

Human Computer Interaction Project Report

**Implementation of an HRI system
using pepper for household's
entertainment**

Contributors:

Maryam Bandali 1861486

Abdolhadi Rezaei 1837982

July 2020

Contribution

Parts related to pepper social entertaining interaction with people have been implemented by Abdolhadi Rezaei and all parts related to pepper emotional interactions have been done by Maryam Bandali. The part related to language selection and preparation of reports has been done by both.

Table of Contents

Introduction	4
Related Works	6
Solution	7
Implementation	8
Details on the implementation	8
Conclusions	18
What we have learnt:	18
Our experience in developing the project	18
Improvement	19
Link to Videos	19

Introduction

Home robots have existed since the 1990s. An early example of this is the 2001 Electrolux robot vacuum cleaner. They are currently used for helping humans with many kinds of domestic chores. In this project, we have explored and implemented one of the major home robot applications which is entertainment (toys and hobby robots).

A 2017 World Robotics report estimates that entertainment robots (toys or hobby robots) sold 2.5 million units in 2017, bringing in a revenue of \$1.1 billion. It also projects that the market size of entertainment robots reach \$7.5 billion between 2018 and 2020, at a CAGR of 20%-25%.

Two examples of AI-embedded home entertainment robots are listed below:

Anki's Cozmo

San Francisco startup ANKI's home entertainment robot Cozmo is an artificially intelligent toy. The company markets Cozmo as a "robot with a big personality and a host of emotions" in its 2016 launch report. According to the report, Cozmo's "personality" is powered by deep AI, robotics, and computer vision. Cozmo can "recognize and remember people" using a built-in OLED camera. The robot comes with an app that is compatible with Apple, android and Kindle Fire devices. It also has built-in gameplay content, which is constantly updated. The report claims that the robot's skill sets with new activities increase over time as he gets more "familiar" with a player.

Emotech Olly

Emotech's Olly is a voice-controlled assistant robot, much like Amazon Echo or Google Home. Emotech claims that this robot has an "evolving personality," meaning it reacts differently to different users. It can "understand" the users' facial expressions, vocal intonations and verbal patterns via computer vision, microphone array technology and an "emotion character engine," and proactively start conversations rather than reacting to users' commands.

It is, however, not very clear how the emotion character engine's technology works. The robot's capabilities also extend to retrieving information and interacting with other connected devices.

Olly can simulate human emotions like empathy (it appears to "see" a user resting his head on the couch and asks him, "Long day?") and predict the music the user might want to listen to according to their mood.

In this project, we get the inspiration from mentioned two hobby robots and develop our own simple robot application by using some tools and platforms like modim, choregraphe, ASR Our Scenario is as follows:

Our robot is able to interact with the households, for example if you open the door he will do greetings with you and ask some questions, for example: how are you? How was your day? And so on.

The pepper has different kinds of entertainment like playing video and music, showing some photos, quiz, dancing and telling stories. Pepper also considers disabled people in the house and can interact with them. Additionally, pepper shows its emotion in some different situations. If someone tells the pepper about his feelings, pepper reacts by suggesting some entertainment and it shows its emotion in some different situations.

Related Works

We have studied the paper entitled “Fribo: A Social Networking Robot for Increasing Social Connectedness through Sharing Daily Home Activities from Living Noise Data”. As mentioned in this paper, the rapid increase in the number of young adults living alone gives rise to a demand for the resolution of social isolation problems. Social robot technologies play a substantial role for this purpose. However, existing technologies try to solve the problem only through one-to-one interaction with robots, which in turn fails to utilize the real-world social relationships. Privacy concern is an additional issue since most social robots rely on the visual information for the interactions. To this end, In this paper, they propose ‘Fribo’, an auditory information centered social robot that recognizes a user’s activity by analyzing occupants’ living noise and shares the activity information with close friends.

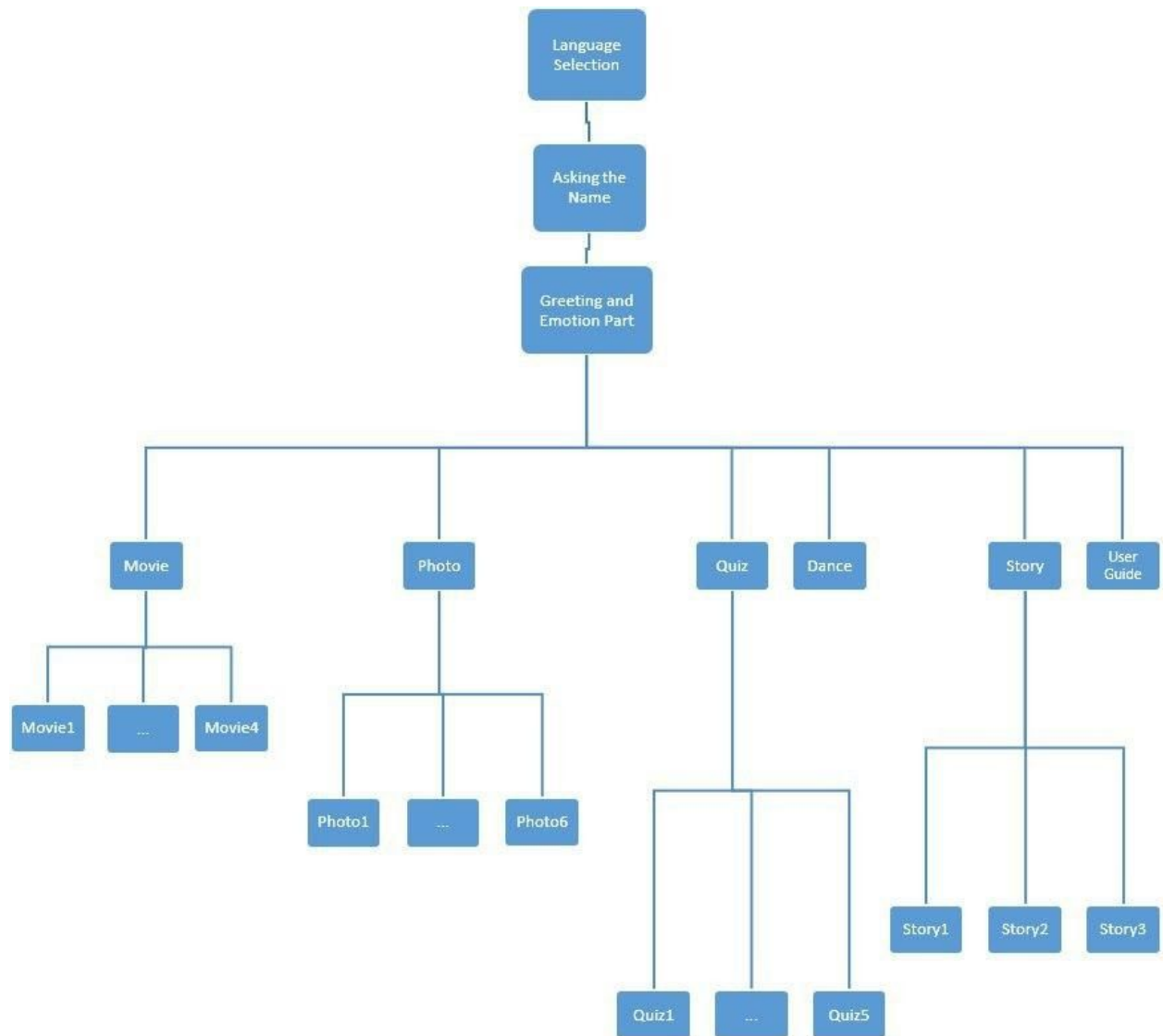
Furthermore, in another paper entitled “Social Interaction with Drones using Human Emotion Recognition”, they have proposed a demonstration that suggests a mapping function for human interpretable drone motions corresponding to the five human emotional states (i.e anger, happiness, sadness, surprise, and fear), using the personal drone’s movements (i.e, changing speed or rotations) instead of anthropomorphizing it.

In the current project, we get the motivation from the above works and made the following contributions: Pepper can interact with households and entertain them properly by playing video and music, showing some photos, making a quiz, dancing and telling a story.

In addition, Pepper shows its emotion in some different situations. If someone tells the pepper about his feelings, pepper reacts by suggesting some entertainment.

Solution

The architecture, components and the connections of different parts of our solution are as follows:



When the application starts to work, firstly, the user can see the welcome page with language selection either for english or italian. Then, on the next page, the robot asks the user's name in order to recognize him. After that, there is a page to ask about the user's feelings. Based on the user's feeling, the robot suggests to the user a variety of entertainment activities

such as watching a movie, seeing photos, dancing, doing a quiz and telling the stories. In each of these sections, there are a number of movies, photos, questions and stories. Additionally, in each part, pepper interacts with the user vocally and communicates with him using speech recognition.

Implementation

Details on the implementation

We have implemented different components of the application as follows:

Language selection: We know that following code is used for language selection:

```
im.setProfile("*,*it,*")
im.setProfile("*,*en,*")
```

However, it does not work properly in our case. So, we have implemented the language selection part in the following way:

For each action, we have considered two files, one for english and another for italian. For instance, in the beginning, when the pepper asks the user "Do you like to switch to italian language?", if the answer is "yes", a1_it action will be executed. On the other hand, if the answer is "No", a1 action will be executed and so on.

```
im.executeModality('TEXT_title','Do you like switch to italian language?!')
im.executeModality('TTS','Do you like switch to italian language?!')
im.executeModality('BUTTONS',[['yes','yes'],['no','no']])
a = im.ask(actionname=None, timeout=4)
    if (a=='yes'):#Italian section
        im.execute('a1_it')
        time.sleep(2)
        a = im.ask('a2_it')
```



```
time.sleep(1)
```

```
.  
. .  
.
```

```
else:#English section*****
```

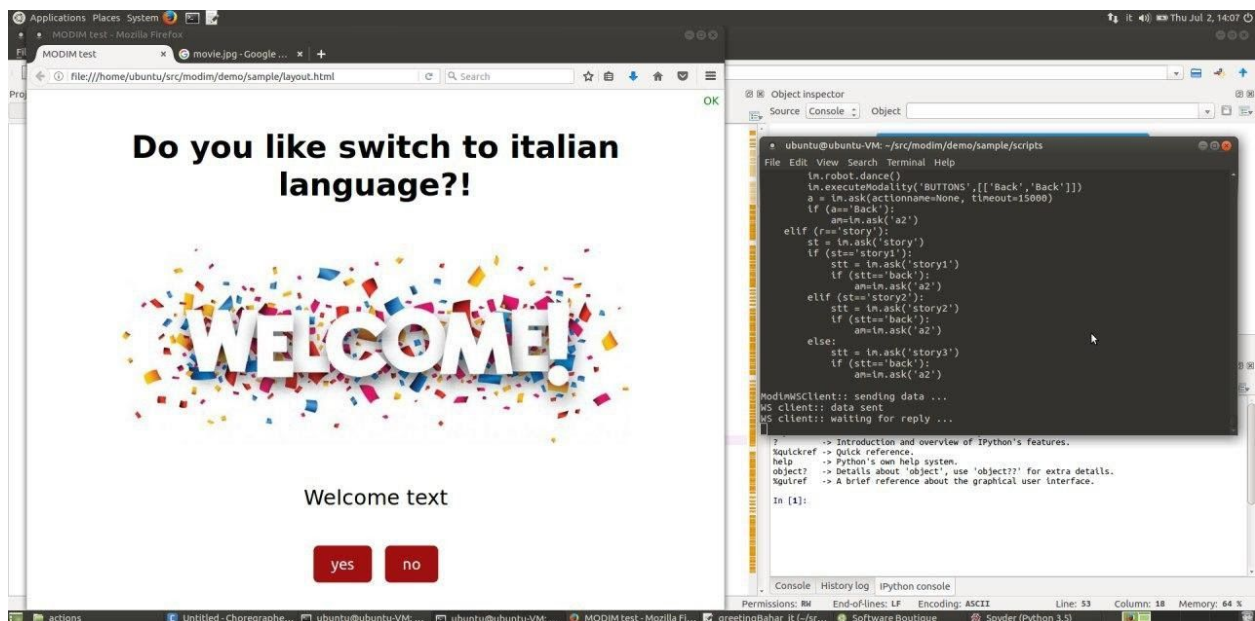
```
im.execute('a1')
```

```
time.sleep(2)
```

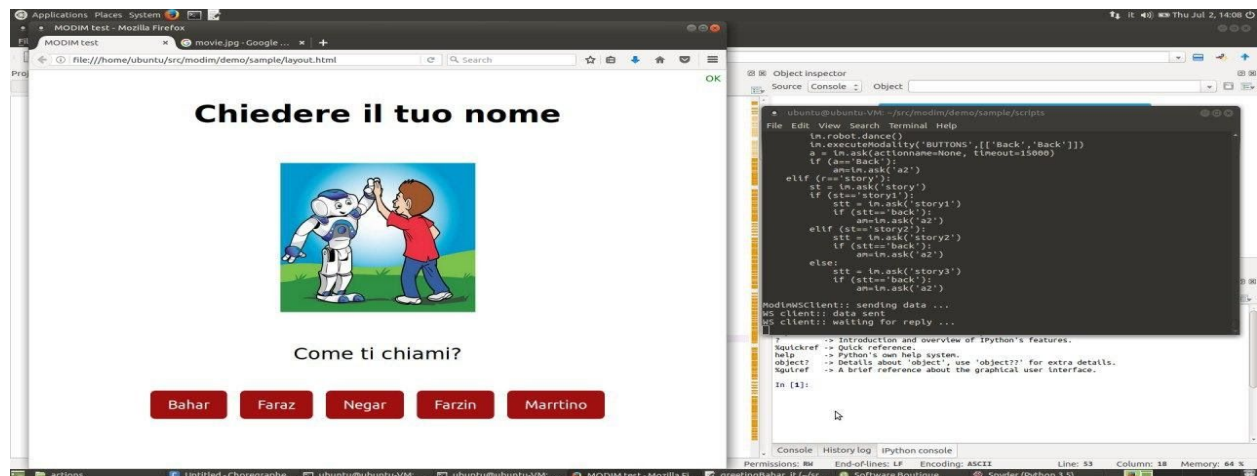
```
a = im.ask('a2')
```

```
time.sleep(1)
```

```
.  
. .  
.
```



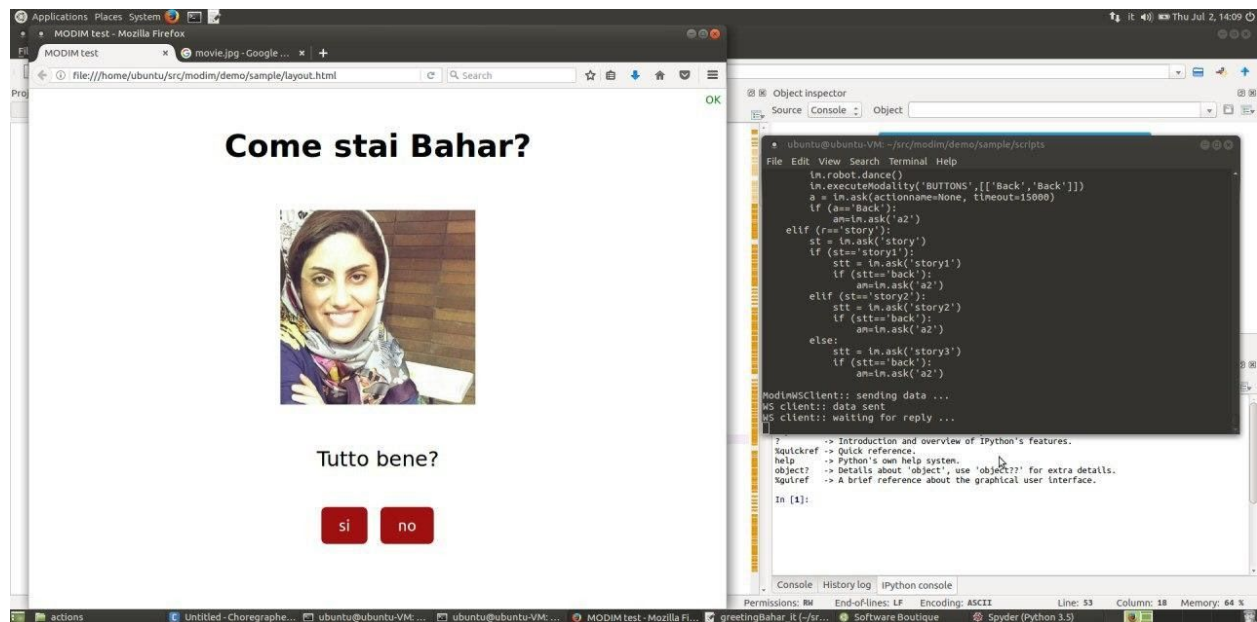
Asking the name: Based on which language has been chosen, the related action will be executed and pepper asks the user's name. The user can see the action either on its tablet or hearing it from pepper. The user has the possibility to select his name by pushing the buttons on the screen or telling his name by voice.



Greeting and Emotion part: When the pepper recognizes the person by knowing his name, it starts to greet him and ask about his feelings. Suppose, the user has chosen “Bahar”, then the pepper asks the user “How are you bahar? Is everything fine?” The user can answer with “Yes” or “No”. If the answer is yes, Pepper tells “I am very happy to hear that so I am going to make you happier! How can I help you bahar?” and at the same time, it shows the user different entertainment such as watching movies and photos, doing a quiz, dancing, telling the story and user guide. On the other hand, if the answer is “No”, pepper tells “I am very sorry, don’t worry, I am going to make you happy, How can I help you bahar?”. In the following code, you can see the implementation:

```
if (a=='Bahar'):
    r = im.ask('greetingBahar')
    im.executeModality('TEXT_default',r)
    if (r=='yes'):
        r=im.ask('greetingBaharYes')
    else:
        r=im.ask('greetingBaharNo')
elif (a=='Farzin'):
    r = im.ask('greetingFarzin')
    im.executeModality('TEXT_default',r)
```

```
    if (r=='yes'):
        r=im.ask('greetingFarzinYes')
    else:
        r=im.ask('greetingFarzinNo')
elif (a=='Faraz'):
    r = im.ask('greetingFaraz')
    im.executeModality('TEXT_default',r)
    if (r=='yes'):
        r=im.ask('greetingFarazYes')
    else:
        r=im.ask('greetingFarazNo')
elif (a=='Negar'):
    r = im.ask('greetingNegar')
    im.executeModality('TEXT_default',r)
    if (r=='yes'):
        r=im.ask('greetingNegarYes')
    else:
        r=im.ask('greetingNegarNo')
elif(a=='Marrtino'):
    r = im.ask('greetingMarrtino')
    im.executeModality('TEXT_default',r)
    if (r=='yes'):
        r=im.ask('greetingMarrtinoYes')
    else:
        r=im.ask('greetingMarrtinoNo')
```



Entertainment activities: This part includes movie, photo, dance, storytelling, doing quiz and user guide.

Photo: If the user selects the photo option, pepper shows on its tablet 6 different photos by time intervals of 3 seconds. As the photos are shown on the tablet, pepper shows its emotion by showing a message on the tablet.

if (r=='Photo'):

 im.executeModality('IMAGE','img/dolphin.jpg')

 time.sleep(2)

 im.executeModality('IMAGE','img/dog.jpg')

 time.sleep(2)

 im.executeModality('IMAGE','img/horse.jpg')

 time.sleep(2)

 im.executeModality('IMAGE','img/orange.jpg')

 time.sleep(2)

 im.executeModality('IMAGE','img/rainbow.jpg')

 time.sleep(2)

 im.executeModality('IMAGE','img/logoeuroboticsweek.png')

 time.sleep(2)

Movie: If the user selects the movie option, pepper ask the user, “which type of movie you want to watch?” and pepper show the user five movies. The user can push the buttons and choose his favorite movie or he can tell movie1, movie2 and so on.

```
elif (r=='Movie'):
    am = im.ask('actionMovie')
    if (am=='movie1'):
        im.executeModality('IMAGE','img/g1.gif')
        im.executeModality('TEXT_default', 'Enjoy...')
        im.executeModality('BUTTONS',[['back','back']])
        #im.executeModality('ASR',[Back])
        a = im.ask(actionname=None, timeout=15000)
        if (a=='back'):
            am=im.ask('actionMovie')
    if (am=='movie2'):
        im.executeModality('IMAGE','img/g2.gif')
        im.executeModality('TEXT_default', 'Enjoy...')
        im.executeModality('BUTTONS',[['back','back']])
        a = im.ask(actionname=None, timeout=15000)
        if (a=='back'):
            am=im.ask('actionMovie')
    if (am=='movie3'):
        im.executeModality('IMAGE','img/g3.gif')
        im.executeModality('TEXT_default', 'Enjoy...')
        im.executeModality('BUTTONS',[['back','back']])
        a = im.ask(actionname=None, timeout=15000)
        if (a=='back'):
            am=im.ask('actionMovie')
    if (am=='movie4'):
        im.executeModality('IMAGE','img/g4.gif')
        im.executeModality('TEXT_default', 'Enjoy...')
        im.executeModality('BUTTONS',[['back','back']])
```

```

        a = im.ask(actionname=None, timeout=15000)
        if (a=='back'):
            am=im.ask('actionMovie')
    if (am=='movie5'):
        im.executeModality('IMAGE','img/g5.gif')
        im.executeModality('TEXT_default', 'Enjoy...')
        im.executeModality('BUTTONS',[['back','back']])
        a = im.ask(actionname=None, timeout=15000)
        if (a=='back'):
            am=im.ask('actionMovie')

```

Quiz: If the user selects a quiz option, pepper will execute the action related to the first quiz and pepper asks “which animal is this?”. The user can answer by clicking on the buttons or by telling the word. In this part, we have 5 questions. Based on the number of questions that the user has answered correctly or wrongly, pepper can give emotional feedback to him. For example, if the user answers just one question correctly out of five questions, pepper shows the sentence “It is very bad, just one answer is right!” and also it shows a sad picture on its tablet.

```

elif (r=='Quiz'):
    c=0 #For understanding how many answers are correct
    am=im.ask('q1')
    if (am=='correct'):
        c+=1
    am=im.ask('q2')
    if (am=='correct'):
        c+=1
    am=im.ask('q3')
    if (am=='correct'):
        c+=1
    am=im.ask('q4')
    if (am=='correct'):

```

```

        c+=1
    am=im.ask('q5')
    if (am=='correct'):
        c+=1
    if (c==1):
        im.executeModality('TEXT_default', 'Its very bad ,just one
answer is right.')
        im.executeModality('IMAGE','img/verybad.jpeg')
    elif (c==2):
        im.executeModality('TEXT_default', 'Bad, just two answer is
right.')
        im.executeModality('IMAGE','img/bad.jpeg')
    elif (c==3):
        im.executeModality('TEXT_default', 'not bad, just three answer
is right.')
        im.executeModality('IMAGE','img/notbad.png')
    elif (c==4):
        im.executeModality('TEXT_default', 'Good, just four answer is
right.')
        im.executeModality('IMAGE','img/good.png')
    else:
        im.executeModality('TEXT_default', 'Congratulation, all answers
are right.')
        im.executeModality('IMAGE','img/congratulation.jpeg')

```

Dance: If the user selects the dancing part, pepper will show his emotion by showing this text “Thank you for selecting this part because this is my favorite part” and telling “I love dancing!” and also it shows a gif on its tablet which is so funny.

```

elif (r=='dance'):
    im.executeModality('TEXT_title','Thank you for selecting this part
because this is my favorite part')

```

```
im.executeModality('TEXT_default','!!! Dancing !!!')
im.executeModality('TTS','I love dancing')
im.executeModality('IMAGE','img/dance2.gif')
im.robot.dance()
```

Story: If the user selects the story part, pepper will react either showing the message or telling “Which kind of story do you like to listen to?”

In this part, there are 3 kinds of stories. The user can either listen to the story or read it on a tablet.

```
elif (r=='story'):
    st = im.ask('story')
    if (st=='story1'):
        stt = im.ask('story1')
        if (stt=='back'):
            am=im.ask('a2')
    elif (st=='story2'):
        stt = im.ask('story2')
        if (stt=='back'):
            am=im.ask('a2')
    else:
        stt = im.ask('story3')
        if (stt=='back'):
            am=im.ask('a2')
```

Libraries/tools

In this project, we make use of libraries such as ws_server to connect to modim server, ws_client that contains some functions for running modim clients, TTS service of the robot for speaking, human_say.py in order to speak with pepper.

Once we have all the settings, we can easily connect to modim server in two ways. In our case, modim is running locally with respect to the client.

When modim client and modim server are running on the same machine, so you just need to create the modim ws_client and then set the path of the demo in a automatic way in this way:

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

In order to see the pepper virtually, we have used choregraphe in a virtual machine. When modim server remote with respect to modim client, so this modim server is running on a different machine with respect to the client, then we have done in this way:

In the virtual machine, we open 2 terminals and we export following dependencies in each of them:

Terminal1:

```
Export PEPPER_IP='127.0.0.1'  
Export PEPPER_PORT=38049  
Export MODIM_IP='127.0.0.1'  
Export PEPPER_TOOLS_HOME=/home/ubuntu/pepper_tools  
Export MODIM_HOME=/home/ubuntu/src/modim  
Cd src/modim/src/GUI  
Python ws_server.py -robot pepper
```

Terminal2:

```
Export PEPPER_IP='127.0.0.1'  
Export PEPPER_PORT=38049  
Export MODIM_IP='127.0.0.1'  
Export PEPPER_TOOLS_HOME=/home/ubuntu/pepper_tools  
Export MODIM_HOME=/home/ubuntu/src/modim  
Cd $MODIM_HOME/demo/sample/scripts  
Python project.py
```

Each time you open choregraphe, you have to change the port number.
In the choregraphe you can find the port in this way:
Edit - preferences - virtual robot tab, then you can see the port.

Conclusions

What we have learnt:

We have gained different experiences in which we have learnt to program for pepper in 2 different ways, one when modim server and modim client are in the same machine and another when modim server is running on a different machine with respect to the client. We have learnt that we cannot use `mws.run_interaction()` in the functions on python modim file, because on the server side, there is an implementation of the interaction manager. So, we can use the above command just on the server side. Additionally, we have learnt how we can use some interaction manager in the client side such as `im.execute('a11')`, `im.askUntilCorrect('a33', timeout=15)`, `im.executeModality('TEXT_default',r)`, `im.robot.dance()` and so on. We have learnt how we can run interactions with the modim server and how to define different kinds of interactions.

Our experience in developing the project

We have encountered some problems in the process of developing the project. We tried to develop the code in a functionality mode; however, we did not get any success. For instance, when we want to make the possibility of jumping between different pages (going backward), we have to write the piece of code related to going backward and call it whenever we need it but this is not possible.

Second problem that we have faced is related to the time of the execution of the program. In fact, when we select the video option and we want to go back to the starting page, it works, however the time of the execution will be up soon. In our opinion, this problem is in close connection with the first problem.

Improvement

What can be done to further extend the functionalities and improve the effectiveness of the project is as follows:

1. The possibility of making functionalities instead of writing if-then-else
2. The possibility of adding and playing videos
3. The possibility of adding games to the application
4. Specific characters such as ò, è, à are supported in the action file; however, they are not supported in the python file.

Link to Videos

<https://youtu.be/3dNEzVKjlwU>

https://youtu.be/hU0-_kaFSNQ

<https://youtu.be/y8iwOhv4F6M>

<https://youtu.be/Lda-ZqxrW10>

<https://youtu.be/n9pVSUoHmwM>

https://youtu.be/9bBVFbb_vbQ