BODY SOUND

Move to create

Creative Programming and Computing
2022 / 2023 Project

Pisanu Nicolò (ID: 10578782)
Bertazzi Andres (ID: 10488849)
Viviani Marco(ID: 10528650)
Kubler Riccardo (ID: 10560345)

October 02, 2023

MASTER OF SCIENCE
MUSIC AND ACOUSTIC
ENGINEERING

Contents

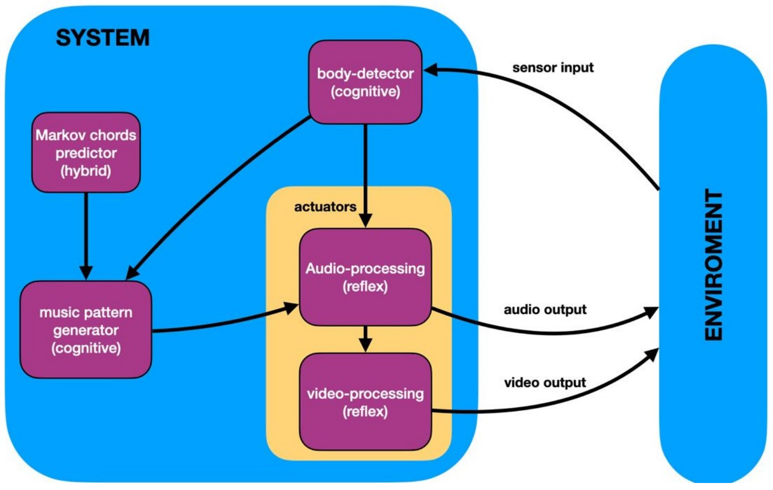
1 Introduction

"Body Sound" is a project whose aim is to develop a virtual musical instrument for an artistic installation.

It can be used by an artist in order to make his performance more appealing thanks to the graphic visualization, but it can also have educational purposes to teach the sounds of the different grades of the scale.

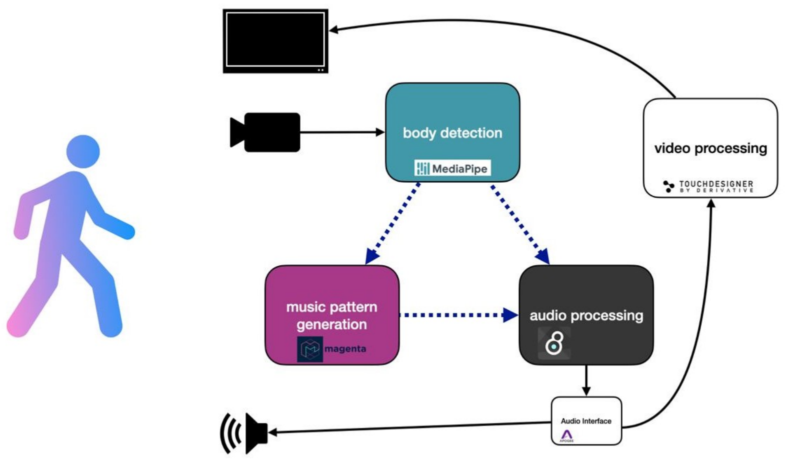
The project is based on a *full-fledged* model in which the user and the system can fully and constantly interact and they are born to learn and adapt.
The idea is that anyone moving his entire body can be involved by generating music.

The interaction between the user and the systems happens thanks to different interfaces which can be contextualized into the *Hybrid Contexts* where an artist creates his own interactive system for his own performance.


SYSTEM
body-detector (cognitive)
sensor input
Markov chords predictor (hybrid)
actuators
Audio-processing (reflex)
music pattern generator (cognitive)
audio output
video-processing (reflex)
video output
ENVIROMENT

The interactive system should be the best to make the artist experience what he/she desires and to make him/her communicate what the audience wants.

The system uses a sensor (camera) that receives information from the external real environment and then a visual representation (MediaPipe) reacts to the inputs given by it showing a graphic representation (TouchDesigner) on a screen and playing some sounds (Max8).



2 Technologies

In this paragraph, we will shortly discuss the technologies used and the challenges, and lessons learned in implementing this project.

2.1 Body Detection

The first module of this project is represented by the ”Body Detection” code.
It deals with identifying some user movements that will modify the generated piece of music in many aspects like the rhythmic structure developed by MusicVAE or some filters and effects generated by Max8.
The music generated by Max8 will then animate an audio-reactive visual implemented in TouchDesigner via OSC.
To do this, we wrote a Python code that makes use of several libraries useful for our purposes.

The first libraries we use are *Mediapipe* and *Mediapipe Holistic*.
They provide Deep Learning tools to recognize the body through a camera. Each part of the body is identified by a landmark which is nothing more than a 2D spatial coordinate on which geometric and non-geometric calculations can be performed.
By exploiting these landmarks we are able to calculate the positions of the affected body parts on the screen and to improve the landmarks' performances we make use of some buffer arrays.
The presence of these buffer arrays allows the user to record a history of the latest values of the calculated quantity and therefore take into account the distance from the screen thanks to the MIN/MAX formula.
At every iteration, a new value is appended to the buffer while the last one is

removed and we compute the min and the max of the vector.

By using the formula $x = (X - X_{min})/(X_{max} - X_{min})$ the values are normalized between 0 and 1 and at the same time the distance is taken into account because the values change with respect to distance from the screen.
Usually, the X value is a mean of the last values of the buffer in order to "smooth" the increase or decrease of the quantity we are calculating.
In this way the quality of the calculated quantity increases significantly.
The second library is Python OpenCV, an optimized computer vision library suitable for our purposes.

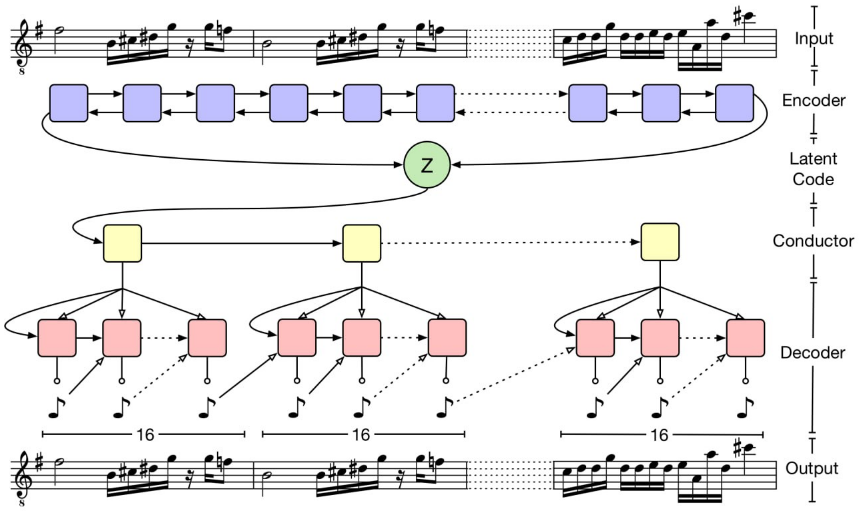
The code also makes use of a function *calcOpticalFlowFarneback*, which allows the user to calculate the optical flow of the screen, and two other functions (*draw_flow*, *draw_hsv*) to see the flow itself and its vectors on the screen.
The optical flow is used to calculate the momentum present in the computer screen and is implemented by calculating the motion of objects in the screen between two consecutive frames.
This motion is represented by several oriented vectors.
A weighted average of these magnitude vectors with respect to the angles is made.
In this way, the average momentum is calculated also taking into account the direction of the body movement.
Once all these quantities have been calculated, all the values (with a frequency of 10 fps) are sent to MusicVAE and Max8 to check the parameters via OSC (*setOSC_Client* function).

Below are all the calculated quantities, which are sent to MusicVAE or Max8, and which parameter they change:

1. Momentum in the screen (via optical flow).
   Modifies MusicVAE temperature
2. Gradual opening of the hand. Modifies high/low by Max8
3. Open/Closed right hand. Max8
4. Right-hand rotation
5. Left-hand rotation
6. Height of the hands
7. x hands????
8. Expansion of the hands

2.2 Music Pattern Generation

A crucial part of the design of the project is the problem of how to generate harmonic and rhythmic structures that we can control.
So we relied on a generative model that outputs MIDI files from which we can play with personalized instruments.
Also, the generation of an audio signal is very difficult that it is coinvincing in reality that's why we looked for models that output just the pattern structure.
A Variational Auto Encoder (VAE) is perfect for our purpose since can be controlled by sampling a latent vector and it is a generative model able to create new data.
For these reasons, our choice fell on MusicVAE.



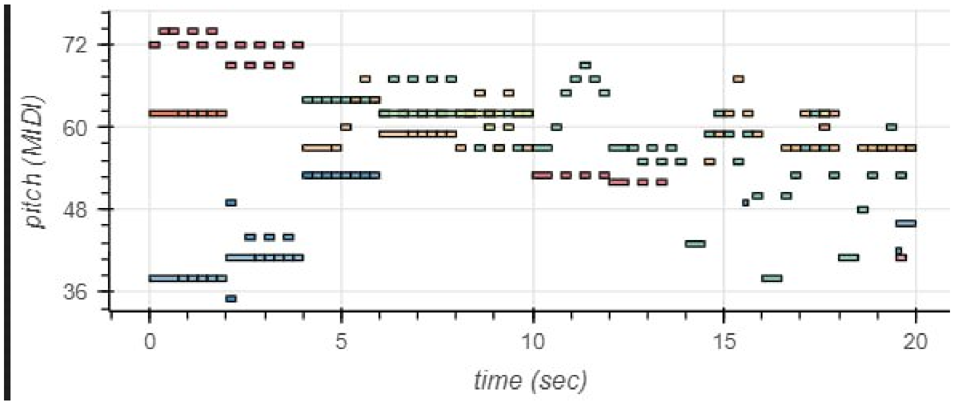
2.2.1 MusicVAE

MusicVAE is a machine learning model that lets us create a melodic and rhythmic that we can condition through some hyperparameters.
This model is the best to accomplish the goal of this project since it is able to generate multitrack MIDI files by decoding a vector sampled from a learned latent space.
MusicVAE has a two-layer bidirectional LSTM network for the encoder and a novel hierarchical RNN for the decoder.
In particular, we relied on the multitrack extension of the MusicVAE model able to handle up to 8 tracks.
A novel event-based track representation that handles polyphony, micro-timing, dynamics, and instrument selection.
It also introduces chord-conditioning to a latent space model which is important for our purpose since we can control it.

Magenta proposes some pre-trained models, for the demo we used
"*hier-multiperf_vel_1bar_med_chords*" since produces convincing rhythmic patterns and has various horizontal melodic possibilities.
The model generates a pattern with a duration of 2 seconds played at 120 bpm.

2.2.2 Interpolation

A latent space can be used to interpolate between two musical sequences. Given 2 vectors we can interpolate between them, then for any number of steps using spherical linear interpolation.



(a) Example of an interpolation in 10 steps between 2 latent vectors that we choose.

2.2.3 Controllable parameters

- Temperature of the softmax

  Increase the randomness approaching a uniform distribution while increases

- Chord conditioning

  The same latent vectors are decoded under several different chords.
  For a given latent vector, the instrumental choice and rhythmic pattern remain fairly consistent, while the harmony changes.
  In this way, we can concatenate multiple patterns to create a harmonically coherent multi-measure sequence.

2.3 Max8

Max/MSP (Max/Music and Max/Signal Processing) is a visual programming language for multimedia and music developed by Cycling '74.

It is commonly used for audio and multimedia processing, interactive art installations, music composition, and sound synthesis.
Max/MSP is a versatile and powerful tool for artists, musicians, and multimedia creators to explore, experiment, and create interactive and dynamic audiovisual experiences.

It's a widely used platform in the world of creative digital arts and music.
Here are some of the key aims and features of Max/MSP used in the project:

- Visual Programming
  Max/MSP uses a graphical interface where users create programs by connecting objects on a canvas. This visual approach makes it accessible to artists and musicians who may not have extensive coding experience.
- Real-Time Audio and Video Processing
  Max/MSP is well-suited for real-time audio and video processing. It allows users to manipulate and generate audio and video in real time, making it popular in the fields of electronic music and multimedia performance.
- Modularity
  Max/MSP encourages modular design, allowing users to create complex applications by connecting various pre-built objects (called "Max objects" or "externals") in a patcher window. This modularity makes it easy to reuse and share code.
- Extensibility
  Users can create custom Max objects using C or JavaScript, allowing for the development of specialized audio and visual processing tools.
- Integration
  Max/MSP integrates with other software and hardware, making it versatile for different applications. It can communicate with MIDI controllers, external devices, and other software through various protocols.
- Community and Third-Party Libraries
  There is a strong Max/MSP user community that shares patches, tutorials, and resources. Additionally, third-party developers have created numerous Max/MSP extensions and libraries to expand their capabilities.
- Cross-Platform
  Max/MSP is available for both macOS and Windows, making it accessible to a broad range of users.

Project Component: Max/MSP Real-Time MIDI Interpretation, Playback, and Integration Module with TouchDesigner Integration

- Introduction
  Within the comprehensive project framework, a critical component is the Max/MSP module, responsible for real-time MIDI sequence interpretation, playback, integration of audio effects controlled by MediaPipe body parameters, synchronization with MusicVAE, and the additional capability to transmit the resulting audio to the TouchDesigner module. TouchDesigner then processes this audio data to generate a synchronized real-time video output.

- OSC Communication and MIDI Sequence Interpretation
  This Max/MSP module interfaces with MusicVAE through OSC, obtaining MIDI sequences representing musical measures. It performs intricate MIDI sequence interpretation to extract essential musical parameters such as program number, instrument, pitch, velocity, onset, and offset time.

- Real-Time Playback with Audio Effects
  A distinctive feature of this module is its dynamic application of audio effects to the generated sound. These effects are meticulously controlled by body parameters obtained from MediaPipe. This integration enhances the expressiveness and adaptability of the musical output, contributing to a more immersive experience.

- Synchronization with MusicVAE via Markov Chain
  Additionally, the Max/MSP module manages synchronization between the MIDI-based music generation and MusicVAE. It employs a second-order Markov Chain to generate chords, which serve as input to request new MIDI sequences from MusicVAE. This synchronization ensures coherence and harmony between the two modules, enhancing the overall musical output.

- Integration with TouchDesigner for Real-Time Video Generation
  A pivotal extension of this Max/MSP module is its ability to transmit the resulting sound to the TouchDesigner module. TouchDesigner processes the audio data in real time and generates synchronized video output accordingly. This integration introduces a multisensory dimension to the project, aligning audio and visual elements in harmony.

- Integration into the Overall Project
  This Max/MSP module, with its multifaceted capabilities, harmoniously integrates into the broader project architecture. It plays a pivotal role in achieving the project's overarching objectives, combining musical creativity with visual innovation.

- Conclusion
  The Max/MSP module, as a central element within the larger project,

showcases the versatility of Max/MSP in facilitating dynamic, interactive, and immersive musical experiences. Its role extends beyond MIDI interpretation and audio effects to include synchronization with MusicVAE and seamless integration with TouchDesigner for real-time video generation. This synergy elevates the project's artistic and creative potential by unifying audio and visual components into a synchronized and multisensory experience.

2.4 TouchDesigner

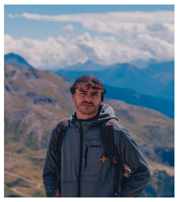
The music generated by Max8 will then animate an audio-reactive visual implemented in TouchDesigner via OSC.
For the occasion, a graphic has been developed.
The idea is to transform a SOP element into two audio-reactive points' clouds.

To do this the incoming audio is processed through an audio analysis process block and from it we isolate in five different channels the presence of low frequencies, middle frequencies, high frequencies kicks and rhythm.
The first cloud, after appropriate transformations, is repeated 4 times as the "outline" of our screen.
It reacts to the presence of low frequencies and low kicks of any instrument.
Low frequencies scale the dimensions of the image, the kick triggers its movement too.

The second cloud is central and represents the melody.
The presence of high frequencies scales the dimensions of the image, the rhythm triggers the movement of it too. The color of the outline clouds and of the central image is modified by the low and high frequencies with a little component of randomicity controlled by white noise.

3 Students

(a)
Andres
Bertazzi
Body Detection


(b)
Riccardo
Kluber
MusicVAE

(c)
Nicolò
Pisanu
Max8

(d)
Marco
Viviani
TouchDesigner

4 Links

GitHub Repository:
`https://github.com/rickykubler/CPAC_Group_7`

Video Demo:
`https://www.youtube.com/`