1. Amendment on the test

The process and databases remain the same. Amendments are about the criteria allowing the choice of the movie (genre and actors). As a consequence, the teams will have to write dialogs using the new criteria.

The movie database contains the following information about each movie:

- Name
- Spoken language
- List of actors
- List of genres
- File name

The list contains fake data (eg: film0 ... film9), and the real content will be provided at the finals.

The user database contains the following information about each user:

1. Name
2. Spoken language
3. Favorite actors

Language & actors are randomly associated to the user each time you start the software.

NAO must:

- Recognize the user
- Find a movie genre from the dialog with the user
- Get the user favorite actors from the database
- Find all the movies with the chosen genre & actor
- Dialog with the user to select one movie within all the suitable ones
- Get the user's spoken language from the database
- Play this movie and add the correct subtitles (if the movie spoken language is not the user's language)

2. Run the TV simulator

For a better readability, we advise to run the python scripts from a command invite. Just navigate in your computer until you get to the "movietime" folder you downloaded, and then run the command "python main.py"

You can read the following logs written by the program:

```
>python main.py
```

The controller (see controller.py) is the main component. It receives the commands from the network and executes the right commands.

```
[MAIN] Starting controller
```

The film library (see movie_library.py) is the database containing all the movies. You can find its content in the "movies.csv" file.

```
[MAIN] Creating a new film library
[MAIN] Registering the film library
```

When registering this component to the controller, the controller will list all the new functions that can be called by NAO:

```
[CONTROLLER] get_all_actors has been registered
[CONTROLLER] get_all_genres has been registered
[CONTROLLER] get_all_languages has been registered
[CONTROLLER] get_all_names has been registered
[CONTROLLER] get_movie_actors has been registered
[CONTROLLER] get_movie_filename has been registered
[CONTROLLER] get_movie_genres has been registered
[CONTROLLER] get_movie_language has been registered
[CONTROLLER] get_movies_by_actor has been registered
[CONTROLLER] get_movies_by_genre has been registered
[CONTROLLER] get_movies_by_language has been registered
```

Then the users' database (see user_database.py) is loaded. This component simulates the knowledge about each user. We could imagine this being based on social network data, or on previous conversations between NAO and the users…

```
[MAIN] Creating a new users database
[MAIN] Registering the users database
[CONTROLLER] get_user_actors has been registered
[CONTROLLER] get_user_language has been registered
```

Here a random list of users' tastes is created. The randomization function will associate each user to one or two actors, and a language.

```
[MAIN] Randomize the database
[CONTROLLER] starting get_all_actors
[CONTROLLER] starting get_all_languages
```

The TV Player is instantiated eventually. This component is the one in charge of showing the video.

```
[MAIN] Creating the TV player
[MAIN] Registering the TV player
[CONTROLLER] play_movie has been registered
```

Once every component is registered, the controller is run, so the server starts listening and receiving the commands from the network.

```
[MAIN] Starting controller
[CONTROLLER] Starting server...
[CONTROLLER] Server started!
```

Note that this v1 version does not contain any graphical interface or real videos. Right now the ten movies are called "film0" to "film9". The next versions will precise which movies are used and films, actors, genres and languages will be replaced by the real values.

Examples

You can find some usage examples in the file "demo.py".
For every request you make on the server (aka controller), you can read the logs in the command window, showing what is happening on NAO.

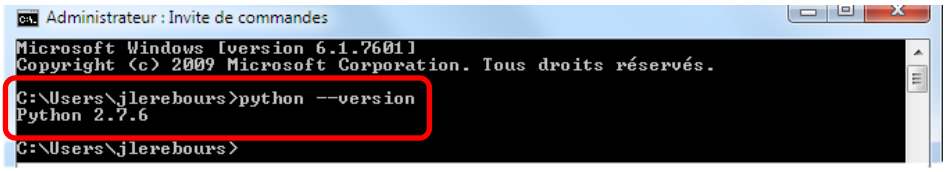
3. Python set-up

Installation

Install Python 2.7 for free on http://python.org. Follow the standard installation process.
Version for Windows: https://www.python.org/ftp/python/2.7.8/python-2.7.8.msi

Check the installation

Once Python setup has completed, open a command invite or console by pressing the keys windows + R, and typing cmd.
Enter the command "python --version" and hit Enter. You should see in the invite the Python version displayed (last version is 2.7.8, but any 2.7.x version will work the same).

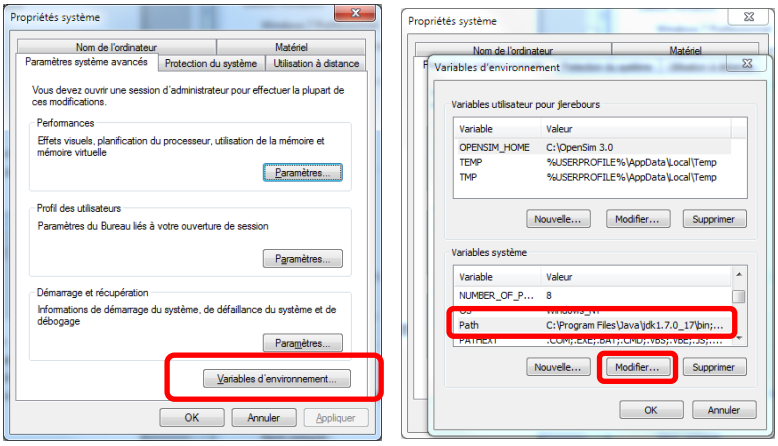


➔ Python command is not recognized?

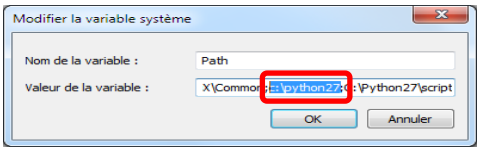
First try again after restarting your computer.
Then you can check that "c:\python27" is specified in you PATH environment variable. If not, you have to add it and restart the command window. You will find many explanations for your operating system about the PATH variable on the web.

Here is the procedure for Windows:

- Right click on "Computer". Select "Properties". Click on "Advanced system properties"
- In the system properties, click on the button "environment variables"
- Locate and select the "Path" variable in the list. Click "Edit".



- Add "c:\python27" to the list. Separate from the other values with a ";".



- Validate and close all windows. Then restart the command invite and try again!