2020

Final Project:
Audio Visualizer Report

DAVID CARDOSO, MARY SOUSA, SORANA I. MARIN

Table of Contents

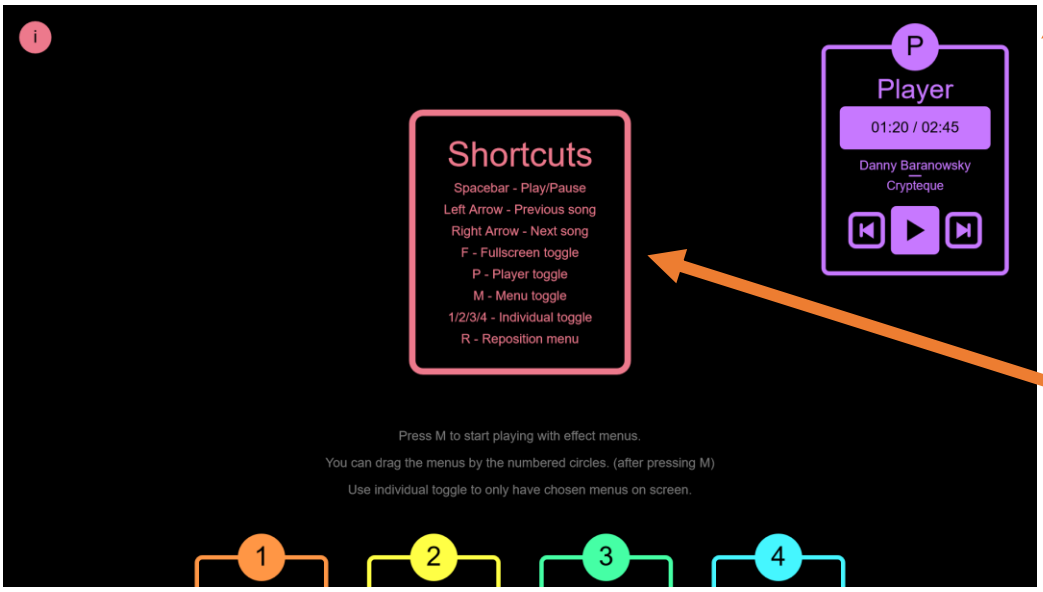
Documentation of Work Done and Implementation

Changes to Base Template	Why? (What have we taken into consideration?)
Fully Rewritten Base Code	Our team consists of 3 members and this project required us to produce at least 2-3 effects each therefore the base template was restrictive. If we attempted to work with the base template, we had to create a second column to display more effects which would have been a bad design choice in our perspective. Additionally, we wanted to challenge ourselves by fully rewriting the base code and practice creating interactive menus for future projects.
New Interactive Menus (Draggable/Clickable)	Due to us all being very determined to do our best in this project, we decided on creating separate menus for all our effects. The menus were created using buttons with both the buttons and the menus being created using classes and linked with our respective effects for ease of updating the code to include all our effects, while also being user friendly. The menus have had various bug fixes regarding making them more user intuitive, and similar but subtle features resembling a floating window manager. Such features include: ▪ the numbered menu being dragged by the user is the top rendered menu. ▪ not being able to activate 2 buttons at the same time by superposing 2 menus. ▪ when the user drags a menu, it should not deactivate the current effect being displayed. ▪ when the menus are moved to a lower position on the screen while on Fullscreen mode and then going back, the screen size difference will make the menus render outside of the canvas thus the Respawn function was implemented (activation letter R); ▪ each menu can be individually turned off/on so that the user has the option to free screen space or even fully deactivate all menus to fully see the effect being displayed.
Loading Songs	In the base template only one song was loading into the audio visualizer. Now we have 9 songs implemented. We first thought that

	songs from a rhythm game would produce nicer effects and it would be easier and smarter to work with. Most of the music in our Audio Visualizer was composed by Danny Baronowsky and is also the soundtrack of his game "Crypt of the Necrodancer", a rhythm game. Initially we have attempted loading the songs with the preload function, however we quickly learnt its limits. At only 3 songs it reached its capacity. To fix this issue was one of our biggest challenges and we managed it by making the Song class which loads songs in using "songList" and the URL of each song on the fly. The song class also keeps track of the song loading and it creates a new "p5.SoundFile" which loads the song. This change increased our song capacity greatly because it allowed on-the-fly loading and it currently allows to load 9 songs with no memory issues. (It works with over 9 songs too)
Player	The player was another challenge for us, since we already implemented the shortcuts necessary for a player such as Next, Previous, Pause/Play. We considered it important to also show these states to the user and give data about the song playing. Whilst we already had shortcuts, it was a challenge having two different ways of letting the user achieve the same output (by clicking the buttons in the player or using a keyboard). We have done this using the Player class which extends the Menu class; the player is a reliable source of information and accessibility to the user. While the keyboard shortcuts were implemented as a convenience to the user.
Effects	When doing the effects, we decided to each store them in our own menu, denoted with our initials (DC, MS, SIM) and Defaults. Each of us was the main coder for our own effects, however we did give feedback and help each other when needed. The extension was a team effect as we all worked on it, and we decided to put it in the Defaults menu and highlight it with a different tint of blue. In the Audio Visualizer each effect is linked to a button inside a menu. The effects themselves have mainly been structured into classes and each effect is its own file inside of our specific effects folder. This was done for ease of access.
File Loader	Due to this being a large project split into a big range of different files (over 24), we believed it would be more efficient if we had a script loading our files into index.html. Thus, we coded a python loader script (makeJs.py) to load all the scripts/classes from within the folders into index.html. This has saved us a lot of time and made managing our project easier.

Final project Screenshots:

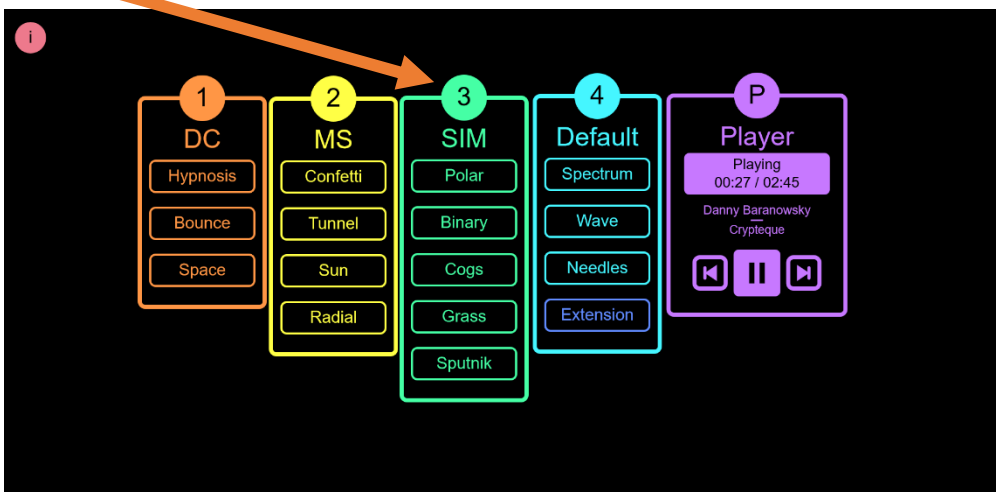
Player Menu, draggable if dragged within the circle at the top of the player.



The info class giving important information to the user, such as the shortcuts they can use. Activated if hovered over the "I".

1.0 Initial look at the Visualizer

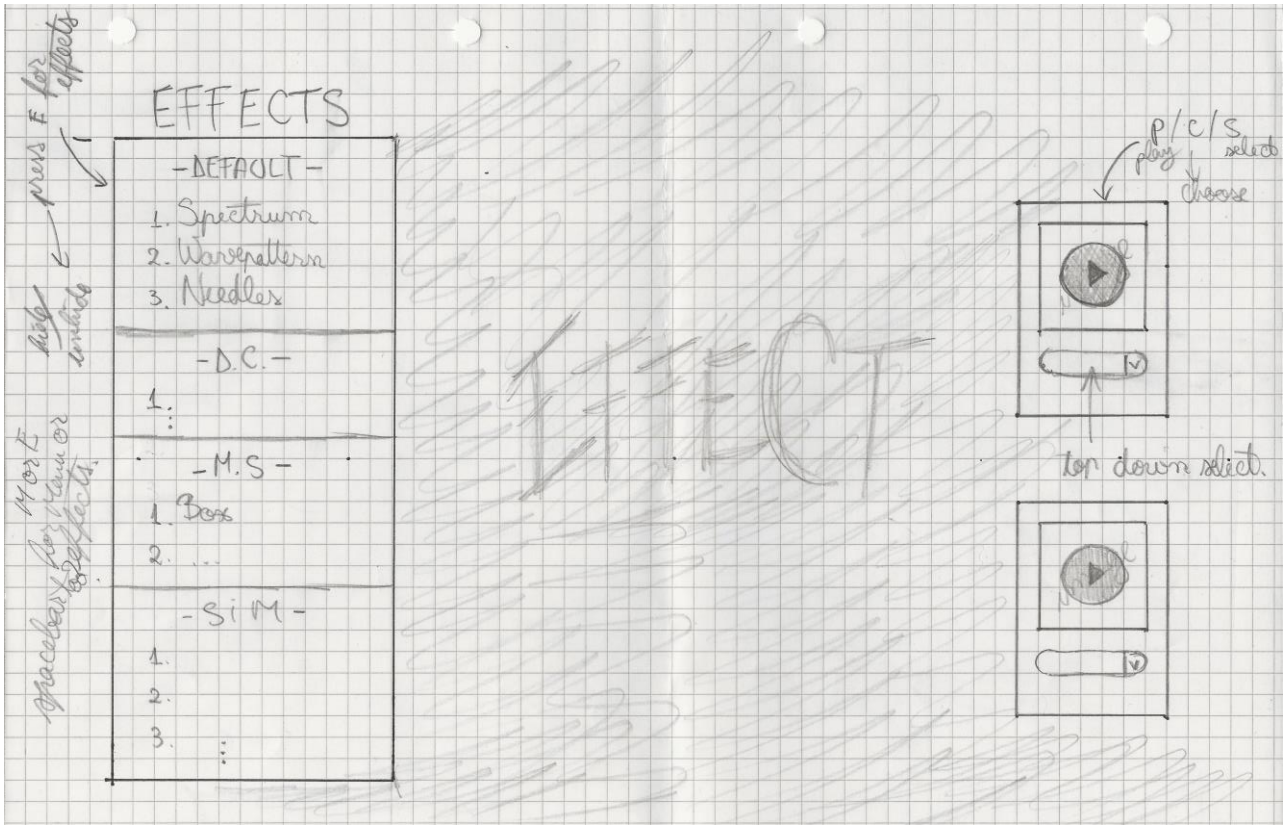
Effect menus also draggable by the circled numbers, these can also be repositioned with the shortcut "R".



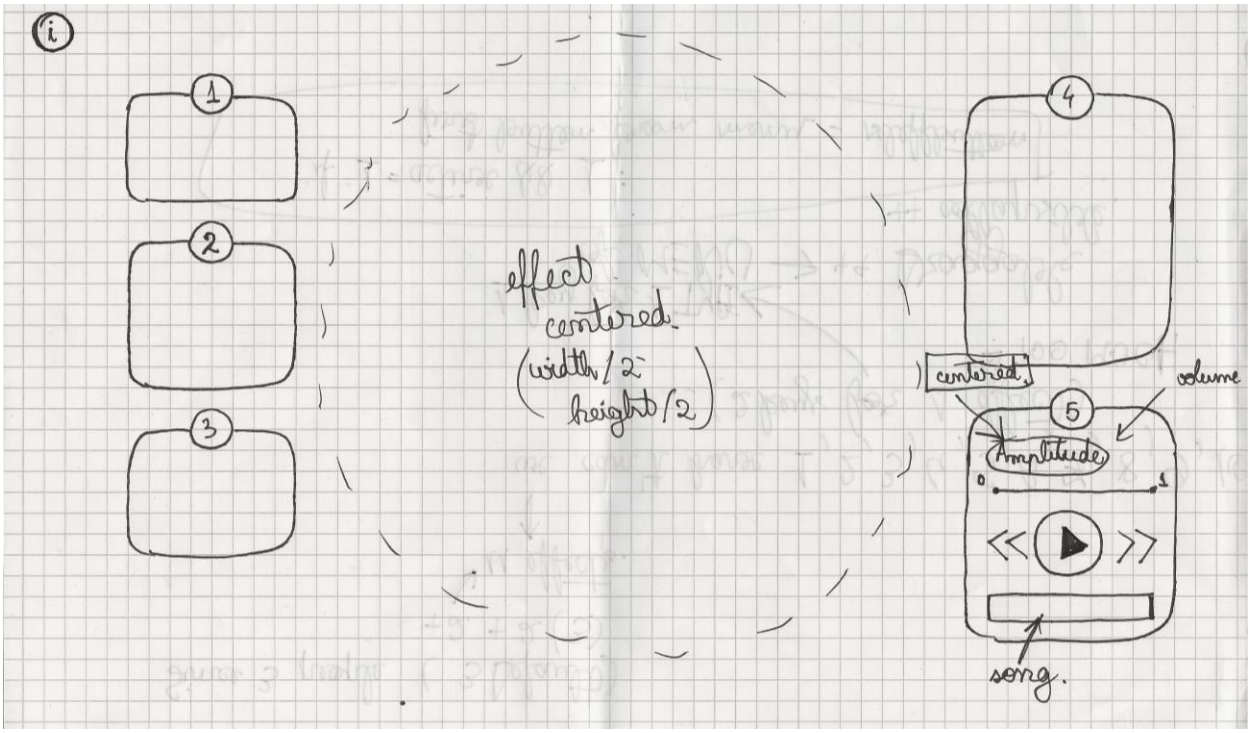
1.1 *Full Menus*

For the user understand the shortcuts and where they are, we chose to attract their attention towards the upper-left corner. We achieved that by initially only rendering the "*i*" circle on the canvas. We found this to be effective with new users, since they are more prone to hover over it.

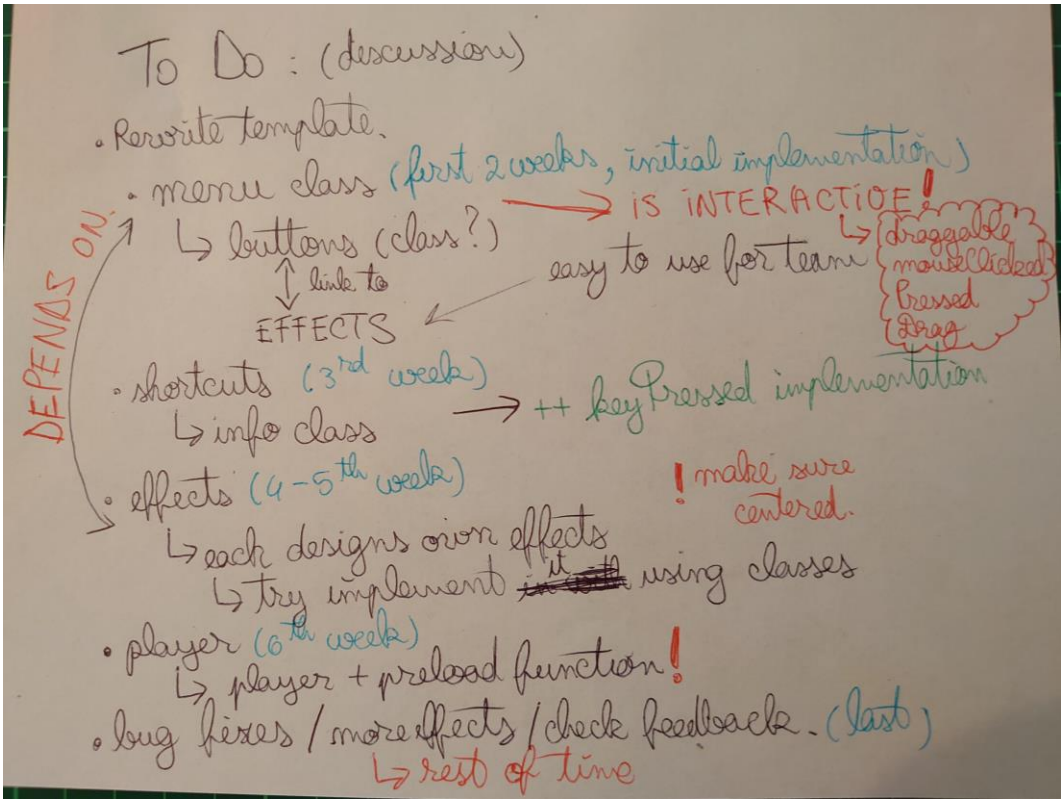
Diagrams/Wireframes/Discussions:



2.0 Initial Wireframe



2.1 Final Wireframe



2.2 Discussion about our project structure

SIM 06/03/2020

```
DONE:
-added the keyPressed function for most or all of the shortcuts
-1,2,3,4 -individual menu deactivation
-M - full menu activation/deactivation
-F - fullscreen
-R - Respawn menu to origin
-left/right arrow previous and next song
-spacebar pauses and resumes
-------------------------------------------------------------------------
-(playlist/ 8 songs added, not loaded)
-Tried adding a sign of whether the menu is on or off, however it looked bad/wrong so i
deleted it, it was just a small text on the upper left corner saying Menu in a white fill,
and grey if the menu is turned off, it didn't look very good and we already have r as a
respawn so yeah
==> Everything works fine

-Player - I don't have the menu for it yet it's just shortcuts in keypressed, maybe this
weekend we can talk about it?
```

2.3 Example of Update to Team through Discord

Evaluation

This project was a constant learning process, not only individually but collectively. We learned how to conjoin each individual agenda and personal limitations to enhance group work quality.
Working as a group requires ability to communicate efficiently, to express our view, listen to other views and find common ground so we can reach our goal. On that matter, we believe that we had a good work environment that, along with the common goal for high marks, enable us to work through our natural differences.
Similarly, we also learned the importance of establishing the program design from the beginning and improving on it, instead of redesigning it completely every now and then with new ideas.

User Review: *"This program is easy to use as there's a clear menu that offers instructions on how to use the program. Furthermore, it is very aesthetically pleasing due to the different colors used and the simplistic look. The animations used in the effects are quite fun and there is a variety of them. I would have liked to have a bit more screen space, so maybe reduce the size of the menus?"*

Regarding this review, we do agree that the menus take a big area of screen space. However, we would have to trade readability for more screen space. We have functions implemented to take care of the lowered screen space, such as the possibility of toggling on and off the menus.
The menus could be a bit smaller and we will consider this in the future updates.

Overall, we are very proud of the final program. It was an ambitious project since the beginning, and we believe that our program delivers the quality and complexity that we idealized.

To further improve our project in the future if given more time, we plan to:

1. Improve the player, adding more functionality for the user such as: giving them the option to add a song or delete it, but also be able to increase/decrease the volume.
2. Improve some effects by making them even more dynamic and more dependent on the music.
3. Potentially adding more effects or redesigning the menus so that they are smaller and neater or on sections depending on the type of effect that you are on. (example: Physics inspired effects in the Physics Sim. Menu).
4. Add a potential extra feature for users who like directly interacting with the effect, such as a Launchpad simulation so users can play with it while also listening to the music and enjoying cool effects.

Appendix (Progress log)

Start date of the project: 25/02/2020

Weeks 1 and 2 (25/02 - 08/03)

Tasks:

- define collective and individual tasks;
- check marking scheme;
- sketch of the main design of the project;
- each member of the group sketch at least 3 individual effects;
- start progress log.

Issues	Solution	Timeline
• Communication platform • Base template restrictions • Individual menu per member	• Discord with dedicated channels • Rewrite entirely the template • Code in classes and link to effects	• 26th of February • 8th of March • 8th of March

Week 3 (09/03 - 15/03)

Tasks:

- create shortcuts;
- create info class;
- update progress log.

Issues	Solution	Timeline
• how to display info in a user friendly way "hidden in plainsight"	• create a info point that will display when and only if hovered	• 15th of March

Weeks 4 and 5 (16/03 - 29/03)

Tasks:

- check marking scheme;
- create readme file;
- code and add extension effect to the program;
- debug program;
- peer review submission (deadline 20th of March);
- update progress log.

Issues	Solution	Timeline
• lagging effects due to too many Sines and Consines (low FPS)	• replace code to improve FPS	• 19th of March

Weeks 6 and 7 (30/03 - 12/04)

Tasks:

- peer review other projects (deadline 27th of March);
- create player;
- update progress log.

Issues	Solution	Timeline
• Link previous shortcuts to player	• Player class extending Menu class	• 12th of April

Weeks 8 to 12 (13/04 - 11/05)

Tasks:

- bugs fixing;
- average FPS per effect optimization;
- effects improvements;
- code cleansing;
- update and finish progress log;
- produce report.

Issues	Solution	Timeline
• Big range of files • Low FPS in some effects	• Code python script to load files into index.html • Reverse enginner the effects code to decrease the computacional power required	• 8th of May • 8th of May

End date of the project: 11/05/2020