U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Digital Interactive Systems
2nd assignment - Audio-visual interactions

Sound Sparks

Teacher: Gilberto Bernardes de Almeida

Students:
Yuka Sakai - up202300600
Tiago Teixeira - up202208511

Index

Context and Conceptual Framework..... 1
Related Work and Artists..... 2
Technical Architecture..... 3
    Input Layer..... 3
    Audio Analysis..... 4
    Visual System (Fireworks)..... 4
    Generative Music..... 4
    Interaction & UI Layer..... 4
    System Integration..... 4
Interactive Framework Analysis..... 5
Final Considerations..... 6
References..... 7

Context and Conceptual Framework

This project was developed as the second assignment for the course Digital Interactive Systems, which required us to integrate computer audition and/or generative music, within a system that reacts to user input and produces coherent audiovisual responses.

Sound Sparks is an interactive instrument where users produce any kind of sound events (claps, whistles, vocals), making it accessible for any type of public. The system performs a real-time audio analysis, such as extracting pitch, intensity, spectral features, and timbre patterns, and maps these characteristics to dynamic firework visuals rendered in real time. Simultaneously, the program produces music that responds to and complements those interactions, forming a closed loop where the user acts, the system responds, and the user adapts.

Unlike simple reactive systems where outputs are fixed or predictable, our project aims to exhibit adaptive behavior, adjusting musical complexity, color palettes, firework trajectories, and harmonic structure in response to ongoing interaction. This addresses one of the core challenges highlighted in the course: designing systems capable of continuous interpretation, not merely event-triggering. Therefore, we believe that our project potentially belongs to an exhibition context, where it has the intersection between art and technology as the main theme, reinforcing the need of an audience to make the art happen.

In terms of the conceptual framework, the project explores:

- How human-generated sounds can be transformed into meaningful aesthetic output;
- How generative music can respond to multimodal cues (amplitude, pitch, temporal activity) to create a sense of agency and co-creation;
- How real-time audio systems can calibrate difficulty and feedback intensity through adaptive thresholds, temporal suppression, and activity-based evolution of music patterns;

Related Work and Artists

Martin Daigle & Pauline Patie

This project performs real-time timbral extraction using machine learning and maps those features to live visual augmentation. It shows that qualities of sound such as the timbre’s brightness and roughness can shape the optical effects. It also helped confirm that audio-visual links don’t have to be literal, and that using short analysis windows is useful for capturing quick gestures such as claps or snaps. This reveals how timbral characteristics can drive more nuanced and musically expressive visuals than amplitude-only systems, inspiring the color mapping to the pitches, and size mapping from intensity, used in our project.

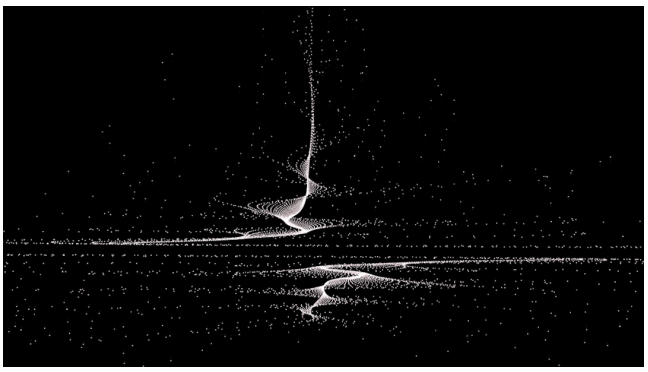

Figure 1: Real-time Timbral Analysis for Musical and Visual Augmentation

HapticSOUND: Interactive Learning Digital Musical Instrument

This system combines gesture detection with sound synthesis and highlights the importance of multimodal feedback in learning and engagement. It reinforced the value of maintaining a continuous feedback loop in which users receive meaningful visual and auditory responses that encourage exploration, motivating features such as the adaptive music complexity system and a temporal cooldown to preserve feedback clarity. The key is that interaction quality depends not only on recognition accuracy but also on the responsiveness, clarity, and pedagogical value of the feedback, which directly shaped the UI design and the spectrum/pitch visualizations.

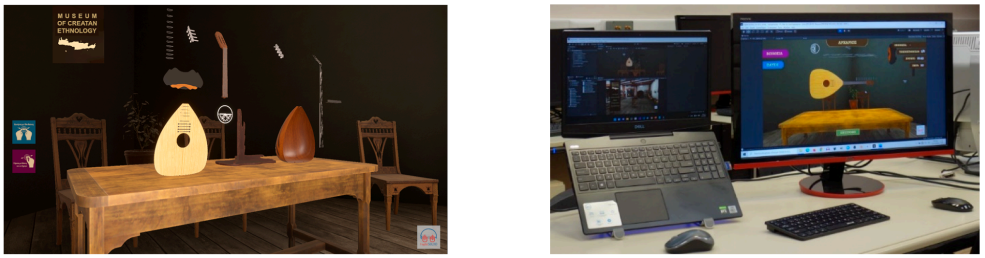

Figure 2: HapticSOUND system and interface

Shimon the Robot

This creation demonstrates how a system can listen to human vocal input and generate musically coherent responses using generative algorithms. Its influence shaped the approach to musical co-creation in this project, focusing not on random generation but on music guided by user pitch, dominant frequency zones, and interaction tempo. The generative music thrives on semantic relevance and parameter-driven mutation, which inspired the evolving melody and bass patterns as well as the mode-shifting behaviors implemented in Sound Sparks.

Figure 3: Shimon rap-battling Dash Smith & Shimon performing “Gospel in Space”

Technical Architecture

Our project integrates audio input, real-time analysis, visual rendering, and generative music into a unified interactive system. Each layer processes user sound and translates it into visual and musical responses, creating a clear feedback loop. The following sections outline the core components and how they work together to produce the system's reactive behavior:

Input Layer

- Microphone input via p5.AudioIn()
- Adjustable sensitivity (manual gain control with ↓ and ↑ arrows)
- Input suppression after sound events to prevent feedback loops
- FFT (64-bin) analysis
- Temporal buffering (250 ms windows)

Audio Analysis

- Pitch detection through:
    - Peak frequency extraction
    - Harmonic pattern analysis
    - Optional ML5.js model
- Feature extraction:
    - Amplitude envelope
    - Spectral centroid (brightness)
    - Harmonic ratio (timbre indication)
- Event detection:
    - Peak detection
    - Temporal windows
    - 300ms cooldown

Visual System (Fireworks)

- Particle system
- Pitch → color mapping
- Intensity → particle count, speed, size
- Spectral brightness → spark dispersion
- Gravity simulation and glow effects

Generative Music

- Four synthesizer layers: bass, melody, harmony, percussion
- Activity-based mutation:
    - Firework frequency → increased tempo and complexity
    - Pitch dominance → melodic contour shaping
    - Low-frequency events → modal shifts

- Pattern evolution every 8 seconds
- Voice-leading and chord progression system

Interaction & UI Layer

- Status overlay
- Real-time spectrum, pitch readout, meters
- Keyboard controls (sensitivity, toggles, resets)

System Integration

Therefore, all modules communicate through a shared event system:

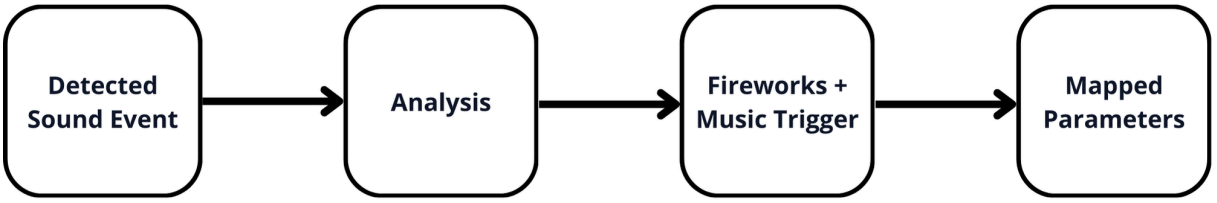

Figure 5: Event Sequence

This creates a continuous and adaptive multimodal loop, emphasized by the diagram below:

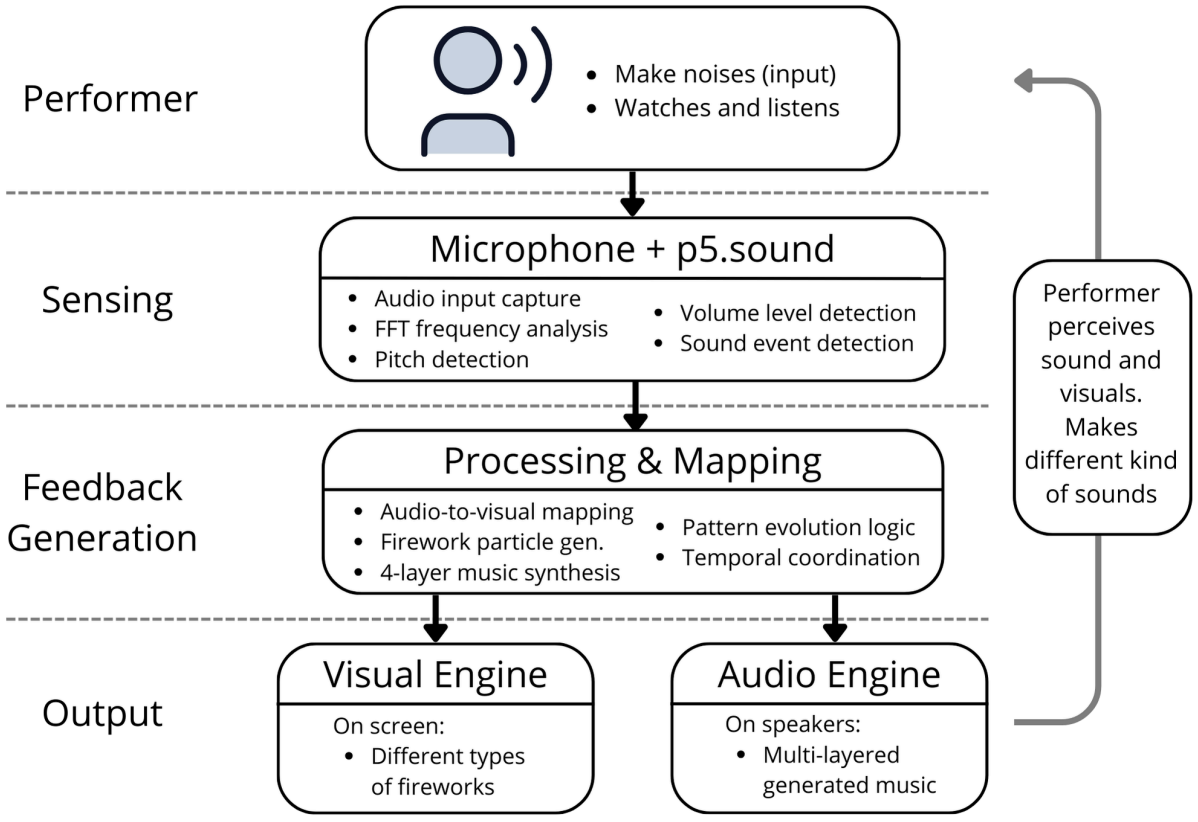

Figure 6: Architecture Diagram

Interactive Framework Analysis

Bongers' Interactive Framework Loop describes interaction as something that develops through a recurring exchange: users perceive, they act, the system reacts, and this reaction feeds back into new perceptions. In Sound Sparks, this loop starts when the participant produces a sound, whether a clap, a whistle, or a spoken voice, which the microphone captures and converts into interaction data.

From there, the program interprets the incoming audio and produces a set of responses that appear visually as fireworks and fluctuating sound meters, and sonically through the generative music produced by the medium. Instead of being a final output, this system behavior becomes the next cue for the participant. Seeing the meters shift and new firework patterns unfold, it encourages them to try different types of sounds, explore variations in pitch, or otherwise experiment with how their actions shape the environment.

Through this ongoing exchange of events and reactions, the participant develops a stronger sense of involvement and connection with the digital system, maintaining a dynamic cycle of perception and feedback that reflects the principles outlined in Bongers' framework.

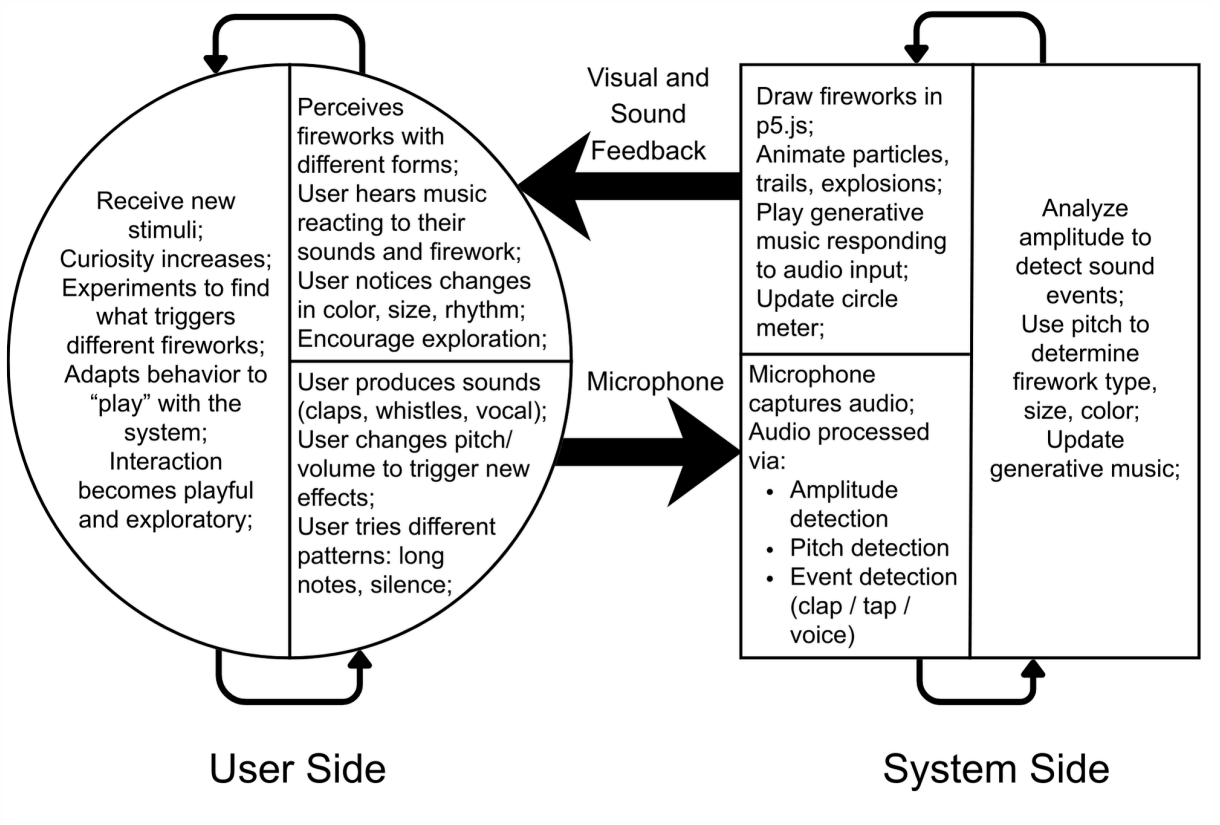

Figure 7: Bongers' Framework

Final Considerations

During the development of the project, we tried to aim for a result that could be pleasing to interact with. The kickoff idea was the easiest part. Even having its own uniqueness, fireworks are an event that we only witness in a specific context, and sometimes they are not even appreciated due to its sounds and air pollution. Thus, we decided to recreate them in a playful interactive way that could also be delightful.

However, we went through some challenges, in which the most difficult laid on the complexity of generative music, something we didn't have experience with. Even though the operation with the microphone was also new, we struggled to make the generative responses be something gratifying and symphonic, and not just random notes mixed in a program. By trial and error, we based our system to be synthesized in four layers (bass, melody, harmony, and percussion) to composition variety, and overall input analysis for the "randomness", achieving our intentions in a possible range.

To conclude, we believe that, while we succeeded to obtain a very interesting result under the topics of computer audition and generative music, there are some features that could be further developed or introduced:

- **Gesture-based interaction through a camera**: In addition to analysing sound, the system could scan user gestures and add a new layer of music.
- **Machine-learning classification of sound types**: This feature would allow for a bigger exploration in music generation.
- **More advanced harmonic analysis or improvisation systems**: Given more time and freedom for complexity, the system's audio analysis and music uniqueness could be improved.

References

https://timbreandorchestration.org/writings/project-reports/real-time-timbral-analysis
https://www.mdpi.com/2076-3417/13/12/7149
https://richardsavery.com/project/shimonraps
https://www.youtube.com/watch?v=wcft3gcspzs
https://www.shimonrobot.com/

Prompts used in Chat GPT:
“Hi, I was thinking of making a project that, through sound (like claps and taps maybe, and/or voice) and its pitch, launches different types or sizes of fireworks that come with music notes. Do you think that idea suits the project where I have to merge computer audition and generative music?”

“Could you help me to build a step by step of how to make the project so I can organize myself? I would like to start with the computer audition part first (starting the microphone, sound test, pitch test, visual response, etc) and then generative music, and I will be using Web/HTML display coding in p5.js.”

Prompts used in Claude:
“I need to build an interactive digital system where sound input (claps, taps, voice, pitch) triggers visual fireworks, starting with initializing the input p5.AudioIn and test microphone volume and display a simple visual meter (bar or circle).”

“I'm noticing that even the loudest noises I can make are being captured as something rather small. I think it is something related to microphone sensitivity or the algorithm itself for pitch, so I will need to put some threshold to the sensitivity or something alike, what would you suggest?”

“Great :), now let’s implement the FFT and/or pitch detection (e.g., ml5.pitchDetection or p5.FFT) so we can detect amplitude peaks to trigger the firework, and detect pitch to determine type/size/color of firework.”

“I got this error as i loaded the page:
Uncaught (in promise) TypeError: ml5.pitchDetection is not a function at initializePitchDetection (sketch.js:66:26)
at setup (sketch.js:47:3)
at _setup (p5.min.js:2:465993)
at _runIfPreloadsAreDone (p5.min.js:2:465046)
at g._decrementPreload (p5.min.js:2:465251)
at g.<anonymous> (p5.sound.min.js:2:98292)”

“I believe there might be a problem related to the sound capturing. Take a finger snap for example. It first produces a low sound and then a high one. I feel like the system has been capturing an interval that is too short leading to fireworks only appearing as red (which makes sense, since most noises are being captured as low and stopping there).”

“Could you make the intensity of overall explosions a bit lower?”

“Let's move on to the next part: Generative Music Layer. For this part, let's start by adding a background generative music engine using p5.PolySynth, and we will have to make it subtle: music should feel like a reaction to fireworks.”

“I reloaded the browser window and I got this printed in the console:
5500/favicon.ico:1 Failed to load resource: the server responded with a status of 404 (Not Found) sketch.js:403
Uncaught ReferenceError: pitchBuffer is not defined at updateMusicParameters (music.js:31:3)
at updateReactiveMusic (music.js:19:5)
at draw (sketch.js:184:3)
at e.default.redraw (p5.min.js:2:542863)
at _draw (p5.min.js:2:466768)”

“Im not getting any music. This is what I get in the console when I start:
Checking ML5 availability...
sketch.js:144 ML5 not loaded, using FFT-based pitch estimation
sketch.js:801 Legacy music system initialized
sketch.js:830 Advanced generative music system initialized
favicon.ico:1 GET http://127.0.0.1:5500/favicon.ico 404 (Not Found)”

“If I make a sound sometimes loud enough, the system picks it up and makes a sound for the music. but that same sound for the music is picked up and used again for the music. Which creates temporary loops.”

“I have a question. I'm noticing that while I have visuals on, when a firework explodes, the text available from pressing H flashes for a moment. Why?”