Music Is People

Creative Programming and Computing

Francesco Piferi, Riccardo Rossi, Ferdinando Terminiello

Professor: **Massimiliano Zanoni**
Assistant: **Luca Comanducci**

A.Y. 2022/2023

POLITECNICO
MILANO 1863

Contents

1 Introduction

Music Is People is a creative application that harnesses the power of collective creativity to compose unique musical pieces. The project revolves around a neural network architecture that generates music based on user feedback. By actively involving users in the composition process, Music Is People aims to emphasize the idea that small individual actions can collectively create something magnificent and inspiring.

Through an intuitive user interface, participants have the opportunity to listen to a previously generated song and provide feedback on their experience. This feedback is then fed back into the neural network, which analyzes and interprets the input to generate a new composition. The newly created song is played for the next user, creating an iterative and collaborative musical journey.

1.1 Objectives

The primary objectives of Music Is People are:

1. Engaging users in an interactive music composition process.
2. Showcasing the creative potential of collective feedback and iterative generation.
3. Exploring the relationship between user feedback and the resulting musical compositions.
4. Fostering a sense of empowerment and artistic collaboration among participants.
5. Conveying the message that individual contributions, no matter how small, can lead to something remarkable and beautiful.

1.2 Artistic Message

At the core of Music Is People lies an artistic message that seeks to evoke a realization among individuals that their seemingly insignificant actions can come together to create something greater than themselves. By involving users in the musical composition process, the project aims to highlight the profound impact that collective creativity can have on the creation of art. It encourages participants to appreciate the value of their contributions, fostering a sense of empowerment, and reinforcing the belief that collaboration and cooperation can lead to extraordinary outcomes.

1.3 Background and Inspiration

Music Is People draws inspiration from the concept of crowdsourcing and collective intelligence. The project is motivated by the idea that diverse perspectives and individual input can contribute to the creation of exceptional artistic expressions. The project team also acknowledges the transformative power of technol-

ogy in enabling collaborative endeavors and aims to explore its application in the realm of music composition.

By building on the advancements in neural network architectures and interactive interfaces, Music Is People seeks to provide a platform for users to actively engage in the creative process and witness the tangible results of their input. The project aims to demonstrate that art can transcend boundaries, unite people, and create a profound emotional impact when rooted in collective action.

2 The Architecture

Here is an image that describes the general functionality of Music Is People.

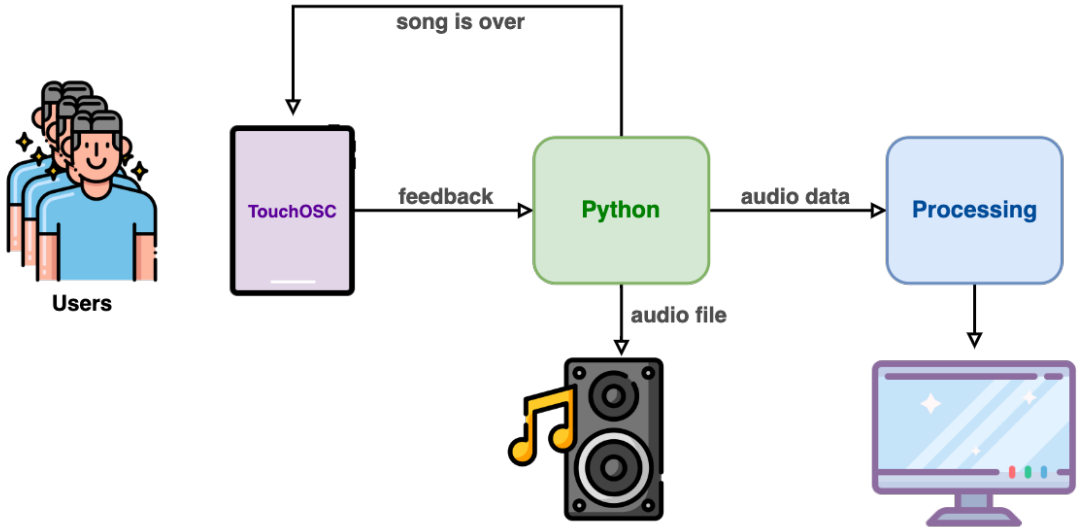

Figure 1: Overall Functionality

Users engage one by one with previously generated songs and provide feedback on their preferences and desired mood for the next composition. This feedback triggers a Python script that utilizes a neural network to generate a new song, which is simultaneously played and sent to a processing application for visualization. Once the current song ends, the next user selects their feedback, and the iterative loop continues, fostering a collaborative and evolving musical experience.

2.1 User Feedback - TouchOSC

TouchOSC is a modular control surface toolkit for designing and constructing custom controllers that can be used on a multitude of operating systems and devices. It can be used on touch-screen mobile devices as well as desktop operating systems using traditional input methods.[1]

With TouchOSC, we were able to develop a simple to use, yet effective Graphical User Interface that could communicate the user's feedback directly to the python

[1]TouchOSC's official website

script via OSC Messages (or MIDI messages, if a cable connection is preferred for stability).

Here is the simple GUI we put together using TouchOSC:

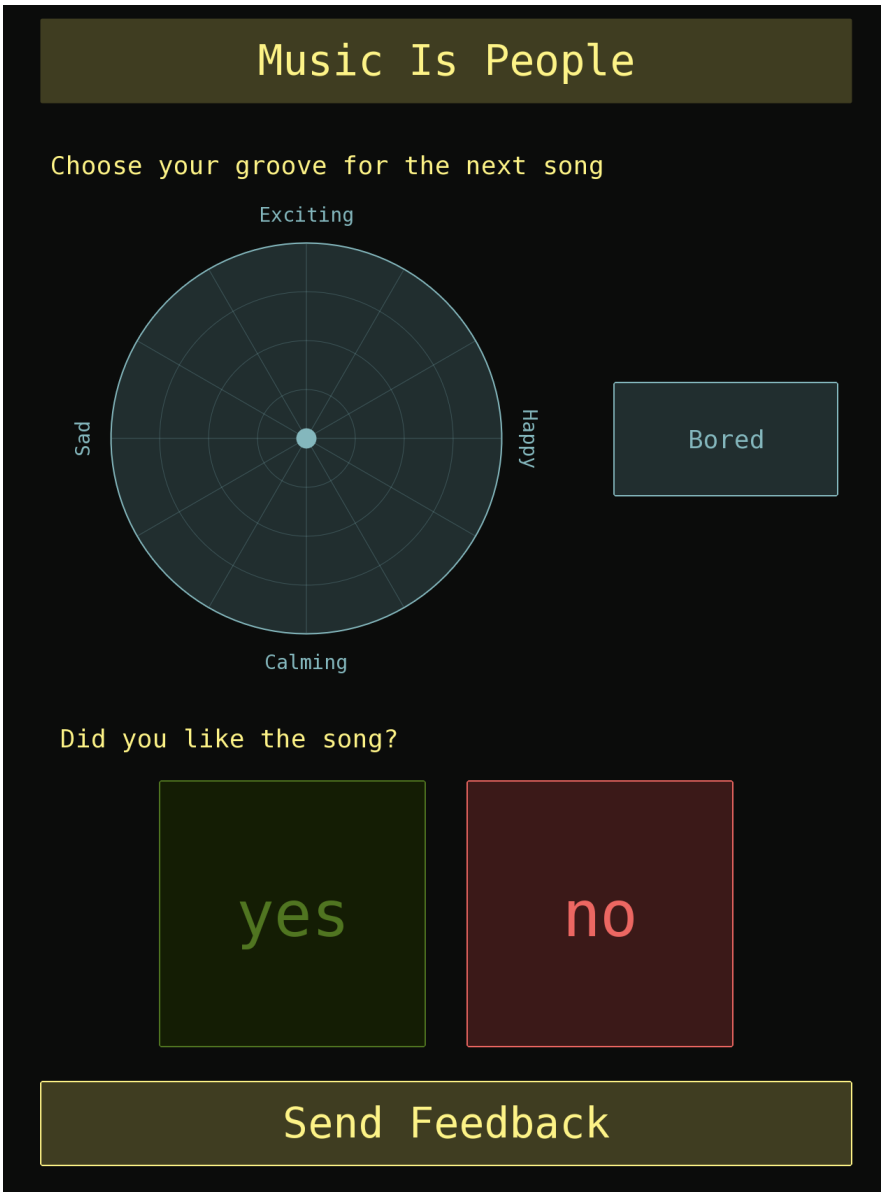

Figure 2: Feedback GUI

The user can move his finger inside the circle to select the mood for the next song (which will be displayed inside the box on the right). The user can also select one of the two options to tell of he liked the song or not. When the user has finished providing a feedback, he can click on the button on the bottom of the GUI to send the feedback over to the python script as an OSC Message.

When the feedback is sent, a new page is displayed (with an animation) to show the user that he has to wait for the song to finish playing before he can provide a new feedback (which technically should be provided by the next user in line).

Figure 3: Waiting GUI

2.2 The Brains - Python

The central processing brain of the application consists in a python script which uses a neural network model to generate a new song based on the user's feedback. The newly generated song is divided in chunks and simultaneously played and sent with a socket for visualization. The neural network used is called MusicVAE and was provided by Google as a publicly available Variational Auto Encoder model. The code used to operate the network is also publicly available in the form of a Google COLAB sheet which we adapted so that it could work locally on our rispective machines. More details on the neural network will be provided in the **Neural Network** section [4].

Here is a scheme to summarize the python functionality of Music Is People:

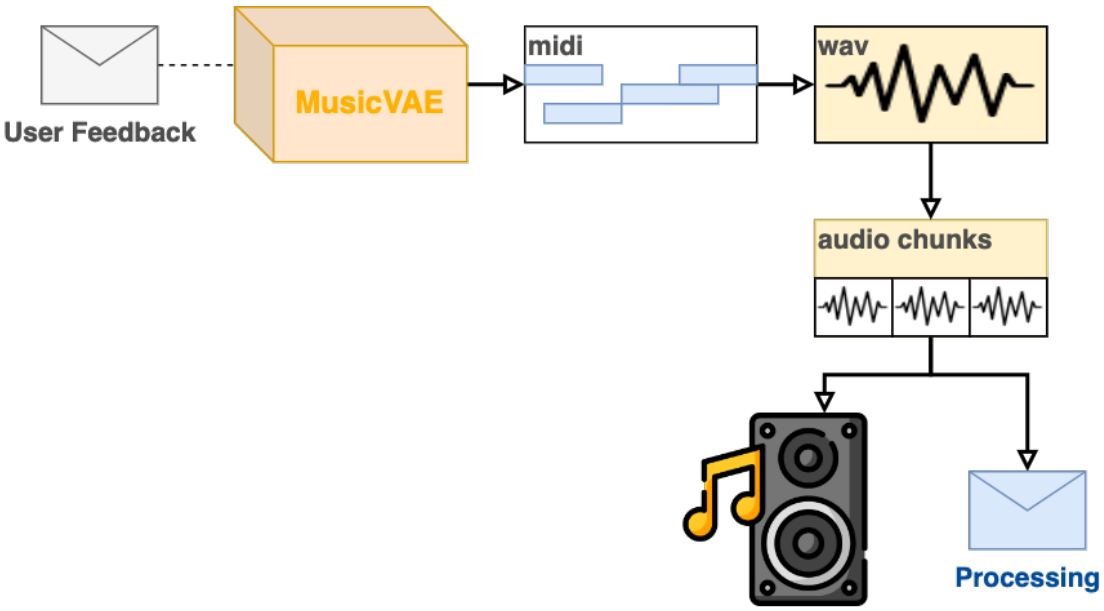

Figure 4: Python Architecture

2.3 Chords Progression Markov Chain

In order to generate music, MusicVAE takes as input a chord progression that is then processed by the neural network and returned as a midi file. This chord progression takes the form of an array of strings, each containing a chord with his tonality. An example of a valid chord progression is:

$$[\texttt{Fmaj7}, \texttt{Cmaj7}, \texttt{G7}, \texttt{Am7}] \tag{1}$$

To enhance the overall user experience, the chord progression to feed to MusicVAE is decided based on the mood that the user picked while in the feedback phase. This makes it so that each mood has its corresponding set of chord progressions, which reflect the mood that the user wants to achieve. If we simply made a list of possible chord progressions for each mood though, there would be two major issues with the system:

1. It would be limited in its ability to create new, interesting patterns (as it would always "move" on the same chords with the same sequence)
2. When changing from one mood to another, it could create unpleasant sounds due to the fact that the previous and next moods' chord progressions would harmonically dissonant.

For this reason, we decided to implement a Second Order Markov Chain that could automatically generate a chord progression based on the two previous chords. This way, the number of possible chord progressions drastically increases and the possibility of having dissonant sounds is greatly reduced due to the fact that the next chord in the sequence is decided based on the previous chords.

For each mood we have a set of possible chords and an associated transition matrix - both for the first and second orders - to tell the probability of going from one subset of chords to the next chord. For example, for a "happy" mood, we have:

- Possible Chords:
  - First order

    `[Imaj7, IImin7, IIImin7, IVmaj7, V7, VImin7, V7/V, V7/vi]` (2)

  - Second order

    `[Imaj7-V7, Imaj7-V7, ...]` (3)

- Transition Matrices:
  - First order

$$\begin{bmatrix} 0 & 0.13 & 0 & 0.13 & 0.42 & 0.13 & 0.13 & 0.06 \\ 0 & 0 & 0 & 0 & 0.25 & 0.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0.27 & 0.09 & 0.09 & 0 & 0.55 & 0 & 0 & 0 \\ 0.16 & 0 & 0 & 0.34 & 0.16 & 0.34 & 0 & 0 \\ 0.16 & 0.16 & 0 & 0.52 & 0 & 0 & 0.16 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

  - Second order

$$\begin{bmatrix} 0 & 0.13 & 0 & ... \\ 0 & 0 & 0 & ... \\ 1 & 0 & 0 & ... \\ ... & ... & ... & ... \end{bmatrix} \tag{5}$$

2.4 Visualization - Processing

A sizable screen showcasing the project's primary graphical user interface finalizes the installation process.

While the user provides feedback, a loading page is presented [5]. After the user finishes compiling the feedback and the Neural Network generates the new song, a fresh page emerges.

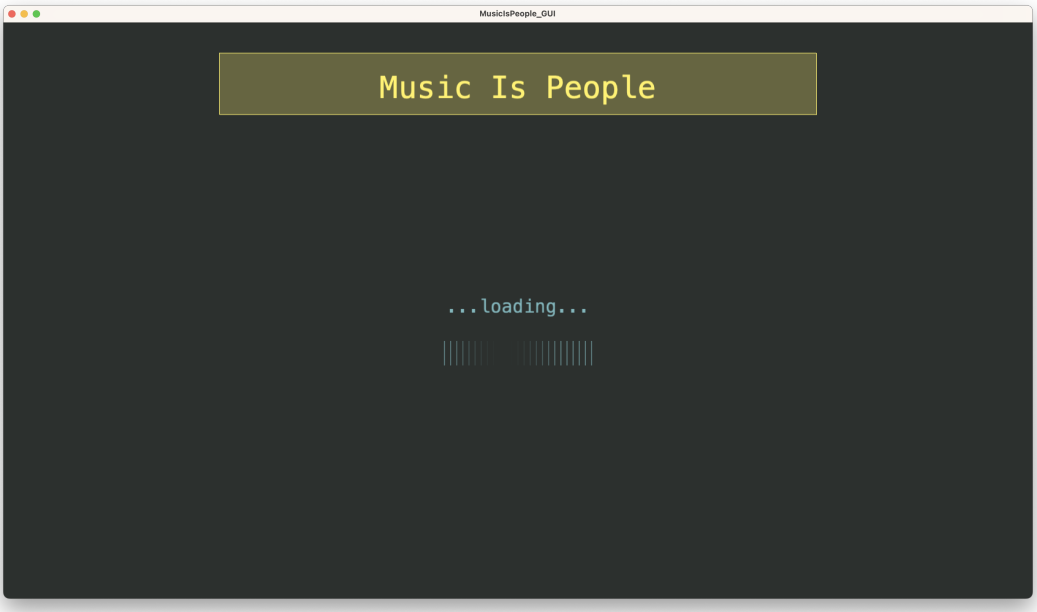

Figure 5: Loading page

As the music begins to play, the screen exhibits a dynamic plane that alters its form in sync with the music. When the song concludes, the loading page is reinstated.

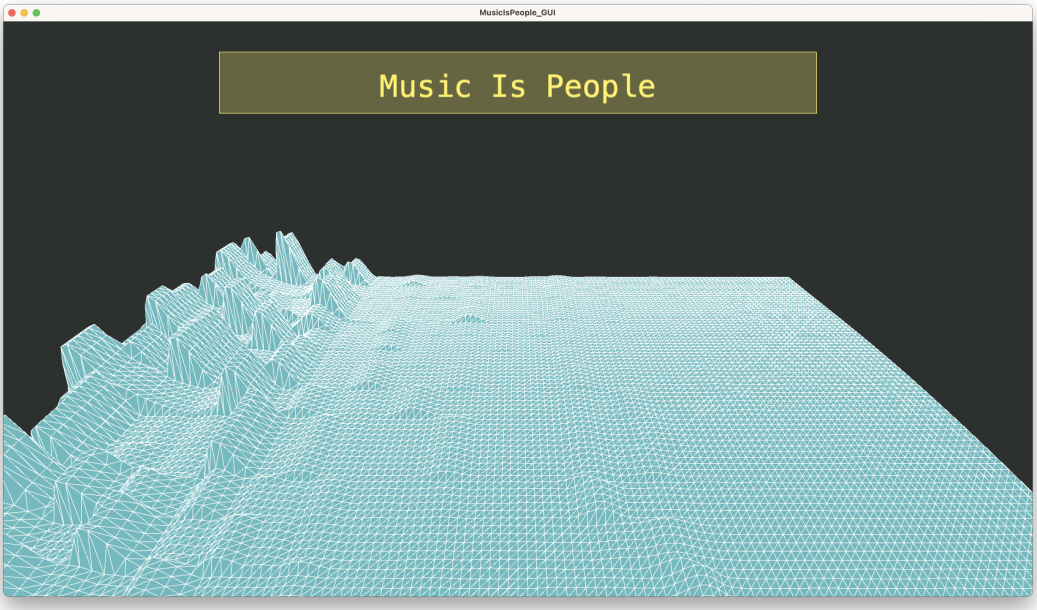

Figure 6: Loading page

3 User Experience

4 The Neural Network

4.1 Variational autoencoders

Variational autoencoders (VAEs) are a type of generative model used in machine learning. They are composed of two main components: an encoder and a decoder.

The encoder takes input data, such as images or sequences of music, and maps it to a latent space representation. This latent space is a lower-dimensional representation that captures the underlying structure and variations present in the data.

The decoder, on the other hand, takes a point in the latent space and reconstructs the original input data from it.

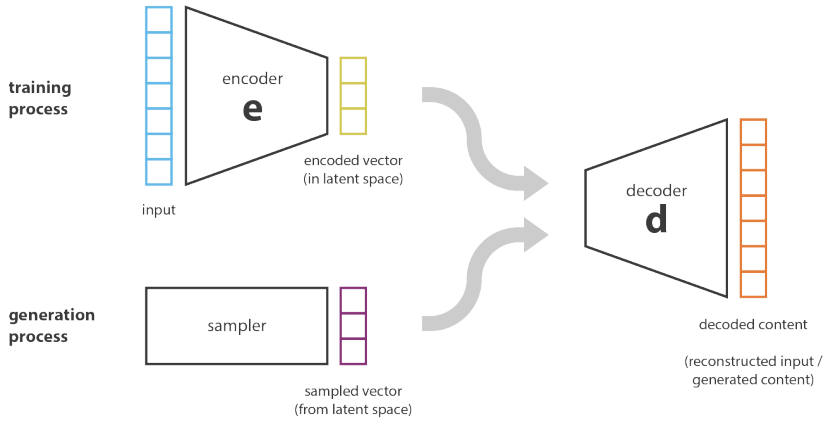

Figure 7: Variational autoencoder

The goal of the VAE is to learn a latent space that captures the essential features of the input data, allowing for effective reconstruction.

The "variational" aspect of VAEs comes from the use of probabilistic modeling: instead of mapping each input to a single point in the latent space, VAEs model the latent space as a probability distribution. This enables the generation of new data samples by sampling points from the latent space and decoding them into the original data domain.

During training, VAEs aim to minimize the reconstruction error while also regularizing the latent space to follow a prior distribution, typically a standard Gaussian distribution. This regularization encourages the latent space to exhibit desirable properties, such as smoothness and continuity.

4.2 MusicVAE

MusicVAE is a neural network model developed by Google's Magenta team, specifically designed for generating musical compositions. It is based on the concept of variational autoencoders.

MusicVAE is trained on a large dataset of musical sequences, learning to encode and decode the latent representations of music. The model captures the statistical patterns and dependencies present in the training data, allowing it to generate new musical compositions that adhere to those learned patterns.

One of the key advantages of MusicVAE is its ability to generate diverse and coherent musical sequences. By utilizing probabilistic techniques during the generation process, the model can produce a range of musical variations while maintaining musical structure and coherence.

The latent space of MusicVAE represents a continuous and structured representation of musical attributes. This allows for various operations such as interpolation and exploration in the latent space, enabling users to navigate and manipulate musical features to generate unique compositions.

5 Music Visualization

5.1 Processing

We developed the main Graphical User Interface of Music Is People using Processing[2] which is a software based on Java mostly used for the creation of artistic installation designed by Casey Reas and Ben Fry in 2001. In our project it receives messages from the Python script. These messages encompass various types, including those signaling the beginning of the song composition, the initiation of playback, and the actual values associated with the song. Through Processing, we effectively handle and interpret these messages to provide a cohesive and interactive user experience.

5.1.1 Socket connections

When the user submits his feedback we utilize a Python code to encapsulate a string message, specifically the string `"CREATING"`, and transmit it via a socket. When the Neural Network has finished creating the song, another message is sent on the same channel, this time containing the string `"START"`. These messages are then delivered to the graphics module, which updates itself with the according page.

We opted for basic sockets over the OSC protocol due to their simplicity, particularly because the message being sent is just a simple string. By opting for sockets, we can effortlessly communicate the necessary information to the graphics module, allowing us to efficiently and effectively update and display the appropriate

[2]Official website: https://processing.org

content. This utilization of sockets streamlines the communication process among the various components involved in the playback, ensuring a smooth and seamless experience.

In regards to the effective message containing the information for visualization, we have chosen to establish another socket connection with Processing. This connection operates on a separate port to prevent interference with the existing communication. Due to how Processing handles OSC messages (concurrently with the update method), we had significant frame drops when trying to use them, hence why we - again - opted for a socket.

The message contains the Short Time Fourier Transform (STFT) of each chunk of audio. These values are then converted from strings (necessary for socket transmission) to floats, enabling their utilization in the visualization process.

When the song is over we send an other message using the first connection, but this time the string is `"STOP"`. Right after receiving the message, Processing will display the "loading" page.

In the figure [8] a summary of what has been said.

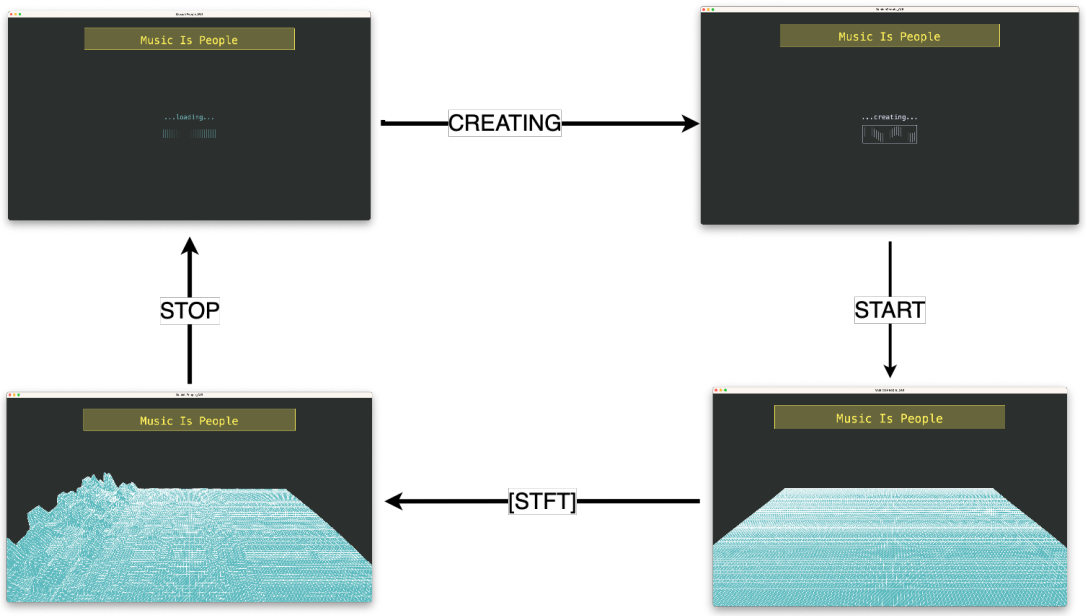

Figure 8: Python-Processing connection (the Python code manages the linkage between the pages.)

5.2 Terrain structure

In order to develop the main page [6], we utilized the 3D rendering capabilities of Processing. This involved defining it within the `setup()` function by specifying `size(width, height, P3D)`. This configuration allows for the inclusion of the z-axis, which represents the amplitude of each wave.

We use a plane mesh 3D structure as the basis of the music's representation. Regardless of incoming data, it follows a sinusoidal movement to avoid stillness in the animation.

The received message has a length of 1025 and needs to be parsed into a float array of the same length. Once the conversion is completed, the array is utilized to compose the waving structure displayed in the images.

Starting from the first line of the plane, the entire line propagates towards the bottom of the screen, creating new shapes with each iteration.

6 Future Improvements

Being the first iteration of Music Is People, there is still much room for improvements in different aspects of the system.

6.1 Multiple Users

Instead of a single user taking decisions for the next song, some sort of voting system could be implemented, where a number of users send their impressions on the previous song and the next song is computed by the average of the user's preferences. This could be implemented via a custom smart phone app (or directly via TouchOSC) and a timing system where users have a short period of time to express their preferences and for the server to collect their votes.

6.2 Advanced AI Model

Since we generate the new songs based on the mood the user wants to perceive, the neural network model could be fine-tuned in order to provide songs that resemble more closely the target mood, which at the moment is only used to generate the chord progression.

6.3 Enhanced User Experience

More freedom could be given to the user in the form of more, different customization options so that the generated music can be further personalized.

7 Conclusions

Music Is People represents a remarkable exploration of collective creativity in the realm of music composition. Through the seamless integration of user feedback, AI-generated music and captivating visualizations, the project is meant to be an immersive and collaborative artistic experience.

By empowering users to contribute their preferences, emotions, and desires, the project can showcase how seemingly small actions can accumulate into something grand and extraordinary. Participants can witness firsthand the transformative journey from individual experiences to a collective artistic expression.

As Music Is People continues to evolve, future improvements and expansions offer exciting possibilities for further enhancing the project's capabilities and impact.

Ultimately, Music Is People could stands as a testament to the boundless potential of human creativity when coupled with technological innovation. It serves as an inspiring reminder that every individual's unique perspective and contribution can shape something extraordinary, transcending the limitations of individual creativity and paving the way for collaborative artistic endeavors that are greater than the sum of their parts.