SIC-2022 course - installation test script

This is a simple test script to test whether you have installed the SIC infrastructure correctly. Before you execute the test script, you should have installed the SIC infrastructure using the instructions here.

Please follow the steps below to test SIC on the machine that you installed the infrastructure on. The script only tests whether you are able to generate speech using your machine's speakers.

1. **Download** the Python skeleton project which can be found here. Use the '*Downloads'* button in the left menu and then select *download repository*. Unzip the folder in a location of your choice.

2. **Open** a terminal and navigate to the unzipped folder of step 1. Then, navigate to the `python` subfolder. From there, navigate to the to the `sic` folder.
   To make sure that the required dependencies are installed, execute the following command `pip3 install .`

3. **Launch** the SIC infrastructure. For doing so, you need to follow the instructions outlined here. In particular, you need to follow the first two or three steps, depending whether you have Docker Desktop installed (first two steps) or Docker Toolbox (first three steps). Simply run the `docker-compose up redis` command to launch only redis (the basic infrastructure) but no additional services.

4. **Open another terminal window.** Navigate to the location of the docker repository that you cloned previously. From there, navigate to the `sic-local` subfolder. If you are using a terminal, type `java -jar computer-speakers.jar` after navigating to the `sic-local` folder and then enter to run the application. A small window labelled **SIC Speaker** should launch (if first a window 'SIC info' opens, just press 'OK').

5. **Launch** PyCharm (or any other integrated development environment you use for developing Python code such as Visual Studio Code 2).

6. **Open** the `python` folder in the Python skeleton project that you downloaded (step 1 above) in PyCharm (or in the other integrated development environment)

7. **Locate** the `my_connector_example.py` file. This file is in the 'examples' folder. **Run** the `my_connector_example.py` file. In the pop-up window select the speaker and press *OK*. You can also check the terminal in which you launched the `computer-speakers.jar`. You should see:

   ```
   AudioLanguage: en-US
   Say: Hello, world!
   ```

   a. If your program only runs the set_language part and you're only seeing `AudioLanguage: en-US` in the Java terminal (the terminal in which you have the java script running), then running your script in the terminal (instead of PyCharm) with `python3 my_connector_example.py` might help.

b. Some systems might not have *espeak* installed, which results in the following error: `java.io.IOException: Cannot run program "/usr/local/bin/espeak": error=2, No such file or directory`.
    i. On a **mac** you can install *espeak* with Homebrew using `brew install espeak`.
    ii. On **linux** (debian), you can install *espeak* by `sudo apt install espeak`. If you still get an error that *espeak* is not found, you should check where *espeak* is installed (probably in `/usr/bin/espeak`) and create a symbolic link to where `my_connector_example.py` is requesting it from (e.g., `sudo ln -sf /usr/bin/espeak /usr/local/bin/espeak`).
    iii. For **Windows** you can download *espeak* here.

8. **Report your result by submitting the Google Docs which you can find here**. If, after a short pause, you hear 'Hello world' in the previous step (step 7) then your installation was successful! You can also get help on Slack in the #installation channel. If you have not joined the Slack workspace, you can do so by clicking on this invite link .