ROBOCUP Project

Cornelius Braun, Seraphin Zunzer, Emilia Schulenburg

Motivation & Goal

Our project goal: Have a robot that can dance to music

For that, the robot needs to...

1. … be able to “listen” to music
2. … be able to recognize what the music is like → music genre recognition
3. … be able to execute dancing movements

Project pipeline

1. **Sensing**
   - **record audio**
2. **Thinking**
   - **analyze audio**
3. **Acting**
   - **dance to music**

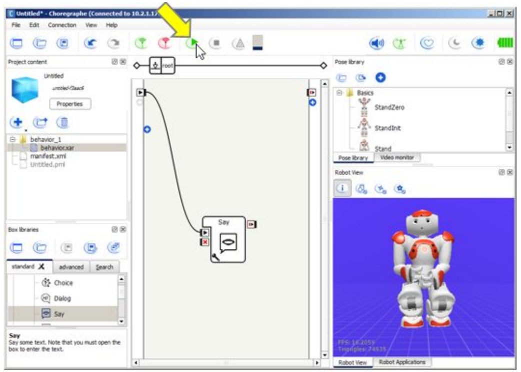
Tools

python™

+ libraries:

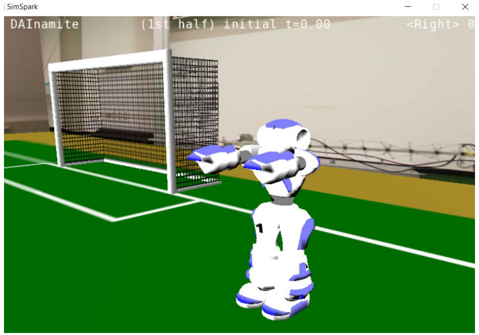
- pyaudio
- sklearn / keras
- librosa
- threading

Choregraphe software:



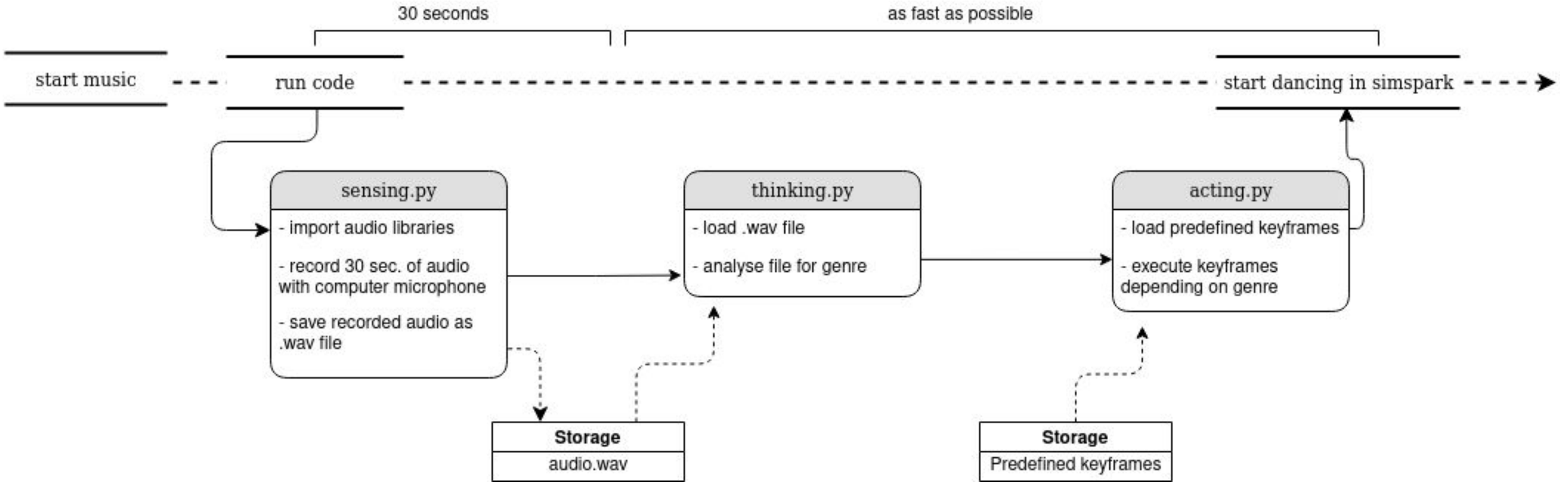
Say
Say some text. Note that you must open the box to enter the text.

Simspark simulator:



DAInamite (1st half) initial t=0.00 <Right> 0

Project Planning



sensing.py
- import audio libraries
- record 30 sec. of audio with computer microphone
- save recorded audio as .wav file

thinking.py
- load .wav file
- analyse file for genre

acting.py
- load predefined keyframes
- execute keyframes depending on genre

User interaction

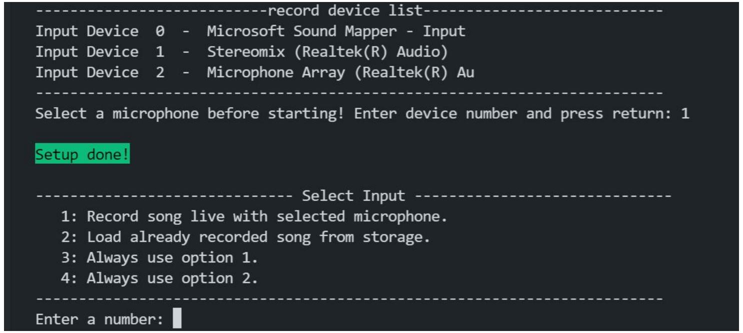
The sensing part implements user interactions:

User interacts with the command line interface:

1. Select a microphone
2. Decide whether you want to:
    - load a music file

    or:

    - record a music file
3. Stop the robot and start again

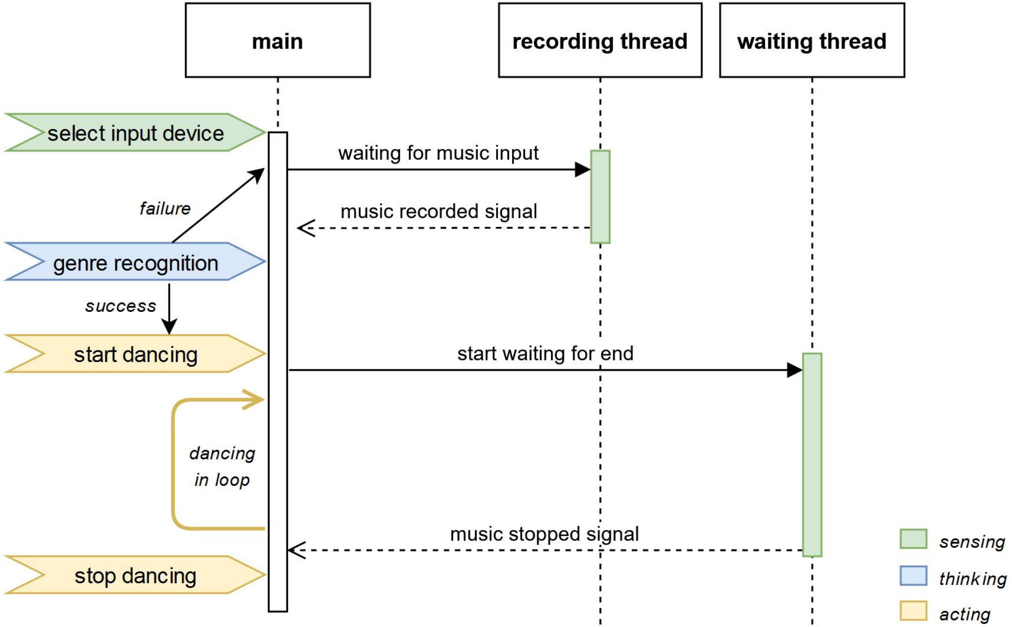


```
---------------------------record device list----------------------------
Input Device  0  -  Microsoft Sound Mapper - Input
Input Device  1  -  Stereomix (Realtek(R) Audio)
Input Device  2  -  Microphone Array (Realtek(R) Au
-------------------------------------------------------------------------
Select a microphone before starting! Enter device number and press return: 1

Setup done!

------------------------------ Select Input ------------------------------
   1: Record song live with selected microphone.
   2: Load already recorded song from storage.
   3: Always use option 1.
   4: Always use option 2.
-------------------------------------------------------------------------
Enter a number: █
```

General structure & Sensing



<u>recording thread:</u>

- select:
  1. live recording or
  2. load music from storage

→ save music and start classification

<u>waiting thread:</u>

- waits until:
  1. user interrupt or
  2. music stopped

→ stop robot

Thinking: Music Genre Classification

Training data:

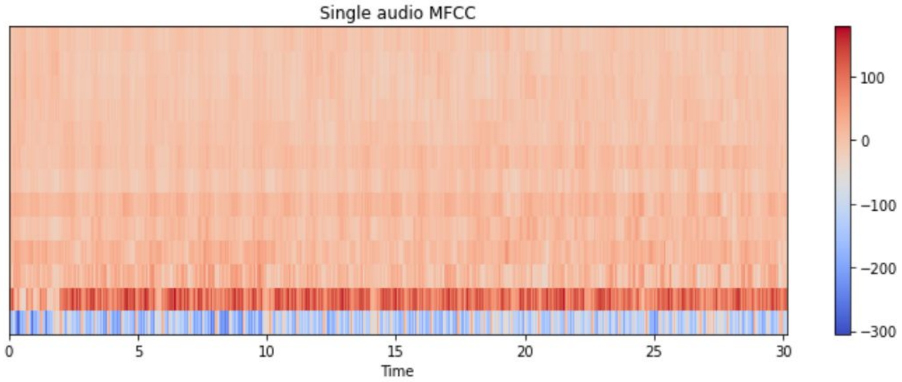
GTZAN dataset

Genres:

Classical, Pop, Metal

Features:

Mel Frequency Cepstral Coefficients (MFCC)

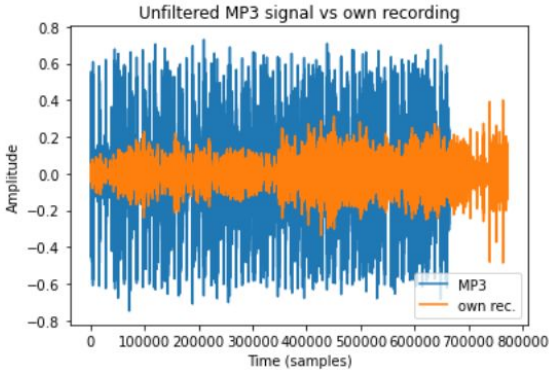


Thinking: Data augmentation

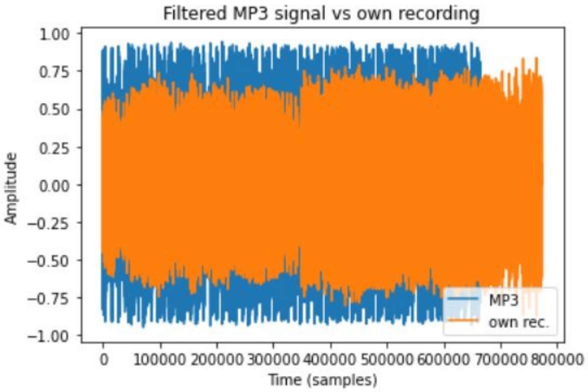
- GTZAN data are snippets of studio recording audio quality

- Computer recordings have worse quality

Solution: use mu-law compressor



→



Thinking: Classification results

We compared 3 classifiers:

Perceptron, Convolutional Neural Network (CNN), Support Vector Machines (SVM)

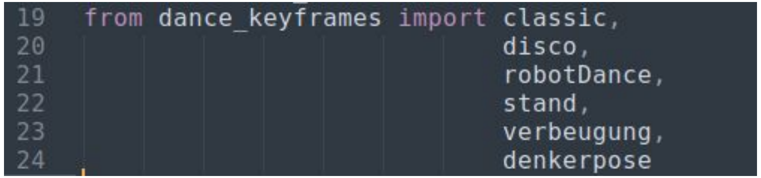
Classifier	**Perceptron**	**CNN**	**SVM**
Mean accuracy	0.97	0.33	0.96

SVM works the best in practice → we use SVM

Acting

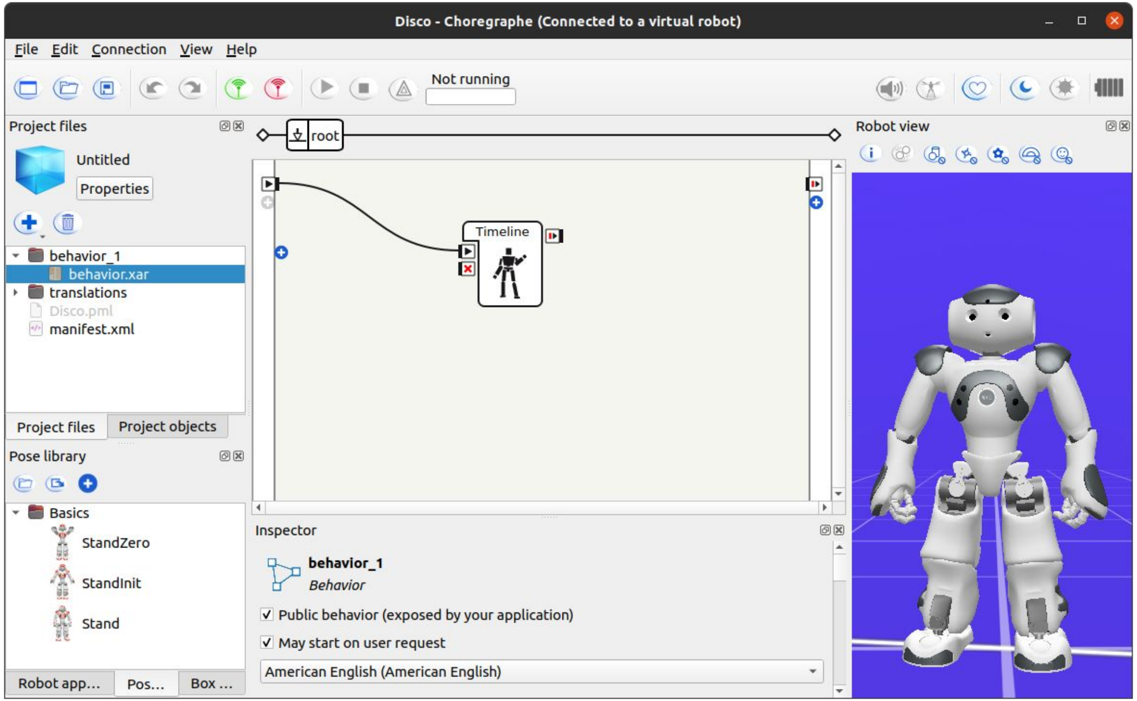
Genres: pop / disco, metal, classic

- software to create keyframes → Choregraphe
- exported in Bezier to Python
- imported in our dancing.py
- simulated in Simspark



```
from dance_keyframes import classic,
                            disco,
                            robotDance,
                            stand,
                            verbeugung,
                            denkerpose
```

Acting



behavior_1

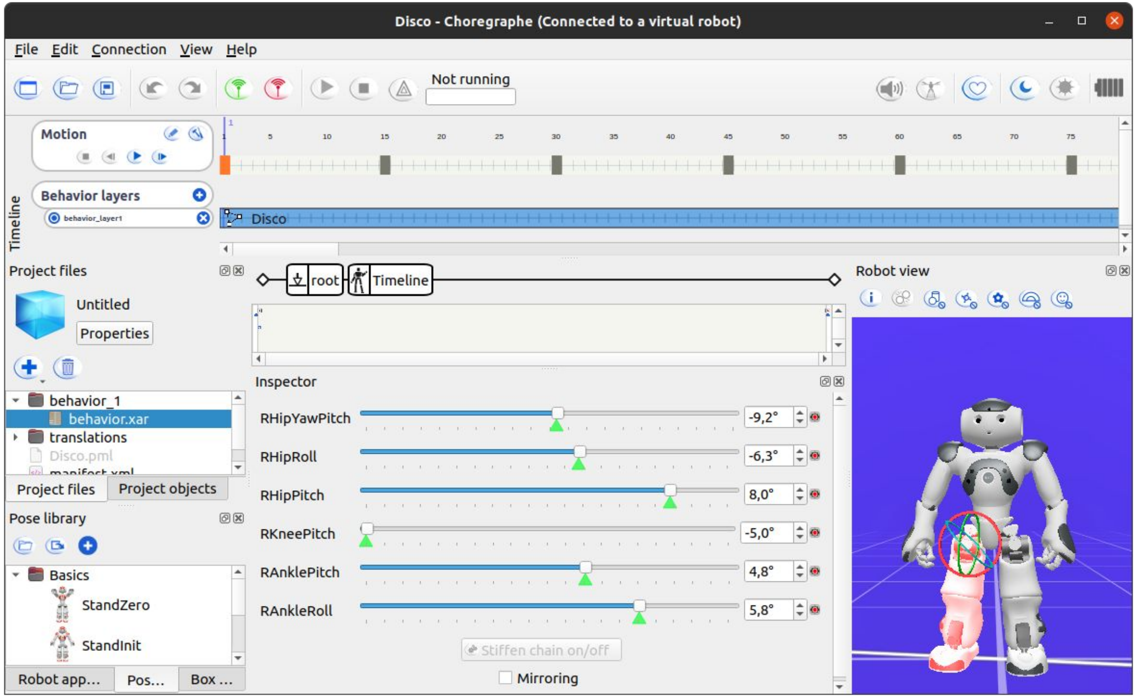
Behavior

☑ Public behavior (exposed by your application)

☑ May start on user request

American English (American English)

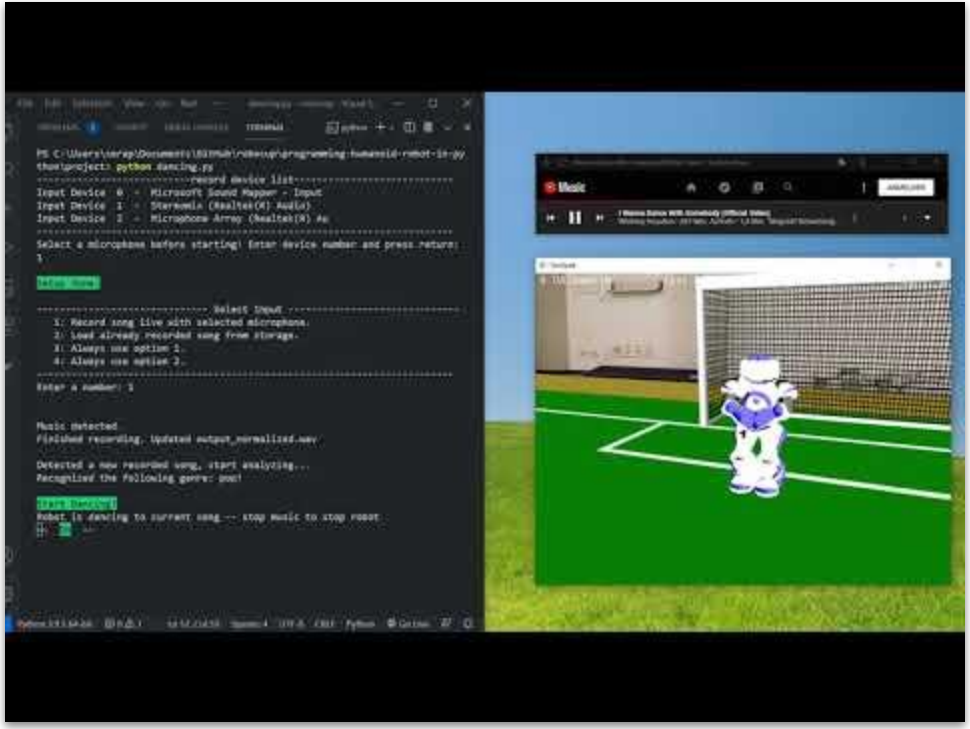
Acting



RHipYawPitch	-9,2°
RHipRoll	-6,3°
RHipPitch	8,0°
RKneePitch	-5,0°
RAnklePitch	4,8°
RAnkleRoll	5,8°

☐ Mirroring

Results



```
PS C:\Users\serap\Documents\GitHub\robocup\programming-humanoid-robot-in-py
thon\project> python dancing.py
----------------------------record device list----------------------------
Input Device  0  -  Microsoft Sound Mapper - Input
Input Device  1  -  Stereomix (Realtek(R) Audio)
Input Device  2  -  Microphone Array (Realtek(R) Au
---------------------------------------------------------------------------
Select a microphone before starting! Enter device number and press return:
1

Setup Done

------------------------------ Select Input ---------------------------------
   1: Record song live with selected microphone.
   2: Load already recorded song from storage.
   3: Always use option 1.
   4: Always use option 2.
---------------------------------------------------------------------------
Enter a number: 1


Music detected
Finished recording. Updated output_normalized.wav

Detected a new recorded song, start analyzing...
Recognized the following genre: pop!

Start Dancing!
Robot is dancing to current song -- stop music to stop robot
```

https://tubcloud.tu-berlin.de/s/J9jYsGp4WJbxTEZ

Project evaluation

Problems we encountered:

- poor audio quality
- recording length needed for classifiers
- maintain robot stability during dancing

Eventually, we were able to manage everything

We reached all of our goals and made the robot dance!

Thank you for your attention!