SIC-2021 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 *download* 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 `python` subfolder in the folder where you unzipped the project. To make sure that the required dependencies are installed, execute the `pip install -r requirements.txt` command in the `python` folder.
3. **Launch** the SIC infrastructure. Follow the first two or three instructions, depending whether you installed Docker Desktop (first two steps) or Docker Toolbox (first three steps), for running the infrastructure here. Simply run the `docker-compose up redis` command to launch only redis (the basic infrastructure) but no additional services.
4. **Perform** the last step of the *running with ...* section on the wiki and launch the `computer-speakers.jar`. If you are using a terminal, type `java -jar computer-speakers.jar` after navigating to the `cbsr-local` folder and then enter to run the application. A small window labelled *Debug Speaker* should launch.
5. **Launch** PyCharm (or any other integrated development environment you use for developing Python code).
6. **Open** the `python` folder in the Python skeleton project that you downloaded (step 1 above) in PyCharm.
7. **Locate** the `my_connector_example.py` file in the folder and **run** it. 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, 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** result through this form. If, after a short pause, you hear 'Hello world' then your installation was successful!

If you had to come up with one more dimension to this framework, which one would you choose and why?