



Interactive Music Science Collaborative Activities
 Team Teaching for STEAM Education

Deliverable 4.1
First Version of Gesture and VR multimodal interaction interface

Date:	02/10/2017
Author(s):	Aggelos Gkiokas (ATHENA), Kosmas Kritsis (ATHENA), Maximos Kaliakatsos-Papakostas (ATHENA), Antigoni Tsiami (ATHENA), Nancy Zlatintsi (ATHENA)
Contributor(s):	Robert Piechaud (IRCAM), Carlos Acosta (LEOPOLY)
Quality Assuror(s):	Petros Stergiopoulos (EA), Rober Piechaud (IRCAM)
Dissemination level:	PU
Work package	WP4 – Core enabling technologies modules of iMuSciCA
Version:	1.0
Keywords:	gesture interaction, multimodal interaction, leapmotion, kinect
Description:	First version of the demonstrator for gesture and VR multimodal interaction interface.



H2020-ICT-22-2016 Technologies for Learning and Skills
iMuSciCA (Interactive Music Science Collaborative Activities)
 Project No. 731861
 Project Runtime: January 2017 – June 2019
 Copyright © iMuSciCA Consortium 2017-2019

Executive Summary

In this deliverable we present the first version of the demonstrator for gesture and VR multimodal interaction interface. These tools can be summarized into two different categories according to two different ways of interaction:

- **Free interaction with the music instrument:** The user interacts freely with the music instrument in the virtual 3D world with his fingers/arms.
- **Gesture based interaction with a music instrument:** The user interacts with music instrument with gestures. Certain gestures are mapped to certain aspects of music instrument performance.

The first version of this demonstrator utilizes two sensors.

- The Leap Motion sensor which allows interaction with the music instrument by using the fingers.
- The Kinect sensor which allows with the music instrument by using the arms.

The environments can be accessed by the following URLs :

- Leap Motion Interaction: <http://tesla.ilsp.gr:1994/leap.1/>
- Kinect Free Interaction: <http://tesla.ilsp.gr:1994/kinect.1/>

Username: imuscica

Password: kidslearn

Software Dependencies:

Leap Motion SDK:

<https://developer.leapmotion.com/sdk/v2>

Kinect SDK:

<https://www.microsoft.com/en-us/download/details.aspx?id=44561>

Kinect Additional Software Needed:

<http://tesla.ilsp.gr:1991/software/kinect/>

Version Log			
Date	Version No.	Author	Change
04-09-2017	0.1	Aggelos Gkiokas (ATHENA)	Initial content
25-09-2017		Kosmas Kritsis (ATHENA), Maximos Kaliakatsos-Papakostas (ATHENA), Antigoni Tsiami (ATHENA), Nancy Zlatintsi (ATHENA)	Input on different sections
27-09-2017	0.3	Aggelos Gkiokas (ATHENA)	Finalized content and sent for internal review
29-09-2017	0.4	Aggelos Gkiokas (ATHENA)	Reviewers comments incorporated
02-10-2017	1.0	Vassilis Katsouros (ATHENA)	Submission to the EU

Disclaimer

This document contains description of the iMuSciCA project findings, work and products. Certain parts of it might be under partner Intellectual Property Right (IPR) rules so, prior to using its content please contact the consortium head for approval.

In case you believe that this document harms in any way IPR held by you as a person or as a representative of an entity, please do notify us immediately.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of iMuSciCA consortium and can in no way be taken to reflect the views of the European Union.

iMuSciCA is an H2020 project funded by the European Union.

TABLE OF CONTENTS

Executive Summary	1
1. Introduction	5
2. Installation and technical requirements	6
2.1. Installation of the Leap Motion Sensor	6
2.2. Installation of the Kinect Sensor	6
3. Description of demonstrator and user manual	7
3.1. Leap Motion Demonstrator	7
3.1.1 Choosing Interaction Type	9
3.1.2. Interacting with the Virtual Music Instrument	9
Free Hands Interaction	9
Gesture-based Interaction	10
3.1.3. Record and Playback of Interaction Performance	11
3.2. Kinect Demonstrator	12
3.2.1. Kinect Free Interaction	12
3.2.2. Kinect Gesture-based Interaction	14
3.2.2.1. Upright Bass/Bowing	15
3.2.2.2. Air Guitar	16
3.2.3. Kinect Free Interaction Recording	17

LIST OF ABBREVIATIONS

Abbreviation	Description
FOV	Field-of-View
HTML	Hypertext Markup Language
OS	Operating System
SDK	Software Development Kit
ATHENA	ATHENA RESEARCH AND INNOVATION CENTER IN INFORMATION COMMUNICATION & KNOWLEDGE TECHNOLOGIES
UCLL	UC LIMBURG
EA	ELLINOGERMANIKI AGOGI SCHOLI PANAGEA SAVVA AE
IRCAM	INSTITUT DE RECHERCHE ET DE COORDINATION ACOUSTIQUE MUSIQUE
LEOPOLY	3D FOR ALL SZAMITASTECHNIKAI FEJLESZTO KFT
CABRI	Cabrilog SAS
WIRIS	MATHS FOR MORE SL
UNIFRI	UNIVERSITE DE FRIBOURG

1. Introduction

In this deliverable the gesture interaction with the Virtual Instruments will be described. The aim of gesture interaction core technology is to provide the students (especially those that are not musically educated) with the ability to perform various Virtual Instruments. To do so, two types of sensors are used in the first version of this core technology which are described as follows.

The first is the Kinect V2 Sensor for Xbox One¹, a motion sensing sensor by Microsoft, which constitutes a good solution to implement high precision motion tracking. For the Kinect sensor two type of music interactions are designed in the current version of this Core Enabling Technology:

a) *Free interaction* with the music instrument by using the arms, allowing the students to navigate in the virtual space where the virtual instrument exists, by means of mapping the movements of their arms to the corresponding coordinates in the virtual 3D space. The current version involves the use of two arms to interact with a stringed instrument.

b) *Gesture-based interaction*, which enables the students to play the virtual instrument by performing specific predefined gestures simulating and resembling real instrument playing movements of the arms. The gesture-based interaction is at the moment a stand-alone application existing in two versions simulating the playing of two different virtual instruments: i) an upright bass using a virtual bow and ii) a guitar using a plectrum. In this mode, the student can thus use gestures that have some referential similarity to the gestures that a musician does when performing the specific instruments.

Secondly, we utilize the Leap Motion sensor² in order to track the user's hands, fingers and gestures. As in the Kinect sensor, two main types of interaction are supported in the current version of this Core Enabling Technology:

a) *Free Interaction* allows the students to navigate in the virtual space where the different virtual instruments exist, by means of mapping their real hand and finger movements to the analogous coordinates in the virtual 3D space. Their hands are modeled by virtual 3D rigged hands which are able to collide with the 3D model of the given instrument. The current version involves, the interaction between the ten fingers of the hands, with a stringed instrument..

b) *Gesture-based Interaction* enables the students to play the virtual instrument by performing specific gestures. Currently, we are mapping the downward movement (i.e. tapping gesture) of each individual finger of both hands (omitting the two thumbs), to the plucking interaction of respective number of strings.

The current deliverable is a demonstrator of the Core Enabling Technology of Gesture Interaction and does not include any connection to any sound creation module. Thus, the output of the

¹ Kinect V2 sensor: <https://www.xbox.com/en-US/xbox-one/accessories/kinect>

² Leap Motion homepage: <https://www.leapmotion.com/>

following demonstrators are commands (textual output) related to the gesture recognition results of the aforementioned technologies.

In the following Section we describe the setup and installation processes of the two sensors.

2. Installation and technical requirements

Both sensors require to go through an installation process for their appropriate SDKs. For the Kinect Sensor, an additional software is needed.

2.1. Installation of the Leap Motion Sensor

In order to utilize the Leap Motion sensor, the user needs first to download the appropriate SDK software, according to the employed OS. The latest version of the Leap SDK is the *Orion* SDK and it is only available for Windows OSs. In order to keep an homogenous architecture and support more target OSs, the 3D interaction environment is based on the “V2” Desktop SDK specifications, that supports Windows, Linux and Mac³. The user can download the SDK and find detailed information at the Developers subdomain of the Leap Motion website⁴.

After downloading the SDK and following the onscreen guidelines, the user should end up with a successful installation of the Leap Motion *V2 Tracking* SDK. After the installation, the user should notice the Leap Motion notification icon at the Windows Taskbar/Mac Finder Menubar, since the Leap Motion Service is running as a startup application. If the Leap sensor is connected and the Leap Service is operating normally, then the Leap icon is colored green, otherwise it remains black if the sensor is unplugged. Other colors indicate errors and abnormal operating conditions (see Fig. 2.1.1).



Figure 2.1.1: Leap Motion Service icon states: a) Leap Controller is connected, b) Leap Controller is unplugged.

2.2. Installation of the Kinect Sensor

In addition to the Kinect sensor hardware, the Kinect for Windows Software Development Kit (SDK) 2.0 is required, in order to enable the interaction with the virtual instrument. The Kinect SDK enables developers and/or users to create and use applications that support gesture recognition. Specifically, the user has to install, in a windows PC running Windows 8, Windows 8.1 or Windows 10, the Kinect SDK v2 (KinectSDK-v2.0_1409-Setup.exe), which can be found in the official Microsoft webpage⁵. First the user has to register so as to obtain the latest resources required for the installation. After the executable for the SDK has been downloaded and successfully installed, the user has to plug the the USB cable from the power hub of the Kinect sensor into a USB 3.0 port on the computer, so as all relevant Kinect drivers are installed. The user can then verify that installation has completed by

³ <https://developer.leapmotion.com/sdk/v2>

⁴ <https://developer.leapmotion.com>

⁵ <https://www.microsoft.com/en-us/download/details.aspx?id=44561>

launching Device Manager and verifying that "KinectSensor Device" exists in the device list. For further information regarding system requirements and installation instructions please refer to the above mentioned webpage.

For the gesture interaction to be enabled the user has to activate the demonstrator interface by running the executables (*.exe files) that are provided at⁶. Specifically, there are three (#3) different executable files: i) one for the *free interaction* mode, named KinectFreeInteraction.exe and two for the *gesture-based interaction* mode and the two different virtual instruments, i.e., the upright bass - named UprightBass-Bowing.exe - and the air guitar - named Guitar.exe. All executables are provided in .rar format with the respective name. The user has to unzip the demonstrator that he/she wishes to test and then run the specific *.exe files, which are essential in order to activate the environments for the free and the gesture-based interaction with the Kinect. All other files included in the zip are needed in order for the demonstrator to run properly, as for instance for the online gesture recognition.

3. Description of demonstrator and user manual

The Gesture and Virtual Reality (G-VR) tools for music interaction allows students to use their hands (using the Leap Motion sensor) or their upper body and their arms (using the Kinect sensor) to perform with a virtual instrument. The user first has to select the sensor of interaction (i.e., Leap Motion or Kinect) and the interaction type (free or gesture-based). For the free interaction, the students are “embedded” in a virtual world where they can see avatars of their hands or bodies interacting directly with the virtual instruments, e.g., virtually plucking the strings, while for the gesture-based interaction the students use pre-defined gestures for triggering interaction events in the virtual instrument. As for instance, if the user selects a bowed string instrument then a gesture is activated that assimilates playing an upright bass, where the right hand simulates the bow gesture (by horizontal movement around the upper part of the waist) and the left hand simulates the fretboard gesture (by moving vertically between head and waist). Similarly, other gestures are available for the Kinect (e.g., guitar-like gestures) and Leap Motion (e.g., piano-like gesture).

3.1. Leap Motion Demonstrator

The Leap Motion Gesture Recognition Demonstrator is hosted in⁷. The demonstrator has been tested and supports all the conventional browsers (i.e. chrome, firefox, nightly, microsoft edge).



Before testing the demonstrator be sure that the Leap Motion is plugged onto the computer and the Leap Service is operating normally (i.e. the Leap Service notification icon is green, see Fig. 3.1.1).

⁶ <http://tesla.ilsp.gr:1991/software/kinect/>

⁷ <http://tesla.ilsp.gr:1994/leap.1/>



Figure 3.1.1: Connected state of the Leap Service Icon at Windows Taskbar and Mac Finder Menu bar.

The Leap Motion demonstrator is a simple HTML webpage (Fig. 3.1.2) which consists of four main areas. In the upper left corner, the *Record and Playback Gesture* area contains the appropriate buttons for recording and reproducing (i.e. playback) a gesture interaction performance. Below the *Record and Playback Gesture* area, lays the *Gesture Type Selection* area where the user can choose the interaction type with which he is going to perform and interact with the virtual instrument, by using either free hand movements or gestures. The main element of the demonstrator that is centered in the window is the *3D Virtual World* where the hands of the user, along with the Virtual Instrument (in the free interaction case only) are presented. In the right part of the screen is the *Algorithm Output* window, which is a text area that logs the output of the gesture interaction algorithm (e.g. *Finger 1 hits String 2* etc).

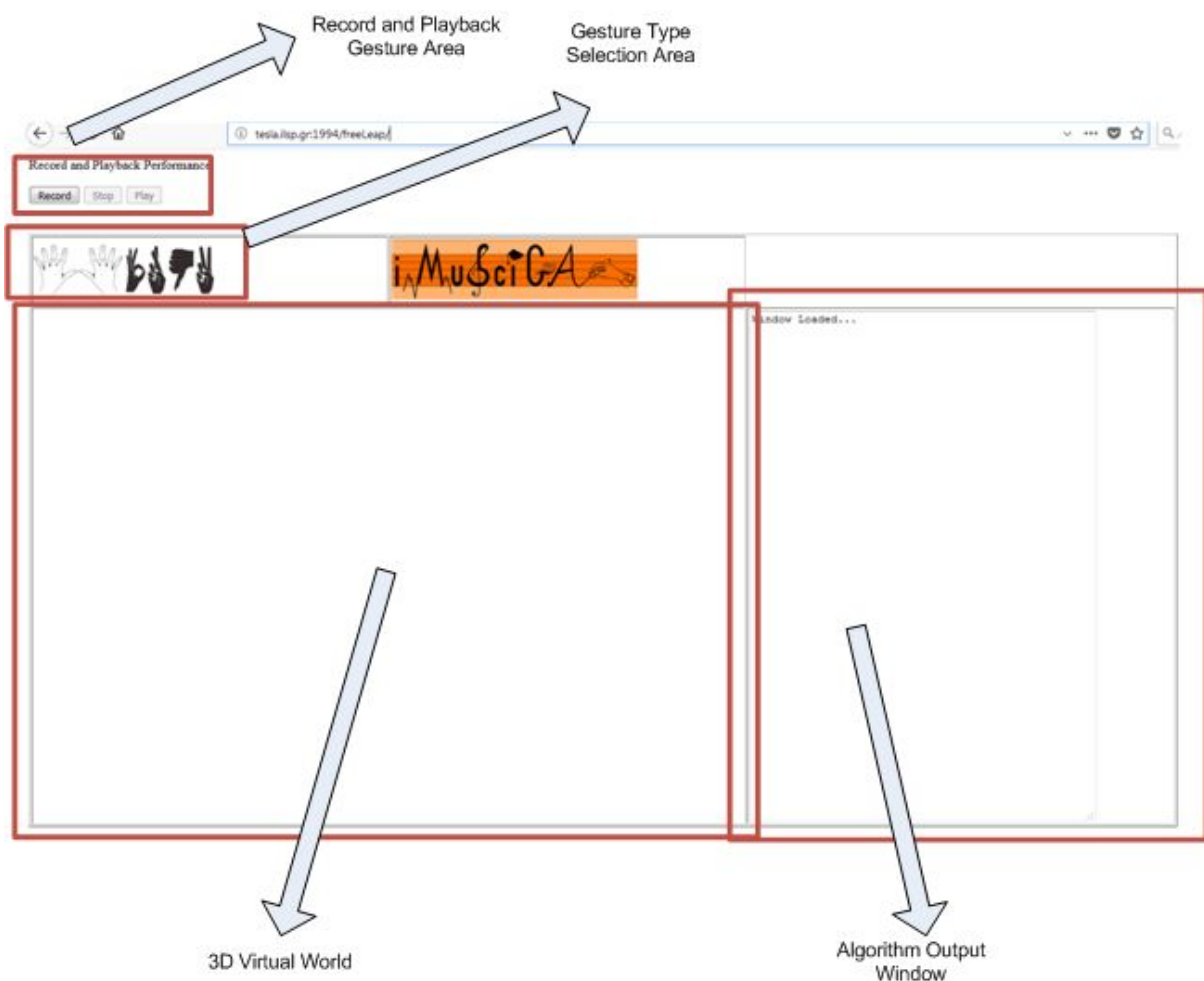


Figure 3.1.2: The Leap Motion Demonstrator interface.

3.1.1 Choosing Interaction Type

The user in order to start performing the virtual instrument he must first make the selection of the Interaction Type. As it was mentioned previously, two types of interaction are available and can be found at the Gesture Type selection area, where two clickable icons represent each of the two interaction types. The leftmost icon stands for the *Free Interaction* option, while on the right side lays the *Gesture-based Interaction* icon (Figure 3.1.3). In the current version of the demonstrator, before the user changes his selection, he must firstly reload the webpage.

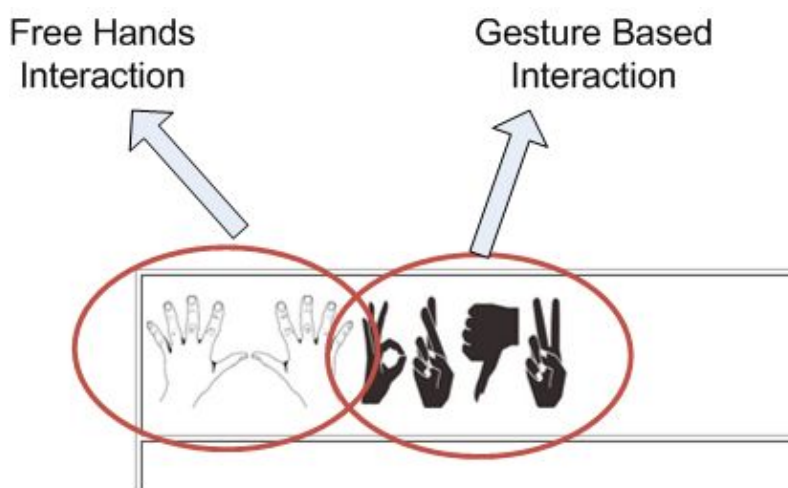


Figure 3.1.3: Gesture Type selection area.



When the user selects an Interaction Type this cannot be undone in the current session. To use another sensor the user must refresh the webpage.

3.1.2. Interacting with the Virtual Music Instrument

Considering the aforementioned types of gestural interaction, we describe briefly the different approaches on the process of instrumental performance that are provided.

Free Hands Interaction

When the user clicks the *Free Hands Interaction* icon (i.e. two white hands icon), he enables the rendering of the virtual instrument, which in the current version of the demonstrator is a four-string instrument. More specifically, four lines are displayed in the 3D Virtual World area, with colors red, purple, blue and green that represent the four strings respectively. Then, the user should place his hands over the Leap sensor in a distance of about 10cm, which by its side would be placed on a steady surface (e.g. a table), with the rear green light facing towards the side of the user. If the tracking of the hands is successful, the virtual 3D rigged hand models will be rendered within the 3D Virtual World at the corresponding coordinates (Figure 3.1.4). Consequently, the user by moving his hands over the sensor's FOV can navigate within the virtual world and interact with the virtual instrument.

Moreover, the collision detection algorithm provides visual and textual feedback. The visual feedback is presented with the blue lines that are displayed along the fingers. When a finger intersects with the virtual instrument, the corresponding blue line turns into red (Figure 3.1.4). Additionally, as a textual feedback, the result of the collision event is logged at the Algorithm Output window, with the format “String X is hit by Finger Y”.

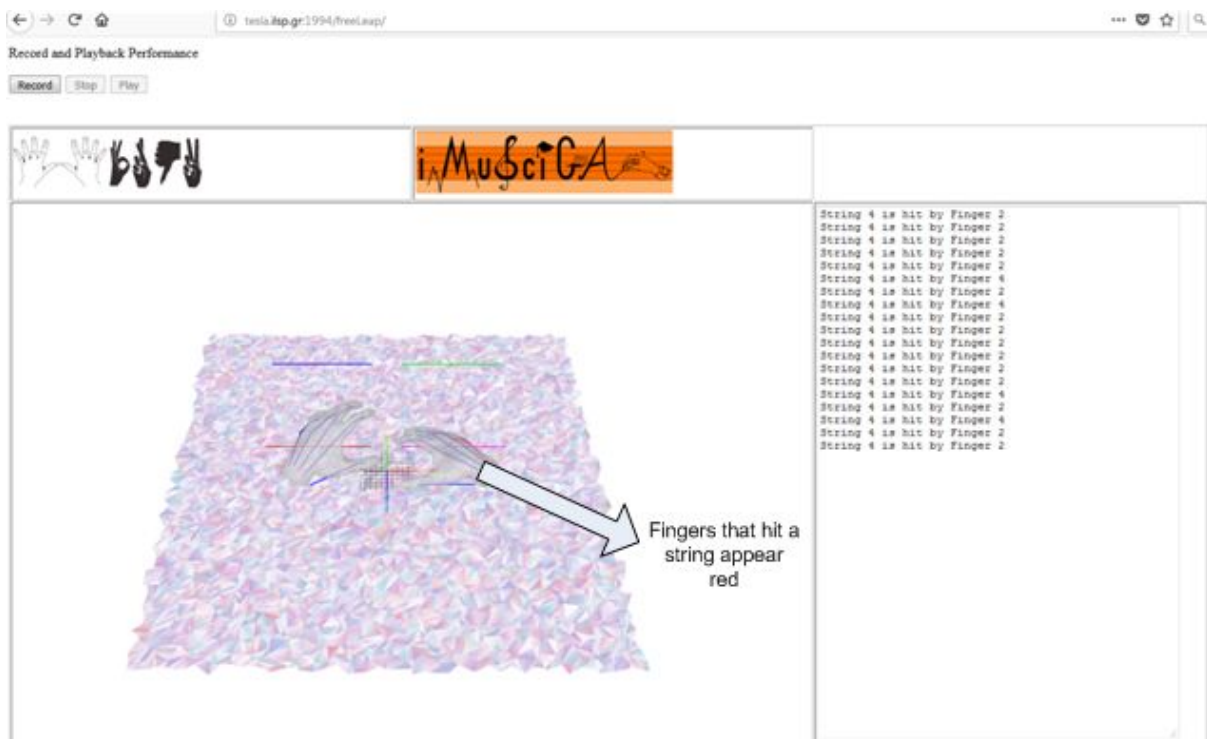


Figure 3.1.4: Leap Motion Free Interaction demonstrator example.

Gesture-based Interaction

In order to proceed with the Leap Motion *Gesture-based Interaction*, the user must click on the respective icon at the selection area (i.e. black hands with gesture signs, Fig. 3.1.3). In this interaction mode, the model of the virtual instrument in the 3D Virtual World is not rendered, since there is no direct interaction of the hands and instrument body. Thus, only the virtual 3D rigged hands models for each recognized hand (Fig. 3.1.5) are rendered. Additionally, we take under account only eight out of ten fingers, discarding the two thumbs, for triggering gestural interaction events. More specifically, we model the tapping gesture (i.e. instant downward movement of the fingertip), for triggering distinct music events, such as mapping each finger either to a plucking string, to a fret of a stringed instrument or to a xylophone key etc. Similarly to the visual feedback provided by the Free Interaction module, the blue lines that model the interaction algorithm, turn red whenever the tapping gesture is recognized. Additionally, a textual feedback is presented as a log message at the Algorithm Output window as a result of the gesture recognition event, with the format “Finger X hits”.

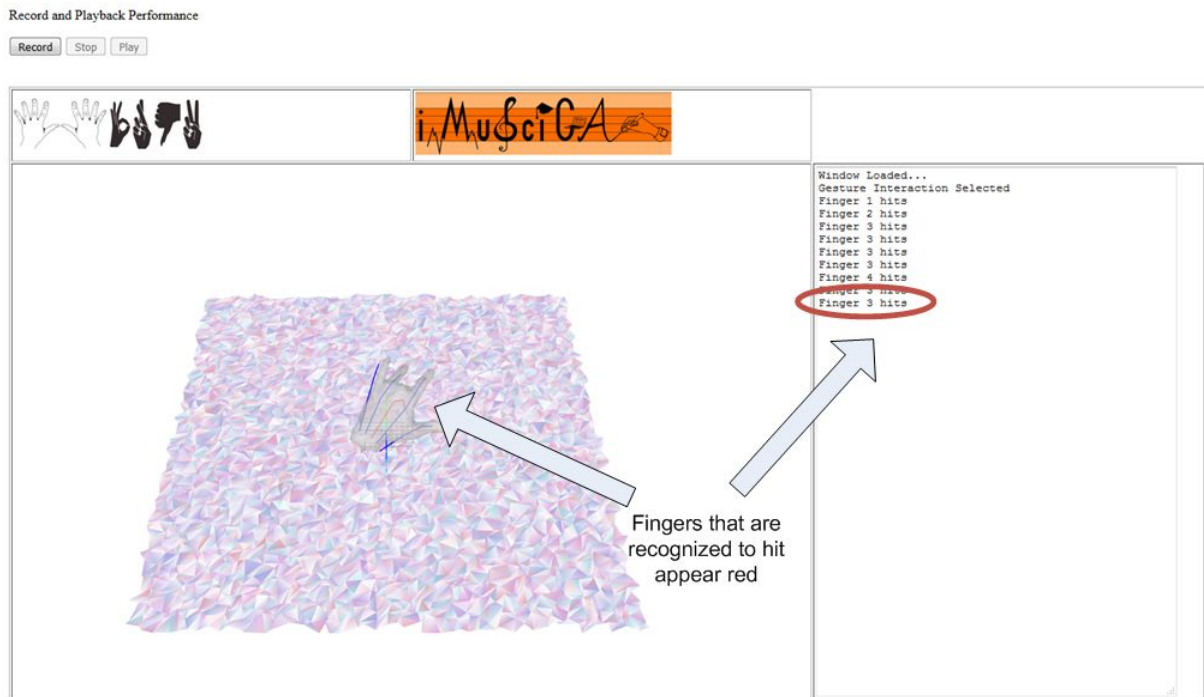


Figure 3.1.5: Leap Motion Gesture-based Interaction demonstrator example.

3.1.3. Record and Playback of Interaction Performance

In order to provide the user with the ability to recreate an Interaction Performance session, a simple recorder that saves the tracking information of the user's hand movements (Fig. 3.1.6) is implemented. When the user after chooses the Interaction Type, then the *Record* button becomes active. Then the user can click the Record button so as to start the session recording. Next, the *Stop* button becomes active in order to enable the user to terminate the current recording process. When the recording is stopped, the *Play* button becomes active.



Figure 3.1.6: Record and Playback demonstrator position.

If an Interaction Performance is recorded, the *Play* button becomes active. By clicking it, the playback of the last recorded session starts, while rendering the consecutive frames on the 3D Virtual World. However, the hands are displayed only by their skeleton and not with the virtual 3D rigged hand models, in order to conveniently present the movement trajectories of the user's hands during the recorded interaction performance, thus facilitating their study (Fig. 3.1.7). During the performance playback, the same interaction messages are triggered as being an actual performance (Algorithm Output Window). Finally, the playback process can be terminated by clicking the *Stop* button.

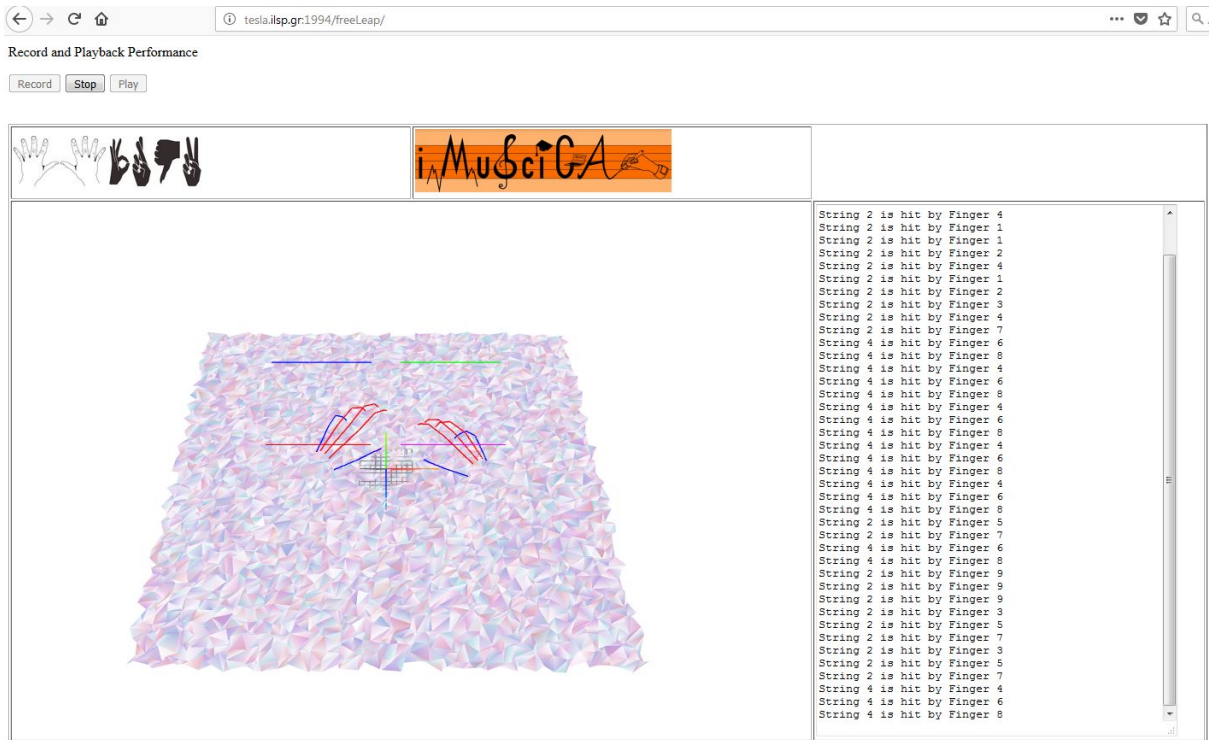


Figure 3.1.7: Example during the Playback process of the demonstrator.

3.2. Kinect Demonstrator

3.2.1. Kinect Free Interaction

Similar to the Leap Motion Free Interaction described in Section 3.1.2, an interface for free interaction with the kinect interface is provided in <http://tesla.ilsp.gr:1994/kinect.1/>. Fig. 3.2.1 shows a screenshot of the interface. As in the interface of Leap Motion, the user can click on the sensor to start the performance.



In order to start a performance, the corresponding executable (see Section 2.2) should be run. The executable is found at <http://tesla.ilsp.gr:1991/software/kinect/FreeInteraction.rar>. Unrar the corresponding file and run the *KinectFreeInteraction.exe*.

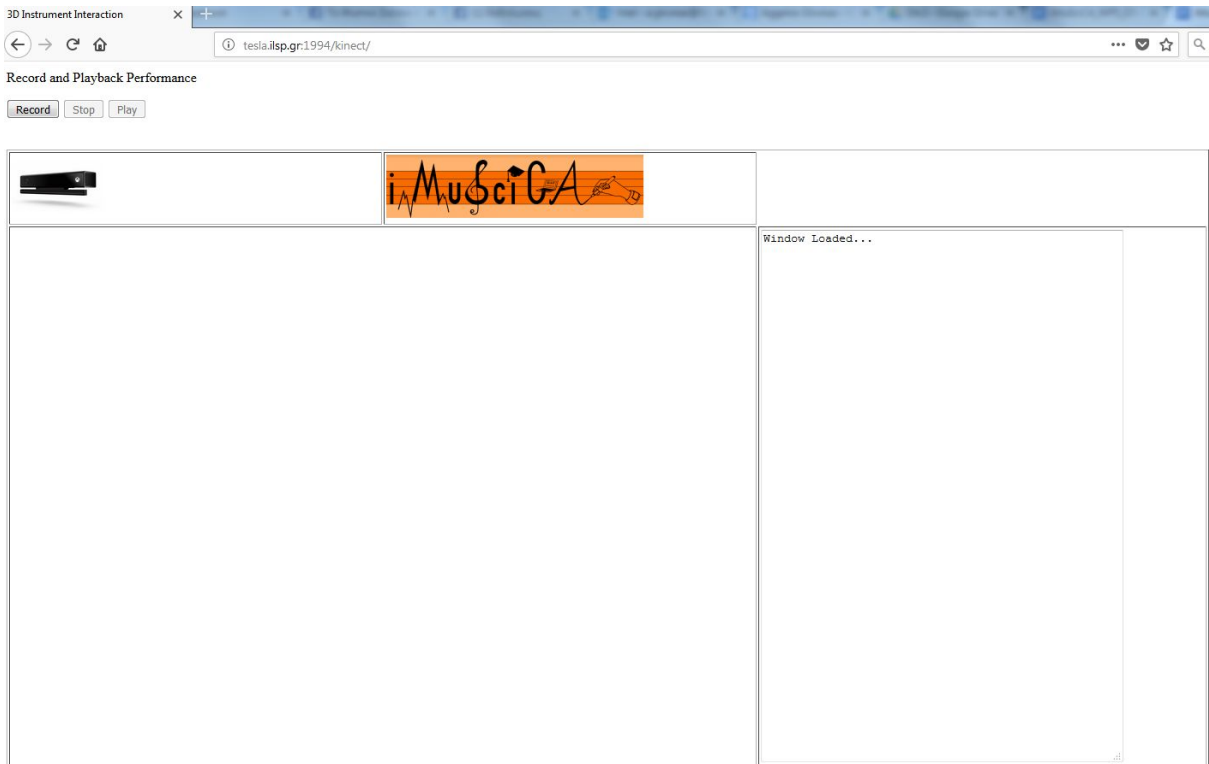


Figure 3.2.1: The Kinect Free Interaction Demonstrator interface.

Once the performance has started, the user is able to see his hands in the 3D interaction world. As in the Leap Motion Demonstrator, the Kinect free interaction applies on a four-string instrument. During the interaction with the instrument, the result of the interaction is shown, e.g. *String 3 is hit by Hand 2*, is shown on the right part of the screen (Figure 3.2.2).

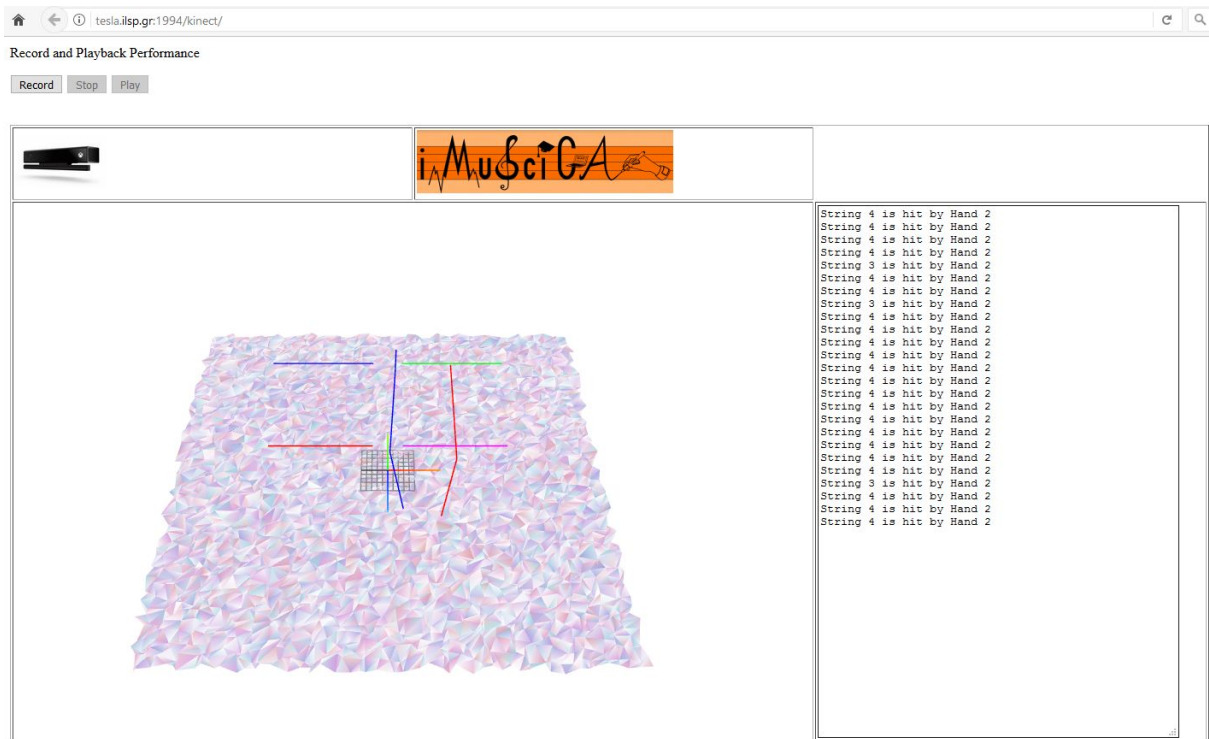


Figure 3.2.2: Kinect Free Interaction with four strings.

3.2.2. Kinect Gesture-based Interaction

The two Kinect Gesture-based Interaction and Recognition Demonstrators (executable files) can be found at ⁸ (for the Upright Bass/Bowing) and at ⁹ (for the Air Guitar). First, the user has to unzip the files and then run the executables found inside the folder (as described next).

The Gesture-based interaction with the virtual instrument, using the Kinect sensor, is at the moment a stand-alone application existing in two versions simulating the virtual playing of two different instruments: i) upright bass using a bow and ii) air guitar. The student can thus select this mode of interaction in order to perform with the virtual instrument using “motion templates”, thus predefined gestures, that have some referential similarity to the gestures that a musician does when performing the specific instruments. Those gesture templates depend on the selected instrument type and they are described in the following subsections.

Next you may find a description of the steps to be followed, in order to test the two demonstrators.



Before testing the demonstrator please make sure that the Kinect sensor is plugged onto the computer and properly installed.

1. The user has to select the demonstrator to be executed, either the Upright Bass/Bowing or the Guitar demonstrator and **run** the UprightBase-Bowing.exe or the Guitar.exe, respectively.
2. The user has to stand in front of the Kinect camera and position him/herself as if he/she is holding the instrument that is visible (overlaid) on the screen of the computer as seen in Fig. 3.2.3 for the two instruments, respectively.
3. Next, the user can test the demonstrator by performing the predefined gestures, which are described in Sec. 3.1.2.1 and 3.1.2.2.



Figure 3.2.3: The Kinect demonstrator interface shown for upright bass/bowing (left) and air guitar (right).

Note!! On the screen the user can see various parameters that change online, thus while moving his/her arms. These parameters, showing information about the performance of the gesture recognizer, are from top to bottom: the pitch (i.e., the pitch of the note that is mapped to each position of the left hand), the right and left-hand positions as measured in the Kinect sensors camera

⁸ http://tesla.ilsp.gr:1991/software/kinect/UprightBass-Bowing_v01.rar

⁹ http://tesla.ilsp.gr:1991/software/kinect/AirGuitar_v01.rar

space and the type of the triggered event (as for instance i) up/down, if the movement of the left hand is upwards or downwards, ii) static, when the left hand is static and a static pitch is intended to be simulated or iii) open string, when the user is not simulating a pitch with the left hand but only triggers a sound event with the right hand). Additionally, the user will be able to see some of the joints of the skeleton that are tracked online while he/she is moving. Those skeleton joints, which are marked with red dots are the following: Head, Neck, Spine Shoulder, Spine Mid, Spine Base, Left Hand and Right Hand.

3.2.2.1. Upright Bass/Bowing

The student by selecting the Upright Bass and the bowing will be able to perform gestures similar to the ones that an upright bass (or a cello) player does.

Gesture 1: In order to enable and activate the “sound” (of the open string) the user brings her/his right hand around the waist height and moves it horizontally (from right to left and the other way around), simulating this way the bowing movements, as if the bow is drawn horizontally across the string, see Figure 3.2.4. As long as the right hand is performing the specific movement/gesture, we assume that the bow is in contact with the string and the open string pitch should be simulated. In the current version, the pitch of the open string is E (or Mi). When no such movement, of the right hand, is performed, a sound event is not triggered.

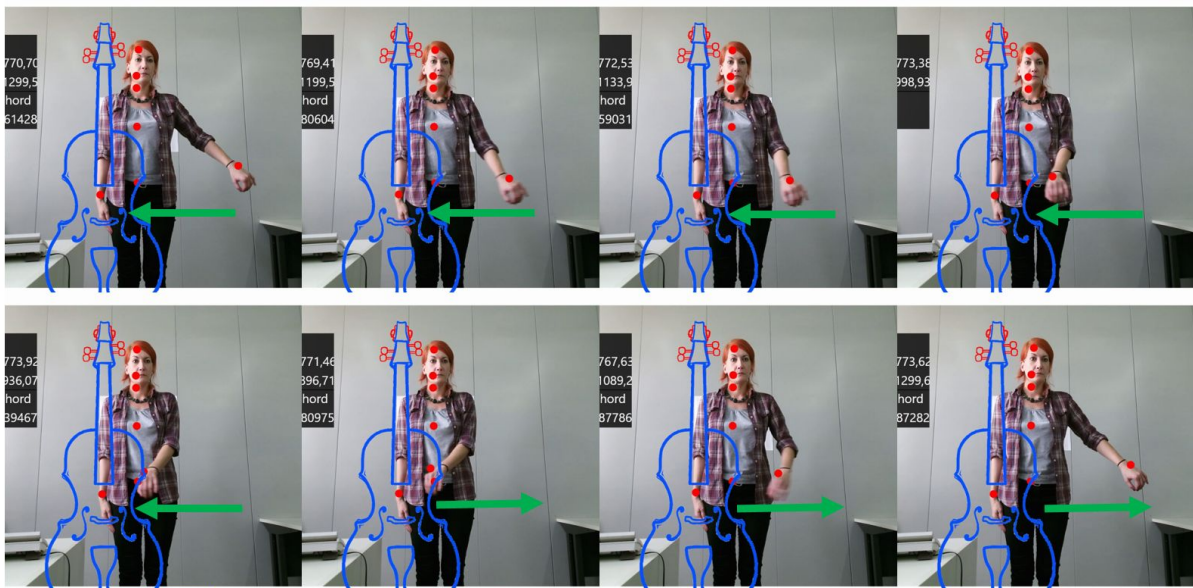


Figure 3.2.4: Bowing Gesture, showing the horizontal movement of the right hand, for triggering a sound event in this case of an open chord.

Gesture 2: In order to be able to change the sounding event, i.e., the pitch of the string, of the upright bass the user has to move the left hand vertically (i.e., downwards or upwards) from head to the waist height, as if it stops the string on the fingerboard determining this way the pitch of the fingered note. This particular gesture is enabled only when *Gesture 1* is active, generating different pitches, see Figure 3.2.5. In the current version of this core enabling technology, the pitches that are simulated depending on the position of the user’s left hand, from top to bottom of the fingerboard, are the twelve pitches of a chromatic scale; considering that the pitch of the open string is E (Mi) then the notes are F, F#, G, G#, A, A#, B, C, C#, D, D# and E (or Fa, Fa#, Sol etc. reaching up to Mi).

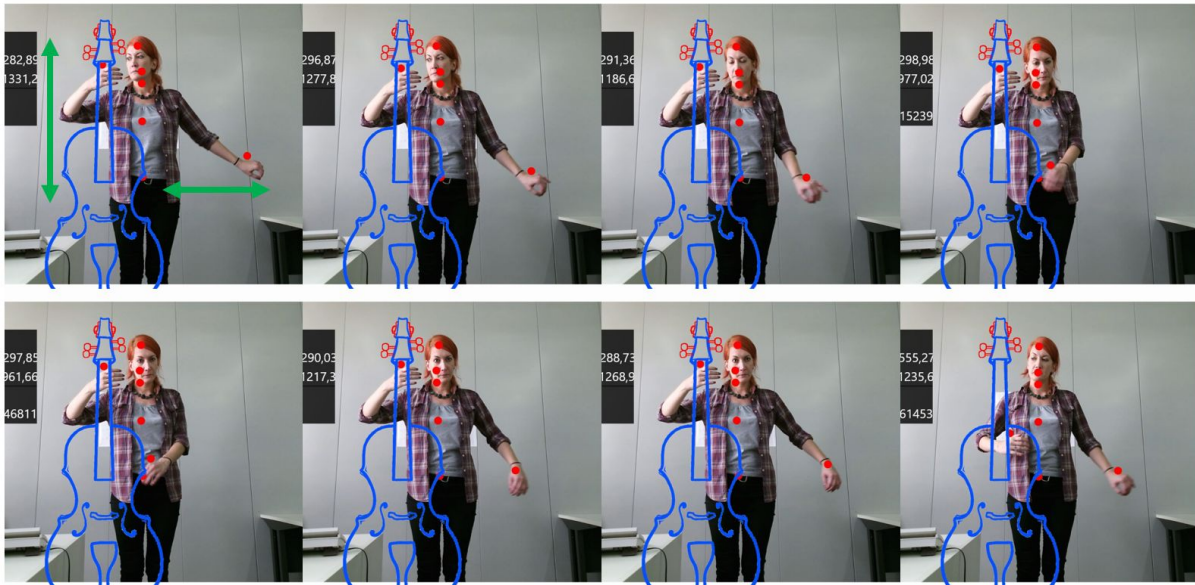


Figure 3.2.5: Combination of *Gesture 1* and *Gesture 2* for triggering sounding events of various pitches. Right hand, showing the horizontal movement, that triggers the sound event and Left hand, showing the vertical movement, that triggers different pitches.

3.2.2.2. Air Guitar

The student by selecting the air guitar will be able to perform gestures similar to ones that a guitar-player does.

Gesture 1: In order to enable and activate the “sound” (of the open string, which in this case is C (Do)) the user brings her/his right hand around the waist height and moves it vertically (downwards or the other way around), simulating this way the moving hand of a guitar player using as for instance a plectrum, see Fig. 3.2.6. As long as the right hand is performing the specific movement/gesture the pitch of the open string should be simulated.



Figure 3.2.6: Guitar Gesture, showing the vertical movement of the right hand, for triggering a sound event in this case of an open chord.

Gesture 2: In order to be able to change the “sounding” pitch of the string the user has to move his/her left hand diagonally from the height of the head to below the waist, as if she/he stops the string on the fingerboard determining this way the pitch of the fingered note. This particular gesture is enabled only when *Gesture 1* is also active generating various pitches, see Fig. 3.2.7. For the first version, we have chosen to map the different position of the left hand to a pentatonic scale including the notes C, D, E, G, and A (Do, Re, Mi, Sol, La).

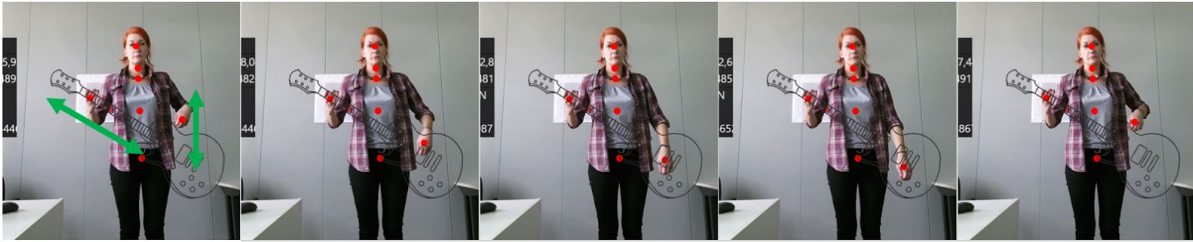


Figure 3.2.7: Combination of *Gesture 1* and *Gesture 2* for triggering sounding events of various pitches. Right hand, showing the vertical movement, that triggers the sound event and Left hand, showing the diagonal movement across the virtual guitar fretboard, that triggers different pitches.

3.2.3. Kinect Free Interaction Recording

Similar to the Leap Motion interaction (see Section 3.1.3), the user is able to record and playback his performance. In the current version of the demonstrator, this is currently available only in the web-based Kinect free interaction interface.