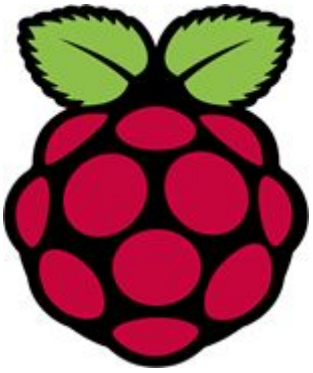
POLYTECH
MONTPELLIER

Project report

Guiding terminal for the Fabre museum



Completed by
Kévin Giordani
Maxime Soustelle

Under the direction of
David Delahaye
Vincent Berry

Academic year 2016-2017

Introduction

We've realized this project along side the Fabre museum at Montpellier. The poject aimed at designing a terminal to help visitor move around the museum. The control interface is a raspberry pi equipped with an ultrasonic sensor, by moving the hand up and down above this sensor the user can move the cursor in a selection of artwork, removing the hand will validate the currently selected artwork.

The data regarding the selected artwork are then sent to another raspberry, which will guide the user, thanks to an LED screen which will print arrows towards his selected artwork. Once the user is arrived the controller will indicate the reverse path.

1 – User guide

In order for the project to work, two raspberry pi are needed, one will be used for the selection of an artwork the other one will guide the user according to the data sent by the first rapsberry pi.

An internet connection is also required to allow the raspberrys to communicate one to the other. A classic alimentation cable is needed for one of the raspberry and an external battery for the command to allow it to move around the museum.

A GrovePi shield to plug in the ultrasonic sensor used for the selection of the artwork. To display the movement the user needs to make we also need a SenseHat which provides an LED screen a button and a gyro sensor.

On the software side we need I2C, sensehat and python3.0 installed on the raspberrys in order to use the sensors. The scripts commandeScript.exp and borneScript.exp must be filled with the IP addresses and the accesses for the pi accounts. The paths leading to an artwork must be filled in interfaceSenseHat.py. The files communications.txt must be initialized to 0. In the file mainBorne.c we must initialise distanceMax to the distance between the ultrasonic sensor and the nearest object before any user puts his hand. Compile the two files mainBorne.c and mainCommande.c and finally execute the on the corresponding raspberry.

2 – Material means

For this project we are using two raspberry pi, one for the interface and another one for the guidance of the user. We chose to use the GrovePi shield with an ultrasonic sensor for to allow the user to move in the selection menu. We also used a Sense Hat to allow the second raspbery to display the direction the user need to take, there is also a button allowing the user to get the next indication and a gyro sensor allowing the raspberry to refresh the direction of the arrow depending on the user orientation.

3 – Human means

We first looked for an idea together then we agreed to answer to the museum's offer. We then established the functional specification and the configuration of the raspberry together so we have a good base for our project.

Kevin Giordani worked on the raspberry guiding the user around the museum while Maxime Soustelle worked on the interface of selection.

We then put our work in common and worked together on the rest of the project which means creating the connection between the two raspberry.

4 – The Code

The project is divided in 7 files, two communication file and two C files, one python3.0 and two exp script.

In the file mainBorne.c we are the necessary function to the display of the user interface. “Show()” which allows to display the interface, “menuUp()” and “menuDown()” allows to move in the menu and “pause()” which allows to pause the display. There is also the function "distance ()" which returns the distance between the ultrasonic sensor and the closest object. There are also funciton to write and read in a file, "writeFile ()" and "readFile ()". The "main()" is an infinite loop that displays the menu until someone puts his hand over the sensor, once the user has his hand above the sensor, the cursor will go up or down in the display according to the current distance of the hand from its previous distance, once the hand is removed, the index of the current item in the menu is written to a file and sent via SSH to the second raspberry. The SSH connection is done through the terminal script.exp script that will run an scp command to the second raspberry so that it can read the file and react accordingly. The main then wait for a response from the command to restart its execution.

In the mainCommande.c file, we find a "script()" function that allows the execution of a python program by giving as parameter an array filled with the name of the current program, the python program, the function to be executed and the differences between them. parameters of the function. This function is taken from the python doc: https://docs.python.org/2/extending/embedding.html. As in mainBorne.c we find the functions of writing and reading in a file. The "main ()" function is also an infinite loop, it first waits for the communication file to be initialize with the index of an artwork, then it starts the main function of the interfaceSenseHat.py program using the index of the artwork as parameter. He then waits until the python program has finished executing to finally communicate via SSH with the first raspberry to tell him to restart his execution and he goes on standby.

The interfaceSenseHat.py file is used to display the arrows to the user, depending on the value the arrow points in a different direction. Paths are pre-recorded in an array of tables. A reversePath provides the paths to return to the starting point. The arrowDirection array contains the direction in which the arrow should point so that even if the user turns on his own the raspberry continues to point in the same direction. One finds the function "textScroll ()" which scrolls a given text in parameters, the function "arrow ()" which displays an arrow pointing in the direction given in parameter, "getYaw ()" returns the difference of angle between l initial orientation and the current orientation of the raspberry, "click ()" which returns 1 if the button was pressed. A function to write to a file. And the function "displayNextDoor ()" which displays the arrow in the direction given in parameter and is responsible that it always points in the same direction. The function "main ()" takes as parameter the index of the work to which it must guide. She will display the arrows one by one until the work, the user must press the button to go to the next indication, once arrived she will display the arrows for the return, once finished the function written in the communication file to allow mainCommand.c to resume execution.

5 – Prospects

The project can still be improved, we can think of integrating a gps to guide the user without having to press a button at each door. We could also add a voice interface for visually impaired people. Regarding the code, it would be necessary to modify the way the paths are recorded to allow the user to add and delete them more intuitively. In general, it would be necessary for the system to be able to take into account the stairs, the rooms having more than one door on the same side. The menu display can also be improved to make it more enjoyable. One could imagine that the command during the guidance gives voice information about the work towards the person is heading. Finally, the system could also create routes rather than just to a works.

To carry out all its improvements, it would probably be a group of two or three people working for 20h each on the project.

The marketing would be possible because it would have a real usefulness in the places like the museum however the improvements mentioned above would make it much more attractive. The market for this type of device is open, so solutions are available today to meet this need.

The price of a terminal and an order currently would be at least € 140 because of the two raspberry, SenseHat, shield, and sensors. However the costs could be reduced by using a gyroscope and a separate screen instead of the SenseHat, we could also replace the raspberry by another less expensive device. Only one terminal is necessary, it would cost minimum around 40 € (without the screen) and it could control several remote that would also cost at least around 40 € each.