```
/*
 * Author  : Dubas Loïc
 * Class   : I.FA-P3B
 * School  : CFPT-I
 * Date    : June 2018
 * Descr.  : serialize and deserialize functions and delete save position
 * Version : 1.0
 * Ext. dll: LeapCSharp.NET4.5
 */

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// References to add
using Leap;
using System.Diagnostics;
using System.Xml;
using System.Xml.Serialization;
using System.IO;

namespace fingers_cloner
{
    [Serializable]
    public class Serialization
    {
        #region Intialization
        // serialize file name, directory name and file path
        private string _positionName;
        private string _dirName;
        private string _filePath;

        // store all positions serialized
        List<MyHand> allPositions;

        // store all the files name
        List<string> allFilesName;

        // hand to serialize
        private MyHand _handToSerialize;
        internal MyHand HandToSerialize { get => _handToSerialize; set =>
          _handToSerialize = value; }
        public string PositionName { get => _positionName; set =>
          _positionName = value; }
        // directory name and file path
        public string DirName { get => _dirName; set => _dirName = value; }
        public string FilePath { get => _filePath; set => _filePath = value; }
        #endregion

        /// <summary>
        /// default constructor - initialize directory name
        /// </summary>
        public Serialization()
        {
            DirName = "serial";
```

```
55
56              Path.GetFileName(DirName);
57          }
58
59          /// <summary>
60          /// serialize a given MyHand object
61          /// </summary>
62          /// <param name="Hand">the hand to serialize</param>
63          public void serialize(MyHand Hand)
64          {
65              PositionName = Hand.Name;
66              FilePath = DirName + "/" + PositionName + ".xml";
67
68              XmlSerializer serializer = new XmlSerializer(typeof(MyHand));
69              StreamWriter file = new StreamWriter(FilePath);
70              serializer.Serialize(file, Hand);
71              file.Close();
72          }
73
74          /// <summary>
75          /// deserialize all xml files in serial directory
76          /// </summary>
77          /// <returns>a list of all the serialize hands</returns>
78          public List<MyHand> deserialize()
79          {
80              allPositions = new List<MyHand>();
81              allFilesName = getFilesName();
82
83              if (Directory.Exists(DirName))
84              {
85                  XmlSerializer serializer = new XmlSerializer(typeof(MyHand));
86                  foreach (string position in allFilesName)
87                  {
88                      FileStream stream = new FileStream(position,
                        FileMode.Open);
89                      allPositions.Add((MyHand)serializer.Deserialize(stream));
90                      stream.Close();
91                  }
92              }
93
94              return allPositions;
95          }
96
97          /// <summary>
98          /// get all the files name in the serial directory
99          /// </summary>
100         /// <returns>a list of all the names of the positions</returns>
101         public List<string> getFilesName()
102         {
103             allFilesName = new List<string>();
104
105             foreach (string fileName in Directory.GetFiles(DirName))
106             {
107                 allFilesName.Add(fileName);
108             }
109
```

```
110             return allFilesName;
111         }
112
113         /// <summary>
114         /// delete a saved position
115         /// </summary>
116         /// <param name="posName">the name of the position to delete</param>
117         public void deletePosition(string posName) {
118             FilePath = DirName + "/" + posName + ".xml";
119
120             File.Delete(FilePath);
121         }
122     }
123 }
124
```