ENGR3420 Analog and Digital Communications Project Presentation:

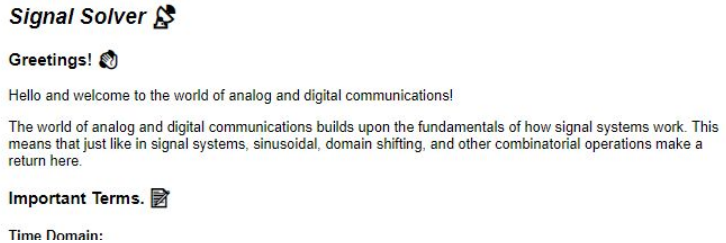
Signal Solver

Tolulope Oshinowo

Olin College
of Engineering

Background

- Wanted to create an educational and interactive experience to showcase my understanding.
- Believed that creating a game would be a great way of accomplishing this objective.
- Figured that designing a game for 14-18 year olds would be an interesting challenge.
- Looked to nostalgia for inspiration.

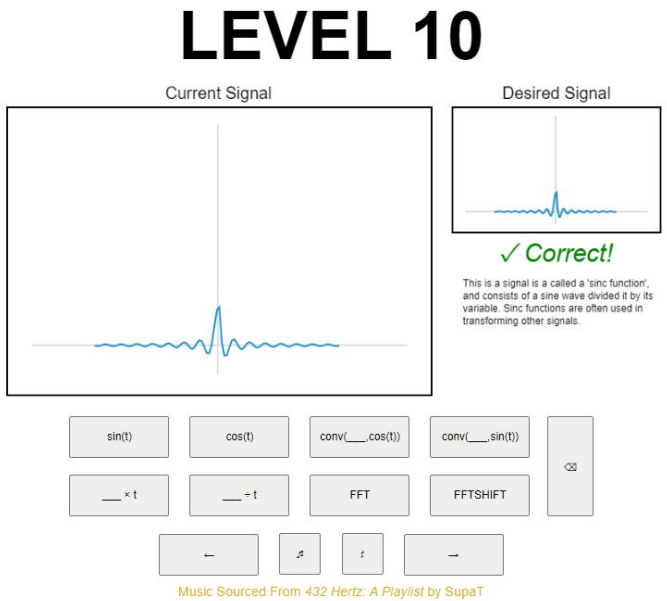


Screencap of the tutorial section of the game.

A gif of the game that served as inspiration.

Project Scope

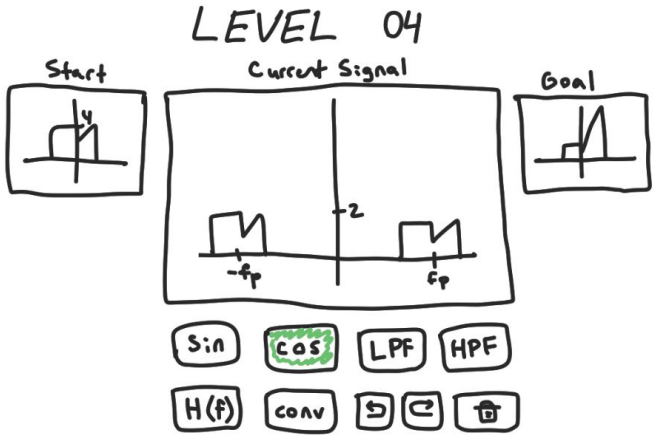
- Undertook an unorthodox approach by making a game.
- Opted to also include a short tutorial to help the user get up to speed.
- *The game's UI:*
  - Current signal, and desired output signal windows.
  - Multiple buttons to apply operations.
- Prioritized depth over breadth for UX.



Screencap of the final game's UI.

Methodology

- Sketched out possible layouts.
- Made a list of desired features for the game.
- Mapped out architecture for code.
- Decided to write in JS to maximize accessibility.
- Reverse-engineered parts of MATLAB's Signal Processing Toolbox and Python's NumPy.
- Utilized beta testing.



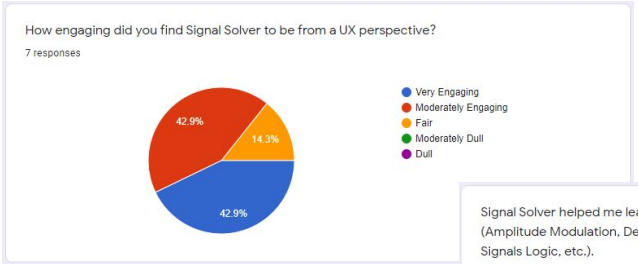
Original game mockup sketch.

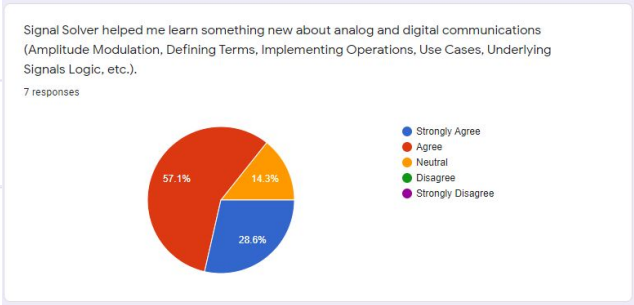
```
const fft = (arr) => {
    var output = new Array();
    for (var k = 0; k < arr.length; k++) {
        var rcomp = 0;
        var icomp = 0;
        for (var n = 0; n < arr.length; n++) {
            rcomp += arr[n] * Math.cos(-2 * Math.PI * k * n / arr.length);
            icomp += arr[n] * Math.sin(-2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

draw(new Array(101).fill(0));

const switchlevel = (cmd) => {
    if ((level == 0 && cmd > 0) || (level == 1 && cmd > 0) || (level > 1 && level
        level += cmd;
    }

    document.getElementById("levellabel").innerHTML = `LEVEL ${level}`;

    let min = -50,
        max = 50,
```

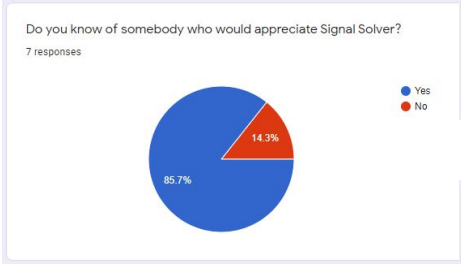
Code snippet of the fft function.

Results

- Collected info from beta testers.
- Gathered feedback from a sent out survey.
- Received generally positive reviews.
- Realized that there is still plenty of room for improvement.



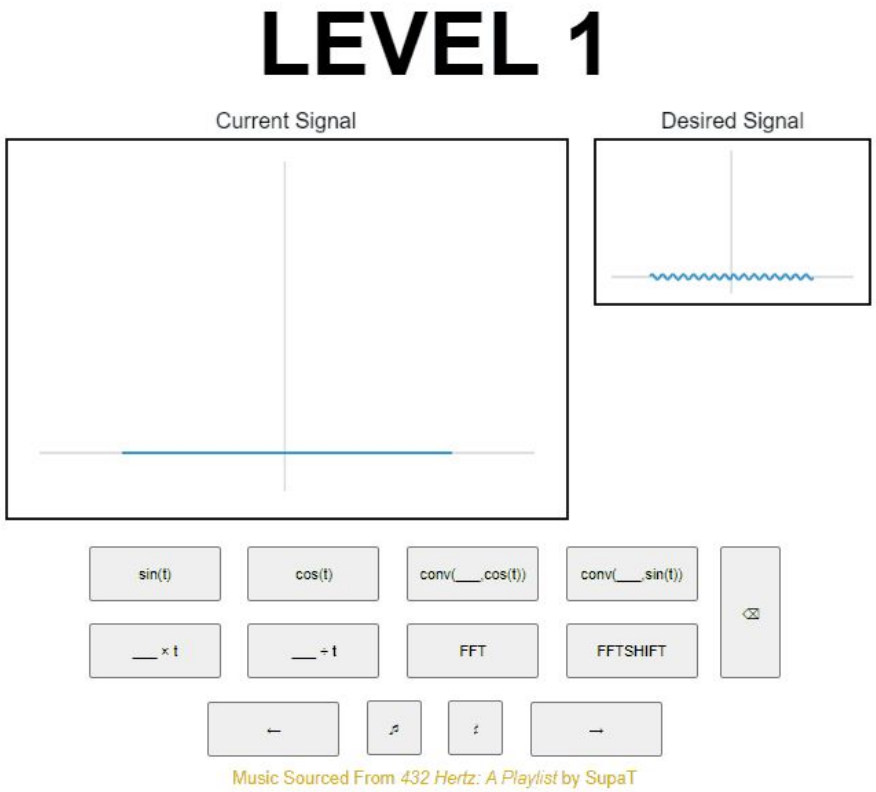




Results from the survey.

Future Work

- Host game on the OSSTP website.🎉
- Figure out how to make the game more ubiquitous.
- Bring back functions I couldn't get to work (LPF).
- Create a story to go along with the game.
- Port the game to other languages.
- Contribute to getting today's youth more interested in AD comms.



A gif of the game in action.

Questions?

- Wanna play?
  https://sites.google.com/view/signalsolver/home
- Repo?
  https://github.com/toluooshy/Signal-Solver
- Shameless plug:
  https://open.spotify.com/album/0EkxPQCZmN6Ujy9MVa8VhA?si=6Tf1WUM3RBCIZrAVIAtk9A

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=6Tf1WUM3RBClZrAVIAtk9A. Accessed 7 Dec. 2021. \