ENGR3420 Analog and Digital Communications Project Report:

Signal Solver

Tolulope Oshinowo

Background:

As the final project for the course, the task of creating an educational and interactive experience was chosen as a means to showcase a deeper understanding of some of the topics covered during this semester of Introduction to Analog and Digital Communications. It was surmised that creating a game would be a great way of accomplishing this objective, and using elements from the first exam—and a bit of nostalgia—as inspiration, work towards developing a puzzle game from the ground up began. In the game the player would be tasked with taking a linear input and using some of the operations discussed throughout this course concerning amplitude modulation to create the desired output signal. Below in **Figure 1** is a screencap from *Factory Balls*, one of the games that helped serve as inspiration for this project[1].

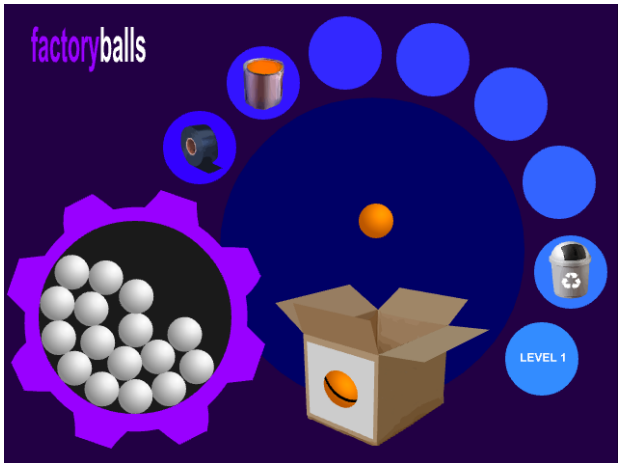


Figure 1: A gif of the game (Factory Balls) that served as inspiration.

[1] games, Coolmath. “Play Factory Balls - Copy Wacky Designs: Coolmath Games.” Play Factory Balls - Copy Wacky Designs | Coolmath Games, https://www.coolmathgames.com/0-factory-balls.

In the figure above, the goal of this game was to recreate the pattern of the ball depicted on the box using a blank sphere and the correct permutation of the available tools. The game that was set out to be created would adopt similar mechanics, but whereas the nostalgic game used balls and tools, this project would use waveforms and amplitude modulation operations.

Project Scope:

In terms of scoping out the project, the first step was to ensure that the core concepts and functionality were solidified. Below in **Figure 2** is a screencap of the final iteration of the game's user interface (UI).

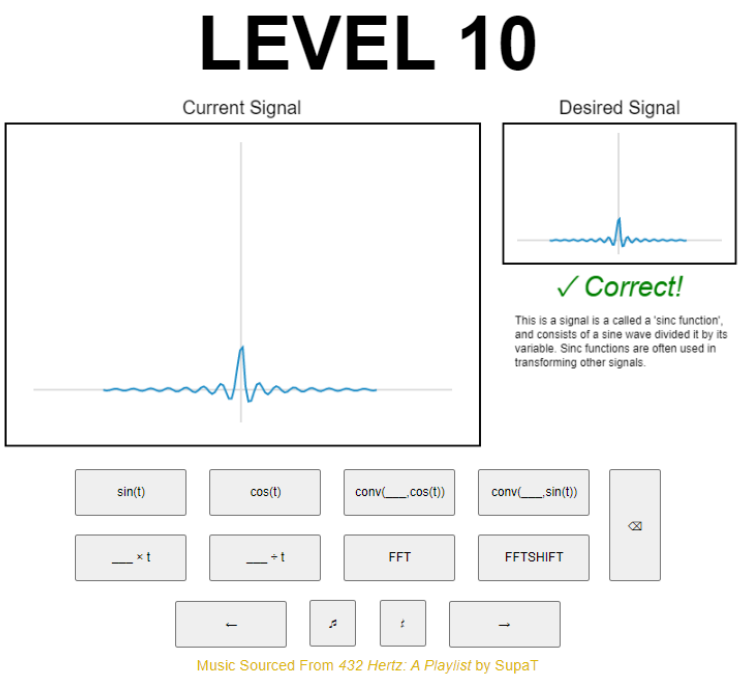


Figure 2: Screencap of the final game's UI.

From this image it can be seen that although the game drew inspiration from the aforementioned *Factory Balls* game, this game was conceived to work with multiple windows; one showing the current signal, and one showing the desired output signal. Additionally there were several buttons of which when clicked applied their labeled function to the current signal. Originally the scope for the game was for it to incorporate more sophisticated functions such as

filters which can be seen in one of the earlier iteration concept sketches shown later in this paper, but due to time constraints and bugs in development the low pass filter and high pass filter buttons were phased out and replaced with a multiplication by x and division by x functions instead. Overall the direction of the game's layout prioritized depth over breadth for the user experience. This meant that at face value the game would seem relatively simplistic, but as the user would interact with the game they with time would realize the versatility encoded within each button press' corresponding function.

Methodology:

To begin the development process of this game a few concept sketches were initially produced that included some of the features mentioned in the Project Scope section, and after a few rounds of trial and error the following layout depicted in Figure 3 was settled upon as the basis for the game's UI.

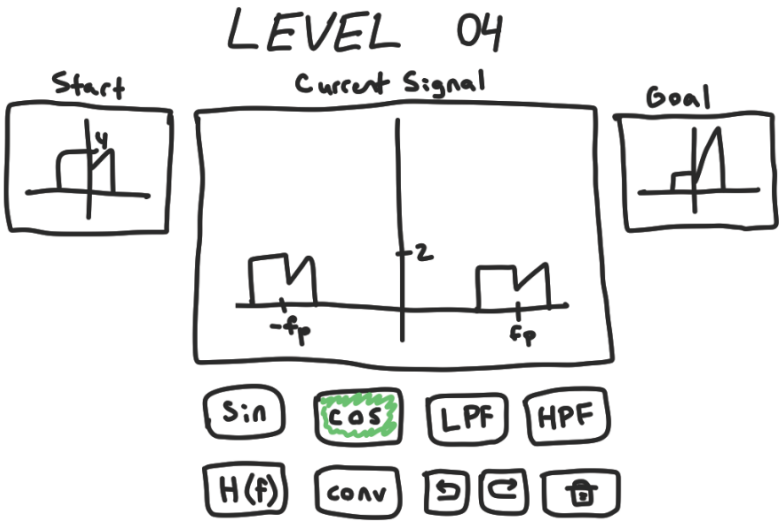


Figure 3: Original game mockup sketch.

Around this time work was also being done on completing the original one page project proposal assignment. As progress was made towards completing that write-up, it was determined to be fitting to explicitly list out some of the features that would be necessary to include within

the development of an educational interactive experience. There were two primary features that stuck out the most from this list: the first being a tutorial to aid new users in becoming familiar with some of the basics of signals and amplitude modulation, and the second being procedural generation so that visuals would be rendered and altered based on user inputs rather than there existing a need to manually create numerous graphics for each unique waveform resulting from all of the possible permutations of operations. Accounting for the former of the two major requirements was moderately straightforward, as essentially the first few chapters from the Analog and Digital Communications course textbook were interpolated into a page worth of fundamental information[2]. A screencap of this tutorial can be seen in **Figure 4**.

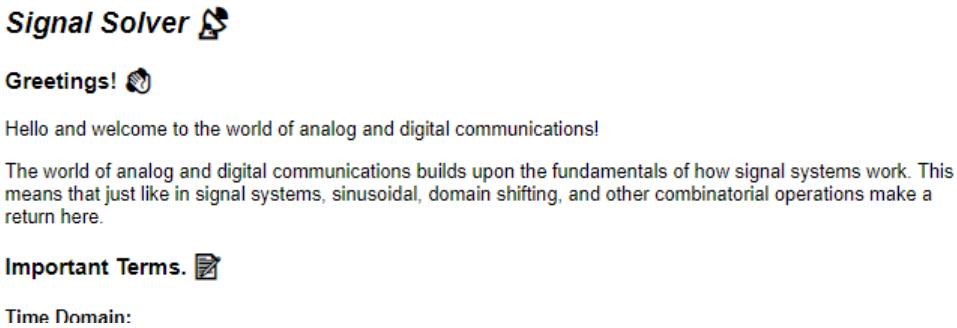


Figure 4: Screencap of the tutorial section of the game.

Through the use of interpolating the course textbook into a short tutorial, the learning curve for the game was made less steep as those who may have otherwise had no prior experience within this space would be able to have a clearer understanding of the terminology. As for accounting for the latter of the two major requirements, it was determined that reinventing the wheel would be counterintuitive and so the procedural generation of waveforms based on functional inputs was reverse-engineered from parts of MATLAB's Signal Processing Toolbox and Python's NumPy library[3][4]. An additional requirement for this project was to maximize

[2] Govindasamy, Siddhartan, et al. Analog and Digital Communications. 1st ed., Olin College of Engineering, 2021.
[3] MATLAB and Signal Processing Toolbox Release 2021a, The MathWorks, Inc., Natick, Massachusetts, United States.
[4] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. (Publisher link).

accessibility. In pursuit of this all the code written for this project was scripted from the ground up without the use of external libraries. Furthermore, vanilla javascript was used within basic HTML as the chosen language environment so that everything would work 'out of the box' and there would be little to no obligation on the end user to make sure that their respective system could run the game properly. In **Figure 5** is a snippet from one of the functions implemented in the code for the game.

```
const fft = (arr) => {
    var output = new Array();
    for (var k = 0; k < arr.length; k++) {
        var rcomp = 0;
        var icomp = 0;
        for (var n = 0; n < arr.length; n++) {
            rcomp += arr[n] * Math.cos(-2 * Math.PI * k * n / arr.length);
            icomp += arr[n] * Math.sin(-2 * Math.PI * k * n / arr.length);
        }
        output.push(Math.abs(rcomp))
    }
    return output;
}
```

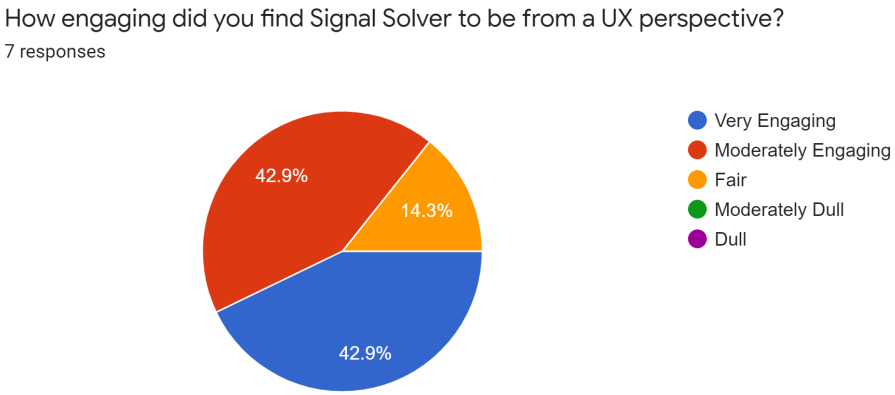
Figure 5: Code snippet of the fft function.

Referencing the code structure and architecture used in some of the relevant functions such as fft and fftshift from Python and MATLAB helped to streamline the development process, and made it so that more time could be allocated to the scripting of the features that allow for users to stack functions, along with the level creation algorithm. Once a completed game was created, the name *Signal Solver* was settled upon because of the game's puzzle-like nature, and beta testing began to collect information on the game and its reception from its target audience of fourteen to eighteen year olds.

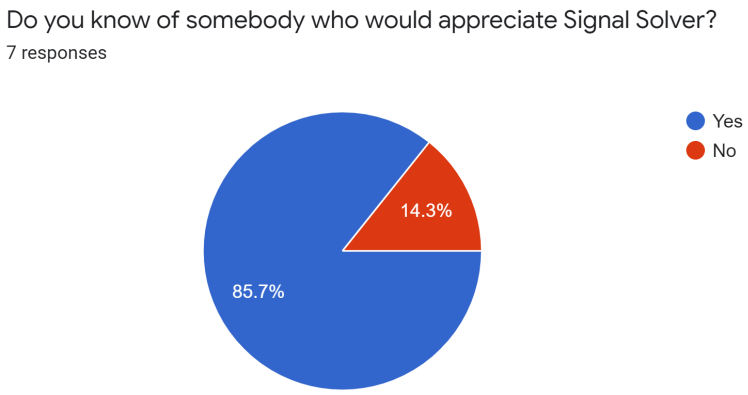
Results

Due to the unorthodox nature of this final project, collecting data on the quality of this project was done with a survey asking various people who played this game how they felt about it. Data was able to be collected conveying significant information from some of the beta testers

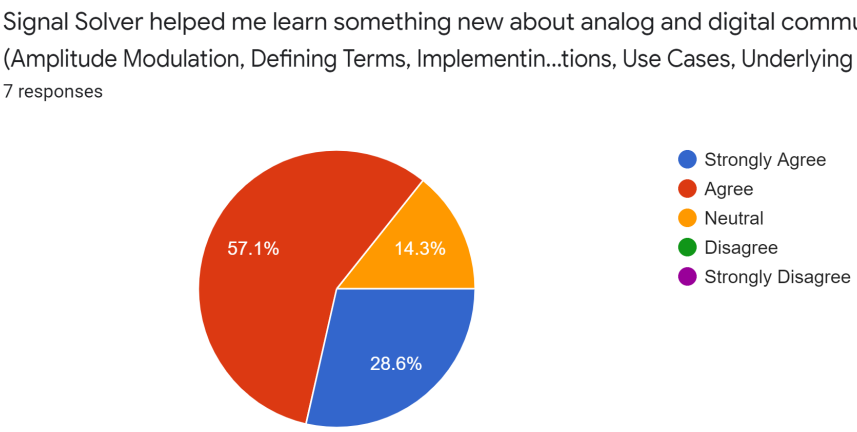
that tried out the game during the development process. Feedback was also gathered from a supplemental survey that was sent out to a group of high school-aged students. In the following **Figure 6** lies some of the relevant data received from a group of seven responders.



6a: User engagement poll results.



6b: User referral poll results.



6c: User learning poll results.

6d: Closing remarks.

Figure 6: Results from the survey.

As can be seen from the above data, there was an overwhelmingly positive review for this project. Quantitatively, a large majority consisting of 85.8% of the total respondents in each poll indicated that they had learned something new from *Signal Solver*, had found the user experience of *Signal Solver* to be engaging, and believed that their peers would enjoy using *Signal Solver*. Qualitatively the majority of the sample size also had optimistic sentiments to share, with multiple respondents using terms like ‘cool’ to describe the interactive experience.

Although there was an abundance of propitious replies there is still plenty of room for improvement. Firstly it is important to understand the reasons behind and what could be done to address the feeling of those who felt as though they did not receive as much from this game as others did. This could be attributed to the structuring of each level being based on self-direction and intuitiveness. In the game's most current iteration as of this paper, once users complete reading through the tutorial they are thrown into the game's first puzzle in level one with little context except for what signal they need to create. This could be a point of confusion, and although the game offers sequential information with each completed level, explaining tasks could be more straightforward. Moreover due to the bare bones nature of the game, it does not present the most mobile friendly user experience due to the lack of responsive layouts. This would result in a limiting of the accessibility that was originally an integral attribute for the game during the planning stage. Considering these points for potential advancements alone, it is clear that the job here is not yet finished, as is often the takeaway from the undertaking of interactive software projects concerning user experience.

Next Steps:

It is fitting that this is the final section of a paper such as this as it serves as a synthesis to everything mentioned before, but is not as close-ended as a conclusionary section would have been otherwise. Regarding the future plans for *Signal Solver*, some details were touched upon within the reflection section but will be elaborated on here. In the reflection section there were two major highlighted regions with room for improvement. The former being the need for additional scaffolding from the end user perspective, and the latter being more responsive programming to accommodate a more diverse range of screenviews.

Implementing supplementary information about what the user needs to do seems like a promising idea, along with further descriptiveness for the in-game axes such as indicators for when the user is working within the time or frequency domain. When it comes to making the game more responsive, a viable solution to patching that issue should not be too challenging either, as most likely objects would just need to be defined by percentages rather than pixels, coupled with the use the CSS Flexbox layout manager with different orientations based on screen views within different predefined ranges.

Aside from the survey response-based edits to be made, some other future plans for this project include hosting the *Signal Solver* on the Olin Satellite + Spectrum Technology & Policy Group (OSSTP) website so that it can be enjoyed by a larger audience; reintroducing functions that were unable to work properly within the allocated time such as the low pass and high pass filters, but then also possibly functions to simulate different types of noise; port Signal Solver to other computer languages as well such as Python, Java, or C# so that the game can receive even further exposure even when offline; expand the breadth of the user experience by making the game's UI more aesthetically pleasing through a more effective use of colors and shapes; and possibly create a greater in-game story to go along with the level progression as a means to give imaginative context as to why these puzzles are necessary. In general the aim of any continuation for this project is to continue in the contribution of encouraging today's youth to become more interested in Analog and Digital Communications.

Reflection:

When I first signed up for ENGR3420 I really did not know what to expect out of it, but in retrospect I can say that I enjoyed this class, and I am sure that amongst my fellow students I am not alone in this sentiment. From the active learning through whiteboard activities, to the

acclimatization of physical radio systems during Software Defined Radio (SDR) Analog Lab, there was never an uninteresting moment. Furthermore, the freedom in vision that was afforded to each student during the final project of this course was something that I thought made this class really manifest the "do-learn" mindset that is native to Olin.

Back when I set out on this project, I stated that beneath my primary objective of creating a working game lay my personal goals of being able to effectively explain some of the fundamentals of signal math and the logic behind signal modulations. I hypothesized that if I could touch upon the first few chapters of the Analog and Digital Communications textbook and incorporate them into an assortment of quirky puzzles I would be able to come away with a more holistic understanding of everything that we had learned during the first half of the semester. Now looking back on how this project turned out I can say with conviction that not only was my initial hypothesis correct, but I also came away with a greater appreciation and respect for high-level math involved in the usage and maintenance of every signal that occupies the air. I was even able to feature some of my music in this project[5]! Working on this project inspired me to get my Technician's class HAM radio license, and if I were to sum up how the work that I did in this project pertains to the real world, I would say that I embarked on a journey to create something that I hope will be able to evoke a similar sense of wonder in the minds and hearts of those who find and play it.

Signal Solver Standalone Link: https://sites.google.com/view/signalsolver/home

Signal Solver GitHub Repository: https://github.com/toluooshy/Signal-Solver

[5] 432 Hertz: A Playlist, SupaT Records, 30 May 2021, https://open.spotify.com/album/0EkxPQCZmN6Ujy9MVa8VhA?si=6Tf1WUM3RBCIZrAVIAtk9A. Accessed 7 Dec. 2021.

References:

- Govindasamy, Siddhartan, et al. Analog and Digital Communications. 1st ed., Olin College of Engineering, 2021.
- MATLAB and Signal Processing Toolbox Release 2021a, The MathWorks, Inc., Natick, Massachusetts, United States.
- Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. (Publisher link).
- "DSP-Kit." Home - Postman Documentation, https://oramics.github.io/dsp-kit/api/index.html.
- games, Coolmath. "Play Factory Balls - Copy Wacky Designs: Coolmath Games." Play Factory Balls - Copy Wacky Designs | Coolmath Games, https://www.coolmathgames.com/0-factory-balls.
- 432 Hertz: A Playlist, SupaT Records, 30 May 2021, https://open.spotify.com/album/0EkxPQCZmN6Ujy9MVa8VhA?si=6Tf1WUM3RBCIZrAVIAtk9A. Accessed 7 Dec. 2021.