
User Guide: Representation and relative positioning from visual information

Submitted by:
ASMA BRAZI

Supervised by:
CÉDRIC HERPSON

Laboratory of Computer Sciences, Paris 6
Sorbonne University - Faculty of Sciences and Engineering

June - July 2019



Contents

1	Introduction	2
2	Getting started	3
2.1	Hardware prerequisites	3
2.2	OS/Software prerequisites	4
2.3	Environment configuration	4
2.3.1	Wireless configuration	4
3	Launching the application	6

Chapter 1

Introduction

This document represents a guide for users. It is necessary to be consulted to ensure a proper configuration of the environment. It is designed to:

- Enumerate the different hardware and software prerequisites.
- Provide a step-by-step instructions to establish the environment configuration.
- Present a brief scenario of how the application runs.

Chapter 2

Getting started

We present in this section the necessary equipment for the use of our application. Also, we specify the essential programs to install.

2.1 Hardware prerequisites

The global system can be seen as two independent physical subsystems in interaction. The first one is composed of a computer, and the second one is what we call the autonomous robot. and whose list of components is showed in the figure below.

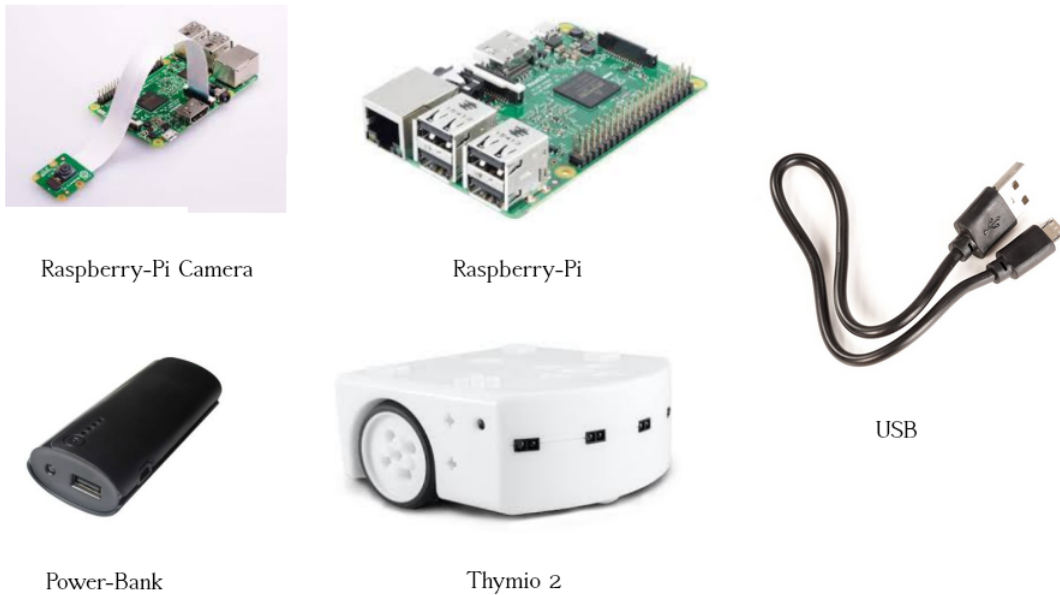


Figure 2.1: The components of the autonomous robot

The computer and the autonomous robot are exchanging via Wi-Fi. Nevertheless, the autonomous robot's components are connecting with USB. So, we have to connect the Raspberry-Pi Camera by inserting the cable into the Raspberry-Pi. In the same way, we do so with the Thymio. After that and in order to power the Raspberry-Pi, we use the power-bank which is connected to it with USB.

2.2 OS/Software prerequisites

We need several programs for both subsystems to be able to launch the application. The programs that we have to install on the Raspberry-Pi are:

- OpenCV 2.
- Python 3.
- Aseamedulla.

Then, those to install on the computer are:

- Java 8.
- JMonkey Engine.
- Java IDE.

2.3 Environment configuration

We chose to establish a connection through Wi-Fi rather than Ethernet, because the autonomous robot have to move in its environment. If the environment is large, even a very long Ethernet cable could not be enough.

2.3.1 Wireless configuration

As mentioned above, the computer and the Raspberry-Pi must be connected on the same network. In the case of the computer, it is easy to establish the connection, by selecting the name of the desired Wi-Fi network. Nonetheless, connecting the Raspberry-Pi to the network does not seem to be that straightforward.

In the case of the Raspberry-Pi, we have to assign an IP address statically to it, so that this IP address is available. In this way, every time the Raspberry-Pi starts, it connects to the router automatically. This is why we opted to fix the IP address.

So, we can assign the IP address of the Raspberry-Pi via SSH, or by connecting it to a screen and a keyboard at least. Once we can manipulate the Raspberry-Pi, we should perform the following steps:

- Open a terminal on the Raspberry-Pi.
- Execute the command: *sudo nano /etc/wpa_supplicant/wpa_supplicant.conf*, in order to modify the file **wpa_supplicant.conf**
- At the end of the current file, specify the router's id and password as follows:

```
network=
{
ssid=network_id
psk=network_password
}
```

- Save and close the file by typing CTRL-X, then Y.
- Load the file to establish the connection with the command:
`wpa_supplicant -iwlan0 -c /etc/wpa_supplicant.conf & dhcpcd wlan0`

At this stage, the Raspberry-Pi is connected to the network. To retrieve its address IP, the command to execute on a terminal is: `hostname -I`

After that, we need to set the IP address attributed to the Raspberry-Pi on the file **config.properties**. We can find this file in the root of the application's directory. Some other attributes may be set like: **RPi_id** and **RPi_password** (To set the id and the password of the Raspberry-Pi).

Chapter 3

Launching the application

After having succeeded the configuration, the application is now ready to be launched. Being on a JAVA IDE, the main class to run bears the name of *Principal.java* whose complete path is: *robot-explo/Code/src/explorator/Principal.java*

The launching of the application opens a JME window. In this window, we can see a representation of the environment explored by the autonomous robot in real-time. We show in the figure below an instance of the execution of the application.

At the beginning, the application will ask the user to enter the name of the target object from the database. We remind that the target object is the object which the robot will try to find in its environment.

After that, the autonomous robot explores its environment, builds it and try to recognize the target.

Finally, the robot indicates whether it finds the target or not.

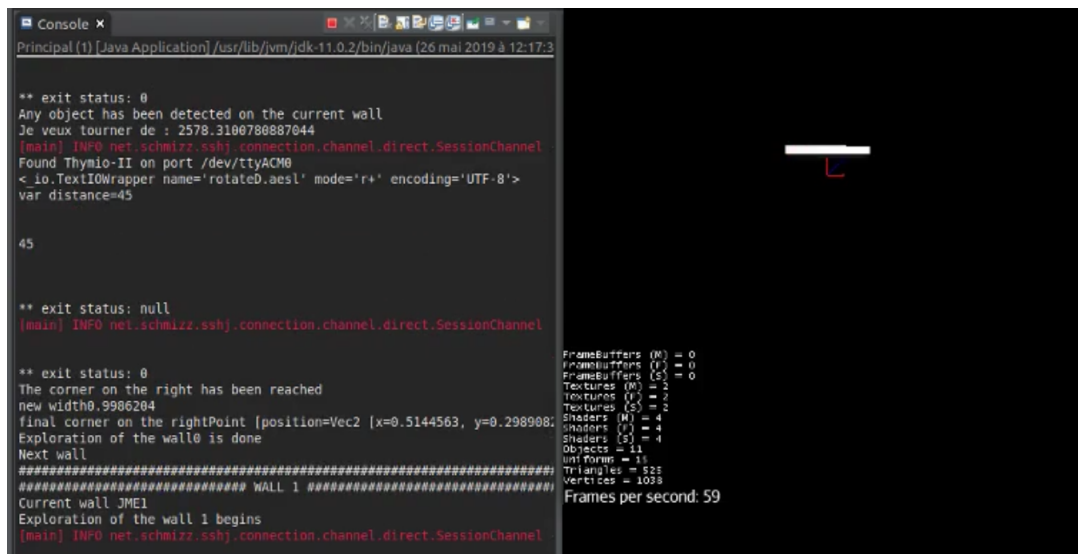


Figure 3.1: An instance of the robot's exploration