POLITECNICO
MILANO 1863

Music And Acoustic Engineering

Computer Music - Languages and Systems
Homework #3
Group: Algorhythmics - A.Y. 2022/2023

YASC

Group Members:

Alice Sironi
Cecilia Raho
Stefano Ravasi
Yan Zhuang

https://github.com/polimi-cmls-23/group5-hw-ID-Algorhythmics

Contents

1 Introduction

The goal of the project is to develop an interactive recorder performance tool, incorporating Nintendo Joy-Con controllers (fig. 2) via web interface for gesture-based inputs and SuperCollider for sound synthesis. The system is able to emulate the recorder experience with real-time, responsive feedback to guarantee an immersive musical performance.

Figure 1: YASC concept

2 Features and GUI

YASC software mainly consists of two parts: the web system and the sound synthesis engine. They are controlled by the user with Joy-Con controllers.

2.1 Joy-Con controllers

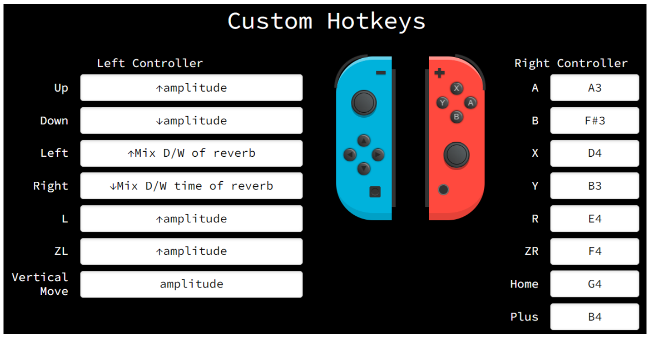
The Joy-Con controllers (fig. 2) consist of two individual units, each containing an analog stick, an array of buttons, a gyroscope and motion sensors (accelerometers). In this project, the left controller is used to control synthesis parameters and the right one is responsible for playing notes.

Figure 2: Joy-Con controllers

2.2 Web system

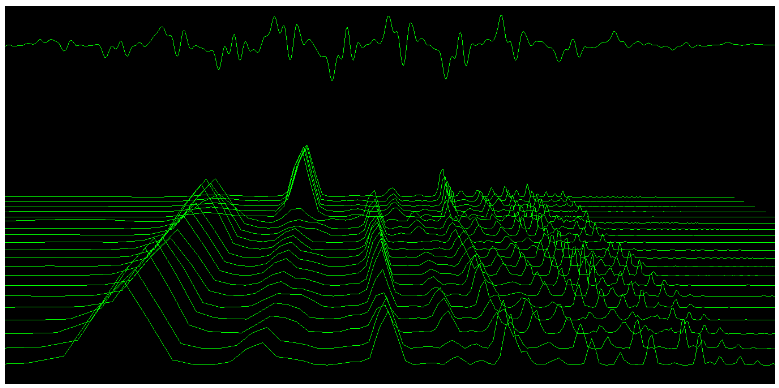
In the web system, users can customize buttons in the hotkey page, improve coordination by playing the Dino game and finally play a song in the performance page.

- The *hotkey page* (fig. 3a) serves as an interface for users to personalize their recorder experience. Through this page, Joy-Con controller buttons can be mapped to different functions: in particular, the left controller is responsible for adjusting recorder parameters in SuperCollider while the right controller is dedicated to play notes.
- The *dino game page* (fig. 3b) serves as an interactive training tool designed to help users familiarize with their custom hotkeys: the jumping action is triggered by the correct Joy-Con button (previously mapped in Hotkey Page). Note that, as the game progresses, the speed increases to raise the challenge level.
- The *performance page* (fig. 3c) provides an immersive audio-visual experience by representing the output sound spectrum. Additionally, a short Mario audio audio sample is prepared for beginners to follow.



(a) Hotkey page detail

(b) Dino game detail



(c) Performance page detail

Figure 3: Web system interface

2.3 Sound synthesis

The recorder sound is generated by SuperCollider. The GUI is divided into four sections (fig. 4).

- The upper part contains the attack and release parameters of amplitude envelope.
- In the second part, there are knobs to control frequency and depth of vibrato.
- The third sections offers the possibility to modify mix between dry and wet signals, time and pre-delay of reverb.
- In the bottom part of the interface, there is a slider to control the output volume and a simple

graphic tool which shows the volume level of recorder sound.

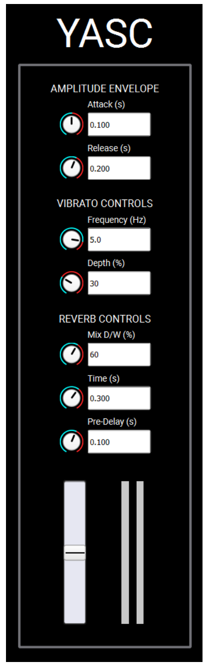


Figure 4: SuperCollider interface

3 Key Implementations

3.1 Overall structure

Referring to the overall structure (fig. 5), the web system tracks interaction with Joy-Con controllers and use their serial messages as input. The web system acts as the central interface, providing users with an intuitive and interactive tool to set up and customize their performance experience. SuperCollider serves as sound synthesis engine: it generates the sound according to the commands sent from web system via Open Sound Control (OSC) protocol. Finally, the audio output can be captured by web system to provide a visual feedback.

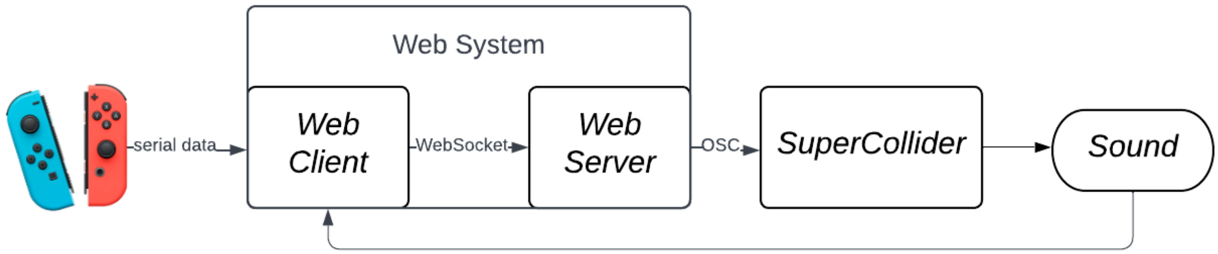


Figure 5: Overall structure

3.2 Web system

There are three main components which run the web system:

- *Parse and Listen*: processes serial data from Joy-Con controllers and translates them into understandable commands. It maintains the connection with the controllers and cts as a message center, registering listener-response events for web pages and cleaning up when the page is left.
- *LocalStorage*: a type of web storage that exists in the browser and provides a relatively small amount of space (5-10 MB) for storing data locally. This leads to faster access times compared to backend databases such as MySQL. Hotkeys configuration changes are stored in it and trigger an update to a global buffer event.
- *Express server*: serves static files (like images and HTML files) and handles OSC messages transmission from the website using WebSocket.

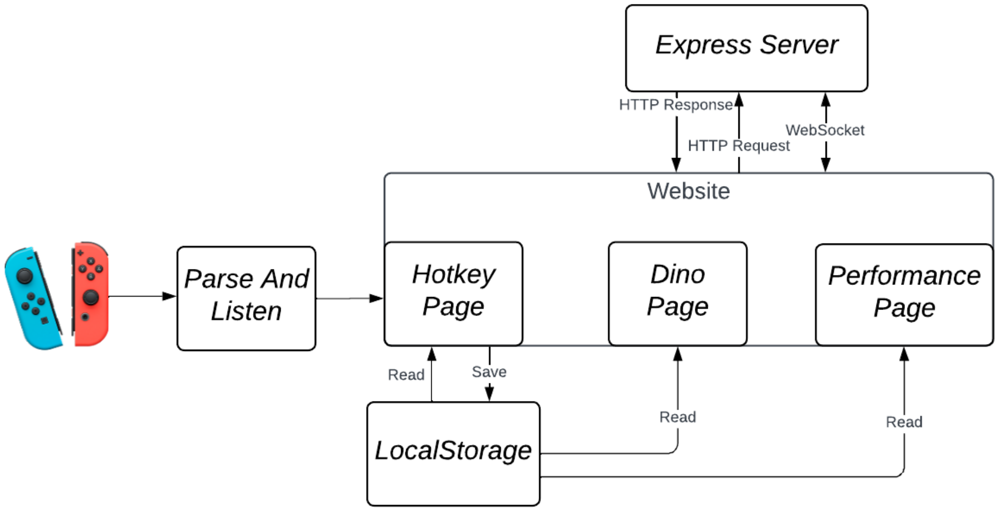


Figure 6: The structure of web system

3.3 SynthDef *flute*

The SynthDef "flute" emulates the sound of a recorder through digital waveguide synthesis. This model is based on Perry Cook's one with the addition of a simple vibrato.

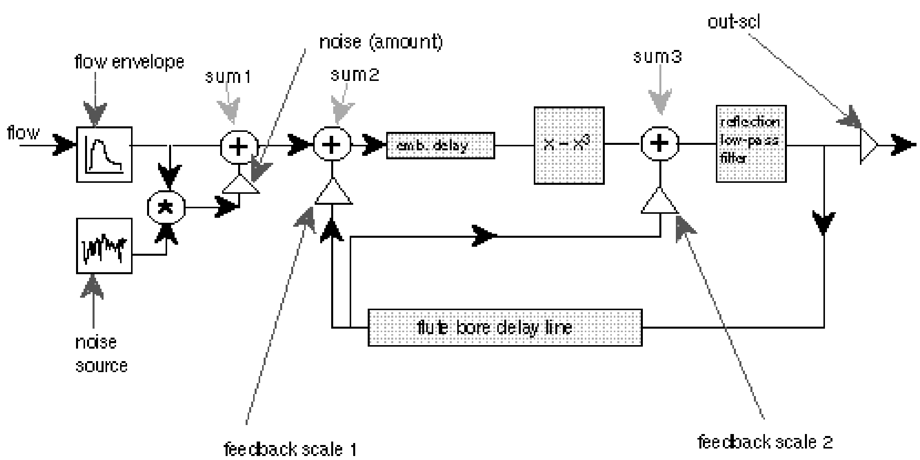


Figure 7: Perry Cook's digital waveguide flute model

The synthesis procedure is summarized by the following steps:

- The initial sound (*sum1*) is generated through a clipped noise modulated by an envelope and summed with a low frequency sinusoidal oscillator to simulate vibrato.
- The second sound (*sum2*) is obtained as the sum of the first one and the actual output processed by a cubic interpolated delay line (used for bore effect emulation, one period's worth of samples long), modulated by a coefficient smaller than one (to prevent instability in the feedback loop).
- The second sound (*sum2*) is processed by another cubic interpolated delay line (used for embouchure effect emulation, half period's worth of samples long) and the cubic polynomial $x - x^3$ is computed for each sample: this result is then summed to the actual output (processed as described in the previous point) modulated by a new small coefficient (always smaller than one for the same reason stated before) to give the third sound (*sum3*).
- Finally, a low-pass filter is applied to simulate the natural decay of high frequencies in a flute sound.

4 Conclusions and further improvements

The current implementation gives satisfying results: dino game playability and visualization effect are enjoyable, the system is easy to control and intuitively understandable for the player.

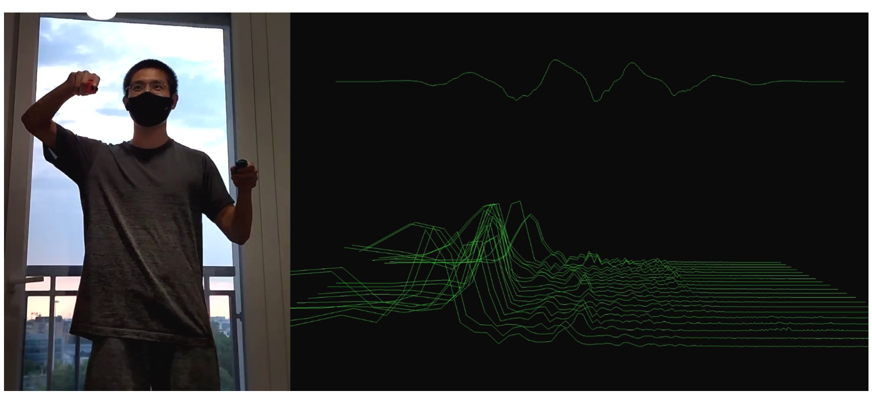


Figure 8: Actual performance example

The following pictures show similarities between the synthesized sound and the real instrument one in terms of spectral components. The main differences are noticeable in low frequencies because the breath noise is simulated.

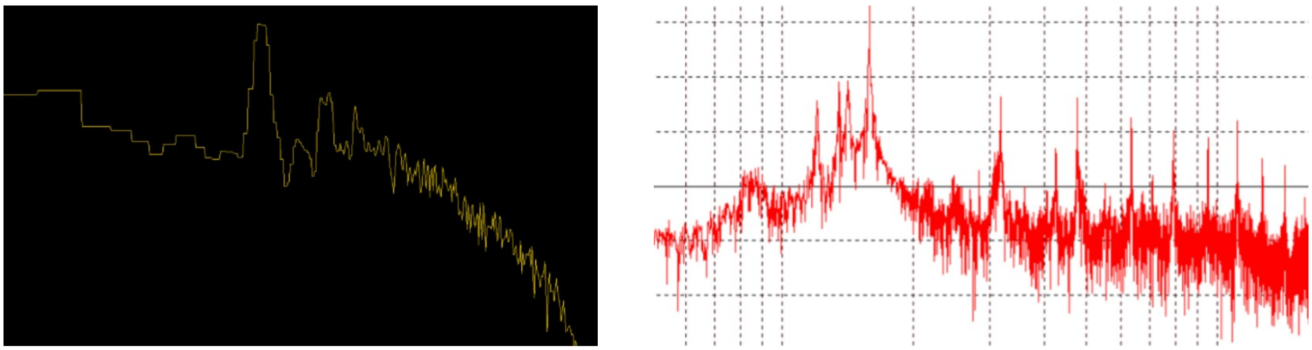


Figure 9: Synthesized sound spectrum compared to real flute spectrum

However, the pursuit of excellence is a continuous journey and the following ideas could be implemented in future.

Regarding the web system:

- Consider the use of JUCE with WebSocket to further reduce latency.
- Explore the possibility of integrating with other motion-sensing devices or MIDI controllers, to expand performance options.
- Implement features that allow users to record, save, and replay their performances.
- Implement a Madgwick filter for improving motion tracking.

Regarding the sound synthesis:

- The waveguide model doesn't behave well at high frequencies. In the short future, multi-modal or multi-dimensional models could be adopted to solve the issue.
- Offer more sophisticated sound parameters for users to manipulate in SuperCollider.
- More accurate results might be achieved by incorporating higher-order non-linearities to simulate the wave bouncing behaviour in the real instrument.