UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

_ TELECOM ESCUELA
TÉCNICA VLC SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

APLICACIÓN HTML5 BASADA EN WEB AUDIO Y VÍDEO EN WEB PARA SOUNDCOOL

Diego Villar Sánchez

Tutor: Jorge Sastre Martínez

Cotutor: Roger B. Dannenberg

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2020-21

Valencia, 21 de julio de 2021

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

VLC/
CAMPUS
VALENCIA, INTERNATIONAL
CAMPUS OF EXCELLENCE

EMAS
Gestión
ambiental
verificada
REG.NO. ES-CV-000030

Abstract

The technology is constantly developing and expanding. Soundcool is the perfect example: a project creation system that seeks to be more than just a unidisciplinary tool.

The Soundcool system was designed with the aim of being in the classroom, being a complementary alternative to classical musical instruments. After the passage of time, its full potential has been verified and the project grows, having more and more applications -not only in music-. At present, it can be affirmed that Soundcool is a multidisciplinary project that is present in the health, educational, musical, and entertainment fields. This work presents the Minimum Viable Product of Soundcool for the browser, being an error-free version prepared for its launch and commercial use. For this, the current version has been tested and consequently the errors have been corrected; without adding any extra functionality. The project has used the JavaScript language combined with HTML5 and CSS, with the React and Redux libraries being very present. Thus, throughout the document the context of Soundcool is exposed and the pertinent modifications are addressed, explaining the methodology and its practical application. Finally, the project conclusions are presented and possible improvements are suggested.

Keywords: Soundcool, collaborative music system, browser, modular system, OSC, HTML5, JavaScript, CSS, React, Redux

Resumen

La tecnología está en constante desarrollo y expansión. Soundcool es el perfecto ejemplo: un sistema de creación de proyectos que busca ser más que solo una herramienta unidisciplinar.

El sistema Soundcool fue ideado con el objetivo de estar en las aulas, siendo una alternativa complementaria a los instrumentos musicales clásicos. Tras el paso del tiempo se ha comprobado todo el potencial que tiene y el proyecto crece, teniendo cada vez más aplicaciones -no solo en la música-. En la actualidad se puede decir que Soundcool es un proyecto multidisciplinar que está presente en el ámbito sanitario, educativo, musical y del entretenimiento. En este trabajo se presenta el Producto Mínimo Viable de Soundcool para navegador, siendo este una versión sin errores preparada para su lanzamiento y uso comercial. Para ello, se ha testeado la versión actual y en consecuencia se han corregido los errores; sin añadir ninguna funcionalidad extra. En el proyecto se ha utilizado el lenguaje JavaScript combinado con HTML5 y CSS, estando muy presentes las librerias React y Redux. Así pues, a lo largo del documento se expone el contexto de Soundcool y se abordan las modificaciones pertinentes, explicándose la metodología y su aplicación práctica. Finalmente, se presentan las conclusiones del proyecto y se sugieren posibles mejoras.

Palabras clave: Soundcool, sistema musical colaborativo, navegador, sistema modular, JavaScript, HTML5, CSS, React, Redux, OSC

Resum

La tecnologia està en constant desenvolupament i expansió. Soundcool és el perfecte exemple: un sistema de creació de projectes que busca ser més que només una eina unidisciplinar.

El sistema Soundcool va ser ideat amb l'objectiu de ser a les aules, sent una alternativa complementària als instruments musicals clàssics. Després del pas del temps s'ha comprovat tot el potencial que té i el projecte creix, tenint cada vegada més aplicacions -no sols en la música-. En l'actualitat es pot dir que Soundcool és un projecte multidisciplinari que és present en l'àmbit sanitari, educatiu, musical i del entreteniment. En aquest treball es presenta el Producte Mínim Viable de Soundcool per a navegador, sent aquest una versió sense errors preparada per al seu llançament i ús comercial. Per a això, s'ha testat la versió actual i en conseqüència s'han corregit els errors; sense afegir cap funcionalitat extra. En el projecte s'ha utilitzat el llenguatge Javascript combinat amb HTML5 i CSS, estant molt presents les llibreries React i Redux. Així doncs, al llarg del document s'exposa el context de Soundcool i s'aborden les modificacions pertinents, explicant-se la metodologia i la seua aplicació pràctica. Finalment, es presenten les conclusions del projecte i es suggereixen possibles millores.

Paraules clau: Soundcool, sistema musical col·laboratiu, navegador, sistema modular, Javascript, HTML5, CSS, React, Redux, OSC

Contents

List of Figures

Chapter 1

Introduction

The ability to evolve is a virtue of human beings, and the evolution of their creations is an inherent characteristic. Applied to the creation and development of software, it is always expanding and there are more and more languages, different methods and new tools.

Most applications or platforms on the web are born with a clear and defined objective, but if they are successful they expand and cover other areas. There are many examples from many different scopes, but perhaps Amazon is the clearest: starting as an online bookstore, right now it is one of the largest online retail sellers and web service provider.

Another example, this time less well known, and setting aside the differences, is Soundcool. This application can be defined as a system that is constantly reinventing itself and looking for new opportunities. Thus, after its success in collaborative music creation, it is expanding into areas such as medicine or education. The great reception that Soundcool has had all over the planet, meaning several collaborations and prizes, has made this quality jump possible.

The Soundcool project is discovering new needs that its platform could cover. In this way, Soundcool can be very versatile: the system could be adapted to a platform in which an opera can be played; its basic functions could be redefined to become an attractive alternative to paper books, and also be connected to interactive blackboards; or the system could deploy and manage a session of the treatment of a neurodegenerative disease such as Alzheimer's.

Thus, after mentioning all the potential of the system and the already tangible applications, in this work a step, within the road map, will be developed. The aim is to improve the transition to the Soundcool website as a collaborative musical creation tool.

The limitations of this project only lie in the time and its financing, since the team is mainly made up of students who want to contribute in a small way.

1.1 Motivation

The COVID-19 pandemic has revealed how important personal relationships are to society. In this way, the virus has been able to spread and infect many people over several waves. That is why we have had to adapt and manage the absence of physical contact. New technologies have undoubtedly been the lifeline during this year and a half, because, thanks to teleworking and video calls, the routine has been followed, as well as possible.

Despite all that the pandemic has entailed, COVID-19 has been a springboard for technologies that were not a priority. And, platforms like Zoom, Microsoft Teams or Skype have never been so powerful or have had so many users. After all, what these programs allow is interaction between individuals, and this is what has been pursued during 2020 and part of 2021.

The motivation for this project lies in this maxim: connecting people with people. Soundcool is the perfect example of a collaborative project, aiming to expand the use of new technologies to generate, edit and produce music creations. Moreover, looking at the trajectory and all the potential it has, the future is promising. Soundcool is getting involved in education, entertainment and medicine. In some way Soundcool wants to help society move forward, and being part of the project is an honor.

1.2 Objectives

Regarding the objectives of this thesis, there is a main objective, which is none other than obtaining a Minimum Viable Product. In addition, an exhaustive explanation of its operation, characteristics and functionalities will be carried out.

To achieve that, a set of more specific objectives will be executed:

- Understand how the system works beyond the code. That means: protocols, information flows, security systems and user usability.

- Get used to the actual web page and its open code from GitHub. Most of the code is already done, so it is important to understand its building and its logic.

- Learn to use React, an specific JavaScript (JS) library used in the project, and Redux, a predictable state container for JS apps.

- Mastering the development tools used: GitHub, Visual Studio Code, Google Chrome Debugger and Redux DevTools.

- Find the errors and correct them. There are many ways to make a correction, but simplicity and effectiveness will always be sought.

After defining the objectives, it can be seen that the work consists of two different parts, which are related but independent. First would be the explanation and description of the operation of the system, and, secondly, the correction of errors and in the case that it would be necessary, some code optimization.

1.3 Thesis structure

This section explains the structure of the thesis, according to the order of the index and the table of contents.

Chapter 1 is the introduction to the thesis. In this section the context of the project is discussed, introducing the key concepts, as well as its motivation and objectives. The structure of the thesis itself also appears at the end.

In chapter 2 reference is made to the state of the art. In this way, the alternatives of the Soundcool project are studied and then the weaknesses are analyzed. Weaknesses or points for improvement are listed in this section. Finally, the proposal is presented and is explained that it is directly linked to the object of the thesis.

Chapter 3 covers everything related to Soundcool. The first section (3.1) talks about history and how it has grown; section 3.2 explains the current Soundcool product and how it is presented. Next, sections 3.3 and 3.4 technically explain how the elements are built and how the application works, with a detailed explanation of the OSC system. Finally, section 3.5 explains the nature of the project and, relating it to Soundcool, what is expected to be achieved.

In chapter 4 the design of the solution is discussed. To do this, the tools that are going to be used are first analyzed, this involves a brief explanation of each one and, if necessary, a comparison. In addition to the tools discussed, the structure of the code itself is detailed, highlighting the essential elements for its operation.

In Chapter 5 the implementation is developed. Since the essence of the work lies in the correction of errors, it is explained one by one how and why the modification has been carried out.

In Chapter 6, as the final chapter, the conclusions obtained are discussed and a series of proposals are suggested.

Chapter 2

State-of-the art

Soundcool is a modular, collaborative and educational online tool but there are some related works launched in to the market already.

Before explaining such alternative systems, it is worth explaining some details of Soundcool. It is modular, which implies that the limits are only set by creativity and imagination; creative construction of instruments, sound processing and interactive audio-video performances, can be carried out by its modules. Furthermore, it is collaborative allowing multiple users to control it through tablets or smartphones (besides the desktop). Finally, Soundcool is educational; modules come ready made with interface.

Regarding education, EarSketch and Blocky Talky are also good tools. In first place, EarSketch teach students how to code through music. The platform let users to make their own compositions while they learn Python or JS. The principal advantages are that there is no needed coding experience, users can use their own sounds and it is free and online [1]. In second place, BlockyTalky performs a very interesting didactic work. The app uses an intuitive programming language for starter developers that allows users to build a wide range of projects, from networked bird feeders to computer music systems [2][3].

Concerning web audio education, iMuSciCA is the most innovative and groundbreaking platform [5]. The project comes from European Union´s Horizon 2020 programme, and its principal aim is to support mastery of academic content on STEM subjects (Physics, Geometry, Mathematics and Technology) through their engagement in musical activities.

Finally, regarding the alternative tools that use Web Audio, three stand out: Synth Kitchen, Zupiter and the Web Audio library modules themselves. These three tools are very similar to Soundcool in terms of user-application interaction, since they are modular. Nevertheless, it is true that they have certain limitations: Synth Kitchen only allows 5 types of effects; Zupiter does not have cloud storage; and the real issue with Web Audio is it is a programming system[4].

2.1 Criticism of the state of the art

The project started in May 2019 and since then there have been more than 7 developers collaborating. Consequently, the code is an amalgam of different styles that without a good guide it is almost impossible to follow and understand the logic. An example is the installation of the project, because even having a guide, it is difficult to get it to work for anyone who is not a professional programmer.

The purpose of this work is none other than to correct current functionality errors, respecting (as far as possible) the project structure and the system configuration. In this way, this section will comment on aspects that could be improved, being these not critical, but recommended. Thus, these suggestions are not intended to change the code, since it is already written, but rather to highlight the current situation of the code to put its difficulty in context.

Here is a list of the most striking features of the code:

1. There is no clear naming convention declared. When defining the names of constants, functions and methods, it is important to follow the same style. In this way, someone who has never used the code can get a clear idea without spending a lot of time looking for what does what. There are many conventions, such as ”camelCase”, ”PascalCase”, ”snake_case” or ”kebab-case”; in the project there is a mixture. Throughout the code, there are jumps between ”camelCase” and ”snake_case”.

2. Good code development practices include the use of native elements, and avoiding other native elements. React is a JS-based framework, but it has its own native elements. In this project it is sometimes taken into account but sometimes not, depending on the style of the developer. An example would be the use of ”class” when defining a class; ”className” would be better.

3. Like the Backend folder do not have Frontend files, this methodology should be extended to the rest of the code. An example could be in the routes code, where there are defined the handlers too.

4. There are many global variables that should be avoided. JavaScript language has three ways to declare a variable: var, const and let. Var is used in very specific events, and it has to be the exception, not the norm.

5. Talking about sql code, there can be complex queries in which there can be logical operations. As a best practice, it is usually assumed that the requests that are made are not trustworthy (emulating a malicious user). To achieve this goal, some variables should be *escaped* within the query by using a question mark (?). In this way, and giving great importance to order, each of the variables is analyzed to avoid security breaches

6. The code has not been optimized for routes. It can be seen in the files dedicated to the routes that the authentication token is checked in all functions. In the case of the implementation of a middleware, that is, the incorporation of a software that is located between the operating system and the application, it would be avoided to be repeatedly checking said token. That would mean an improvement in the fluidity of the system. An implementation that seems

viable would be the use of this tool, the middleware, before the routes.

7. Some variables are stored in cookies, and not session variables. This complicates communication with the server and data management. If they were session variables and you want to change something, it is updated instantly; but since they are cookies, the data is already saved and the cookies themselves would have to be manually updated. In this way, if you initialize the system, even if you register with a user, you only have access to the files in the cloud, as there will be no files uploaded.

8. In some cases, in order to be more comfortable for developers, the style of some elements is defined in the .jsx file (React) itself. Not in all the code, but there are some parts where a dedicated CSS file could have been created.

2.2 Proposal

The proposed proposal is directly linked to the objective of this Final Degree Project, which is to achieve a Minimum Viable Product for Soundcool in the browser version. The work, and in essence the proposal, goes beyond the introduction of new functionalities or style corrections, but seeks a version in which there are no functionality errors to launch on the market and test on a large scale.

Chapter 3

Soundcool

In this chapter it is going to be described everything related to Soundcool. From start to date, and even the direction and goals for the future.

3.1 History of Soundcool

It all started in 2012, with the creation of an own degree from the Polytechnic University of Valencia (UPV): ”Electronic music, electroacoustic and video creation”. The promoter of this studies, Jorge Sastre, along with other students and teachers, began to work with research projects on the man-machine relationship in education. In this way, they began to shape an idea that tried to change the traditional conception of music education.

In 2013 the idea began to materialize: Soundcool had obtained the support of the UPV and it was already a project. In order to research and develop ideas, the Spanish Ministry of Education, Culture and Sports facilitated Jorge to carry out a research stay at Carnegie Mellon University (USA). It was there where he met Roger Dannenberg, popularly known as the co-founder of Audacity.

Throughout 2015, it was not only present in some Spanish schools, but thanks to the Erasmus + K201 program, Soundcool was used in countries such as Italy, Portugal or Romania. Thus, such had been the success that the program had, it had been repeated another two times (2015, 2015-2017, 2018-2019)

In 2016 took place the first great performance with the exclusive use of Soundcool, La Mare dels Peixos, premiered in Valencia, Spain; This work was re-presented in 2019 [6]. In addition, this year the project was awarded with the 2016 SIMO Education Prize and received financing from the Daniel and Nina Carasso Foundation.

The Soundcool project received the NEM Art award in 2017 of European creative industries of the New European Media association. The value of the project was also recognized through the Telefónica Chair. In addition to both awards, Soundcool was used at the Barcelona electronic music festival Sónar 2017 and at the Kikk Festival in Namur in Belgium.

2018 was the year that Soundcool took the step to other platforms: Augmented Reality (AR). For the presentation of this proposal, ”HoloSound” was created, which was premiered at the MEVArt 2018 Festival of the UPV[1]. The work was created by the artist and composer Stefano Scarani, a member of the team. The play HoloSound was also represented at the World Science Festival in New York.

As an anecdote, Soundcool will appear for sure in the history books, this is due to the fact that in 2018 the Guinness record was achieved for the ”greatest music lesson in history”, after more than 4,000 students (between the ages of 8 and 16) participated in a class introduction to the Soundcool system.

The historical thread of Soundcool and how it has been financed appears on the project’s own website[2].

3.2 What is Soundcool

Soundcool is a modular collaborative creation system. This means that for its configuration and proper operation, use is made by predefined modules. In order to run it, it is necessary to download the application from the official website[3]. Currently, the latest version released is Soundcool 4.0.

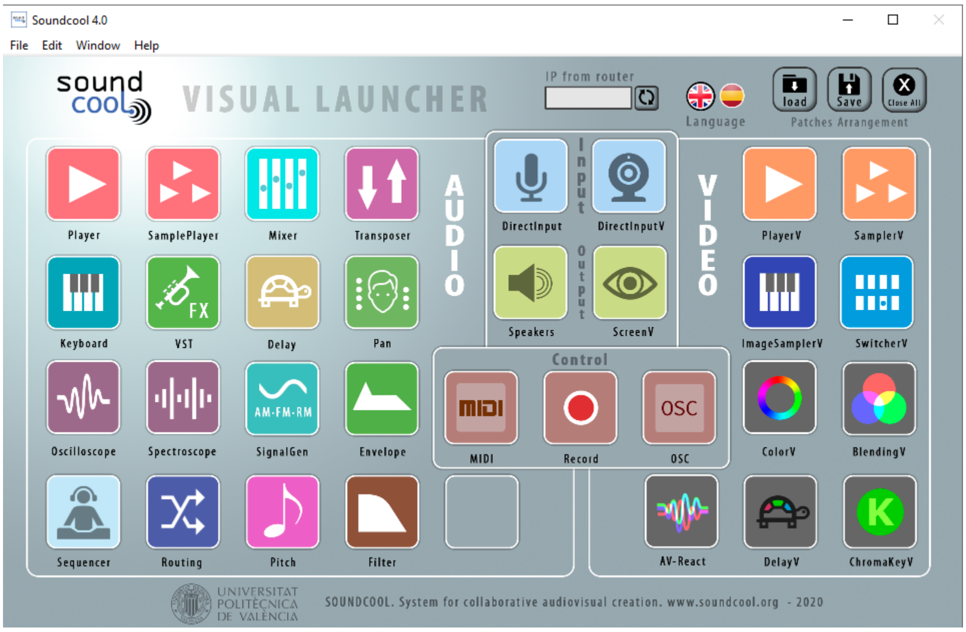
The modules can be broken down into three groups: audio, video and control. The first group, respectively, consists of 16 modules: Player, SamplePlayer, Mixer, Transposer, Keyboard, VST, Delay, Pan, Oscilloscope, Spectroscope, SignalGen, Envelope, Sequencer, Routing, Pitch, and Filter. The video group, as its name suggests, is aimed at managing images with audio, and they are: PlayerV, SamplerV, ImageSamplerV, SwitcherV, ColorV, BlendingV, AV-React, DelayV and ChormaKeyV. Finally, the control group is made up of the MIDI, Record and OSC functionalities, as well as the output and input of signals: DirectInput, DirectInputV, Speakers and ScreenV. In Figure 3.1 can be seen the distribution of these buttons in the interface.

Each module has a specific function; for a more detailed explanation, there are tutorials for each functionality on the official website of Soundcool;. However, the potential of Soundcool lies in the combination of modules; to learn more about all the possible combinations, García Talavera, L. made a whole project explaining the best module mixes [7]. Each of them, depending on its function, has an output or an input (in or out button), which allows them to be connected to each other. In addition, the system allows a multiple connection, without the need for it to be in a chain, that is, one ”out” can be connected with several ”in”.

One of the virtues of the application, in addition to the versatility provided by the modules, is that you can use your own sounds. In other systems there are usually predefined sounds, but in Soundcool you can play with the sounds that are on the computer.

The interface on the desktop has already been explained, but the part that makes Soundcool dynamic and collaborative is provided by the app for tablet or mobile. This application is available for Android and IOS, in their respective stores, and in the Figure 3.2 you can see its interface. It has a simple and clear design, where it

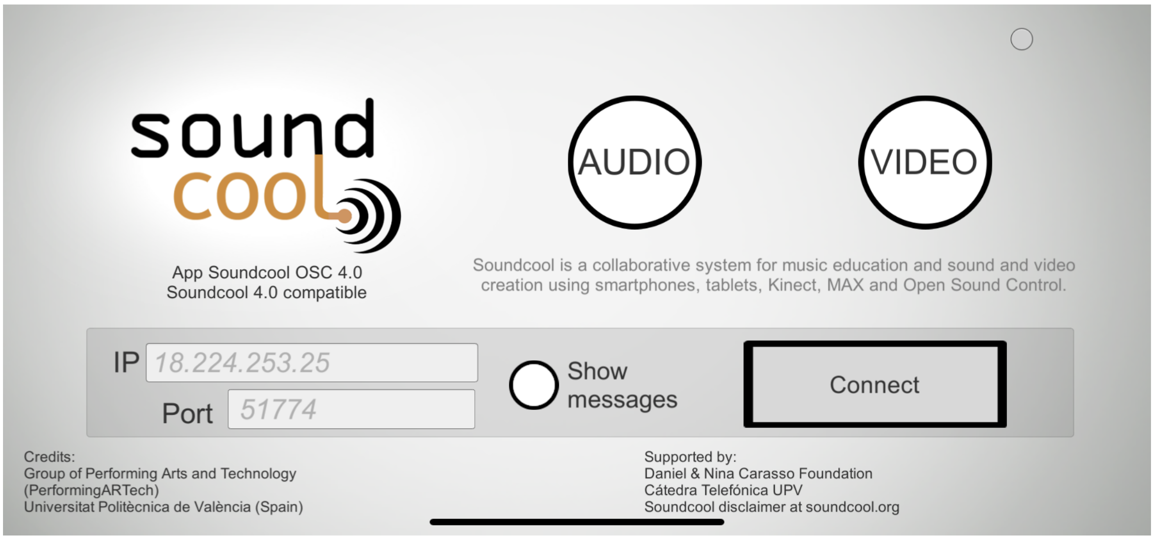
[1] https://youtu.be/ZJF4kl8-TDI
[2] www.soundcool.org/financiacion
[3] www.soundcool.org



SOUNDCOOL. System for collaborative audiovisual creation. www.soundcool.org - 2020

Figure 3.1: Soundcool‘s 4.0 interface

asks the user for the IP, port and mode (audio or video). If the connection is correct, the circle in the upper right corner will appear in green, otherwise it will appear in yellow warning the user of a problem.



App Soundcool OSC 4.0
Soundcool 4.0 compatible

Soundcool is a collaborative system for music education and sound and video creation using smartphones, tablets, Kinect, MAX and Open Sound Control.

IP 18.224.253.25
Port 51774
○ Show messages

Credits:
Group of Performing Arts and Technology
(PerformingARTech)
Universitat Politècnica de València (Spain)

Supported by:
Daniel & Nina Carasso Foundation
Cátedra Telefónica UPV
Soundcool disclaimer at soundcool.org

Figure 3.2: Soundcool‘s 4.0 app interface

Depending on the chosen mode, one type of functionalities or others will be accessed, so the audio and video cannot be managed at the same time from the same device. The appearance of each module on the mobile device is identical to the desktop modules and (if the connection is correct) they are synchronized.Allowing in this way an endless number of collaborators

3.3 How is it made

Currently, the Soundcool platform is built using different technologies, one type aimed at the desktop and another towards the mobile application.

The desktop platform is based on the MAX environment, with the addition of the MSP / JITTER extensions, developed mainly by the Cycling'74 company; despite not being open, the code allows the adoption of new routines created by third parties. This software is specific for the production of multimedia content and does not need a professional background knowledge to be able to develop it, indeed, it is a visual programming language [8]. MSP is an extension of MAX, allowing digital audio to be manipulated in real time and allowing the creation of synthesizers and effects - processors by users. In the same way, JITTER supposes another extension of Max, but this time of video. On this occasion, the new software allows to process video, 3D and arrays.

Actually Soundcool has a similar functioning as MAX, furthermore, both work with interconnected modules (in MAX, connections are more explicitly shown using lines, whereas in Soundcool, inputs are marked with labels to indicate the source). What Soundcool provides are the predefined modules, since MAX is a programming language and Soundcool a teaching tool. How has been developed Max in Soundcool‘s project can be seen in the Final Degree Project of Calabuig Benítez, L [9].

Regarding the mobile application, Unity3D has been used mainly. The reason lies in the versatility it offers and the simplicity of the software tool, since it allows compiling on iOS and Android without making modifications. The amount of possibilities that Unity3D offers is enormous, as are its components; the ones that have been used in this project are: Text, Image, Button, Slider and Toggle. The document by Moncho Moncho, S. describes how the app has been implemented and the use of each of the components [10].

In addition, Unity3D has also been used for AR development, specifically with the ARFoundation framework, which includes the augmented reality libraries from Apple and Google, making compilation easier. The way in which this part of Soundcool has been carried out is documented in the memory of Garijo Contreras, P [11].

3.4 How does it work

It has already been explained what Soundcool is and how is built, now it will be explained how it works. The encoding used in the project is Open Sound Control (OSC) for communication between devices; on the official website of OSC: www.opensoundcontrol.org, they do not define it as a protocol, but as an encoding.

3.4.1 About OSC

OSC was created around 2002 by the Center for New Music and Audio Technologies (CNMAT) for the networking of synthesizers, computers and multimedia devices in general. The community of developers of the musical environment uses this encoding as an evolution to the Musical Instrument Digital Interface (MIDI). Although OSC was not designed to solve all the drawbacks of MIDI, OSC has multiple advantages [12], such as:

- OSC has intuitive names when assigning points in the mapping, using the URI format.
- MIDI has a limited number of names, all possibilities are 0-127 on channels 1-16. The limitation of OSC lies in the channel capacity and saturation, beyond the naming.
- About the address ranges, OSC namespace feature can define a wide range of points in a single message, in other words, it uses a pattern matching language allowing multiple clients to receive a message.
- Simultaneity in the effects of encapsulated messages.

Apart from the advantages of one system over another, the main characteristics of OSC are the following:

- OSC messages are most commonly sent via UDP, but OSC does not specify any particular transport.
- The message format allows for the machine-independent transmission of integers, floats, strings, and other data types.

In Figure 3.3 you can see the composition of a generic OSC message, what is divided in three parts. The first is dedicated to the address, and its format is very similar to the path of a Unix file system. The second part is for data type management. And finally, thirdly, the space dedicated to the message.

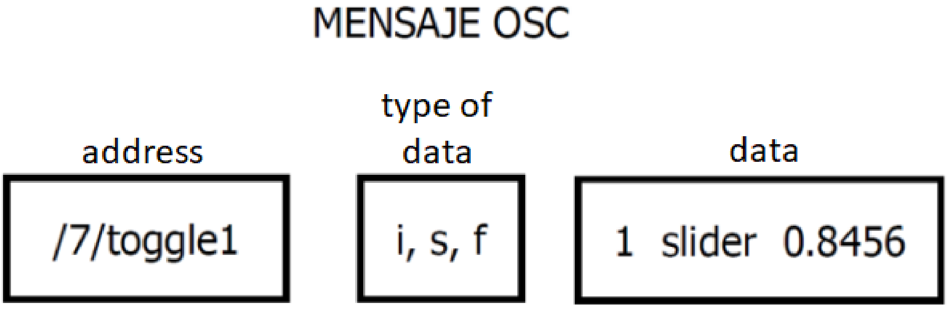


Figure 3.3: Generic OSC message

3.4.2 OSC applied to Soundcool

After explaining what OSC is, it will be shown how it is applied for this work. In a first version TouchOSC was used, but in subsequent versions a specific script for Soundcool was made. This particular code was made with the UniOSC tool, developed by Unity, which implements the OSCsharp library. Due to how the application was developed, it has two different scripts, one dedicated to the desktop - device connection, and the other for the control elements of the modules [10].

Regarding the connection between devices, first, an IP is assigned to the project, which will have the name of IP OSC and unless you want to modify it, it will be the IP of the host. Each Soundcool module is independently assigned a different port. In this way, a multitude of devices can each control a module. As Dannenberg R. wrote in one of his reviews about Soundcool [12], the idea in the future is to automate this task (assignment of IP and ports), the solution he proposes is to use Zeroconf [14].

Afterwards, Soundcool can be used locally or telematically. An example of a local use could be a music class, in which the host (the teacher's computer) manages all the session while each of the devices connects to it, controlling a module. Another option is remote use, in which each of the participants can use a device, but also needs a computer on which to have the session. In this case, it is necessary to make a copy of each of the messages (OSC) that any device emits seeking for the maximum synchronism. In Figure 3.4 are defined the types of protocols used; TCP in order to avoid loosing data, and UDP to reduce the delays [13].

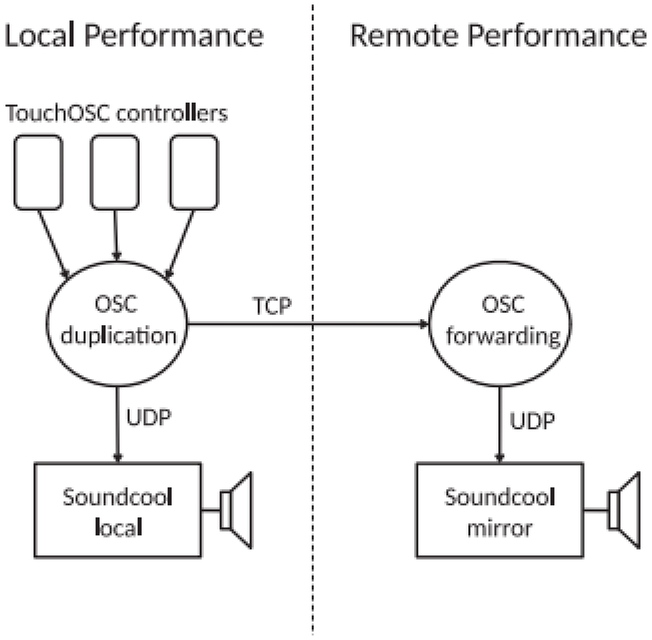


Figure 3.4: Telematic performance of Souncool

3.5 Minimum Viable Product

After the 2020-2021 pandemic, it has been possible to verify the potential that Soundcool has in being able to be telematic. The downloadable application is in version 4.0 and its good performance has been verified; the next step that the Soundcool team wants to achieve is to develop the project in a browser version.

This new version aims not other than to be more comfortable for users. Being able to have files in the cloud linked to a profile and being able to use it from another device, share the compositions created with other profiles, reduce latency with getting rid of screen delays or obtain audio files directly from YouTube, are just some of the advantages of this new format. So, this project, which is based on AudioWeb, has been under construction for a long time and has the same functionalities as the downloadable application plus some advantages such as those already described.

The work involved in this project is to get a product that could already be put on the market and could be tested. Thus, without having to add any more functionality, the bugs or errors that currently existed had to be solved.

The instructions are simple: test the modules and different functionalities, and together with the bugs already discovered, fix them. This list of required fixes is posted on the Wiki of the project repository on GitHub[4] . The list of the bugs is going to be described in the appendix.

[4] www.github.com/rbdannenberg/soundcool/wiki/Alpha-Release-Task-List

Chapter 4

Solution design

The solution has been designed based on the project already developed, and there has not been much room for maneuver since the objective is to correct errors, not modify the nature of the project.

Soundcool for the browser uses the javascript framework itself as a base tool: ReactJS, in addition, as complementary tools there will be: HTML5 and CSS 3 for design and layout; JQuery as an external library for data processing and animations; and MySQL as database language, using its Workbench as database manager.

In the current market there are other types of tools with which these solutions can be implemented. Next, the tools used are analyzed and the alternatives are presented; all this based on the analysis, comparisons and user experiences.

4.1 Tools analysis

Right now, browser Soundcool development is written in JS, HTML5 and CSS. It should be noted that within the JS environment a very specific library called React is used. In addition, it makes use of the Redux library. These two libraries are often used together.

JavaScript

JavaScript is one of the most widely used programming languages in the world along with Java, C and Python. JS mainly serves to create effects and perform interactive actions, that is, it allows you to implement complex functions on web pages, displaying static information for you to see with content updates or creating interactive maps, 2D / 3D graphics animation, etc.

Some of the characteristics of this language are: it is dynamically typed, meaning, it uses classes and objects as structures to organize itself; it is non-typed, so it is not necessary to specify the data type; moreover, it runs on the client's own machine through the browser [16].

React

React library is an open source and was devised by Facebook developers in 2010 and the prototype was created in 2011. In 2014 the library was already expanded and lectures on it began to be given; since then, the number of users who use it has only grown. This library aims to help creating interactive user interfaces, generating simple views for each state of your application, as defined in the official page[1]

The main element of the React library is the component, which in turn has its own state. This state is a representation of the component at a specific time. Another significant difference is that React creates its own DOM, that is, it creates a virtual one called ReactDOM. In this way, only the part of the DOM that has been modified is updated and not the entire real DOM.

The great alternative to React is Angular. This Javascript framework is one of the most used to create Web applications. Its characteristics include the use of Typescript, the possibility of using declarative templates, applying dependency injections and creating reusable components. In Table 4.1 are described the main differences between both frameworks [15].

COMPARATIVE TABLE	REACT.JS	ANGULAR.JS
Author	Facebook	Google
Type	Open code	Open code
Language	JSX	JavaScript,HTML
Learning curve	Low	High
Packaging	Strong	Weak
Positioning	Server side	User side
Application architecture	Flow	MVC
Data link	Unidirectional	Bidirectional
Universality	Web and mobile	Web and mobile
Companies using it	Apple-Nike-HBO	Twitter-Netflix-Uber
Directives	Complicated for outsiders	Easy with JS knowledge
DOM	Virtual DOM	Regular DOM

Table 4.1: Comparative table React vs Angular

Redux

The Redux tool is also an open source library and its main mission is to manage the state of the applications. This library works by having a ”Store”, ”State”, actions and reducers. The Store centralizes the storage of information, the State is read-only and to make changes to the State is through actions and reducers [17].

MySQL

MySQL is an open source relational database management system with a client-server model. A database is nothing more than a collection of structured data, and the client-server model refers to the relationship between the computer or device (client) and where the data is stored (server). Relational databases (software called

[1] www.es.reactjs.org/

RDBMS) work through the use of several tables and offer a methodical view of the data; RDBMS is the most standardized and used system.

The MySQL competitor par excellence is PostgreSQL. This alternative system for database management is characterized by its high performance, broad support for a large number of languages, support for distributed databases and its object-relational operation, in which it treats data as an object. The main drawback for a project like Soundcool, with a minimal budget, is the fact of having to buy the license. In Table 4.2 both options can be compared [18].

COMPARATIVE TABLE	MySQL	PostgreSQL
Type	Open code	Open product
Implementation language	C/C++	C
Storage engine	Multiple	Single
Full-text search	Yes(limited)	Yes
Data types	Only SQL-standard types	Advanced types
IP address data type	No	Yes
Table inheritance	No	Yes
Connection Scalability	New connect., new thread	New connect., new process
Analytic functions	No	Yes
Programming languages	SQL:2003 syntax	Perl, Python, SQL, JS, etc.
ACID	Not always	Yes

Table 4.2: Comparative table MySQL vs PostgreSQL

HTML5

HTML5 (HyperText Markup Language) is the fifth revision of the HTML language. This language together with CSS3, and to a certain extent JS, defines the new web development standards, redesigning the code to solve problems and thus updating it to new needs.

The advantages of this version are mainly a greater interaction between our web pages and the media content (video, audio, among others), as well as greater ease when coding our basic design. The most significant features of HTML5 would be: first, local storage on the client side; second, integration of scalable vector graphics (SVG) to replace generic ones; and thirdly, new multimedia elements such as audio and video, for example [19].

CSS3

CSS (Cascading Style Sheets) is a graphic design language for defining and creating the presentation of a document. Along with HTML and JavaScript, CSS is the third component for the creation of most websites, specifically CSS allows the creation of visually attractive pages, user interfaces for web applications and GUIs for many mobile applications.

CSS is essentially designed to differentiate content from the document's presentation form, mainly colors, fonts and layouts. This language seeks to improve the accessibility of the document, provide more flexibility and control in the specification of presentational characteristics and reduce the complexity and repetition of code in the structure of the document.

JQuery

jQuery is an open source, cross-platform JavaScript library. This library aims to simplify the task of programming in JavaScript and allows you to add interactivity to a website.

jQuery is largely plug-in and essentially a Document Object Model (DOM) manipulation library. The DOM is a tree-structured representation of all the elements on a web page. Plugins, free or paid, can be created by third parties to solve specific situations in relation to the layout of a web page.

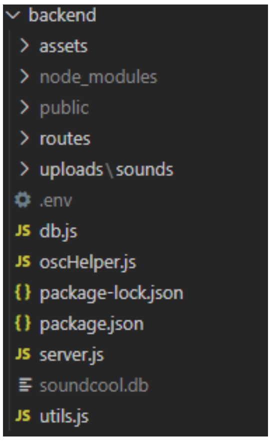
The main features of JQuery are: first, it allows the separation of JS and HTML, so these two languages can be programmed in different files, making the code much more readable and understandable; second, JQuery encourages clarity and brevity with abbreviated names and chainable functions; in addition, it manages inconsistencies between browsers in order to eliminate incompatibilities; finally, extensibility stands out, since new functionalities can be added easily and quickly removed [20].

After the analysis of the current tools and a brief comparison, it has been concluded that the type of software used is justified for this project and would be the optimal one.

4.2 Software architecture

In this section it will be explained how the code is structured. The project can be broken down into 2 parts, like most projects: Frontend and Backend; in addition, there is a third folder that refers to the DB, but only initializes the MySQL DB.

First of all, there is the Backend, which would be the part that connects to the database and the server that uses the web page, in other words, it runs on the server side. To achieve this connection between the two parts, an API that appears in the "routes" folder is used, as can be seen in Figure 4.1.

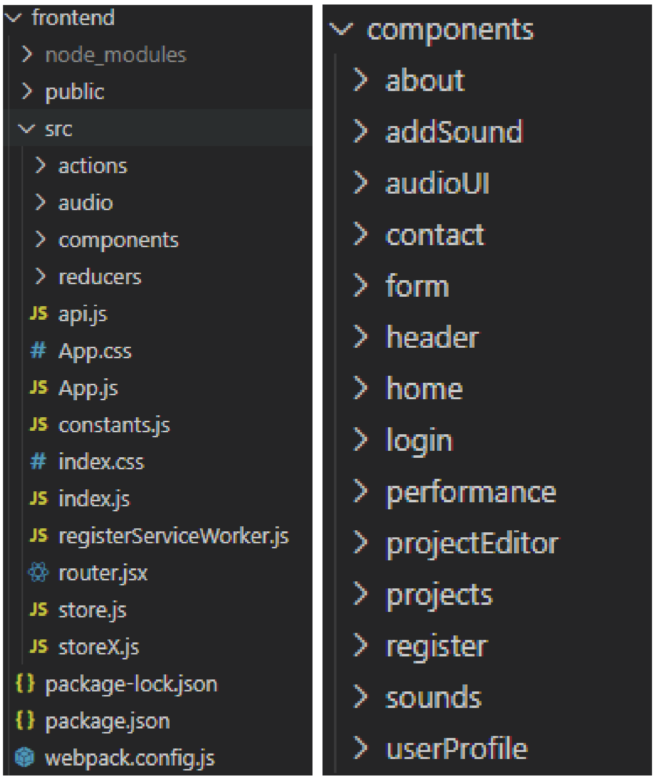


```
backend
  > assets
  > node_modules
  > public
  > routes
  > uploads\sounds
  .env
  db.js
  oscHelper.js
  package-lock.json
  package.json
  server.js
  soundcool.db
  utils.js
```

Figure 4.1: Backend folder

Second, Frontend refers to the part of the web page that interacts with users, meaning, on the client side. Since the tool to build is React, the project is based on components, placing those in the Frontend.

In Figure 4.2 could be seen all the folders, among which are the actions, the audio and the reducers; plus API paths and an index.js that integrates everything. Inside the "src" folder are placed components, being these, together with their states, the main elements of the project.



```
v frontend
  > node_modules
  > public
  v src
    > actions
    > audio
    > components
    > reducers
    JS api.js
    #  App.css
    JS App.js
    JS constants.js
    #  index.css
    JS index.js
    JS registerServiceWorker.js
       router.jsx
    JS store.js
    JS storeX.js
  {} package-lock.json
  {} package.json
     webpack.config.js
```

(a) Frontend folders

```
v components
  > about
  > addSound
  > audioUI
  > contact
  > form
  > header
  > home
  > login
  > performance
  > projectEditor
  > projects
  > register
  > sounds
  > userProfile
```

(b) Components folder

Figure 4.2: Broken down Frontend folder

Finally, there is going to be explained the data management of the project. In Figure 4.3 it can be seen the arrangement of the data within the MySQL Workbench environment, and it is as follows: the "users" element is the most global, since it is where the users are stored, in this element the configurations are stored , such as name, email and password (plus an ID that is generated automatically). During the session, each user generates a series of elements, such as their own project, a predefined project or storing a sound. Within the project, stored in "projects", appears the name, the date, the description, its content (modules + connections) and the option of your privacy; as well as information for internal management. For the predefined configurations, saved in the "presets" item, the configuration is simpler, and only the name, its location and the date are saved. Finally, the "sounds" element allows you to store music uploaded from a local file or from the Internet itself (using YouTube for example); likewise, it has another indexed element that is "audiosharing" to specify if the music can be shared with other users.

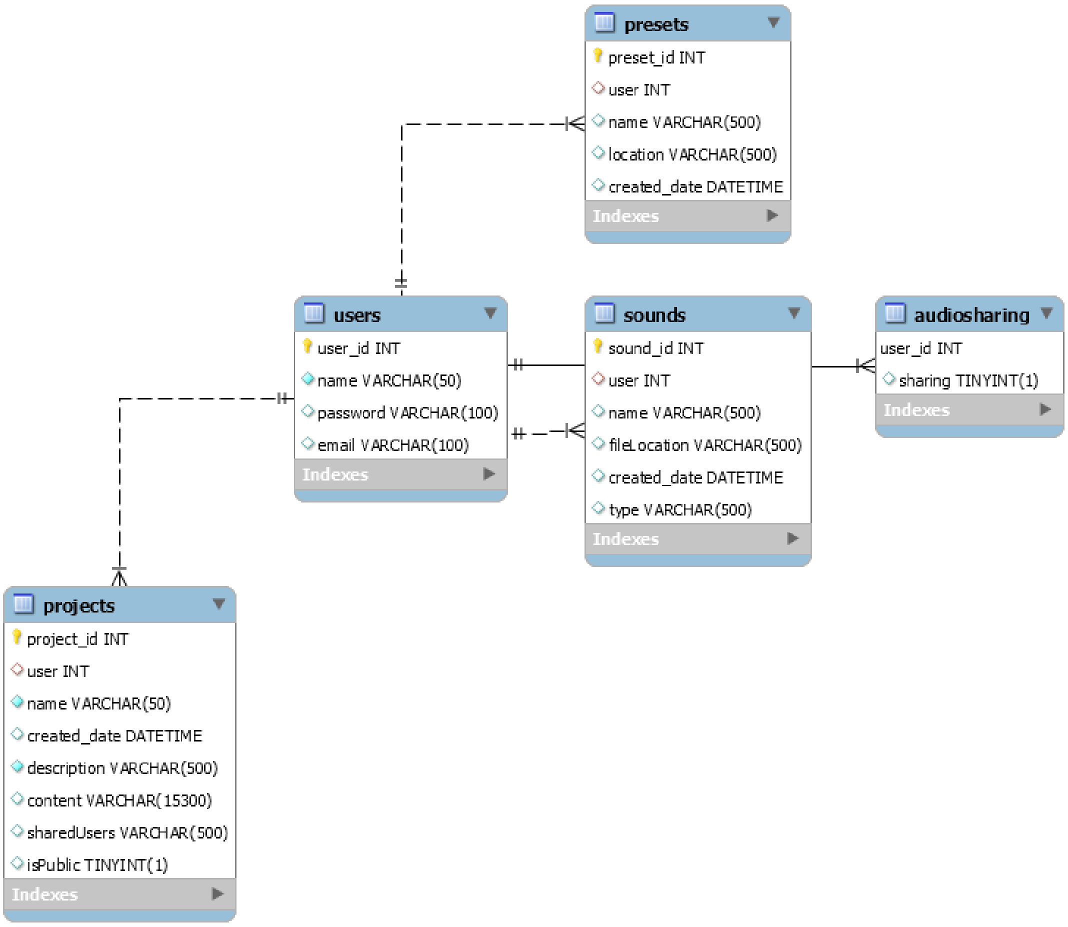


Figure 4.3: Database design in MySQL Workbench

Chapter 5

Implementation

This section presents the necessary implementation for the creation of the Minimum Viable Product. All the required modifications are listed below, and in each section, the existing problem is presented and subsequently its solution. Moreover, the list of errors, as it was presented in the Wiki, is listed in the Appendix.

5.0.1 Modification 1

Display the name of the current project, maybe to the left of the current user ID.

Solution

Implemented right in the tool panel of the project itself instead of in the header. The name of the project is taken from the localStorage and in the case of a new project, that is not already saved, no name is displayed.

5.0.2 Modification 2

Current user is a menu, but Edit Profile item is (always?) grayed out and inactive.

Solution

This is, by far, the most complex correction. First, the button was disabled, so it was needed to change its mode. Then, a page has been elaborated to edit the profile and a path has been created linked to the button. The new page shows in the screen the current user name, mail and password, and let the user change it.

It is necessary to connect the Frontend with the Backend, so an API an its route is needed. In the route is created an authentication token and after verifying it, the user configuration is updated with its respective hash. Highlight that all the process is based in the primary key: user_id, meaning that it is unique and no other user could have the same. Finally, after having changed the configuration, it was intended to see the new elements displayed in the screen.

Moreover, there was a synchronization problem between the localStorage, that manage the data, and the new updated information in the server. The most efficient solution lied on setting a timer that after 1,5 seconds, reloads the page.

5.0.3 Modification 3

If you move off a project page, it says ”WARNING: you are leaving the project editor. If you leave now, you can resume the current progress in Projects -¿ Resume Local Project, or you can stay on the current page and save.” But:

1. You get this message even if you save. Should only get warning if there are changes.

   Solution

   It has been implemented using React properties, that is, a state has been created for each project in which it is determined whether or not there have been changes. This state, depending on the modifications made, is variable and is the condition for the warning notice to appear. The state called: ”isProjectChanged” is initialized as false and if any change has taken place, it change to true.

2. ”Local Project” shows up a a recent project, but it is not listed on the complete list of projects. Let’s make this rule: the last project on the project editor is always saved as the Local Project, which you can get to by clicking on ”Resume Local Project.” It can be called Local Project.

   Solution

   The last unsaved project is called Local Project and can be accessed from the projects page. As soon as the project is saved with a name, it is no longer Local Project. What’s more, if Local Project is one of the last three projects, it also appears in the recent projects drop-down.

5.0.4 Modification 4

Floating view has only a tiny drag icon you have to use. (And the drag behavior is different in the two views: column view has the open hand icon where you can drag and anything that’s not a control serves as a point you can drag; floating view must drag by the little arrows in the corner, and when you hover there you see an index finger icon instead of a hand icon.).

Solution

The solution has been implemented with a modification of the CSS file. It was needed to find the elements in the floating view through the classes. Since most of the elements share classes and attributes, it was needed to use two classes, to be more specific. So, the classes: .boxContainer and .react-draggable were used, and then the cursor was changed to : ”grab”.

Moving away from the own resolution of the error, it is necessary to comment that the floating property was never included in the MVP. Now, since several errors have been corrected that improve the fluency, it could be included within the MVP. The reason for including it, as well as the reason why it is so problematic, is the freedom offered to the user, which I consider relevant.

5.0.5 Modification 5

Floating view editing region has strange behavior and a visible border. Maximum width should expand to width allowed by window. Region has big right margin (try dragging something to the far right.) Nested scroll bars are strange.

Solution

This was a problem for specific modules, because not all of them provoke this error. The solution is based on hiding the scroll bar through adding an attribute to the style, in particular, overflow: "hidden". This attribute eliminate the error of the white margin, whithout making interferences in the column mode.

5.0.6 Modification 6

The top of columns could be aligned with the top of the window and the top of the module palette – why the space at top?

Solution

In order to adapt to any screen, a constant called middlePoint was defined. This variable used the window.innerWidth method to calculate the midpoint of the screen. In this way, the elements were placed based on this midpoint. It is important to highlight that the position of each of the buttons was absolute, so it is not a responsive design.

To correct this alignment, it has been decided to remove the value that defined the midpoint, so that the buttons do not move even if you reduce the height of the screen.

Then, in order to make the buttons responsive, as the modules, they have been included in a div. The principal problem was related to the type of position: "absolute", because each absolute element is linked to its parent. The solution lied into changing the father´s position: "absolute" to "relative".

5.0.7 Modification 7

Repositioning: Create a speaker. Create a Player. Save. Exit and reopen project from Projects screen. Observed: Player is on top. Expected: Speaker is on top the way it was originally created and saved.

Solution

The implemented solution has been based on the analysis of a property called "typeId", as it has been done with the solution for the error number 15. With this property it can be saved the order.

5.0.8 Modification 8

It would be nice to have recent projects appear in the drop-down menu to allow a single selection to open a recently edited project.

Solution

A drop-down has been made that collects the latest localStorage projects and allows access to them. In the React environment, the if statement does not work, so it was used the function .map, in order to make a loop. Moreover, it would be advisable to introduce a functionality to verify if there are projects, because when there are less than three projects, the Local Project option appears.

5.0.9 Modification 9

Export patch: Instead of tabs, use 2 spaces in JSON.

Solution

There was a wrong declaration with the JSON.stringify method. There was a tabulation space declared with ”\t”, and the solution lied on the Unicode characters. The most common regular expressions, as ”\s” or $< br >$ did not work. The Unicode used was U+0020, but it is also correct to use the HTML code: ”&#32” or the CSS code:”\0020”.

5.0.10 Modification 10

Patches need to keep track of when changes happen so that you don’t get warnings about unsaved patches if you haven’t changed anything and therefore it’s already saved.

Solution

This problem has been solved by using the same fixing as the implementation number 3. It has been created a state, using React, that track changes and it determines the warnings in relation to the unsaved projects.

5.0.11 Modification 11

Project:New – does not work. Go to Projects page. Click Start New Project button. Create a Player. Select Project:New menu item. Expected: ”you are leaving this page” dialog, followed by blank screen. Observed: No dialog, Player is still in the edit window.

Solution

There is a structural problem with the new projects and the Local Project. In this case, since a new project has the path of ”... / new”, the system does not allow users to access again. The most efficient solution has been to compare routes, and if you were in ”... / new”, and you wanted to access again, delete the modules that have not been saved.

5.0.12 Modification 12

Project:New – if you connect SignalGen to Speaker, make a sound, and select Project:New, the sound does not stop.

Solution

The problem raised has not been simple because there is no button that allows you to pause all the modules at the same time. So, the speaker module, which can only be one in each project, has been analyzed and it has been found that it has a property called "suspended", that can be true or false. This property allows you to mute or activate the module. So, knowing this, a function has been created that allows to know if the user have exit from the project editor page or not, by studying the routes, and consequently activate or deactivate the sound.

5.0.13 Modification 13

Project:Save As – create a new project. Create a simple patch. Project:Save As. Dialog says "Create New Project" but we are "saving as" (should not change terminology given to user). Fill in name and description. Click "Create" (but this should be "Save"). Dialog pops up: "Stay on Page" or "Leave Page" (no dialog should be given here). Click Stay on Page and bad things happen.

Solution

Two different problems have been solved in this section. In the first place, the "Save" and "Save as" popups have been modified, in which, depending on the mode, user interaction is different. Also, as an additional contribution, the "Save as" option has been blocked when it is a new project.

Second, the popup comment that made the browser asking you if you want to stay or leave has been blocked. The solution has been to use a property, from the library that is already in use, that disable these comments from the browser: "disableNative".

5.0.14 Modification 14

Project:Save As: If you save as in floating mode, you return to the editor in column mode.

Solution

This section is not a real problem, but it is a question of design. On the Projects page, there is a button for the configuration which is the default view, with two unique modes: column and floating. This setting has more power than saving a project in one mode or another. Therefore, you have to consider whether you still want this default configuration or not. Because in the event that the mode is saved, this default setting would be useless.

5.0.15 Modification 15

Modules are numbered sequentially, so if you create S1, S2, S3, S4, delete S1 and create a new one, it gets numbered S5 instead of reusing S1. Each class of module should pick the first available integer for each new instance.

Solution

The implemented solution has been based on the analysis of a property called ”typeId”. So, it is checked if the length of the array, where all modules of this type are stored, is the maximum, then it is assumed that a new module is added. However, if the check does not verify the maximum extent, it looks for where the gap is in the sequence and fills it.

5.0.16 Modification 16

You cannot upload .aiff files.

Solution

All types of files can be uploaded with the uploadSound method, that is, from an audio file to a JSON. Thus, it would be advisable to make a filter. A similar possible filter is placed in the sharing toggle, where the project could be imported or exported, there is a filter that only accept .json.

It takes the audio file as payload and upload it to the server, using the API with the POST method. All types of files can be stored on the server, but not all can be played: the most popular audio formats are played, such as .mp3 and .wav, but others, such as .wma or .ds2, are not. However, it can be played .mp4 files, since the video functionality is not implemented, but it will be soon.

5.0.17 Modification 17

You can upload .aif files (same as .aiff, but different extension), but the sound will not play.

Solution

This error was not a code error, but a web browser limitation; in the linked page can be found the limitations between browsers and audio files[1].

Moreover, although it was a browser problem, this functionality has been corrected. The solution has been the following: if a file is .mp3, it is saved in the DB, but if not, using the ffmpeg library (which is obligatorily required in the installation of the project itself) it is transformed into .mp3 and saved in the DB. In the sounds page the original format appears so as not to confuse users, but in the database it is saved as .mp3.

[1] www.developer.mozilla.org/enUS/docs/Web/Media/Formats/Audio_codecs

5.0.18 Modification 18

Connect a player to a speaker. Hover over Speaker In button. Player dims to indicate connection. Click on Speaker In button. Connection is broken upon mouse up event. Move mouse away from Speaker In button. Player is still dim. Expected behavior: On mouse up event, connection is broken and the Player display is restored to normal (bright) look.

Solution

There were problems with going over the buttons. The solution has been to implement a condition in which every time a connection is canceled, the rest of the modules are taken and their opacity is reset to 1.

Chapter 6

Conclusion

To conclude the work, the conclusions obtained will be discussed. Moreover, these conclusions are related to the objectives proposed at the beginning of the thesis. The conclusions are presented below:

1. First of all, highlight that the MVP has been achieved. This has been possible by solving the critical problems presented by Soundcool web, among those that stand out, the use of all types of audio files and the possibility of the user to change their data. So, it could be positive to launch the product into the market to make a bigger testing, and find the bugs that only appear after having done a real use of the product.

2. Regarding to the Soundcool MVP itself, it is strongly believed by the author of the project, that almost all of the current properties and functionalities should be included in the MVP to give to the user an easy, but not simple, experience.

3. Regarding to Soundcool Web, during the realization of this project, new bugs have been discovered that, due to lack of time, or due to their relevance, have not been carried out. The list of them has been updated in the project's Github repository.

4. After numerous tests with the code, in order to understand the structure and operation, to make the necessary changes, it has been concluded that the code is a very well organized project. Although Soundcool web is the result of developing a large project with programmers with different styles and methodologies, the project has been done in a very professional way. Although the programmers are still students, the passion for good practices and excellence is denoted.

5. Git technology is very powerful, however without the right knowledge it can be disastrous. The code is open and when it comes to making updates there is only one development branch, making it perfect for developers who don't usually use Git. During the development of the project, the author of the work has had problems with the merging of local work and new updates.

6. A minimum of knowledge of Js, the basis of the project, has been essential for the project. In addition, it has been necessary to study and understand

the bases of React, including Redux. The logic applied by this framework can be very powerful when optimizing, but it is necessary to understand the components and their states to do so.

7. This job would have been nearly impossible without the development tools. GitHub, Visual Studio Code, Google Chrome Debugger, Redux DevTools and Insomnia have made possible the testing of the code and a comfortable integration in the project.

8. Finally, as regards the aim of the work: the correction of errors, due to the limited knowledge of the author, has been carried out mainly through trial and error. Lack of experience has been the cause of many errors (mainly syntax) that have caused an unnecessary prolongation of development in time. Despite this, the objectives have been met.

6.1 Work relationship with studies completed

When it comes to relating the work done with the studies completed, doubts arise about the correlation between them.

In the first place, the Telecommunications degree is not aimed at web programming - which is essential nowadays-. The basic knowledge of Java is explained in order to have a base of how a language is structured, but it does not go further. The used languages as Matlab or Verilog, for example, are aimed at more technical aspects.

The protocols and in general, the way in which the systems are connected, are the pillars of the degree. For this reason, although it was not the direct object of this work, the OSC protocol that is currently used has been studied in depth to a minimum.

There is no doubt that working methods have been internalized during the degree, different ways of writing have been learned and the student has been prepared to adapt and solve constant challenges, however, it could be said that the knowledge obtained has not been directly necessary in the degree.

6.2 Improvement proposals

In this section, some possible improvements for the project will be suggested. This will only be proposed, since the budget of the work and its limitations are known.

1. First, although the final product will be integrated into a website and does not require a download (one of the main reasons for integration), developers need to do a series of steps that do not always work well. Multiple languages, libraries, and programs need to be installed. The exact same steps have been performed on two different computers, with one computer running smoothly and the other not, despite numerous attempts. It is necessary to change the method.

2. The modules and their functions are by far the most powerful part of the system, however, it would be advisable to have a function that is capable of

activating all the modules, or even more practical, deactivating them all at the same time.

3. One element that does not contribute at all would be the Local Project. This element is used to refer to new projects that are not save already, but it is not necessary. Having Local Project in Soundcool means issues with the local storage and issues related to making new projects.

4. The use of cookies and localStorage data management could be a problem, not only synchronization. As a suggestion, cookies should be changed and eliminated. First, if the user eliminates the cookies, there would be no token, so the session would expire. Second, the token is generally made by the server, not by the code, meaning that there could be several security breaches.

   As long as the project director wants to continue using cookies, they should be signed on the server side with cryptographic schemes to guarantee the security of the platform.

5. Continuing with security, the authentication token is based on the user mail. Meaning that could be decoded and obtained. It is not a critical problem, but for sure could be avoided.

6. During the user data update process, it has been implemented a timer for reloading the page. A serious improvement would be to create a global state using Redux, that let making changes immediately.

7. Right now the server is a test server supplied by AWS. Since it is a university project, the server can be transferred to the university domain. As long as the platform will add new users, and some will pay, paying servers could be an option; otherwise, there will be a capacity problem.

8. It would be a plus to periodically update the problems and bugs that are found, if possible, describe them in an organized way in the Wiki of the GitHub page. Thus, the use of a tool that allows us to see what task each member of the team is doing would be recommended.

Finally, the author of this document would like to encourage the whole team to keep fixing the errors that had appeared during this time. I pledged to make a number of corrections, but more have come out in the process and much more would appear with a bigger testing.

Chapter 7

Appendix

- Alpha Release Task List

  1. Display the name of the current project, maybe to the left of the current user ID

  2. Current user is a menu, but Edit Profile item is (always?) grayed out and inactive.

  3. If you move off a project page, it says "WARNING: you are leaving the project editor. If you leave now, you can resume the current progress in Projects -¿ Resume Local Project, or you can stay on the current page and save." But:

     – You get this message even if you save. Should only get warning if there are changes.

     – "Local Project" shows up a a recent project, but it is not listed on the complete list of projects. Let's make this rule: the last project on the project editor is always saved as the Local Project, which you can get to by clicking on "Resume Local Project." It can be called Local Project internally, but aside from the "Resume Local Project" button, users should not see it anywhere.

  4. Floating view has only a tiny drag icon you have to use. (And the drag behavior is different in the two views: column view has the open hand icon where you can drag and anything that's not a control serves as a point you can drag; floating view must drag by the little arrows in the corner, and when you hover there you see an index finger icon instead of a hand icon.)

  5. Floating view editing region has strange behavior and a visible border. Maximum width should expand to width allowed by window. Region has big right margin (try dragging something to the far right.) Nested scroll bars are strange.

  6. The top of columns could be aligned with the top of the window and the top of the module palette – why the space at top?

7. Repositioning: Create a speaker. Create a Player. Save. Exit and reopen project from Projects screen. Observed: Player is on top. Expected: Speaker is on top the way it was originally created and saved.
8. It would be nice to have recent projects appear in the drop-down menu to allow a single selection to open a recently edited project.
9. Export patch: Instead of tabs, use 2 spaces in JSON
10. Patches need to keep track of when changes happen so that you don't get warnings about unsaved patches if you haven't changed anything and therefore it's already saved.
11. Project:New – does not work. Go to Projects page. Click Start New Project button. Create a Player. Select Project::New menu item. Expected: "you are leaving this page" dialog, followed by blank screen. Observed: No dialog, Player is still in the edit window.
12. Project:New – if you connect SignalGen to Speaker, make a sound, and select Project:New, the sound does not stop.
13. Project:Save As – create a new project. Create a simple patch. Project::Save As. Dialog says "Create New Project" but we are "saving as" (should not change terminology given to user). Fill in name and description. Click "Create" (but this should be "Save"). Dialog pops up: "Stay on Page" or "Leave Page" (no dialog should be given here). Click Stay on Page and bad things happen.
14. Project:Save As: If you save as in floating mode, you return to the editor in column mode.
15. Modules are numbered sequentially, so if you create S1, S2, S3, S4, delete S1 and create a new one, it gets numbered S5 instead of reusing S1. Each class of module should pick the first available integer for each new instance.
16. You cannot upload .aiff files
17. You can upload .aif files (same as .aiff, but different extension), but the sound will not play.
18. Connect a player to a speaker. Hover over Speaker In button. Player dims to indicate connection. Click on Speaker In button. Connection is broken upon mouse up event. Move mouse away from Speaker In button. Player is still dim. Expected behavior: On mouse up event, connection is broken and the Player display is restored to normal (bright) look.

Bibliography

[1] Magerko, B., Freeman, J., Mcklin, T., Reilly, M., Livingston, E., Mccoid, S. and Crews-Brown, A. *"EarSketch. ACM Transactions on Computing Education"*. In: vol.16(4), page 1–25. (2016). doi:10.1145/2886418

[2] Deitrick, E., Shapiro, R. B., Ahrens, M. P., Fiebrink, R., Lehrman, P. D. and Farooq, S. *"Using Distributed Cognition Theory to Analyze Collaborative Computer Science Learning"*. In: Eleventh Annual International Conference on International Computing Education Research - ICER '15. (2015). doi:10.1145/2787622.2787715

[3] Shapiro, R. B., Kelly, A., Ahrens, M. and Fiebrink, R. *"BlockyTalky: A Physical and Distributed Computer Music Toolkit for Kids"*. In: the 2016 conference on New Interfaces for Musical Expression. Brisbane, Australia. (2016)

[4] Kleimola, J., Larkin, 0. *"Web audio modules"*. In: Proc. 12th Sound and Music Computing Conference (SMC15). Maynooth, Ireland.

[5] Katsouros, V., Fotinea, E., Frans, R., Andreotti, E., Stergiopoulos, P., Chaniotakis and Liwicki, M. *"iMuSciCA: Interactive Music Science Collaborative Activities for STEAM Learning"*. In: Designing for the User Experience in Learning Systems, 123–154 (2018). doi:10.1007/978-3-319-94794-5_7

[6] Sastre Martínez, J., Murillo Ribes, A. and Lloret Romero, N. *"SOUNDCOOL: SMARTPHONES Y TABLETS PARA LA EDUCACIÓN MUSICAL, ÓPERA LA MARE DELS PEIXOS Y LA DANZA"*. In: IV Congreso Nacional y I Internacional de Investigación en Danza ,UPV (2016)

[7] García Talavera, L. *"SOUNDCOOL: SMARTPHONES, TABLETS Y OTRAS INTERFACES PARA LA CREACIÓN AUDIOVISUAL"*. In: Final Degree Project. Valencia: Universitat Politècnica de València. (2016)

[8] Manzo, V. J. *"Max/MSP/Jitter for Music: A Practical Guide to Developing Interactive Music Systems for Education and More"*. In: Oxford University Press. (2011) doi:10.1093/oso/9780199777679.001.0001

[9] Calabuig Benítez, L. *"Desarrollo de módulos para la plataforma Soundcool: conexión entre lenguajes visual y sonoro"*. In: Final Degree Project. Valencia: Universitat Politècnica de València.(2018)

[10] Moncho Moncho, S. *"SOUNDCOOL: OPEN SOUND CONTROL, SMARTPHONES, TABLETS Y OTROS INTERFACES PARA LA*

CREACIÓN MUSICAL Y VISUAL". In: Final Degree Project. Valencia: Universitat Politècnica de València.(2018)

[11] Garijo Contreras, P. *"REALIDAD AUMENTADA PARA LA CREACIÓN COLABORATIVA"*. In: Final Degree Project. Valencia: Universitat Politècnica de València.(2019)

[12] Fraietta, A. *"Open Sound Control: Constraints and Limitations"*. In: Proc. 8th Int. Conf. New Interfaces for Musical Expression (NIME 2008), Genova, Italy

[13] Scarani, S., Muñoz, A., Serquera, J,. Sastre, J,. Dannenberg, R.B. *"Software for Interactive and Collaborative Creation in the Classroom and Beyond: An Overview of the Soundcool Software"*. In: Computer Music Journal 2020; 43 (4): 12–24 (2020) doi:10.1162/comj_a_00534

[14] Guttman, E *"Autoconfiguration for IP Networking: Enabling Local Communication"*. In: IEEE Internet Computing 5(3):81–86. (2001)

[15] Saks, E *"JavaScript frameworks: Angular vs React vs Vue"*. In: Bachelor's Thesis: HAAGA-HELIA University of Applied Sciences (2019)

[16] Wei, S *"Practical Analysis of the Dynamic Characteristics of JavaScript"*. In: Bachelor's Thesis: Virginia Polytechnic Institute (2015)

[17] Freeman, A. *"Using a Redux Data Store"*. In: Pro React 16. Apress, Berkeley, CA. (2019). doi:10.1007/978-1-4842-4451-7_19

[18] Klimek, B,. Skublewska-Paszkowska, M *"Comparison of the performance of relational databases PostgreSQL and MySQL for desktop application"*. In: Journal of Computer Sciences Institute, 18, 61-66. (2021). doi:10.35784/jcsi.2314

[19] Lu, Q. *"Library Mobile Service Innovation Under Html5 Environment"*. In: Hu B. (eds) Human Centered Computing. HCC 2016. Lecture Notes in Computer Science, vol 9567 (2016). doi:10.1007/978-3-319-31854-7_95

[20] Wang, Y. *"Design on jQuery Evaluation System Framework of Movements Resource"*. In: Proceedings of the 2014 International Conference on Computer Science and Electronic Technology (2014). doi:10.2991/iccset-14.2015.44