Vincent Van Pepper - Your Museum Guide

"Sapienza" Università di Roma

Giovanbattista Abbate - 1875271
Mario Fiorino - 1871233

Spring Semester, 2020

Declaration

We hereby state that all the members equally contributed to the project.

Contents

Abstract

In this work, we faced the problem of guiding an English speaking visitors in a museum, who are not necessarily great art experts, and that potentially enter that museum for the first time (Section 1). In particular, we developed a solution to this problem using a Pepper robot, named *Vincent Van Pepper* to empathize with the humans interacting with it (as explained in Sections 2 and 5), inserted in an environment that recreates the Museum of Modern Arts based in New York "cut to the bone". In our experiments, visitors are welcomed by Vincent, and in Section 5 more results of these interactions have been shown. Since this work has been done at the time of COVID-19 emergency, real experiments on the Pepper robot were not available, nevertheless in Sections 3 and 4 we explained how we dealt with this problem. Finally, in Section 6, we have defined a questionnaire for the evaluation.

1 Introduction

Many times, visiting a museum, visitors know only partially, if at all, the history of it and the most important works in the museum. In order to give a solution to this project, we thought about developing a version of Pepper for museums.

Our purpose is to help the visitors, making their visit as pleasure as possible, avoiding them to be stressed or upset for the lack of information. Moreover, the interaction with Pepper can be more engaging and funnier than a simple audio guide, because of the gestures and the anthropomimetic appearance of the robot.

This leads to our idea: Pepper is placed at the entrance of the museum and will:

- Give an explanation of artworks, that is understandable even to not art experts.
- Give directions for reaching the artworks.
- Guide the new visitors in discovering history of the museum.
- Show to the new visitors a tour of the main masterpieces of the museum.

We think that this problem is important because knowledge about art cannot be lost and should not be ignored. Moreover, the design of a robot receptionist can be important because it makes the museum experience more interesting for visitors and economically advantageous for museum (with respect to hiring new human staff).

2 Related Works

To get an idea of what it meant to design a receptionist robot and what are the main problems involved(in particular the expectation of people regarding receptionist) we read the work by Trovato et al. [1, 2].

In the work by Wiese et al. "Robots As Intentional Agents: Using Neuroscientific Methods to Make Robots Appear More Social" [5] we understood how attribution of intentionality (robot is perceived as intentional agents by user) produces positive effects in human interaction: it can create a pleasant and more engaging experience (for more information, see Note in Section 5). We understood also how avoid producing the sensation of ambiguity in the user (when the agent is perceived as "human-nonhuman"). So, basically, do not fall into the uncanny valley phenomenon.

From the work "Toward Anthropomimetic Robotics: Development,Simulation, and Control of a Musculoskeletal Torso" by Wittmeier et al. [4], we better understood the concept of embodiment, together with the physical and cost limits of making an anthropomimetic robot as Pepper. These concept were useful to comprehend how and why Pepper were able to perform actions as gestures.

The final questionnaire was inspired by reading "Theory and evaluation of human robot interacion" by J. Scholtz. [3].

3 Solution

Our proposed solution to the problem is developed using the following architecture:

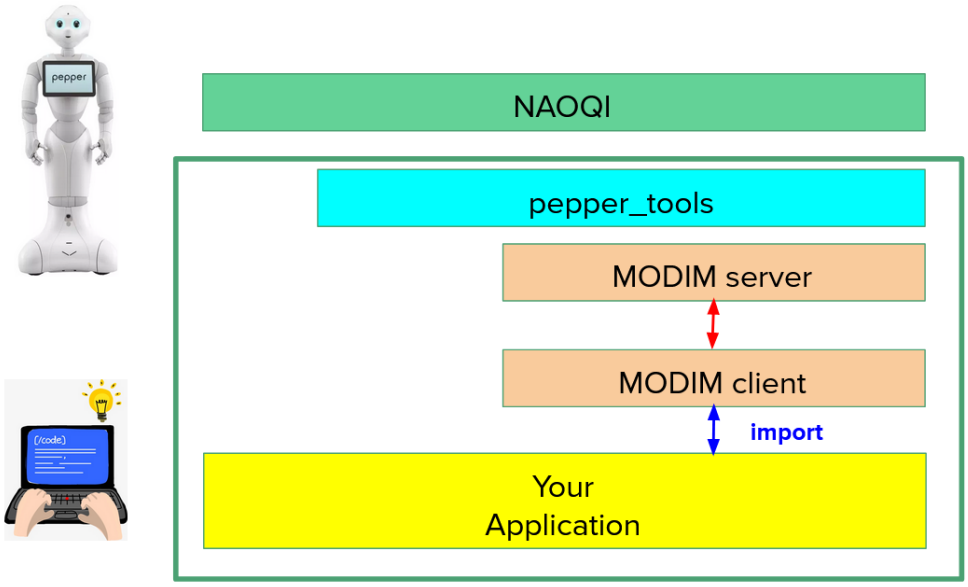

Figure 1: Architecture of the solution

The entire solution runs on a NaoQI server with Python SDK, which emulates the behaviour of Pepper, since it is not possible at the time of this project to interact with the real robot. The tablet of Pepper is emulated through MODIM, which allows to emulate it through a HTML file, that it is possible to dynamically change using a Python script. Through MODIM it is also possible to emulate on terminal most of the features of Pepper using `pepper_tools`[1], like Autonomous Speech Recognition, Text-To-Speech and Gestures. This solution has been implemented using the Docker image provided during the course[2]. More information about the solution and the implementation of it will be given in Section 4.

Unfortunately, other solutions to better emulate Pepper behaviours like Android SDK were not available for performance reasons. Moreover, the physical distance between the members of this project and the University prevented the use of a PC in the department.

[1] https://bitbucket.org/mtlazaro/pepper_tools/src/master/
[2] https://bitbucket.org/iocchi/hri_software/src/master/

4 Implementation

In this section, we are going to present how we implemented our idea presented in Sections 1 and 3. First of all, NaoQI and MODIM servers must be running, as presented in 3. To do so, two simple bash scripts have been written:

```
#!/bin/bash
cd
cd /opt/Aldebaran/naoqi-sdk-2.5.5.5-linux64
./naoqi
```

Listing 1: startNaoqi.sh

```
#!/bin/bash
cd $MODIM_HOME/src/GUI
python ws_server.py -robot pepper
```

Listing 2: startModim.sh

4.1 HTML

After starting NaoQI and MODIM, a HTML page should be opened, using the preferred browser. In this way, MODIM can have an anchor to execute the code and use it as an emulator for the Pepper tablet. During the implementation of our project, the standard HTML file provided by MODIM repository has been used. The CSS syntax in the `<style>` block has not been changed, the only changes made were to remove the `status` element to hide the connection status to the MODIM server (used only for debug purpose). All the buttons were removed, leaving only the `<div>` element to define a placeholder in the page to add buttons using MODIM actions. The most important part in the HTML file is the JavaScript script to create a link with the emulated robot in the NaoQI server, that is:

```
var ip=window.location.hostname;
if (ip=='') ip='127.0.0.1';
var port = 9100;
wsrobot_init(ip,port);
```

Listing 3: Part of index.html

4.2 MODIM Actions

To perform dynamic changes to the HTML page without writing hundreds of rows in Python, we chose to use MODIM actions. We developed 27 actions, and used the following modules to dynamically change the HTML view of the starting file during the execution of the project:

- **TTS**: Text-to-speech, unfortunately the audio is not available in the emulator, and only a text output in the terminal is produced. In the video result, we added the Ivona TTS audio during the edit of the video.
- **IMAGE**: Change the image with id `image_default`.
- **TEXT**: Change the content of the tag with id `text_default`.
- **TEXT_title**: Change the content of the element with tag `<h1>`.
- **ASR**: Start an Autonomous Speech Recognition action. Unfortunately, we would have liked to use the microphone of Pepper to perform this action, but due to COVID-19 emergency we only were able to emulate it through the module `human_say` in `pepper_tools` library. To facilitate the user in interact with Pepper, a bash script has been developed:

```
#!/bin/bash
cd ~/src/pepper_tools/asr
python human_say.py --sentence "$1"
```

Listing 4: asr.sh

- **BUTTONS**: For each line, add a button in the `<div>` with id `buttons`. Each button has an output, which is going to be useful in the Python script.
- **GESTURE**: Emulate Pepper gesture. Using NaoQI, not all the gestures are available[3], but the most important ones are supported.

Each module has an instance `<*,*,*,*>`, that is indexed as:

```
<access_grade, sex, language, role>
```

Listing 5: Instance syntax

Among these four indexes, in our project we only care about language, but we decided to focus only on English language, since it would have required a simple translation into other languages, specifying the language in the same module (for example italian) as `<*,*,it,*>` and translating every English part about text, TTS and ASR. Each module has a finisher syntax `----`. A sample syntax of MODIM actions is the following:

```
TEXT_title
<*,*,*,*>: What to write in <h1>
----
IMAGE
```

[3] http://doc.aldebaran.com/2-4/naoqi/motion/alanimationplayer-advanced.html#nao-nao-list-of-animations-available-by-default

```
<*,*,*,*>: Image to use
----
GESTURE
<*,*,*,*>: /path/to/gesture
----
TEXT
<*,*,*,*>: What to write in text_default
----
ASR
<*,*,*,*>: {'recognized_word': ['different','ways','to say it']}
----
TTS
<*,*,*,*>: Pepper say this
----
BUTTONS
output1
<*,*,*,*>: Text1
output2
<*,*,*,*>: Text2
----
```

Listing 6: MODIM action sample

4.3 Python

The core of the project lies in the Python script `start.py`. This script indeed starts the interaction with the MODIM server, and manage the behaviours of Pepper and its emulated tablet during the execution of the instructions. In the script, it is possible to see the key features of our project:

- Start the interaction with MODIM and NaoQI. Assuming that `intStart` is the main interaction of the project:

```
mws = ModimWSClient()
mws.setDemoPathAuto(__file__)
mws.run_interaction(intStart)
```

- Initialize the user through `im.init()`, where `init` is a plain text file, similar to a MODIM action, in which the user is initialized. In our case is:

```
URL: index.html
PROFILE: <*,*,en,*>
TEXT_title: HRI - Abbate | Fiorino
IMAGE: img/diaglogo.jpg
```

Listing 7: init file

- Wait the presence of a human. The human must be in front of Pepper for at least 2 seconds. This is achieved using the system library `time`. Moreover, through MODIM it is possible to use `im` command to access `pepper_tools`, therefore we managed to check the sensors through it:

```
im.robot.startSensorMonitor()
while not flagP:
  while not detected:
    p = im.robot.sensorvalue()
    detected = p[1] > 0.0 and p[1] < 1.0
  if detected:
    print('*Person Detected*')
    time.sleep(2)
    p = im.robot.sensorvalue()
    detected = p[1] > 0.0 and p[1] < 1.0
    if detected:
      print('*Person still there*')
      flagP = True
    else:
      print('*Person gone*')
im.robot.stopSensorMonitor()
```

Listing 8: Person detection inside start.py

On the user side, since real sensors of Pepper were not available, we developed a bash script to help the user emulate his presence in front of Pepper. To achieve this we used the function `sonar` from `pepper_tools`:

```
#!/bin/bash
#Usage: ./front.sh <duration> <value>
cd ~/src/pepper_tools/sonar
if [ $# -eq 0 ]
then
  #default: value=0.75, duration=3
  echo no value no duration
  python sonar_sim.py --sensor SonarFront
elif [ $# -eq 1]
then
  #default: value=0.75
  echo no value
    python sonar_sim.py --sensor SonarFront --duration "$1"
else
  echo full
  python sonar_sim.py --sensor SonarFront --value "$2" --
     duration "$1"
fi
```

Listing 9: front.sh

- Pepper asks the user if it is the first time in the museum: if it is, then Vincent Van Pepper explains the history of the MoMA, and asks if the user wants a tour. If the user wants a tour, a sequence of MODIM actions is executed through `im.execute()` to show the user a brief tour of the museum (better explained in Section 5). If it is not, or simply the user does not want the tour, then the user is redirected to the map of the museum, where he can choose an action to execute just by clicking on a button or by talking to Pepper. Some interactions are different for the user, depending on whether it is the first time he/she visits that page or not. In particular, in the page where the full map of the museum is shown, as described in Figure 2, inside the previous loop there is an if statement to check the previously introduced condition:

```
if not flag2:
  im.executeModality('TTS','<first time>')
  im.executeModality('TEXT_default','<first time>')
  flag2 = True
  else:
        im.executeModality('TEXT_default','<not first time>')
        im.executeModality('TTS','<not first time>')
```

Listing 10: Different behaviours between first time and other times

- Moreover, from the python script a while loop is started. If Pepper does not understand any word from the ASR vocabulary, the visitor is adviced of this misunderstanding, and he/she is asked to repeat what he said:

```
while not flag:
  a = im.ask('fullmap',timeout=5)
  while a=='timeout':
    im.executeModality('TTS','I did not understand, can you
       repeat please? You can also use the tablet.')
    a = im.ask('fullmap',timeout=5)
    if flag2:
      im.executeModality('TEXT_default','<not first time>')
      im.executeModality('TTS','<not first time>')
  if a=='goodbye':
    flag=True
    im.execute('goodbye')
    break
  else:
    im.execute(a)
    a2 = im.ask(a,timeout=999)
      if a2=='fullmap':
      im.execute(a2)
    else:
      im.execute(a2)
```

```
20          time.sleep(4)
21          im.execute('fullmap')
```

Listing 11: Core of the project

In this loop, better explained in 5, the user chooses an artwork, then it is requested if he/she wants to know directions for it, and in the end the tablet shows the initial map. Rows 6-8 let the user see something different if it is not the first time he/she visits that page.

- In the end, when the visitor says goodbye to Pepper, the script waits 6 seconds, then start from the beginning. In this way, more visitors can use Pepper without manually initialize it each time.

5 Results and Explanations

The main goal of the project is to create a robot receptionist to inform and help the guest in the museum, to ensure that he, or she, can have a more pleasant and engaging experience.

For greater clarity of our intent, we took for environment, in which the robot acts, the structure and artworks of a real art museum, very famous: The Museum of Modern Art (MoMA) located in Midtown Manhattan, New York, but "cut to the bone": composed of just one level and 8 artworks. Clearly, the process can be developed for multiple levels and more than 8 artworks.

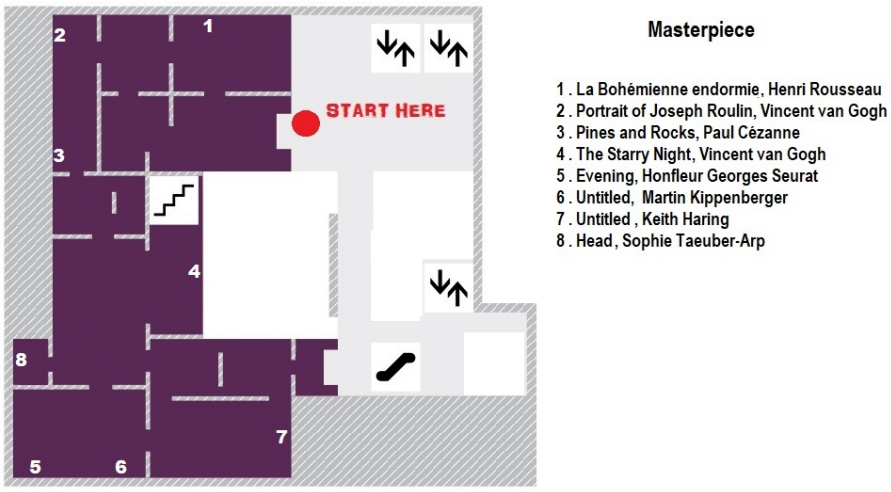

Figure 2: Setting - MoMa reduced map. The numbers in white point the position of artwork in list."Start Here" in red is the entry and the position of robot.

The robot waits the visitor at the entrance ("Start Here",on the map). When the presence of a guest is detected by the sensors, the interaction procedure begins. Note: the robot does not move from "Start Here", and does not walk around the museum with visitor. The robot assists exclusivity at the entrance; trying to inform (reception) as many users as possible there.

The diagram below summarizes the entire interaction process, and the different possible scenarios.

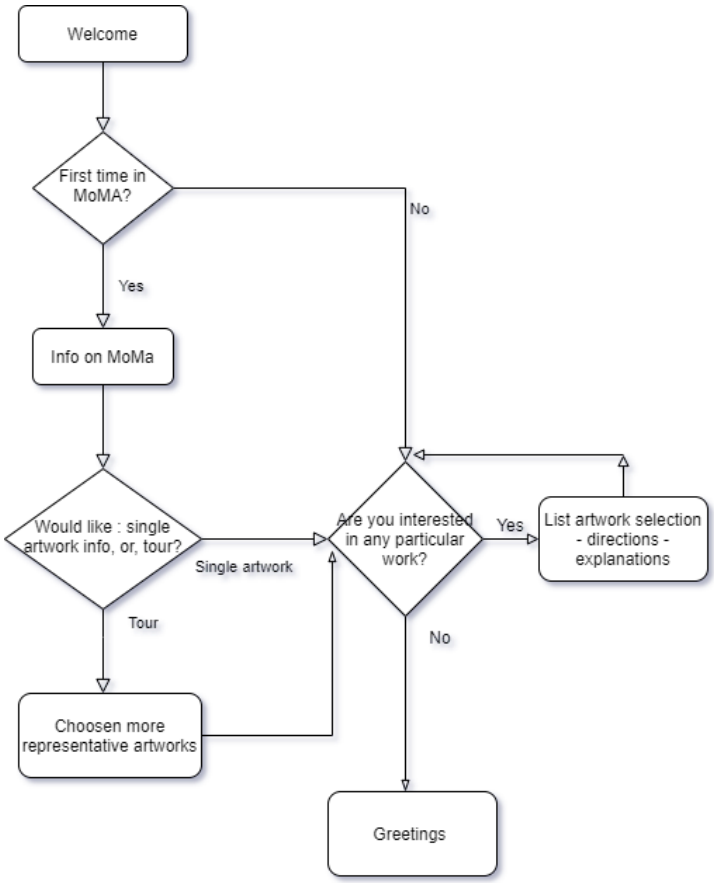

Figure 3: Flowchart of receptionist interaction process

For instance, starting from a simple scenario, where user is the first time he enters in MoMa; once the presence of user is detected, start the Welcome procedure:

Robot Speech : "*Hello! Welcome to MoMa museum!*
I'm Vincent Van Pepper your museum assistant.
I'm here to inform and help you during your visit.
You can interact with me just talking, or, if you prefer clicking the tablet"

Robot Moving : moves its arm in greeting.

Note :

1. We gave a funny name to the robot: Vincent van Pepper, aimed to create more empathy with user. The name takes inspiration from the museum's most valuable work : The Starry Night of Vincent van Gogh
2. Similar discourse for movement of robot, gives it vitality and sympathy.

Then the first interactive question begins.

Robot Speech : "*Is this your first time at MoMa?*"

Note : This helps to involve the visitor more in the visit, trying to establishing a connection with him/her

A user response is waited. He/she can choose to talk or click the button on the tablet screen (Note: all interaction procedures will be accompanied by the tablet for greater clarity). If the speech answer is different from "Yes", "No" , or takes more than 5 seconds, the robot will output: "*I did not understand. Can you repeat please?*". This latter checking procedure will repeat adequately, for each iteration.

For instance user replies: "*Yes*"

The robot briefly presents the history of the MoMa

Robot Speech : "*Fine! Let me tell you something more about the MoMa: It was founded in 1929, and was the first museum devoted to the modern era. Today MoMA's rich and varied collection offers a panoramic overview of modern...*"

Robot body movements accompany the speech trying to boost the communication.

Then, when the historical introduction is complete, the tablet shows the map of museum (Fig.2) with the list of all artworks on the right, and the robot invites to a second interaction:

Robot Speech: "*Do you want to take a tour on the main artworks, or is there anything you want see in particular?*"

That means: the user is asked if he/she is interested in a brief review of the most important works in the museum (the "top highlights"; or in case of a very large museum: the procedure Tour could show the main artworks of a certain section (e.g., Egypt, Italian artists of the 1500s,...), collection, level), or if there is one in particular he/she would like to see.

A user response is waited by talking or clicking on the tablet or talking. For instance, response: "*Tour*"

From this moment, on tablet screen, the museum's main works are shown in succession (exactly 3 works, in our context of "reduced museum"), while the Robot comments quickly on them, and, finally displays the indications to reach them. This is the output procedure:

Tablet display: The Starry Night, Van Gogh

Figure 4: The Starry Night,Van Gogh

Robot Speech : "*Fine. We will begin with Starry Night by Vincent Van Gogh. Van Gogh's night sky is a field of roiling energy. Below the exploding stars, the village is a place of quiet order. Connecting earth and sky is the flamelike cypress, a tree traditionally associated with graveyards and mourning. But death was not ominous ...*"

Robot body movements accompany the speech.
Once the explanation is finished, the instructions to reach the work appear on the screen:

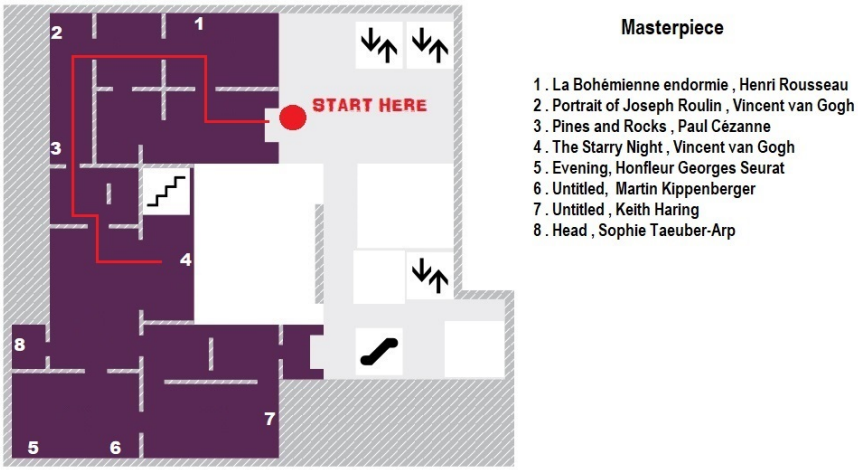

Figure 5: Map for reach The Starry Night

And he robot simply says : "*This is how to get The Starry Night*"
This is repeated for the other two works of the review; precisely, "*The second work we are going to see is The Sleeping Gypsy by Henri Rousseau...*" and "*Finally, we will see Untitled for series Lieber Maler by Martin Kippenberger...*"

Once the procedure Tour is over, the robot starts a third interaction. The tablet shows again the map of museum (Fig.2) with list of artworks on the right.

Robot Speech : "*Are you interested in any particular work?*"
A user response is waited by clicking on the tablet or talking. In this case the speech answer can be: a number, a ordinal numbers (the first, the second,...), the name of artwork. (As in all interactions, the checking procedure is applied in case of misunderstandings).

For instance, replies *"Untitled by Haring"*
The procedure starts: Tablet shows the image of artwork:

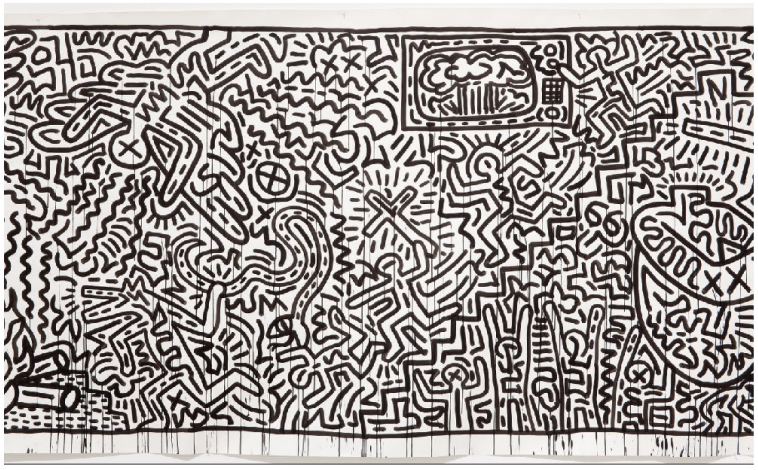

Figure 6: Untitled , Keith Haring

Robot comment on artwork: "*Cartoonish creatures were combined with highly stylized abstract patterning to create a dense, rhythmic carpet of imagery, and spaceships, hearts, crawling babies, and winged figures appeared frequently, as they do in this large, friezelike drawing....*"

Robot body movements accompany the speech.

After the explanation, the robot asks if the user is interested in how to reach the work. If the answer is positive, the map is shown

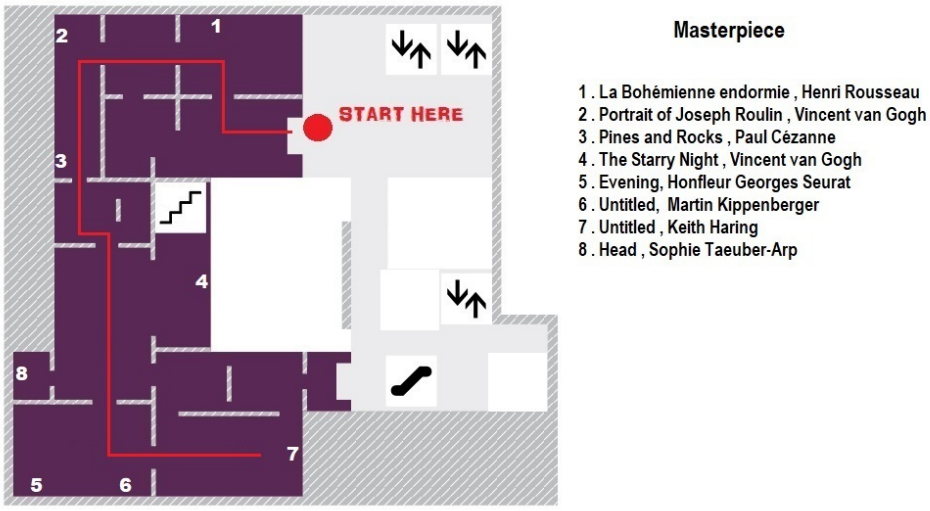

Figure 7: How to get Untitled , Keith Haring

otherwise skip this step.
Then again, the robot begins a new interaction.

Robot Speech : "*Are you interested in any other work?*"

Note: In this case, the robot does not repeat the sentence: "Are you interested in any particular work?", it could have seemed too mechanical, we tried to give it more vitality and fluidity adding in speech : "any other work?" ; as if the robot were aware of what the user has just chosen and seen. And it improve a lot the quality of interaction

A user response is waited. For instance, response "*No*"

At this point we have reached the end of the interaction, the robot wishes a good visit and greets

Robot Speech : "*Enjoy the visit! Bye!*".

The robot moves in greeting
User moves away, and the robot waits for another user.

5.1 Comments on the video

The video could be found by clicking here or on the link in the references [6]. The video is self-explanatory, since text comments on it were added during the post-editing process, but we will also comment in this section some of the behaviours:

- 0:00 Docker image is started, together with scripts for starting NaoQI and MODIM, explained in Section 3 and implemented in Scripts 1 and 2.
- 0:57 The first interaction starts using Script 9. In this interaction, the user will be a visitor that visits the MoMA for the first time, chooses a tour, than chooses work 7 and chooses to have directions for it, all of these interactions by button. Moreover, at 5:03, it is possible to see the change in the text shown on the virtual tablet with respect to the starting text at 4:13.
- 5:14 The first user walks away, and the second user walks into Pepper (always using Script 9) at 5:23. This user is an "expert" visitor, meaning that it is not his/her first time at MoMA, therefore he/she is redirected to the full map of the museum with a joke. The visitor chooses to see work 5, then refuses directions and says goodbye to Pepper. All of these interactions are made by voice and ASR, using the Script 4.
- 7:04 The second user walks away, and the third user walks into Pepper (always using Script 9) at 7:16. This user represents the visitor that visits the MoMA for the first time, refuses the tour, and then finishes the interaction without seeing anything from Pepper. All of these interactions are made by voice and ASR, using the Script 4.

6 Questionnaire

Evaluation made by the visitors:

- The robot is pleasant to use
- I was able to understand how to interact with the robot
- I think that the way to interact with the robot is simple and uncomplicated
- I was always aware of what the system was doing
- The system appeared to freeze or pause at intervals
- The robot behaved in an unexpected way
- I felt disorientated while using the robot
- The graphical interface (tablet display) is pleasant
- The graphical interface (tablet display) is useful
- I thought the voice of robot quality negatively influenced me
- I felt the receptionist had a personality
- I felt the receptionist succeed in establishing a connection with me
- It is good as receptionist
- Its behavior is appropriate to its rule.
- It gives clear (easy to get) information
- It gives useful information
- I think this robot improve the quality of my visit
- After all, I am satisfied with the service robot

7 Conclusions

We dealt with (and solved) the main problems related to the design of a robot receptionist acting in a museum; with the aim of satisfactorily informing the English speaking user about both the artworks and their position inside the museum, and making his/her visit pleasant and engaging. We have used a series of strategies to make "less robotics" the interactions with Pepper, (i.e., less mechanical and frightening) and more friendly, so as to be accepted even by a reluctant and timid audience.

Regarding the implementation part, we learned how to deal with both NaoQI and MODIM servers to handle a virtual environment for Pepper robot. Unfortunately, as stated several times during this report, due to *force majeure* we could not use the real Pepper robot, therefore we could not try real Pepper ASR, TTS and gestures, but only their virtual equivalent. Moreover, we think that a deeper study on body language moves of Pepper, as well as on the tones of the voice, a greater range of gestures available (on NaoQI virtual environment only a few of them were available) and selectable tones of voice could have made the experience of interacting with Pepper even more lively ("less robotic") and engaging. Despite these problems, we found this project very interesting and funny, and we hope to be able to interact with the real Pepper in the future.

Future works starting from this project should consider the possibility to extend the number of available languages, to extend the range of users capable of interact with Pepper. Moreover, they could think about making Pepper as a robot guide following a group of people to assist them, but we do not think this is achievable by now, considering Pepper's cost. In the end, we think that this project could be easily scalable to bigger museums.

Finally, we learned the techniques (questionnaire) to evaluate our work in a interaction with human.

References

[1] G. Trovato, J. G. Ramos, H. Azevedo, A. Moroni, S.Magossi, H. Ishii, and R. Simmons, "Designing a receptionist robot: Effect of voice and appearance on anthropomorphism," *Institute of Electrical and Electronics Engineers Inc*, 2015 November. 2

[2] G. Trovato, J. G. Ramos, H. Azevedo, A. Moroni, S.Magossi, H. Ishii, and A. Takanishi., "A receptionist robot for brazilian people: study on interaction involving illiterates.," *Paladyn, Journal of Behavioral Robotics*, 2017. 2

[3] J. Scholtz, "Theory and evaluation of human robot interacion," *Proceedings of the 36th Annual Hawaii International Conference on System Science*, 2003. 2

[4] S. Wittmeier, C. Alessandro, N. Bascarevic, K. Dalamagkidis, D. Devereux, A. Diamond, K. J. M. Jäntsch, R. Knight, H. G. Marques, P. Milosavljevic, B. Mitra, B. Svetozarevic, V. Potkonjak, R. Pfeifer, A. Knoll, and O. Holland, "Toward anthropomimetic robotics: Development,simulation, and control of a musculoskeletal torso," *Artificial Life - MIT*, vol. 19, no. 1, pp. 171–193, 2013. 2

[5] E. Wiese, G. Metta, and A. Wykowska, "Robots as intentional agents: Using neuroscientific methods to make robots appear more social," *Frontiers in Psychology*, vol. 8, art. 1663, pp. 1–18, October 2017. 2

[6] G. Abbate and M. Fiorino, "HRI project - Abbate | Fiorino." https://youtu.be/9yI6hpiFtwI. 5.1