Study of movement coordination in human ensembles via a novel computer-based set-up

Software usage

Francesco Alderisio[1,†], Maria Lombardi[2,†], Gianfranco Fiore[1] and Mario di Bernardo[1,2,*]

[1] *Department of Engineering Mathematics, University of Bristol, Bristol, United Kingdom*
[2] *Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy*
[†] *These authors have contributed equally as first authors*

[*]Correspondence:
Mario di Bernardo, Merchant Venturers Building, Department of Engineering Mathematics, University of Bristol, Woodland Road, Clifton, Bristol, BS8 1UB, United Kindom
m.dibernardo@bristol.ac.uk

In this document we explain how to run the Java application that allows to perform coordination tasks among human and possibly virtual players. In particular, three kind of coordination tasks can be performed:

1. *Solo experiments*: this task involves one agent at a time only. Participant is asked to generate a spontaneous movement of his/her preferred hand, so that his/her motor signature can be recorded;

2. *Dyadic interaction*: this task involves two agents at a time only. Two kinds of trials can be performed:

   - HP-HP trial: human participants can either interact in a Leader-Follower condition or in a Joint Improvisation condition;
   - HP-VP trial: a human player is asked to either lead or follow a virtual agent, described by an appropriated mathematical model.

3. *Group interaction*: this task involves three or more agents, coupled through a structure of interactions that can be set at will. In particular, the topology of the interactions can be either *undirected* or *directed*. Two kinds of networks can be implemented:

   - HP networks: three or more human participants are asked to synchronize the motion of their preferred hand with that of the others they are topologically connected with;
   - mixed HP-VP networks: as HP networks, but one or more participants are virtual agents.

The human player (HP) can perform a trial by means of either a mouse or a Leap Motion controller.

1 How to use your Leap Motion

The instructions written here below are compatible with both an OS X machine and a Windows machine (the correct Leap Motion drivers can be downloaded from *https://dibernardogroup.github.io/Chronos/download.html*).

1. Download the Leap Motion drivers from the provided support material (the name of the folder for OS X machines is *LeapMotionKit_mac*, while for Windows machines is *LeapMotionKit_win*);

2. Run the Leap Motion drivers (for OS X machines is *LeapMotion_Installer_mac_x64_-2.3.1.dmg*, while for Windows machines is *LeapMotion_Installer_2.3.1_winx86.exe*) and, when instructed, connect the Leap Motion to your computer. At this stage, your Leap Motion should be able to properly interact with your machine;

3. Download the folder of the Leap Motion Software Development Kit (SDK), named *LeapSDK*, and copy it in a folder that you prefer.

In order for the software to run correctly when using the Leap Motion as input device, you need to have version 2.3.1 of the Leap Motion library (provided in the folder *LeapSDK*); more updated versions are not compatible with the software application.

2 How to run the code

Since the whole software is written in Java, the Java Virtual Machine (JVM) is needed to launch it. Version 8 of Java is used to implement the software: you can download the correct JRE (Java Runtime Environment) according to your operative system from the Oracle website:

http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html

Also, you can download the compatible Java SE Development Kit from the following link: *http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html*

2.1 Executable files: Administrator and Player

In order to launch a Leap-enabled program, Java needs to find the Leap Motion native libraries at runtime. *LeapJava.jar* must also be on the classpath. You can set Java's *java.library.path* parameter to identify the native library. The command line syntax is slightly different between Mac and Windows. More importantly, on Windows, you have to specify either the 32-bit or the 64-bit libraries to match the architecture of the JVM you are using.
On Mac, you can run the **Sample** program using the following command:

*java -classpath ".: **LeapSDK** /lib/LeapJava.jar" -Djava.library.path= **LeapSDK** /lib -jar **Sample.jar***

On Windows, you can run the **Sample** program using a 64-bit JVM with the following command:

*java -classpath ".; **LeapSDK** /lib/LeapJava.jar" -Djava.library.path= **LeapSDK** /lib/x64 -jar **Sample.jar***

where:

- ***Samples.jar*** is the name of your executable file and its path;
- ***LeapSDK*** is the location of your Leap Motion SDK folder.

If you plan on using the mouse as input device, a double-click on the executable file is enough to launch the application.

You can terminate the application either by clicking on the red "x" on top left of the screen, or by using the following command on Mac:

pkill ***nameApplication***

and the following command on Windows:

taskkill /f /im ***nameApplication***

where ***nameApplication*** is the name of the executable file.

2.2 Executable file: Server

The Server has neither a graphic interface nor the need for the Leap Motion support. For this reason, in order to launch the supplied executable *Server.jar* it is enough to write the command line:

java -jar ***Sample.jar***

where ***Samples.jar*** is the name of your executable file and its path.
Alternatively, since the Server does not use the Leap Motion, you could double-click on the executable file to launch the application.

On Mac, you can terminate the application by using the following command:

pkill ***nameApplication***

while, on Windows, the command is:

taskkill /f /im ***nameApplication***

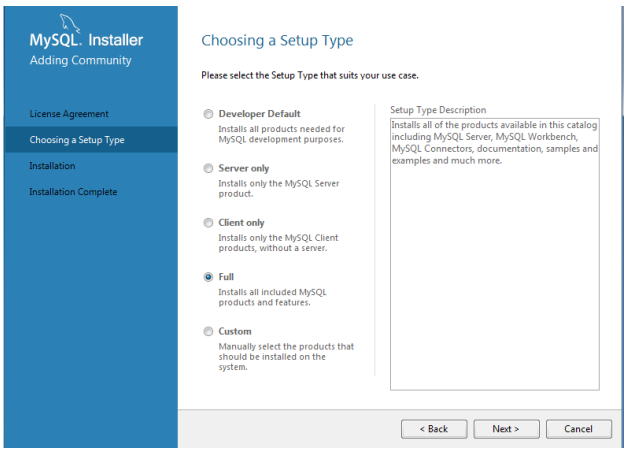
where ***nameApplication*** is the name of the executable file.
The machine where Server runs hosts the signatures database too. Therefore, before beginning any trial, you need to start the database (see Section 2.3).

2.3 Database

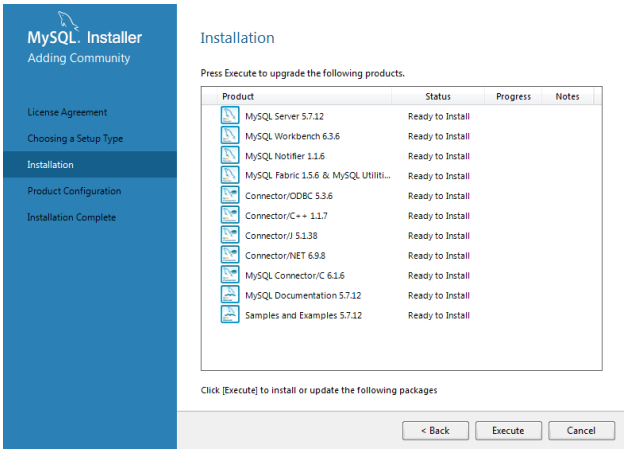
We provide the signature database through a file *.sql*. In order to correctly read this file and be able to interact with it, you need a DBMS (DataBase Management System); specifically, you need a DBMS called MySQL.

Installing MySQL on a Windows machine

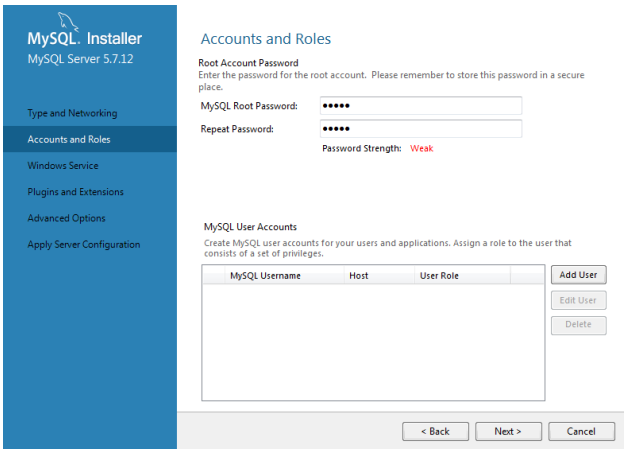
- Download MySQL Installer Community from the website *http://dev.mysql.com/downloads/installer/*. MySQL Installer is a user-friendly wizard that Oracle provides to install the mean products to ensure the correct operation. It includes the installation of both MySQL Servel and MySQL Workbench.
- Run MySQL Installer and follow the instructions of the wizard. In Figure 1 we show how to correctly install the application:
  1. Choose a full setup type;
  2. Check whether both MySql Server and MySql Workbench are in the installation list;
  3. Enter "Pipp0" as password;
  4. Enter "root" as User and click on *Check*.



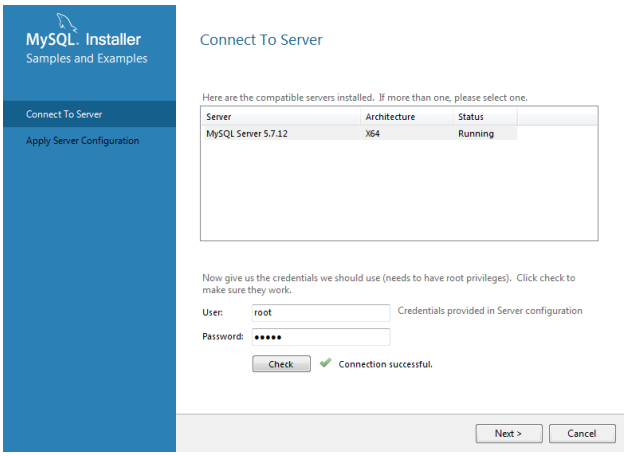
(a) Step 1



(b) Step 2



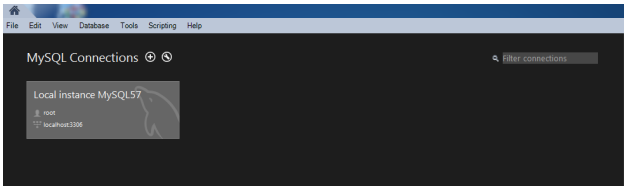
(c) Step 3



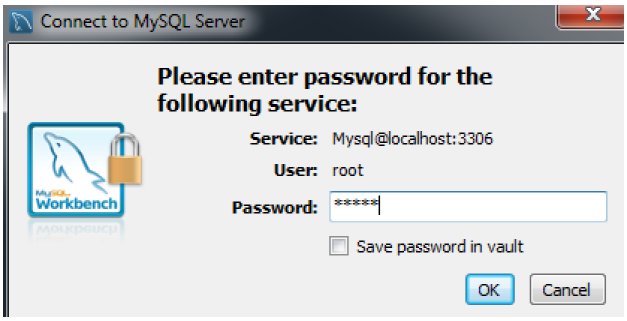
(d) Step 4

Figure 1: Screens of the wizard during the installation of the DBMS

- Now the DBMS is installed. Run MySQL Workbench and click on the connection created by the wizard (Figure 2(a)). When required to insert the password to establish a connection, enter "Pipp0" (Figure 2(b)).



(a) Connection created by the wizard on the start screen of MySQL Workbench



(b) Enter the password "Pipp0" to establish the connection

Figure 2: Screens of MySQL Workbench

Installing MySQL on a OS X machine In the case of OS X machines, MySQL Server and MySQL Workbench need to be installed separately.

- Download and run MySQL Community Server from the website *http://dev.mysql.com/downloads/mysql/*;
- **Save the provided temporary password** (it cannot be recovered) (Figure 3) as it is necessary for the first logging in MySQL;

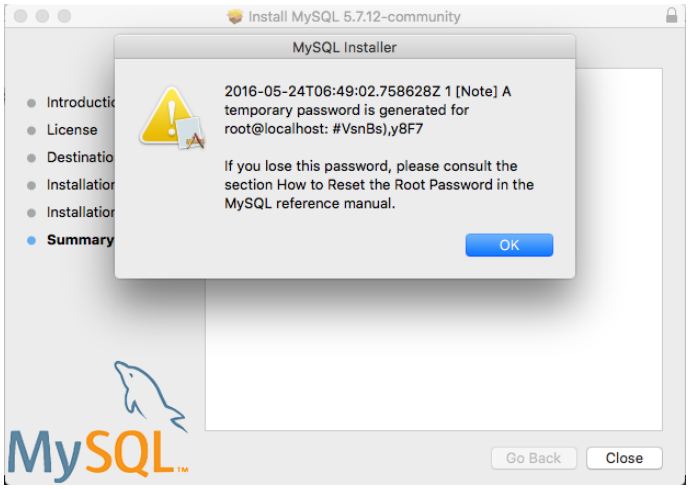


Figure 3: Temporary password for the first logging in MySQL

- Check whether the installation of MySQL Server has been completed successfully. In System Preferences of you computer, you should see the icon of MySQL Community Server (Figure 5);

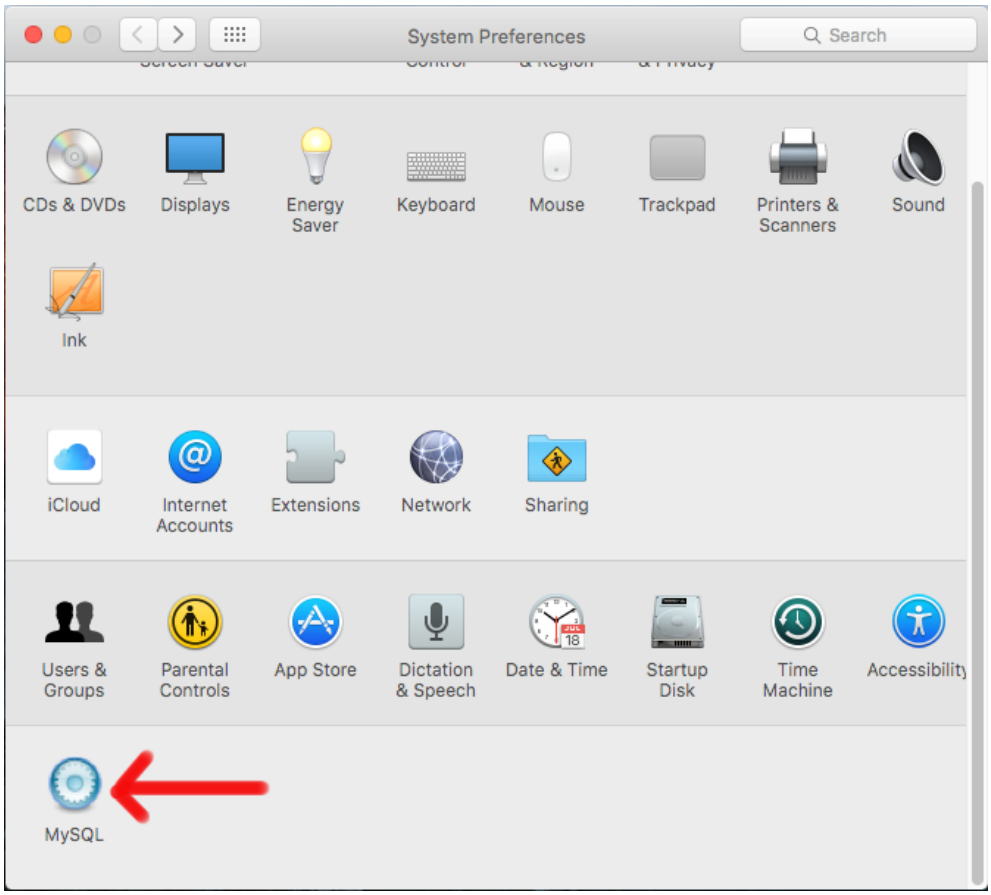


Figure 4: MySQL Server icon.

- Download and run MySQL Workbench from the website *http://dev.mysql.com/downloads/workbench/*;
- Once the installation of MySQL Workbench has been completed succefully, launch it and click on the button "+" to create a new connection. Insert a Connection name at will and check whether the parameters of the connection are set as follow (see Figure 5):
  - Hostname: 127.0.0.1
  - Port: 3306
  - Username: root

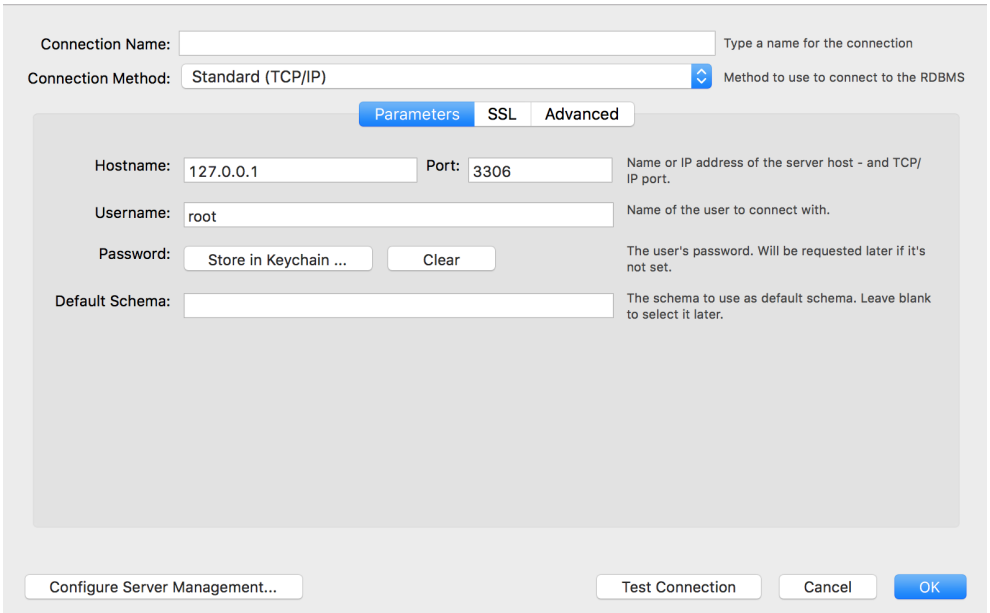


Figure 5: Connection parameters.

- Click on the connection to open it. When presented with the password to establish the connection for the first time, write the temporary password saved before. In our example the temporary password is "$\#VsnBs), y8F7$". You will then be presented with the screen shown in Figure 6 to set a new password. Enter again the temporary password in the box "old password", and enter "Pipp0" in both the box *New Password* and *Confirm*.

Figure 6: Old and new password to establish a connection with the database.

Import the provided database in the workbench You can now start MySql Server and import the provided database, named *DBgame.sql* in your workbench (Figure 7). These steps are the same for both Windows and OS machines:

1. run MySql Server as shown by arrow 1;
2. import the .sql file into the workbench as shown by arrow 2;

3. refresh the side menu of MySql Workbench; you should be able to see the database *DBgame* in the schema list, as shown by circle 3.

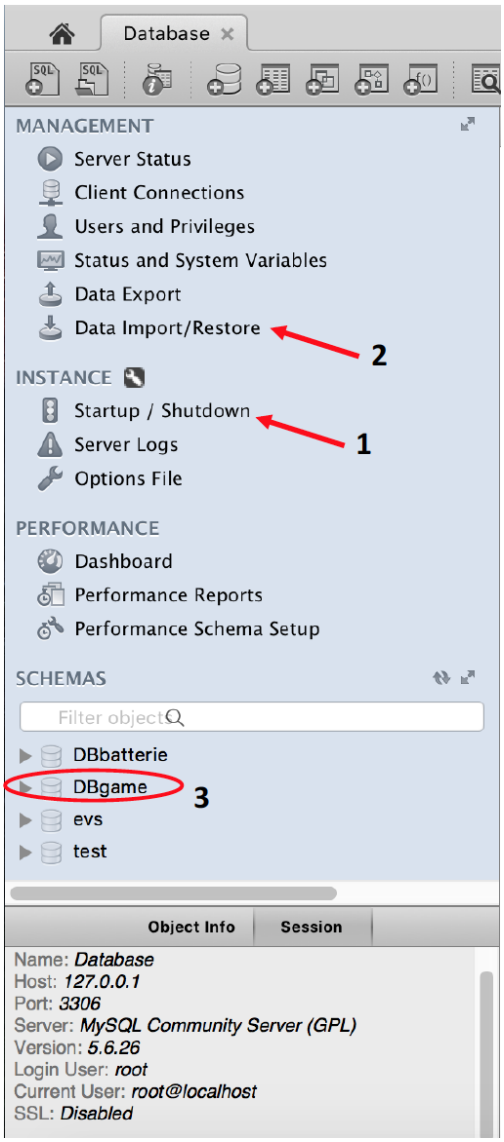


Figure 7: Side Menu of MySql Workbench.

3 How to use the application

In this section we show two examples of a correct use of the software. The former is an example of a two-player session, specifically a leader-follower trial between a HP leader and a VP follower (modeled by HKB equation and PD control); the latter is an example of a group session among three human players.
Both the Server and the database are necessary, as at the end of the trial data is sent to the Server to be saved, and the virtual player needs a motor signature stored in the database. The administrator is needed only for the group session to assign the structure of interactions among

the players. It is also needed for recording new signature into database. Note that Server and Administrator can be run on the same machine.

3.1 Two-player session: HP leader and VP follower

The correct setup for the HP to use the Leap Motion as input device is shown in Figure 8. Since the Leap Motion has a cone-shaped space of data capture, the hand should be placed at least 10 cm above the device (20 cm ideally).
The human player's motion is graphically represented by a blue circle on the game screen, while the virtual player's motion by an orange circle.

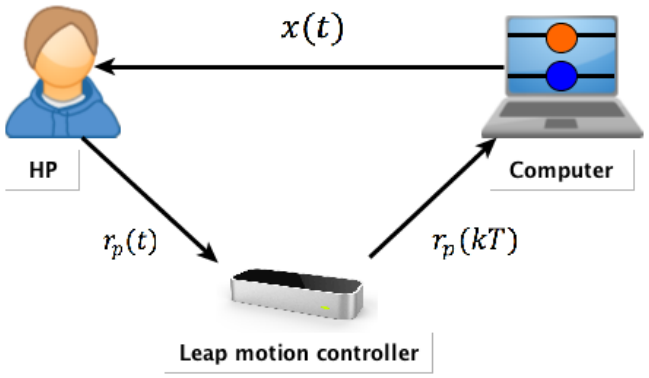


Figure 8: Mirror game setup with leap motion controller. The human participant waves his/her hand over the leap motion controller providing the position time series $r_p(t)$. Then, the sampled position $r_p(kT)$ is sent to the laptop computer. The position of the players are represented on a screen as circles with two different colors (blue: HP , orange: VP).

In this two-player session, two machines are needed (one where the Server runs, the other for the HP). In the case of a two-player session between two human participants, three machines are necessary instead (one for the Server, the other two for each of the HPs). All the computers have to be connected to the same Wi-Fi network. If the network is institutional, i.e. a university Wi-Fi with specific security settings, a portable Wi-Fi router can be used. An external data connection (sim card) inserted in the router is not necessary, as the router creates a WLAN (wireless local area network) without using Internet.

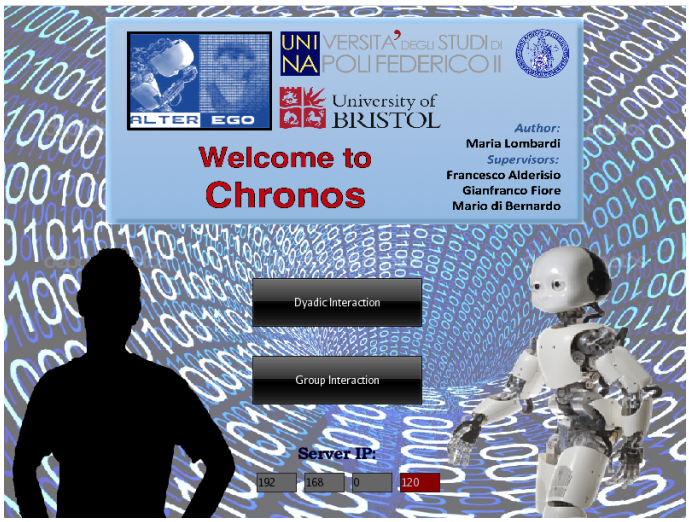
3.1.1 Server side

Before the trial between HP and VP can begin, it is necessary to launch the Server executable file (*Server.jar*) and start the signature database.

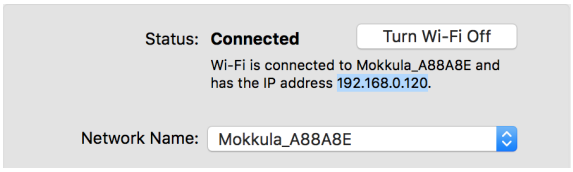
3.1.2 Player side

On a different machine, the human participant launches her/his executable file (*Player.jar*) so that the starting screen is presented (Figure 9(a)). Here, the player can choose between *dyadic interaction* and *group interaction*. On the bottom of the screen there is a text box to insert the

IP address of the Server.
The IP address is made up of four natural numbers from 0 to 255. Each number must be inserted in a single text box. The Server IP can be found in the settings of the machine where the Server application runs. Alternatively, write the following instruction in the command line: *ifconfig* for a OS X machine (Figure 10(a)) or *ipconfig* for a Windows machine (Figure 10(b)).



(a) Starting screen of the player



(b) Example of the IP address of the computer

Figure 9: Starting screen with game menu when the application is launched

```
Marias-MBP:~ marialombardi$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether a4:5e:60:d2:14:7d
        inet6 fe80::a65e:60ff:fed2:147d%en0 prefixlen 64 scopeid 0x4
        inet 192.168.0.120 netmask 0xffffff00 broadcast 192.168.0.255
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
en1: flags=963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX> mtu 1500
        options=60<TSO4,TSO6>
        ether 4a:00:01:09:84:60
        media: autoselect <full-duplex>
        status: inactive
```

(a) Find the IP address of an OS X machine. The IP address is the number at the item *inet*.

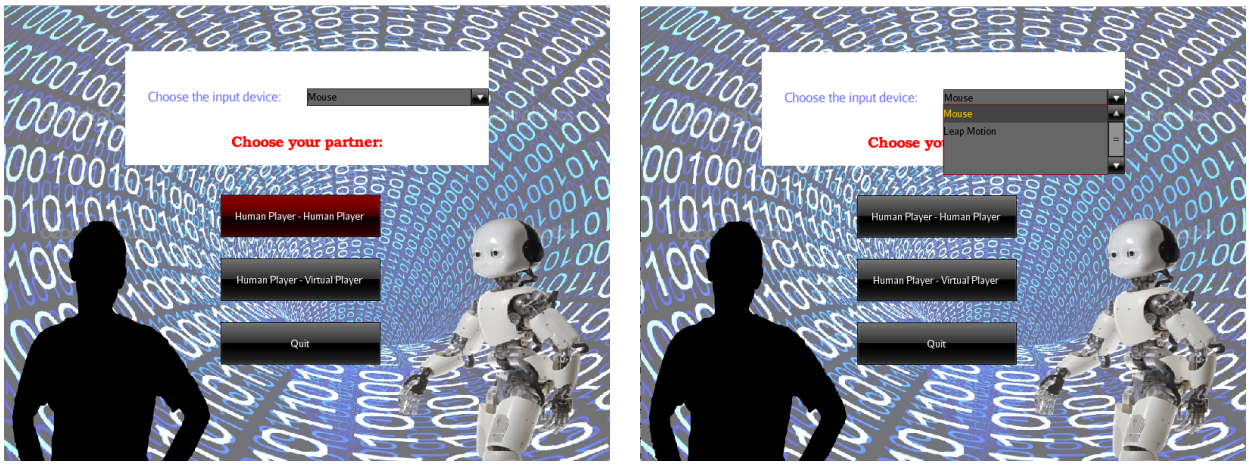
```
C:\Users\Maria Lombardi>ipconfig

Configurazione IP di Windows

Scheda Ethernet Connessione di rete Bluetooth:

   Stato supporto. . . . . . . . . . . . : Supporto disconnesso
   Suffisso DNS specifico per connessione:

Scheda Ethernet Connessione alla rete locale (LAN):

   Suffisso DNS specifico per connessione: localdomain
   Indirizzo IPv6 locale rispetto al collegamento . : fe80::f929:286d:f3a:bbf8%1
0
   Indirizzo IPv4. . . . . . . . . . . . : 172.16.149.128
   Subnet mask . . . . . . . . . . . . . : 255.255.255.0
   Gateway predefinito . . . . . . . . . : 172.16.149.2

Scheda Tunnel isatap.localdomain:

   Stato supporto. . . . . . . . . . . . : Supporto disconnesso
   Suffisso DNS specifico per connessione: localdomain
```

(b) Find the IP address of a Windows machine. The IP address is the number at the item *Address IPv4*

Figure 10: Find the IP address of own machine from the prompt command

If no IP is inserted, the application sets it to the default value, that is *"localhost"* (it is the IP address of the machine where the current application has been launched). In Figure 9 an example of how to correctly enter the IP address in the application is shown. In this example, the HP's computer is connected to the network *Mokkula-A88A8E* and the IP address is *192.168.0.120*.

We choose *dyadic interaction*. The next screen (Figure 11) allows the player to choose the input device (either mouse or Leap Motion), and the type of partner (HP or VP).



(a) Screen with choice of the partner **(b)** Screen with the input choice

Figure 11: Screen where the player can choose the input device and the type of the partner (HP or VP)

We then choose *HP-VP* trial. The next screen allows to set the virtual player as a leader or as a follower (Figure 12).

Two choices are possible:

- the HP is the leader and the VP is the follower;
- the HP is the follower and the VP is the leader.

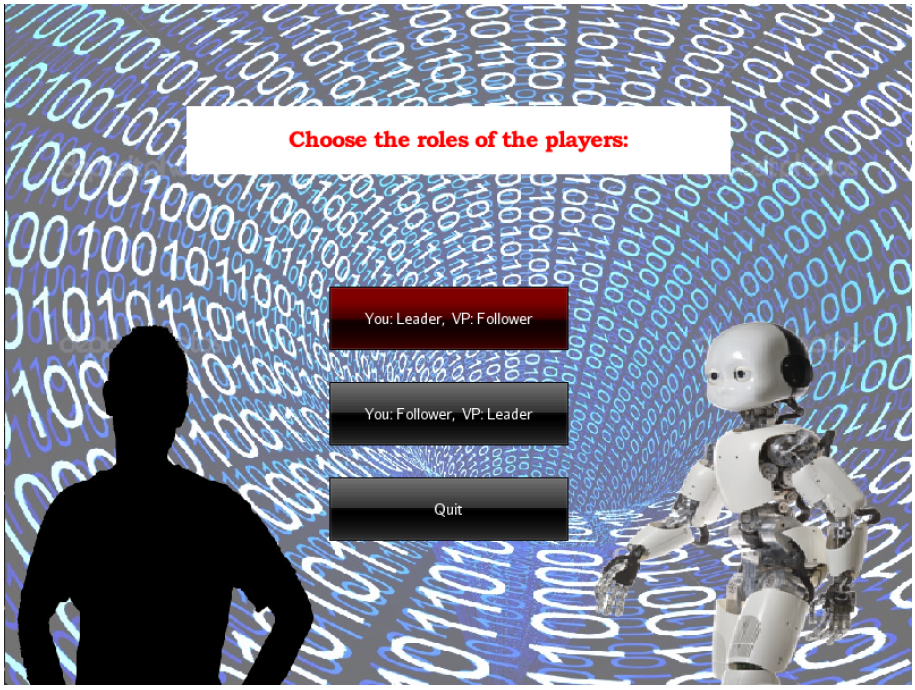
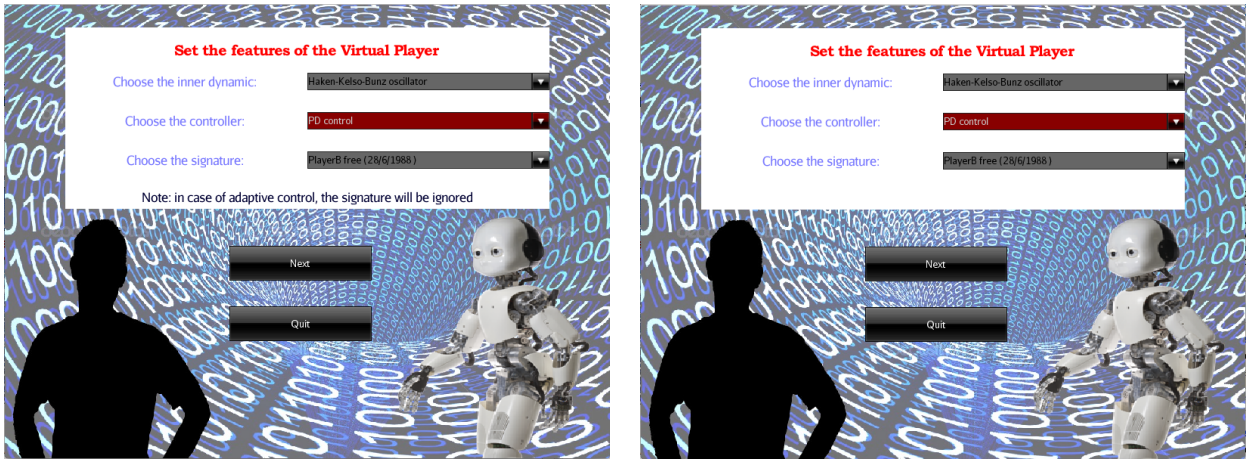


Figure 12: Screen where the player can choose the roles (leader or follower) in the interaction.

In the next screen (Figure 13) we can choose inner dynamics (among HKB oscillator, harmonic oscillator and double integrator), control (between adaptive control and PD control) and motor signature for the virtual player. The list of possible signatures is imported from the database.
Only in the case of VP as follower with adaptive control, the signature will be ignored: an alert message is shown on the screen (Figure 13(a)).

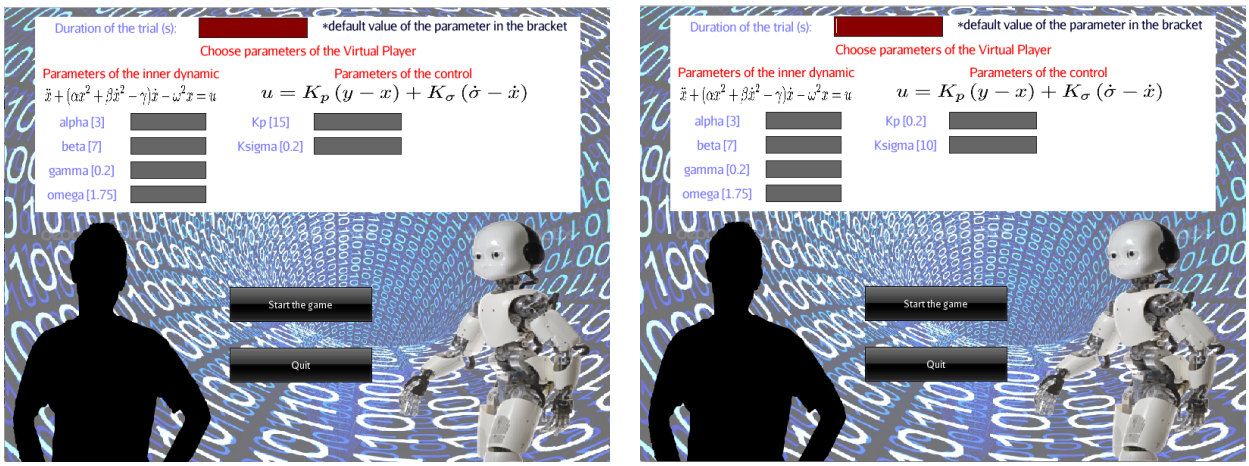


(a) Screen with the choice of inner dynamics and control for the VP

(b) Screen with the choice of motor signature for the VP

Figure 13: Screen where the player can choose inner dynamics, control and signature for the virtual player

In this case the VP acts as a follower, so we select Haken-Kelso-Bunz oscillator as inner dynamics and PD control as controller. The next screen (Figure 14) allows to select all the parameters characterizing the chosen mathematical models. The mathematical laws of inner dynamics and control are displayed above the textboxes. Furthermore, the duration of the trial can be set (in seconds).



(a) Screen with the choice of parameters for VP when acts as a follower

(b) Screen with the choice of parameters for VP when acts as a leader

Figure 14: Screen to set trial duration and parameters for the VP.

If no values are entered in the textboxes, the software automatically assigns default values for the virtual player. The default values for each option are shown between brackets near each parameter and detailed below:

- **Duration of trial**: 30 seconds;
- **Inner dynamics**:
  - HKB oscillator: $\alpha = 3, \beta = 7, \gamma = 0.2, \omega = 1.75$;
  - Harmonic oscillator: $a = 1.5, b = 3$;
  - Double integrator: no parameters.
- **Control**:
  - Adaptive control: $C = 20, K = 40, \psi(0) = \chi(0) = -10, \delta = 0.5$;
  - PD control (VP follower): $K_p = 15, K_d = 0.2$;
  - PD control (VP leader): $K_p = 0.2, K_d = 10$.

In Figure 15 we show the screen seen by the player during the trial. On the right-hand side the countdown is shown, while on top of the screen the type of the trial and HP's role is shown. In this case we have a Leader - Follower trial where the HP acts as leader and the VP as follower.

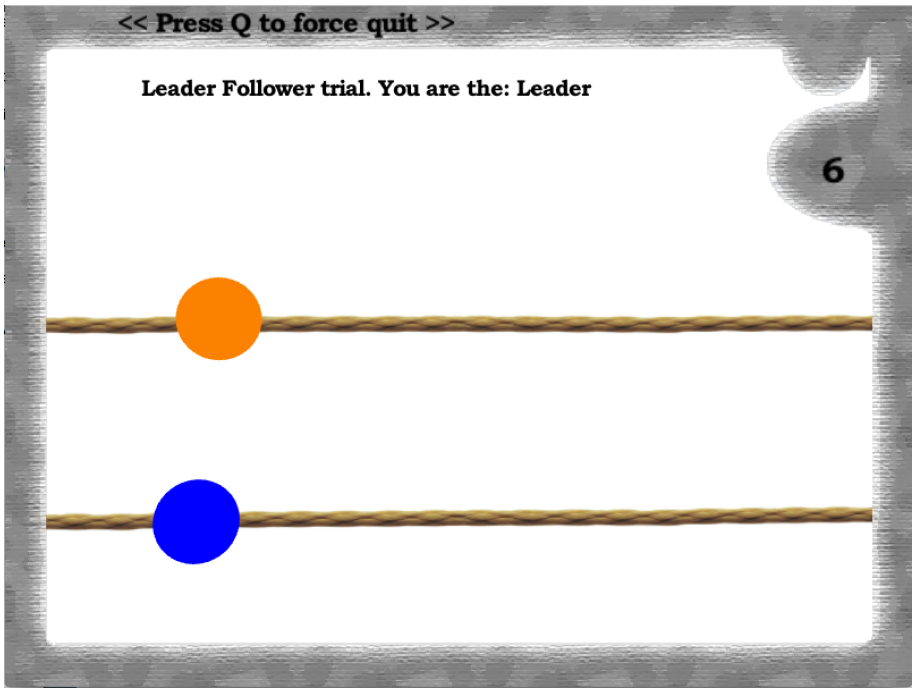


Figure 15: Playing screen with two different circles. The blue circle is HP, while the orange circle is VP

At the end of the trial, the last screen is shown (Figure 16). Here, the player can select the plot that he/she wishes to be visualized. Several plots are possibile: position time series, velocity time series, position PDF, velocity PDF, error position and velocity position.

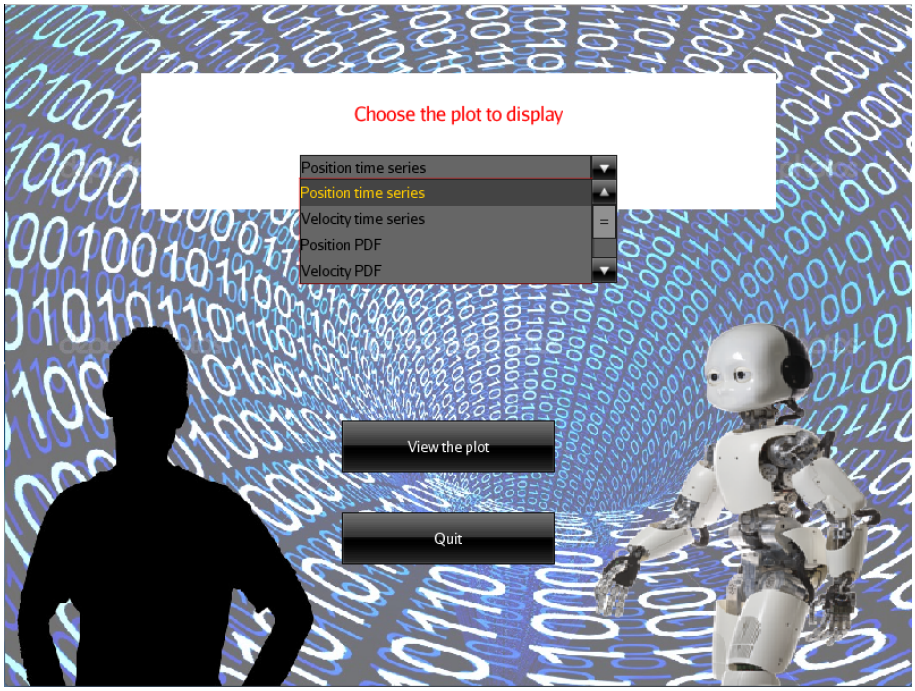


Figure 16: Screen for the choice of plots that the player wishes to see.

By clicking the button "Quit" the player returns to the main menu and can then start a new trial.

3.2 Group session

Differently from a two-player sessions, in the case of multiplayer trials (more than three agents) the role of the administrator is necessary for the network topology to be set before the trial begins. For a correct use of the software, the administrator should use the software on a different computer and should not be one of the players.
As in the case of dyadic interaction, also for group interaction all the computers must be connected onto the same Wireless Local Area Network (WLAN, Figure 17).

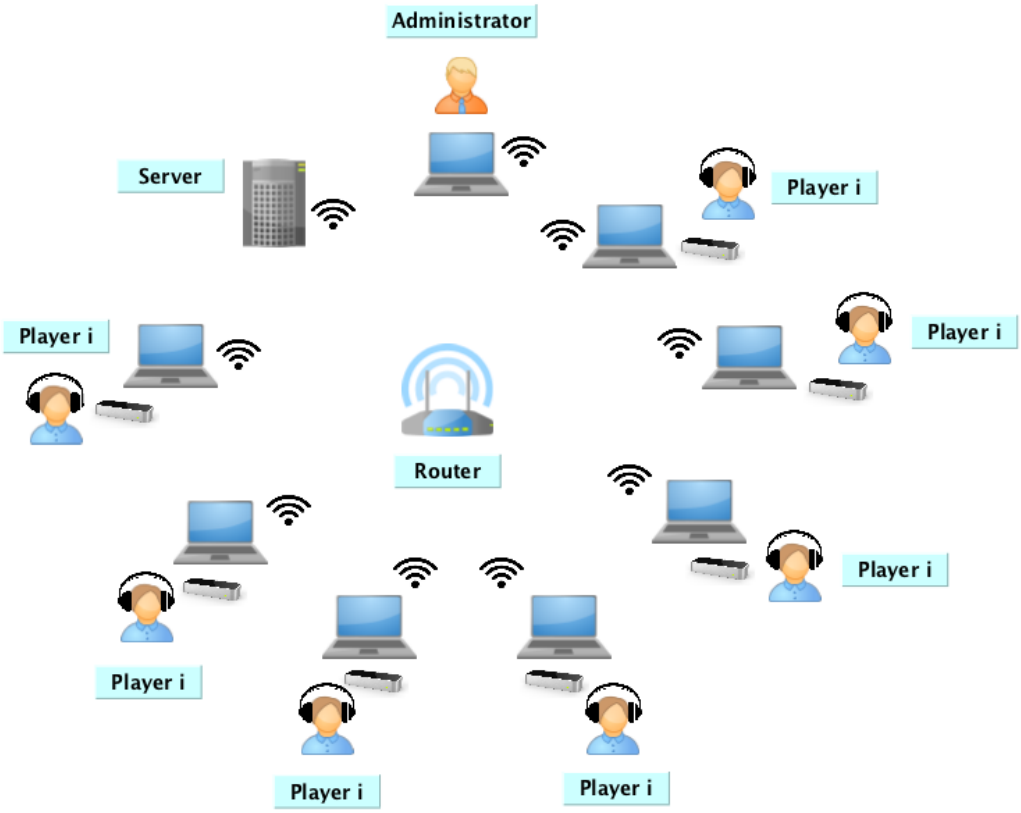


Figure 17: Semplified diagram of the experimental setup for a group session

3.2.1 Administrator and Server side

As in the case of dyadic interaction, before the trial can begin it is necessary to launch *Server.jar* and start the database. It is then necessary for the administrator to run her/his own module (*Administrator.jar*), possibly on the same machine, so that the starting menu shown Figure 18 is presented. If Server and Administrator are run on two different machines, it is necessary to enter the IP of the former in the appropriate textbox.

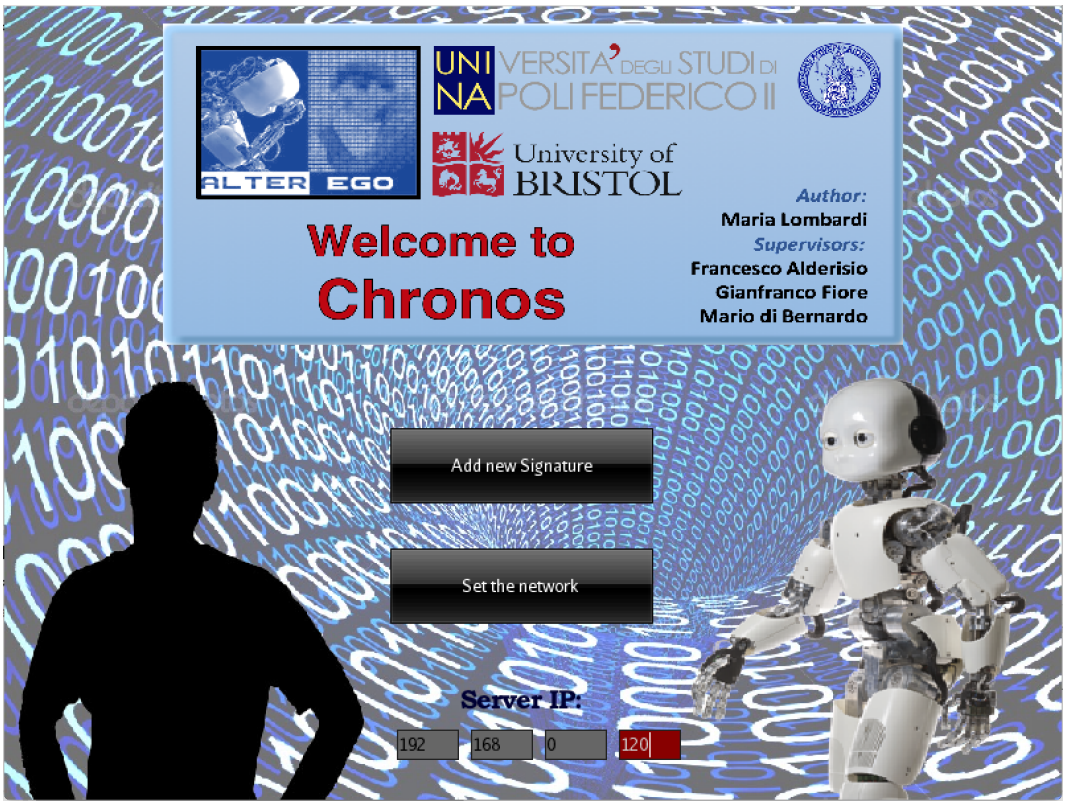


Figure 18: Starting screen of the administrator

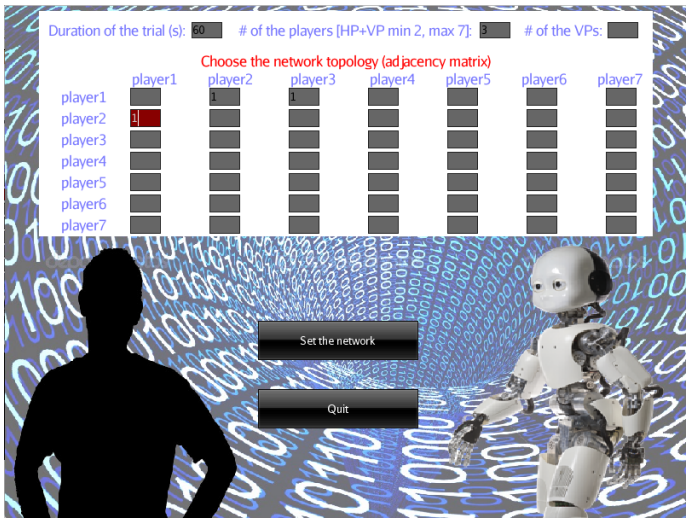
After selecting "Set the network", the next screen in Figure 19(a) shows how the administrator can set all is needed to run a group interaction trial. At the top of the screen there are three different text-boxes:

1. in the first textbox the administrator inserts the duration of the trial in seconds. If no value is entered, the duration will be set to the default value of 30 seconds;

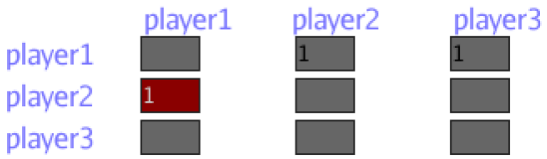
2. the administrator inserts the total number of players (HPs and VPs, combined) for the current trial. If the textbox is left empty, the setup will terminate as the default values are set to 0;

3. the administrator can choose a mixed network, where HP and VP play together. In this case the number of VPs has to be specified (if 0, the total number of players corresponds to only human participants).

Below, the administrator can set the structure of the interactions among the players through the adjacency matrix. By setting the cell (i, j) to 1, player i sees the motion of player j. Each player is identified by a unique natural number from 1 to 7. Figure 19 shows an example for a network with three human players.



(a) Screen to set all is needed for a group interaction trial



(b) Adjacency matrix among the three human players

Figure 19: Screen where the administrator can choose number of players, duration of the trial and structure of interactions among the participants.

If the number of VP is greater than zero, for each VP the administrator can choose inner dynamics, control, motor signature and index in the network. Futhermore the administrator can choose whether VP will act as a leader or as a follower and for each matematical model can set its parameters.
When the trial has ended, the last screen is shown (Figure 20). Here, the administrator can select the plot that he/she wishes to be visualized. Also, the administrator can check the players that he/she wishes to be reported in the plot. In the current example, player 1 and player 3 are visualized.

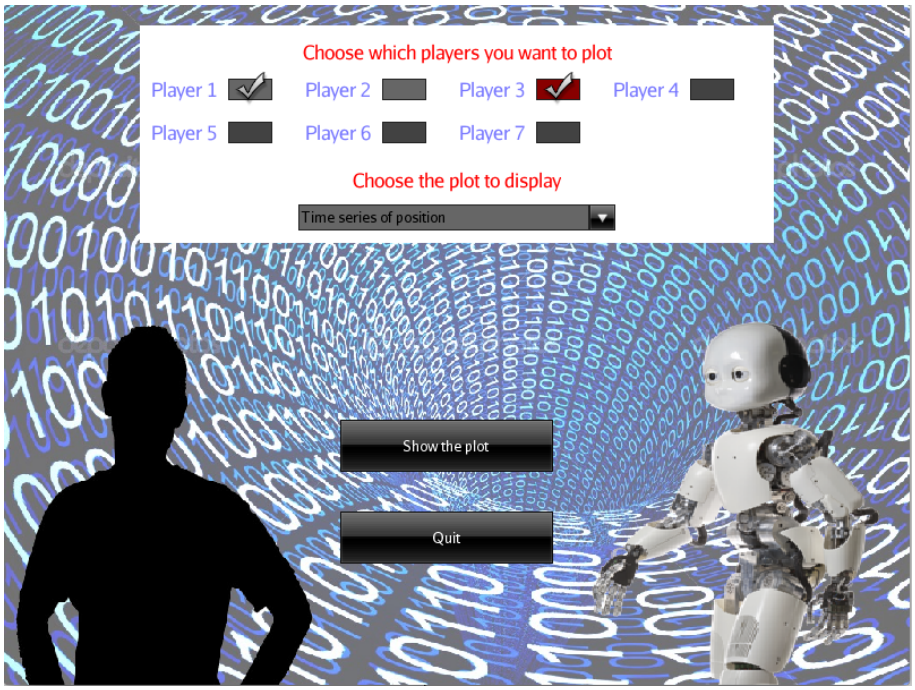


Figure 20: Screen for the choice of the desidered plot

By clicking the button "Quit" the administrator returns to the main menu.

3.2.2 Player side

Each player runs her/his module (*Player.jar*) on their own machine. Then, the starting screen of Figure Figure 9(a) is presented, and players can select *Group interaction* so that the screen of Figure 21 is shown. Here, as usual the players can choose the input device; moreover, they need to enter a reference ID (natural number between 1 and 7) in order to be uniquely identified in the network. Then they can click on "Play on the network".

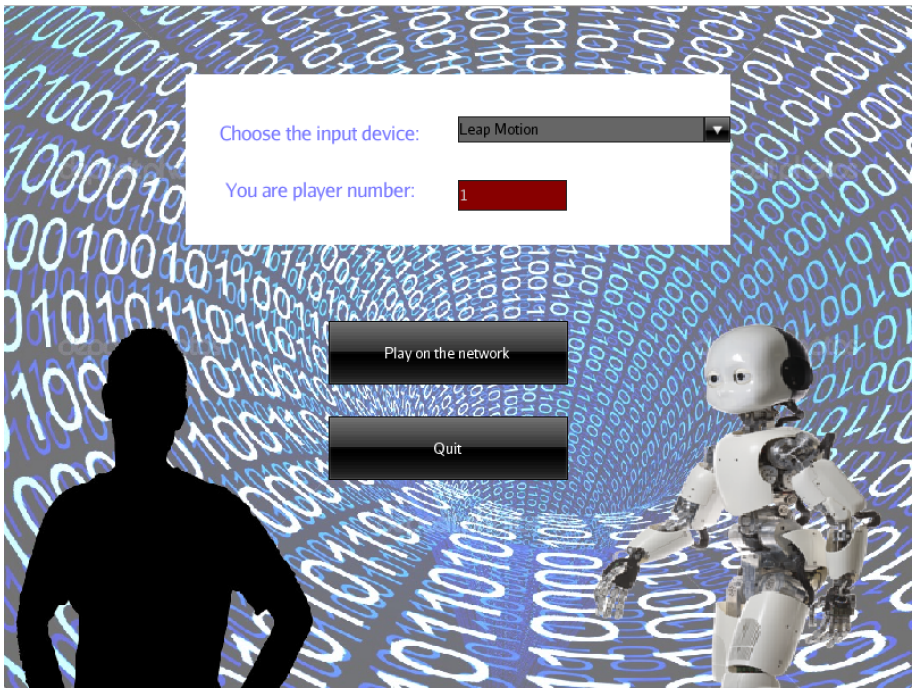


Figure 21: Screen to choose input device and index of the HP in the network.

The next screen (Figure 22) is an awaiting screen that allows all the players to be connected to the Server. During this time, players can correctly place their hand over the Leap Motion.

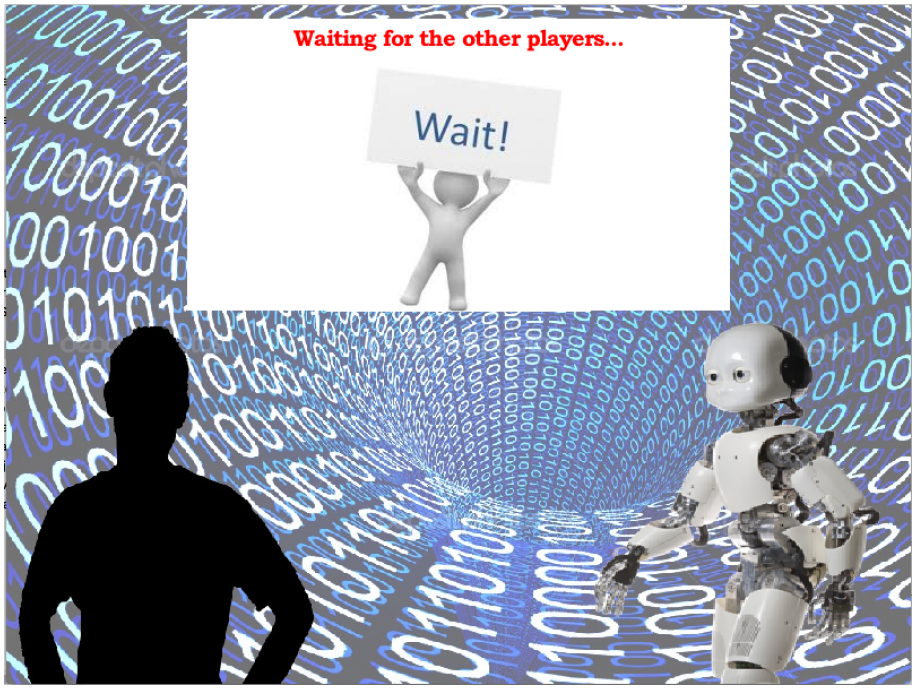
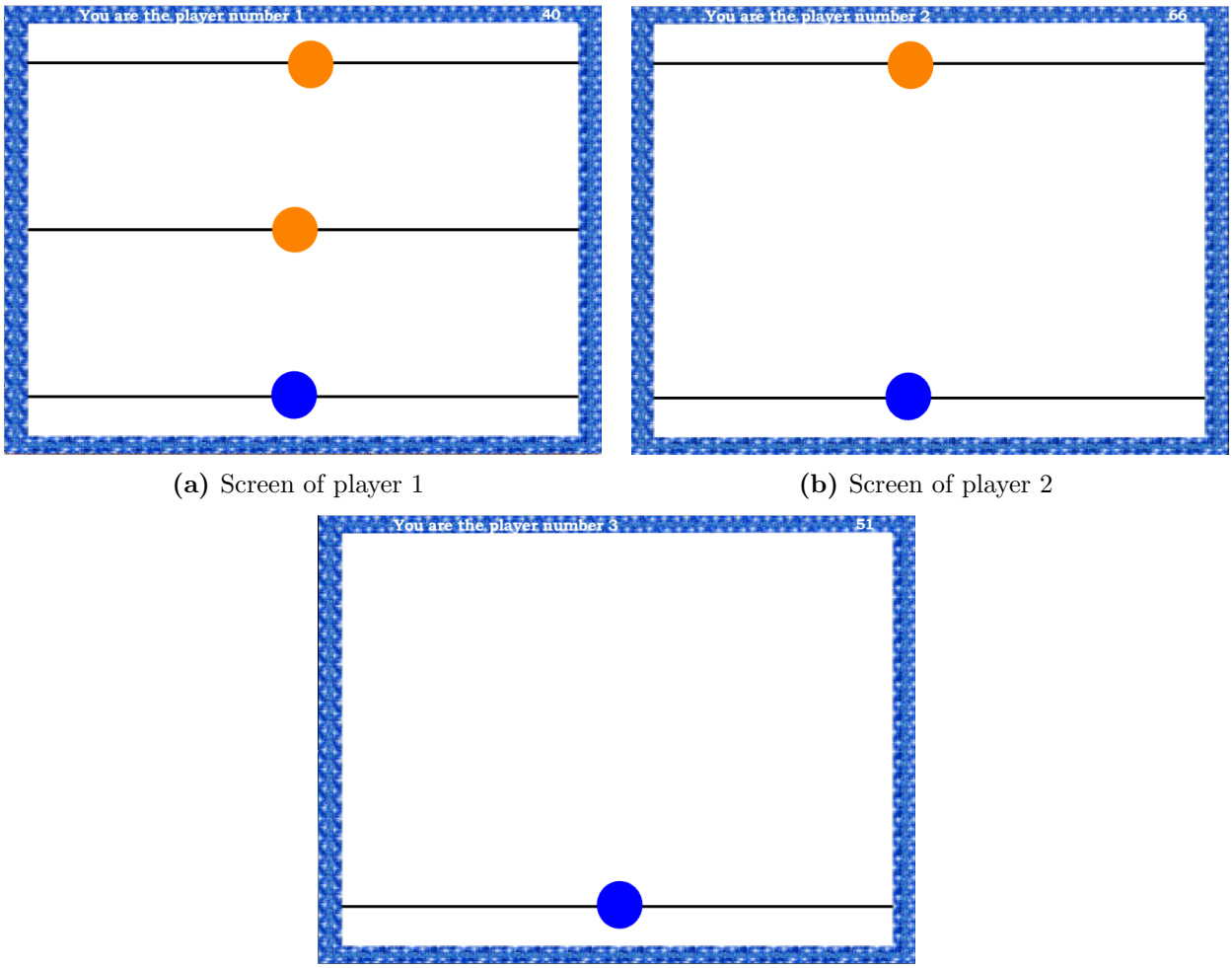


Figure 22: Awaiting screen for all the players.

Once all the players are connected, the software goes on with the game. At this point, each player sees a different visual perspective of the game according to the network topology

assigned by the administrator. Their own circle is always blue and shown in the bottom, while the others are shown in orange.



(a) Screen of player 1

(b) Screen of player 2

(c) Screen of player 3

Figure 23: According to the network topology, different players have a different visual perspective of the game

When the trial ends, each player returns to the initial screen with the game menu.

4 Data saving

For each trial, the application saves data on the server's machine. In particular it creates a text file with two column vectors: the former contains the time samples (in *ms*), while the latter contains the sampled positions (in *dm*). In the case in which a signature is recorded for a human player, one more column containing the sampled velocity (in *dm/s*) is added.
The .txt file created by the application does not contain the number of the current trial, so data can be saved correctly only by running on the Server's machine the provided shell script, named *saveData*, at the end of each trial. The script appropriately renames all .txt files of the current trial, and when necessary it also creates a structure of folders.

We provide the shell script both for OS machines (*saveDate.sh*) and Windows machines (*saveData.bat*).
On Windows, you can run the batch script just by writing the name of the script *saveData.bat* on command line.
On Mac, you can run the bash script with the following command lines:

chmod +x saveData.sh
saveData.sh

The name of the files has this structure: $PN_p\_0N_t\_Z\_1d$ where

- N_p is the total number of players involved in the trial;
- N_t is an integer index identifying the trial;
- Z is a uniquely identifier the players involved in the trial.

In the case of *Solo experiments*, a further parameter is added to identify the kind of motion (sinusoidal or free) performed by the human player.

For instance:

1. in *Solo experiments*, *P1_03_Sample_free_1d.txt* refers to the third trial of a player called *Sample* who performed a free motion in isolation (*sinusoidal* instead of *free* if the motion performed was a sine wave);
2. in *Dyadic interaction (Leader-Follower condition)*, *P2_03_L_1d.txt* refers to the third trial of the leader (F instead of L for the follower);
3. in *Dyadic interaction (Joint improvisation condition)*, *P2_03_JI1_1d.txt* refers to the third trial of the player identified with index *1* (*JI2* instead of *JI1* for the other player);
4. in *Group interaction*, *P5_02_4_1d.txt* refers to the second trial of the player identified with index *4* in a network of *5* participants.

The sampling frequency is set inside the code to 10 Hz, so according to the duration of the trial you have around *numSamples = time*frequency*. The exact number of samples depends on the performance of the machine where the application runs.