNormPos, FingersNormPos);
94              }
95          }
96      }
97  }
98
```