POLITECNICO
DI TORINO

System Design Project - Securing Robots and Exoskeletons

LaserBot Battle - User Manual

Cirici	Stefano	s243942
Panini	Francesco	s242547
Ledda	Luca	s237447
Spigarelli	Edoardo	s241486

Academic Year 2017-2018
June 17, 2018

Contents

1 Introduction

This document contains all essential information for the user to make full use of the application, including a description of the system functions, capabilities and alternate modes of operation, and step-by-step procedures for system access and use.

The project scenario consists in a laser battle involving robots remotely driven from a client browser, which have to move and shoot in order to survive. A robot is composed by a Raspberry connected to an Arduino, whose goal is to manage IR lasers, sensors and stepper motors attached to its chassis.

2 Overview

In order to setup the game you have to first configure each robot (section 4) and the server (section 5). After that, power on the robots and launch the docker image on the server pc (section 5.2).

The application will startup and robots will automatically connect to the network.

From now on, every device connected to the wireless network can access the web page http://laser_bot_master.local:5000 and play the game (see section 6 for consulting game instructions).

3 Docker

All dependencies needed to make the robot and server working are provided in two separate Docker images. To be able to run these images, the Docker platform must be installed on the Raspberries and on the server. This section explains step-by-step how Docker could be installed.

Docker installation is needed for section 4 (robot environment configuration) and section 5 (server environment configuration). The complete guide can be found at the Docker website.

3.1 Install

The following commands must be launched on a shell.
To automatically install Docker CE:

```
$ curl -fsSL get.docker.com | sudo sh
```

In order to use Docker as a non-root user, it is necessary to add the user to the "docker" group issuing:

```
$ sudo usermod -aG docker $USER
```

Finally, to make operative this change, a log-out and log-in is needed.

To check if the installation has terminated without errors, we can issue:

```
$ docker --version
```

This will return the current installed version of Docker (e.g. `Docker version 18.05.0-ce, build f150324`).

3.2 Uninstall

For uninstalling the Docker CE package:

```
$ sudo apt-get purge docker-ce
```

Images, containers, volumes, or customized configuration files on the host are not automatically removed. To remove them run:

```
$ sudo rm -rf /var/lib/docker
```

4 Robot Environment

In this section all the steps needed to configure the robot are covered.
Each robot is composed of three main parts connected together: Raspberry, Arduino and sensors-actuators. Detailed information for each part will be provided in the related subsection.

4.1 Arduino

The application is distributed to work on an Arduino Uno, Arduino nano or Arduino Mega boards. Other Arduino boards have not been tested but may be however working.

It is required to have Arduino IDE installed on a working PC to be able to properly flash Arduino with the application sketch. Use the link provided to download and install it before proceeding with the following configuration.

4.1.1 Libraries

Some libraries are required to be included in the sketch. Download the AccelStepper library, the Timer library and the ros_lib library.
Go to "Sketch" → "Include Library" → "Add .ZIP Library..." and select the downloaded zip files.

4.1.2 Flash the firmware

Download the sketch with a right click on this link and select "save as" [1].
In order to correctly flash Arduino the following steps must be performed:

1. Connect the Arduino to the pc and go to "Tools" → "Board" and select the correct model of the Arduino in use
2. Go to "Tools" → "Port" and check if `/dev/ttyACM#` or `/dev/ttyUSB#` is present in the available ports. If not, there could be a problem with installed USB driver [2].
3. Open the provided sketch `"rosserial_node_arduino.ino"`
4. Use the Upload button to deploy the sketch to Arduino

4.1.3 Connections

Once Arduino has been properly programmed, it is necessary to make all the necessary connections between the involved components. The connections proposed in this section are coherent with the proposed arduino sketch and project architecture, but they have not been tested since all the hw components were not available yet.

[1] If this option does not shows up, simply click the link. It will open the source file in a browser. To download it right click on the background and select save as.
[2] Check the Arduino guide for further information

4.1.3.1 Connection to Raspberry

Connecting Arduino to Raspberry is done by mean of USB cable. This will enable serial communication between them, such that all command messages from the master will reach Arduino, but also sensor data messages will arrive to the master.

4.1.3.2 Connection of motors

Each stepper motor (right and left) has four Arduino pins reserved, as shown in Figure 1. The proper connection involves these steps:

1. Connect, following increasing order, the four reserved pins of Arduino from the IN1 to IN4, by mean of four wires. Respectively for the left and right motors, the connections would be: 8-9-10-11 to IN1 IN2 IN3 IN4 (orange pins in Figure 1) and 4-5-6-7 to IN1 IN2 IN3 IN4 (purple pins in Figure 1)
2. Connect the flat cable of the motor to the respective socket in the driver
3. Choose the intended power supply voltage to be used (5V - 12V), then in the driver board move the jumper in the respective position and connect the power supply, following positive and negative polarity. Refer to Figure 2, 1 and 2 highlight the stepper dedicated pins

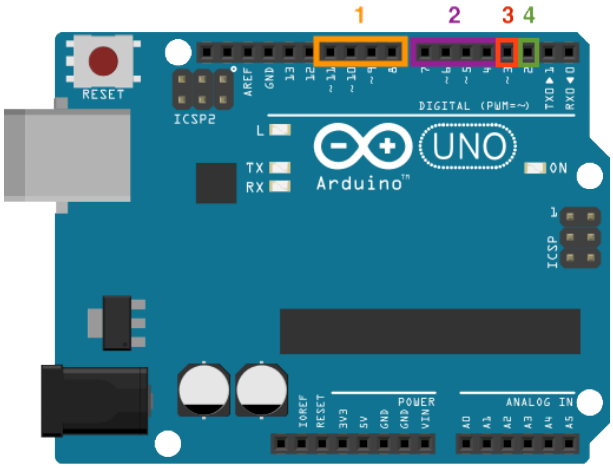


Figure 1: Arduino pins

Figure 2: Stepper motor and driver

Connection of IR Transmitter

Connect the output (terminal 2 in Figure 3) of the IR transmitter to pin 3 of arduino (red pin in Figure 1) and the ground (terminal 1 in Figure 3) to a GND pin of arduino (see Figure 1).

Figure 3: IR Transmitter

Connection of IR Receiver

Connect the output (terminal 1 in Figure 4) of the IR transmitter to pin 2 of arduino (green pin in Figure 1) the ground (terminal 2 in Figure 4) to a GND pin of arduino (see Figure 1) and the power (terminal 3 in Figure 4) to Vdd (a 5v pin in Figure 1).

Figure 4: IR Receiver

4.2 Raspberry

This installation procedure must be followed only once. The application will be updated, if needed, on its startup.

In order to make Raspberry properly operate the following steps must be performed :

1. Rasbian OS : It is required the Rasbian operating system up and running on the Raspberry. Use the link provided to download and install it before proceeding with the following configuration.
2. Configure the Raspberry to automatically connect to a wireless network following this guide. The network must be the same on which all Raspberries and the Server will be connected on game start.
3. The dependencies needed to make the robot working are provided in a Docker image. Follow Docker installation guide provided in section 3.

To automate the application launch on Raspberry startup, the following command can be executed:

```
$ echo "sudo docker-compose run robot" >> /etc/rc.local
```

On Raspberry startup the script will run: if the application image has not been downloaded before or has been updated since the last launch, the new version will be downloaded.

5 Server

In this section description, of how to install and how to launch the server are reported. The server will run on a pc with Ubuntu.

5.1 Install

The dependencies needed to make the server working are provided in a Docker image. Follow Docker installation guide provided in section 3, before proceeding.

5.2 Launch

Now that Docker is installed, the Docker image to launch the Server can be run issuing:

```
$ sudo docker-compose run server
```

After few seconds (in which the application will auto.update and setup), if no error arises, the Server application is now up.

6 Application

This section provides a description of system functionalities.
It is important to notice that previous setup steps as section 5.2 (for server to be up), section 4.2 (for raspberry to be up), section 4.1.3 (for arduino to be connected to raspberry) and section 4.1.2 (for arduino to proper handle robot hardware) should be performed in order for the user to access to a working environment.

6.1 Access

User can access to the application services through the Home web page.
From a browser connected to the same wifi network from which the server is connected to, go to the following address: `laser_bot_master.local:5000`. The following page should be displayed.

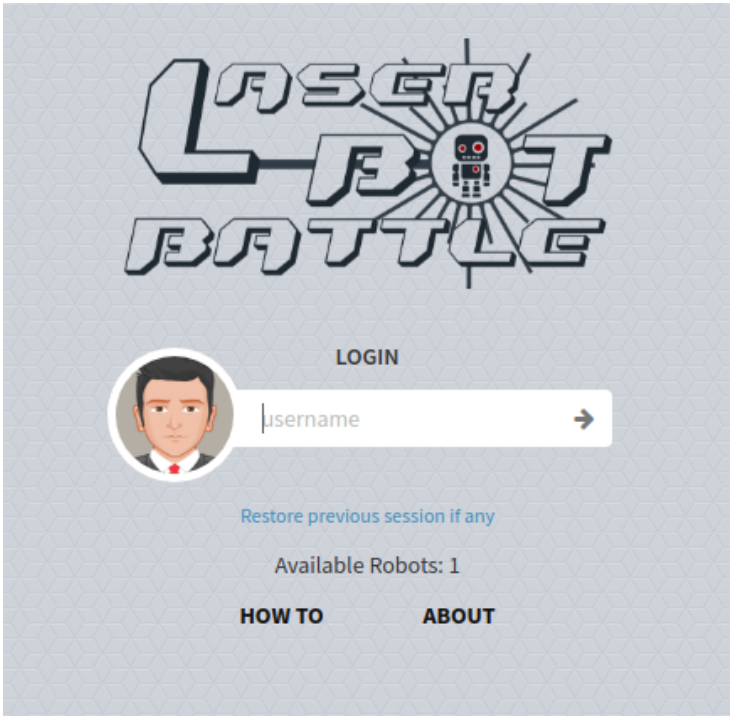


Figure 5: Index page - View

From this page, "HOW TO" alert and "ABOUT" page can be consulted.
The "HOW TO" alert reports information related to how to play the game, as the following image shows :

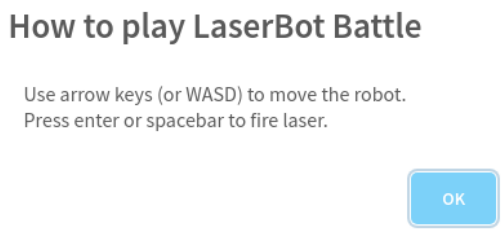


Figure 6: "How to" alert - View

The "ABOUT" page involves the team description, aims and motivation of the project.

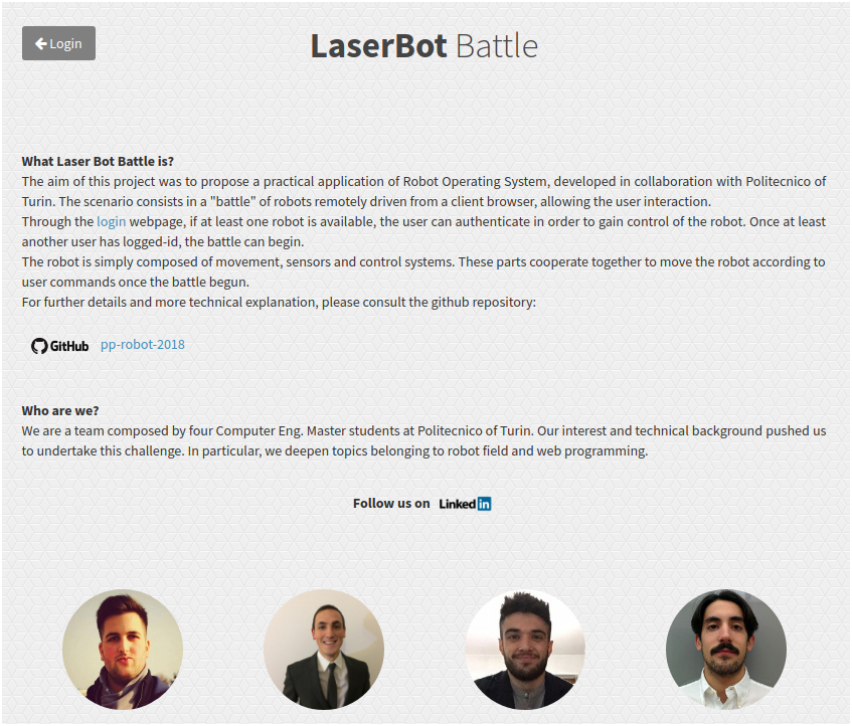
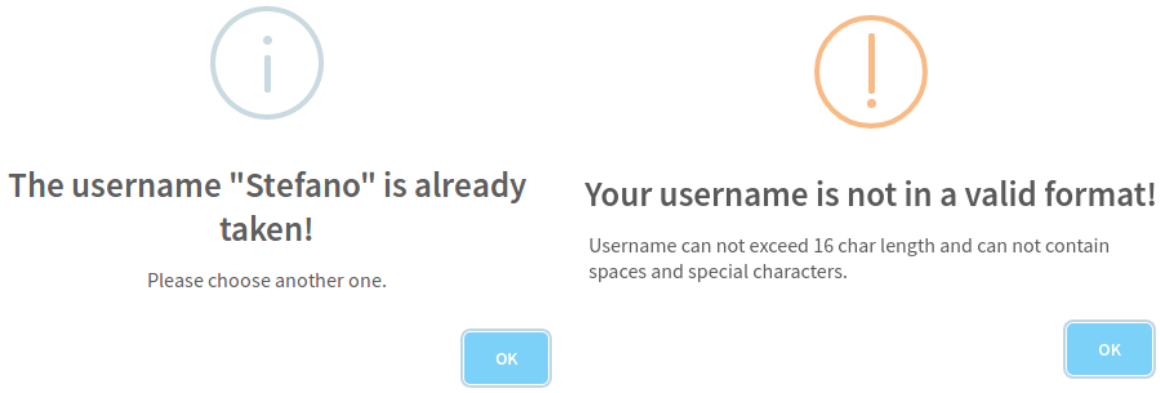


Figure 7: About page - View

In addition, the number of "Available Robots" notifies to the user if there is the possibility to login to the application (section 6.2).

6.2 Login

User can login to the application after accessing the index web page (section 6.1).
If the number of "Available Robots" is greater than 0, the user can login to the page after having chosen a username and an avatar. Username has to be filled in the "username" form while the avatar is chosen by clicking on the "avatar" image. Respective errors as validation or uniqueness of the username are displayed on the page (in form of pop-up messages), as for example the following ones:



(a) Login - Uniqueness error

(b) Login - Validation error

Figure 8: Pop-up login errors

If the login is performed without any errors, a robot is associated to the user and the Home web page, from which the user can be involved in a battle, should be get (section 6.3).
Moreover it is possible from this page to restore previous sessions by clicking on the respective link. If the user has already logged in without making any sign out, a session restoring is performed. It consists in a redirection to the Home page with all previous login information restored (as username and avatar).

6.3 Home

The Home is the page which is get after login.

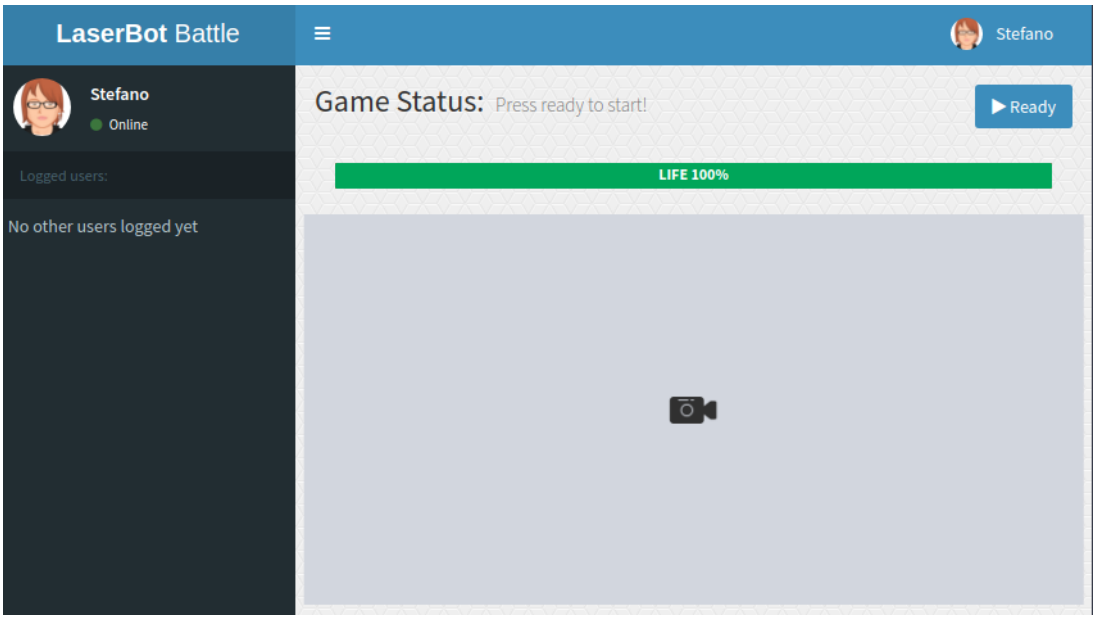


Figure 9: Home page - View

This is a personal page from which the user can logout, initiate a battle, consult users informa-

tion and watch the streaming video coming from the associated robot camera.

6.4 Sign Out

By clicking on the right-top icon, a drop down menu appears, as the following one :

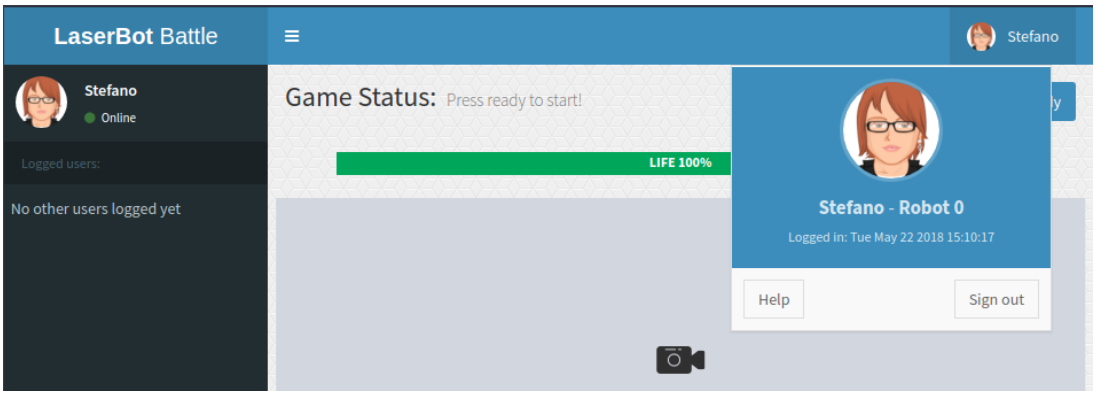


Figure 10: Personal menu - Home

From this menu the sign out action can be performed. After sign out, the user is logged out and redirected to the index page (Figure 5). Moreover, the "HOW TO" alert can be opened (Figure 6) to consult game instructions.

6.5 Users information

Users information can be retrieved on the left dark column.

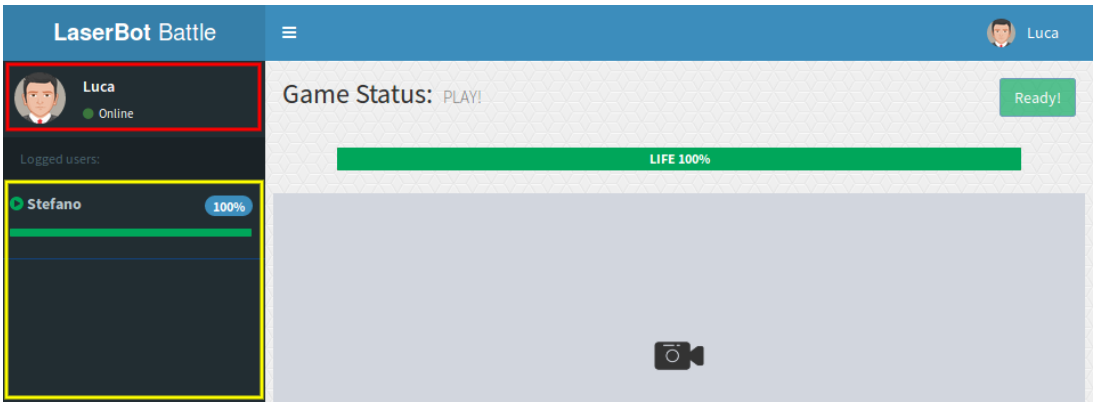


Figure 11: Users information - Home

Information reported are :

- **Personal info** (red shape on Figure 11) : At the top, personal user info as "username", avatar and status are reported.
- **Users list** (yellow shape on Figure 11) : Other users are listed here. For any of them the following attributes are reported :

- **username** : Username of the user.
- **status** : Status of the user
  - `Ready` : waiting for a battle to start.
  - `Online` : just logged in.
  - `Fight` : involved in a battle.
- **life percentage** : percentage expressing the remaining life.
- **life bar** : bar expressing the remaining life.

6.6 Game

Game information can be retrieved on the right side column.

The top space is reserved to show the game status (red outline in Figure 12).

On the right side the user can press the Ready button (green outline in Figure 12) to set his status to ready (or from ready to unready).

Under it there is the user life bar. Each user will start with 100% of health and each shoot received from another robot will decrease the hit robot health. When a user robot is dead (its life is at at 0%) the user has lost the game.

At the end of the game, when all robot but one are dead, a pop-up will appear showing the current user position and global rank. Dismissing this message, an user can begin another game setting again his status to ready.

6.6.1 Starting the game

If at least another user is logged (and ready), a 15 second countdown will start (see Figure 12). In this time other users can join the game but currently ready users can't change their status anymore.

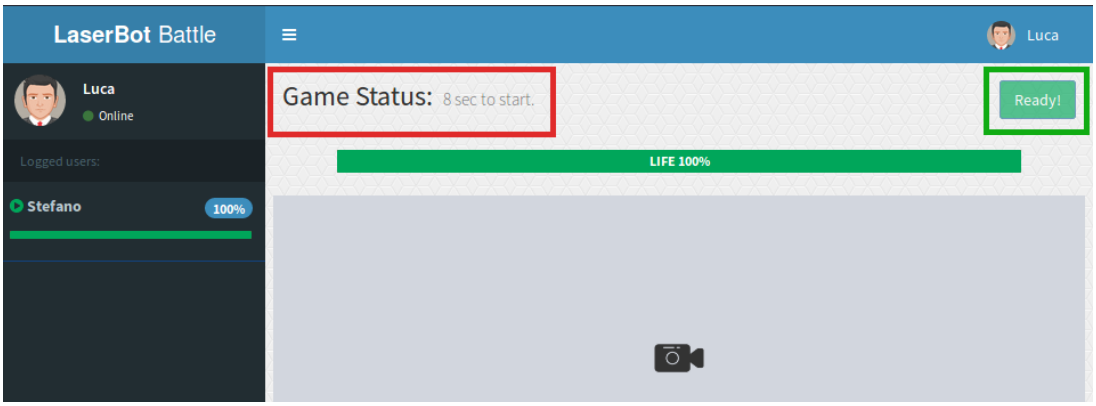


Figure 12: Countdown - Home

Once the game is started, the "Game Status" will change to "PLAY!" as the Figure 13 and the users will be able to control and move their respective robots.

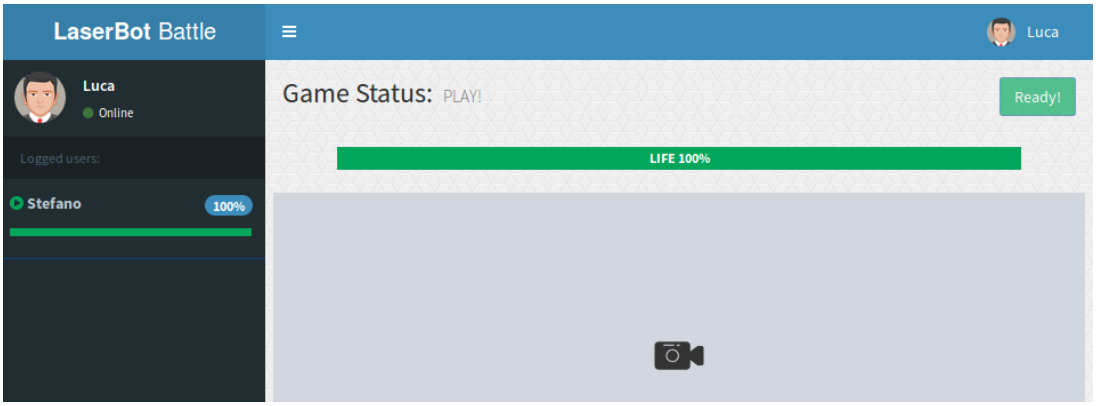


Figure 13: Play - Home

6.6.2 Controlling the robot

The user associated robot can be moved only during the PLAY status of the game.

To control the robot movements the user can use both the directional arrow on his keyboard or the `WASD` keys to respectively move the robot forward, left, backwards and right. Combination of keys (e.g. `W+A/D` or `S+A/D`) are allowed, permitting the robot to move in the resulting direction. Conflicting combinations of keys (e.g. `W+S` or `A+D`) result in no movement of the robot.

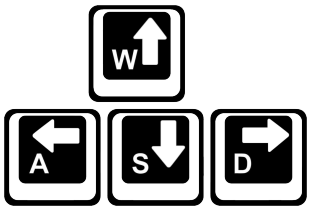


Figure 14: Robot controls - Home

To shoot, simply press the `SPACEBAR` or the `ENTER` keys.

6.6.3 Viewing the robot

The robot web cam stream will start automatically on the user login and will be displayed under the user life bar.