Sprint 1
Review

Yankee Doodle Pigeon

BURTON Nidishlall,
CHOUIYA Asmae,
EL HACHIMI Asmae,
MARTY Axel,
PIQUES Nicolas,
RAMIARA Maxime

INSA TOULOUSE

Contents

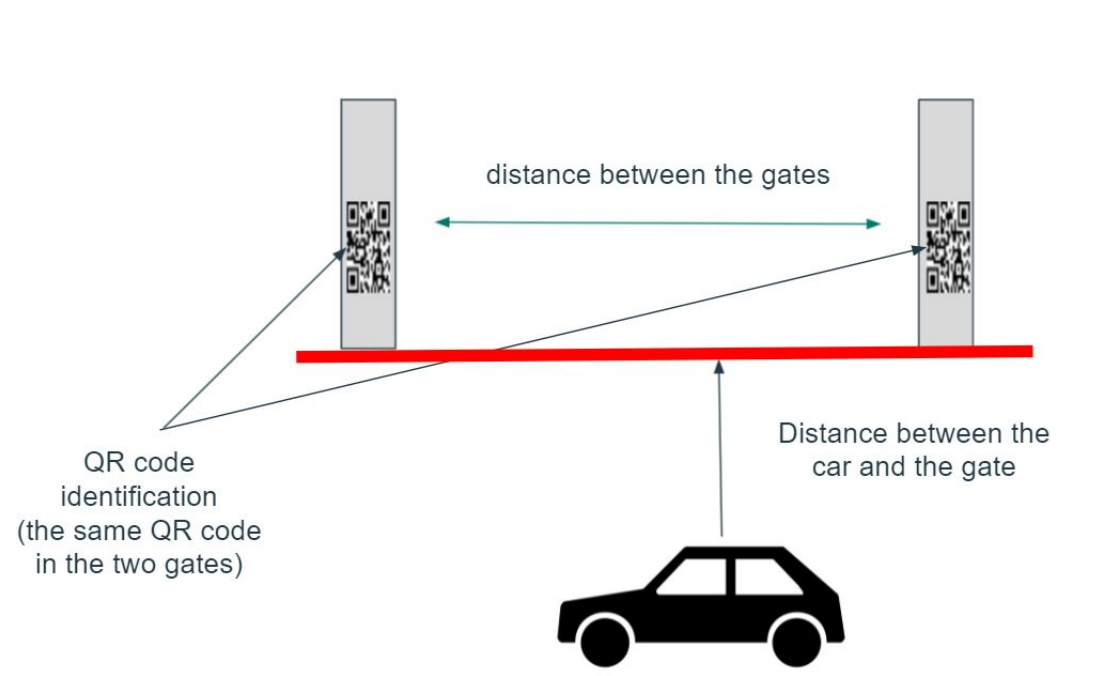
Reminder of the objectives of the project

The **autonomous** car is designed for storage companies. It will **transport goods** to its corresponding warehouses.

Each aisle will be identified by **a gate.** The car will recognize the right aisle by detecting the **gate's QR code.**

The car will be equipped with a **GUI in order control it** in case of emergency.

Reminder of the objectives of the project



Sprint 1

Reminder of the sprint 1 ’s objectives

- Create a graphical interface to control the vehicle manually :
  -> a simple interface containing movement buttons (Walk, stop, turn right, turn left, AU)

- Establish the WIFI connection between the interface and the vehicle:
  - ensure the WIFI connection with the Raspberry Pi
  - send data from the interface (phone) and be able to retrieve it on the Raspberry Pi

- QR code detection:
  - detect markers
  - specify the size of the QR code
  - set the average distance between camera and QR code
  - camera calibration

- Bibliographical research on the calculation of the trajectory

Project organization

- QR code detection:
  - Asmae El Hachimi
  - Maxime Ramiara
  - Asma Chouiya

- WIFI connection :
  - Nidishlall Burton
  - Asma Chouiya

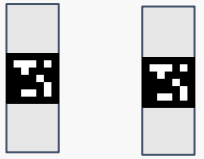
- Graphical interface :
  - Nicolas Piques
  - Axel Marty

- Bibliographical research:
  - Nidishlall Burton

QR Code detection

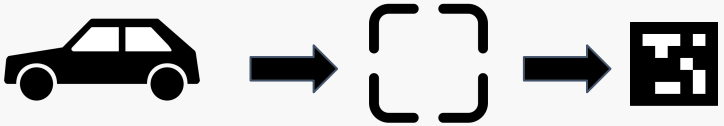
1) **Context and objectives**
2) Camera calibration
3) Tracking
4) Tests
5) Demonstration

Context :



- Gates marked with **QR codes**
- Tell if it's the **right way or not**
- Choice of a "correct" QR Code -> pass through the gate associated

Objectives :



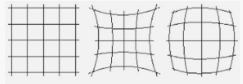
- Detect it **as early as possible**
- Recognize **each ID** without errors

QR Code detection

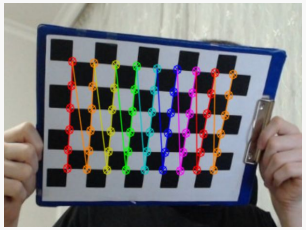
1) Context and objectives
2) **Camera calibration**
3) **Tracking**
4) Tests
5) Demonstration

Tools used : **OpenCV** and **Aruco** libraries

Camera calibration :



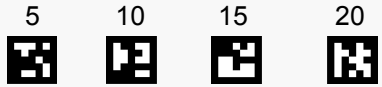
- Avoid **distortion effect**
- Make it able to **detect corners**
- Get **matrix coefficients** from the calibration



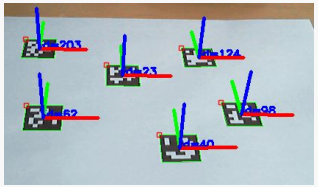
Calibration on the chessboard

Tracking :

Examples of marker IDs



- Choice of a **Aruco dictionary**
- Open camera -> tracking of Aruco codes
- Detection of **corners**
- **Get IDs** of detected Aruco codes



Tracking of Aruco markers

QR Code detection

1) Context and objectives
2) Camera calibration
3) Tracking
4) **Tests**
5) **Demonstration**

Demonstration :

- Still a **problem of Aruco library** on Nano Jetson -> demo on computer
- **External camera** connected by USB
- Calibration -> camera coefficients
- **Choice** of a gate
- **Tracking** and differentiation

Acceptance tests :

- Test 1 : Distance of **2 m from the QR code**
- Test 2 : Delay of detection **less than 500 ms**

Manual control using a GUI

1) **Context and objectives**
2) Design of the GUI
3) Communication between the GUI and the car
4) Tests
5) Demonstration of the feature

Context :

- The customer will have a manual control of the vehicle in case of emergency.
- This manual control will be available by using a GUI.
- GUI will be used for our trajectory tests.

Objectives (at sprint 0) :

- Create a simple Graphical User Interface with direction buttons and an emergency button.
- Establish a Wi-Fi connection between the car using the Raspberry Pi and the GUI. Send data from the GUI to the car.
- Process incoming data to translate orders into mechanical movement.

Manual control using a GUI

1) Context and objectives
2) **Design of the GUI**
3) Communication between the GUI and the car
4) Tests
5) Demonstration of the feature

How to develop a Graphical User Interface ?

Tool used :

The Integrated Development Environment QT Creator using C++ language

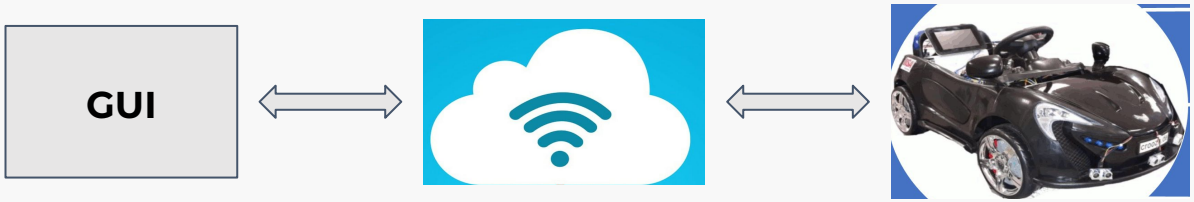
Method :

- Create a window where buttons will be shown
- Create button instances for each direction of the car
- Create an emergency button
- Associate each button to an instruction represented by a prompt command

Manual control using a GUI

1) Context and objectives
2) Design of the GUI
3) **Communication between the GUI and the car**
4) Tests
5) Demonstration of the feature

Wi-Fi Communication



MQTT:

- Client Server publish/subscribe messaging transport protocol

Mosquitto:

- MQTT server open source (broker)
- Device to publish and subscribe to one another

Manual control using a GUI

1) Context and objectives
2) Design of the GUI
3) Communication between the GUI and the car
4) **Tests**
5) **Demonstration of the feature**

Demonstration :

- Run **QT Creator** for the computer
- **Subscribe** to the server from the Raspberry Pi and put the answers into a **text file**
- Click on different buttons on the **GUI**
- Check if the text file is **updated**

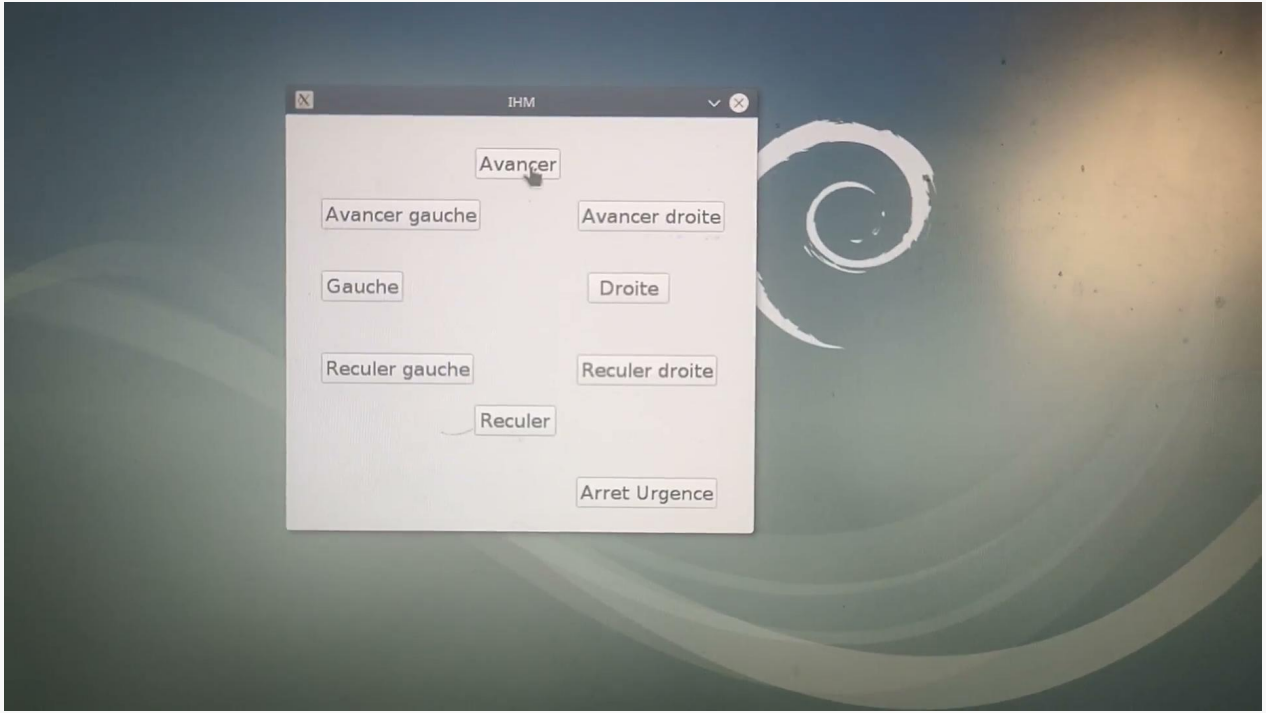
Acceptance tests :

- Test 1 : remote manual control with a range of **0 to 10 m**
- Test 2 : response time **< 1 s**

Manual control using a GUI

1) Context and objectives
2) Design of the GUI
3) Communication between the GUI and the car
4) **Tests**
5) **Demonstration of the feature**

Demonstration :



Sprint results

Sprint successes :

- Establishment of the QR code detection
- The design of the graphical interface
- Establishment of the WIFI connection between the vehicle and the GUI

Improvement for next sprint :

- Implement the QR code detection into the JETSON
- Communicate between the Rasberry and the Nucleo to have a mechanic control

Sprint 2

Sprint 2

1) **Sprint 2 ’s objectives**
2) **The planned tasks**
3) Acceptance tests
4) Demonstrations planification

Objectives :

- Moving the car forward on a simple trajectory after detecting the corresponding QR code
- Gate identification using camera
- Gate detection using a Lidar
- Communication between the Rasberry and the Nucleo

Tasks :

- Identification of a gate with 2 identical QR Codes (Maxime Ramiara, Asmae El Hachimi)
- Detection of the distance between the car and the gate (Nidishlall Burton, Nicolas Piques)
- Move in a straight line until an identified gate is detected (Axel Marty, Asma Chouiya)

Sprint 2

1) Sprint 2 ’s objectives
2) The planned tasks
3) **Acceptance tests**
4) Demonstrations planification

- Calculation of the trajectory to follow -> cross the gate:
    - Delay < 500 ms to calculate the trajectory, middle of the car aligned with the center of the gate, 5 cm tolerance
- Identification of a gate:
    - Delay < 500 ms
- Manual control using a GUI
    - Command response time < 1 second

Sprint 2

1) Sprint 2 ’s objectives
2) The planned tasks
3) Acceptance tests
4) **Demonstrations planification**

- Show that the camera identifies a gate using its QR code.
- Show that our program returns the value of the distance between the car and the gate.
- Show that the car is able to go in between the gate
- Control the car from a distance with graphical interface

Thanks !

Any Questions?