Thesis Project Portfolio

Voice-Controlled Pong

(Technical Report)

Danger of Generative AI

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Edward Oline

Spring, 2023

Department of Electrical and Computer Engineering

Table of Contents

Sociotechnical Synthesis

Introduction:

My technical capstone project and my STS research both revolved around speech technology, but were not directly related to one another. As a computer engineering and electrical engineering double major, I worked in a team for my technical work. We created an adaptation of the classic video game Pong, and changed the input so that the user moved their paddle up with a high-pitched noise and moved the paddle down with a low-pitched noise. My STS research drew from the dangers of other speech recognition technologies that record your voice and how with new generative AI, the right to privacy is more important than ever before.

Technical Summary:

Development of our capstone project took place throughout the entirety of the Fall 2022 semester. The project was a success, and we were able to apply concepts learned from our coursework along with additional research to create a well-functioning final product. The home-console type system was comprised of a Raspberry Pi as the main computer, a custom designed PCB, an LCD touchscreen display, and a machined aluminum housing. We used the PCB as the hardware for analog processing of the audio signals, and code libraries on the Raspberry Pi for digital signal processing. Each member of the team was able to focus on their strengths and it resulted in a product we were very proud of.

STS Summary:

Within the past year alone, the field of generative artificial intelligence has expanded exponentially, the most notable of which being the release of Chat GPT-3 during that fall semester. My research discovered how dangerous these technologies can be. The VALL-E AI from Microsoft can simulate a person's voice saying anything with only a three second audio clip as the input. While our project doesn't actually record the audio input from the player, this forced me to think about that possibility. The huge prevalence of smart home devices like Amazon Alexa, which do record full audio in some fashion, provide large sets of audio data. If a malicious actor gains access, these AIs could replicate voices to near perfection. The dangers are bigger than ever.

Conclusion:

I used to be indifferent towards the lack of privacy around my data. I did not care if my activity online was being tracked to target ads towards me. But with all of this new generative technology, my research opened my eyes to what people have been saying for years and how important data privacy really is. In relation to my capstone, that experience taught me a lot about how to work in a team and how see the design of a project through from start to finish. For future engineers in the electrical and computer engineering department, I recommend taking a page from our group and make your project something fun. It makes working on the project more enjoyable, and sharing the work you've done with others much more fun too.

Voice-Controlled Pong

Abstract

This project involved constructing a physical single player video game module with vocal controls. The module runs a version of the classic game Pong, but with a twist. The user moves their paddle up and down based on the pitch of their voice. Based on an initial calibration, the user sings a relatively high pitched note to move their paddle up, and a relatively low pitched note to move their paddle down. The opponent is a computer controlled paddle on the opposite side of the screen. The module consists of a small monitor in a display console with an external microphone for the user to vocalize their input into.

Background

The project was chosen to incorporate a wide variety of electrical engineering, computer engineering, and design skills, while remaining creative, engaging, and fun. As a group, implementing an interactive game that most users could understand and play without background knowledge was a more engaging task than a project in a non-game, or noninteractive format.

Similar projects in the past include a mobile wrist watch video game module [1]. Two iterations of this project featured two games: pong and asteroids. The game was not interactive, however. It displayed gameplay from these two games which corresponded to the current time, so that the device functioned as a wrist watch. It did not accept user input.

Another related past technology is the Atari, on which the video game pong originated [2]. The module utilized joystick input for player movement, and ran the game on a microprocessor in an apparatus which the user would attach to a monitor.

A third related technology is speech recognition. One project implemented voice recognition technology to control classic video games on a microcontroller [3]. The project distilled human speech into features, and used these features for processing and eventual game control.

Our project is separate from these projects primarily through user input. The first similar project, the wrist watch, accepted no user input. Our project will be controlled by a user through recognition of their voice's pitch. The second and third similar projects did accept input, but in a different form. The difference from a joystick input is clear, but the difference between the speech recognition technology arises in how the audio input is used. Our project will not be concerned with the words the user is speaking, but instead it will isolate the primary pitch of their voice. This pitch recognition and utilization as input differentiates our project from these previous works.

This design will require new technical capabilities while drawing on previous coursework. For processing the microphone's input, filtering and amplification will be required. These tasks will draw on filter and amplifier design skills learned in the electrical and computer engineering fundamentals courses: ECE 3750 and ECE 2660 (FUN II and III). Next, a printable circuit board (PCB) will be required to implement the filtering and amplification of the microphone input, drawing on circuit design and PCB layout skills from ECE 2630, 2660, and 3750. Since a microcontroller will be used as the primary processor for implementation of Pong, microcontroller experience and embedded system design skills learned in ECE 3430 will be used. Through programming tasks to develop the functional Pong game on a microcontroller, best programming practices and object oriented programming learned in CS 2110 and 2150 will be used. Finally, project troubleshooting and circuit debugging skills developed during ECE 2630, 2660, and 3750 will be used while testing and troubleshooting each component and the system as it is assembled.

Physical Constraints

Design Constraints

Our video game module will be relatively easy to manufacture on a large scale. The PCB is the only thing that needs to be fabricated. The Raspberry Pi, temporarily disregarding procurement concerns detailed in the next section, solely needs to be flashed with the OS and program images. This makes the microcontroller fast and cheap to manufacture, beyond the wholesale cost of the unit and constraints of the manufacturer supply chain. The other components are “off the shelf”, with again the only external constraint here is component availability. Given the limited number of connections in the design plan, this product could be reproduced at a relatively low cost using standard assembly and fabrication equipment.

The main materials needed are the electrical components, printed circuit board, cables and connectors, casing for the module and the external microphone. In the process of identifying necessary componentry, the state of the global supply chain was at the forefront of thought. The availability of our selected parts is dependent on the stock of electronics supplier DigiKey Electronics [7] - our primary supplier. We have identified several acceptable replacements for parts that are low in stock and potentially will be out of stock in the future. The primary headache revolves around sourcing our selected microcontroller, the Raspberry Pi, through their authorized dealers. This microcontroller is currently in high demand and has limited availability. Both the external microphone casing are available through multiple manufactures and have several acceptable substitutes.

The constraints imposed by the physical device include the processing speed and graphical processing capabilities of the microcontroller. In a single thread it proved impossible for us to create a working version that ran at 60 FPS so multithreading was implemented to account for this. The audio processing was run on one thread while the game logic and graphics were run on another. The microphone also proved to be a constraint, as well as the physical laws pertaining to audio sampling especially when it comes to frequency analysis. We spent significant time tuning the sampling frequency from the microphone, the sampling frequency of the pitch detection algorithm, and the method of calculating pitch from the raw input audio signal.

Cost Constraints

The primary economic constraints of this project are the $500 budget specified in the project description and current component pricings affected by recent supply chain limitations. The largest items on our budget include the Raspberry Pi microprocessor as well as the LCD screen, which totaled $167.94 and $75.89 respectively.. Since we only produced a single unit for our project, we did not have significant difficulty keeping within our budget. If this device were to go into large-scale production, economies of scale would decrease the individual unit price down well below the $500 maximum afforded in this project through bulk part pricing and consolidated labor. The device also requires external power through a Universal Serial Bus (USB) Type-C port, so long term use would incur energy costs on the part of the user.

Tools Employed

Raspberry Pi 4 (4 GB RAM): The Raspberry Pi 4 is a single board computer which functions as a microcontroller and the primary processing power for game functionality. It also served to interface with the audio input from our PCB, and produce an audio output. As a tool, it was used in testing software, and generating power from its available 3.3V and 5V pins. While Isaac took the lead role on the embedded development necessary, both John and Teddy had experience with microcontroller interfacing.

On the software front we used a custom build of Visual Studio Code [8] to write and test code natively on the Raspberry Pi. However before doing so we had to use BuildRoot [9], a tool used for baking custom Linux images, to create a custom operating system image for our Raspberry Pi. This image was then flashed to a micro SD card and loaded onto the Raspberry Pi. The native UART (universal asynchronous receiver-transmitter) port was useful during this stage to manually check data and ensure the installation process was successful.

Many useful software tools and techniques were used to create the finished product. First of which, the use of an FSM, was a technique learned in our embedded programming courses, but the specific application for our project (especially using Python) required some significant additional skill development. Additionally, the creation of the Linux image that we ran our game on required learning BuildRoot as well as what an RTOS (real time operating system) is and how to effectively use one to create a functional product. Several synchronization primitives including multithreading, locking, and data synchronization were used to improve performance and ensure no data loss between threads that were used to handle different tasks.

The circuit schematic and the final PCB layout of the components were designed using the KiCad software [18]. This was the team's first time using this software as previous projects in the electrical and computer engineering curriculum had used other applications for the PCB design process. In addition, the analytical design work for the PCB was done in part with assistance from the Analog Devices Analog Filter Wizard [19]. Once filter specifications were determined, the tool was used to acquire component values for the Sallen-Key [20] filter that was implemented on the PCB.

The case itself was bought from Hammond Manufacturing [17]. The aluminum case required several holes that were made with a milling machine. Before getting certified in the Mechanical Engineering Machine Shop, a CAD file was downloaded from Hammond Manufacturing and modified with the extrusions required. This modification of a CAD file required the use of Autodesk Inventor, a computer-aided design application for 3D mechanical design, simulation, visualization, and documentation. John had previous experience in CAD and was able to complete this task effectively. After the design was completed, a 3-axis vertical mill was employed to make the cuts. This was done under the supervision and guidance of the Lead

Engineering Technician responsible for running the Machine Shop. John had experience with rapid prototyping and fabrication. These skills were helpful in the novel domain of machining metal required by our project.

Societal Impact Constraints

Environmental Impact

The main environmental impacts this product will have are in relation to its manufacture and disposal. During manufacture, both our custom PCB connector board and the Raspberry Pi will produce the most waste of their lifetime from the subtractive manufacturing methods employed [10]. Recycling the device will be challenging since it will be composed of multiple parts that are not all recyclable. As long as the device is not physically destroyed, the Raspberry Pi is eligible for factory restoration and recycling [11]. Additionally, we have sourced electronics disposal companies that provide free disposal of LCD screens [12]. Otherwise, during its operational lifetime, the game can be replayed with minimal environmental impact during its useful lifetime.

Sustainability

While this video game console uses electricity, we do not believe this presents an issue from a sustainability perspective. Given the lack of mechanical componentry and battery this product will have limited deterioration and thus an extended lifespan. Raspberry Pi's and PCBs can last 10-30 years if adequately designed and enclosed [11]. The electrical parts, including the screen, microprocessor, PCB, and connectors, could be replaced if there is electrical or physical damage. If this our product was being produced at the volume where sustainability would be pertinent, the product would be manufactured such that skilled labor could effectively replace components. This requires design choices that would allow access to the components, such as using screws to seal the enclosure instead of permanent methods.

Health and Safety

In order to ensure consumer safety, materials and manufacturing techniques will be employed to protect the electronic devices from possible water contamination. Additionally, a strict following of standards outlined in the proceeding section will ensure the product is not harmful to consumers. Measures including adequate handling of ground faults, shorts, and rated equipment will protect the user in the event of an unexpected electrical malfunction[13].

Ethical, Social, and Economic Concerns

One concern about this device relates to the disposal of the LCD this module uses. LCD disposal is dangerous and recycling them is expensive and difficult [12]. This leads to a massive export of end of life LCDs to developing countries where environmental and workplace regulations are more relaxed. This is an unfortunate truth that our device could be subject to if not given to an ethical recycler.

This device will be focused on playing a video game with a human opponent versus a computer. This process will not require or benefit from the use of data collection or profiling, therefore, there should be no ethical issues regarding race, gender, ideology and so forth. The only issue may regard the premium nature of the specialized video game console which would put a barrier between the product and poorer individuals who would struggle to justify the cost.

External Considerations

External Standards

With the use of electrical equipment and a PCB designed for handheld operation, one of the standards will be to have an enclosure following the NEMA/IEC type 1 standard. This protects the user from an electrical shock and ensures the electrical equipment is free from physical debris, including dust [14].

Both the OS modifications and the video game program will be written in C. Therefore, the Barr C coding standards will be utilized to help minimize bugs and allow for formal standards to facilitate ease of coding and communications [15].

Another standard will be required because of the power supply utilized by the circuit. We will be designing a Class 1 circuit which is defined as having an output power that does not exceed 30V or 1000 volt-amps, and can use a 120V power source from an outlet as preferred by OSHA standards [16].

Intellectual Property Issues

Patent No. 2611782: PONG [21]

This patent encompasses the usage of the word “pong.” If we were to bring this product to market our current project title “Voice Controlled Pong" would be an infringement of this copyright. However, as game mechanics are unable to be patented, the gameplay experience could remain untouched in a commercial version.

Patent No. D926749: Raspberry Pi [22]

The only patent owned by Raspberry Pi Ltd. is regarding a computer case. We will have no issues with the usage of a Raspberry Pi in our commercial product. However, we cannot use branded trademarks and logos of Raspberry Pi such as the name “Raspberry Pi” and their logo in certain manners. Their website [27] has a detailed description of the allowed usage of their trademark.

1.5

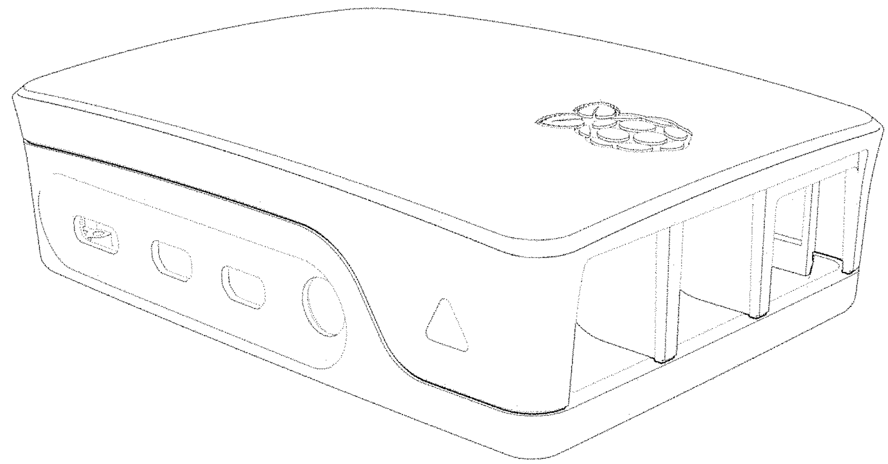

Figure 1. Raspberry Pi Ltd. Computer Case Patent

Patent No. US 2004/0138888A1 : Method and Apparatus for Speech Recognition [23]
This intellectual property in this patent discusses cepstral tuning techniques, and leveraging the cepstrum for speech recognition and conversion. The cepstral technique they use is similar to that used within our project, but the object of intellectual property is their method for using the produced cepstral coefficients to recognize speech, which does not pose a problem for the intellectual property of our project.

In summary, we are limited by the copyright rules and claims to the Pong and Raspberry Pi names. This project is likely difficult to patent, as it does not use any of its components in a completely novel way. The Raspberry Pi, monitor, and ICs utilized are fulfilling their intended manufacturing purpose, and thus this integration of the various components may not be considered nonobvious for the purposes of patenting.

Detailed Technical Description of Project

PCB Schematic Design

Summary of Function

The total function of our PCB was to accept the microphone signal provided by our implemented lapel microphone, and convert it to a decipherable signal for the raspberry pi to use in processing. For this to be done, the function was split into three stages: amplification, filtering, and analog to digital conversion. The small signal from the microphone needed to be amplified, then any frequencies outside of the range of the human voice were to be filtered out, then the signal must be converted from the analog to the digital domain, as the raspberry pi could accept only a digital signal. Finally, the PCB also housed the connection for the pushbutton implemented on the project. The circuitry for each of these functions are detailed in the sections to follow.

Amplifier

Amplifier selection was somewhat inhibited by available information for the selected microphone. A lavalier (lapel) microphone was selected for user friendly implementation and flexibility in movement, but a well documented lavalier microphone was not available. However, the lavalier microphone chosen is an electret microphone, thus output impedance was approximated to be 2.2 kOhms, with output voltage in the 20 mV range, as is typical for electret microphones used in this context. An amplifier was then selected on the basis of automatic gain control (AGC) inclusion, input impedance, gain available, and price. The MAX 9814 [24] was then selected, as it includes AGC, has an appropriate input impedance of 100 kOhms, offers three gain selections of 40, 50, and 60 dB, and boasts a price of under $2.20. Standard audio amplification from stipulates a minimum amplifier input impedance of 5-10 times output impedance [28], thus the 45 times ratio the MAX provides should be more than adequate.

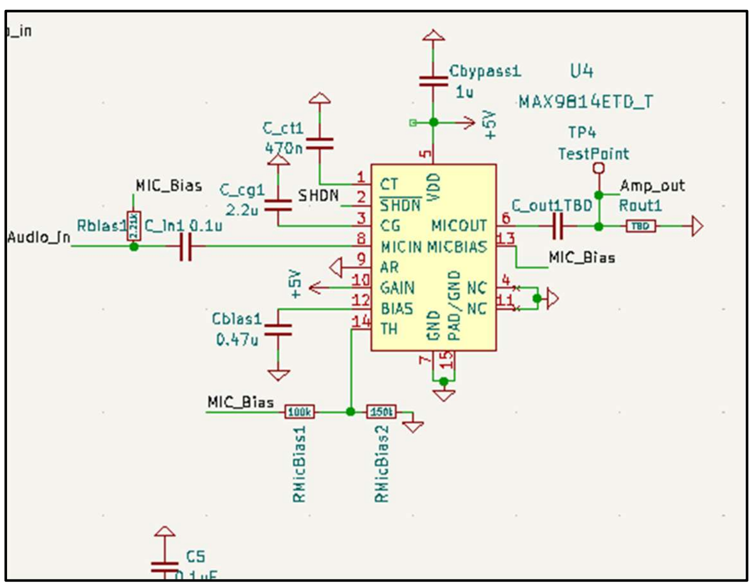

Figure 2. Amplifier Circuitry

Sallen-Key Filter

Before filter design could begin, the performance required of the hardware filter needed to be defined. For audio, industry standard sampling rate is 44.1 kHz [29], so this rate was chosen for our implementation. With this sampling rate in mind, Nyquist theory dictates that any frequency above 22.05 kHz in our sampled signal would contribute to aliasing. Thus, the primary function of this filter would be to reduce the effect of aliasing. To reduce aliasing, frequencies above 22.05 kHz would need to be attenuated by at least 30 dB. In addition, it was determined via spectral analysis of a vocal sample that the majority of signal power in a human voice will be between 20 Hz and 3.4 kHz, thus 3.4 kHz must be contained within the filter passband.

Filter design utilized the Analog Filter Wizard from Analog Devices [19]. Upon initial analysis, our specifications determined above resulted in a fifth order filter. However, given that the Sallen-Key architecture can produce a second order filter from a single Op-Amp, and that dual Op-Amp chips would be utilized, it was determined that the entirety of the Op-Amp capability of our two dual Op-Amp chips would be used through the construction of an eighth order filter. This eighth order filter provided us additional attenuation of frequencies that would not contribute to aliasing, but would contribute to the noise of the signal (beyond the frequency range of human speech). The filter need not fully attenuate all noise, as additional filtering can be included via a software implemented finite impulse response (FIR) filter. The resulting frequency response and component values of the filter as given by the Analog Devices tool are displayed below.

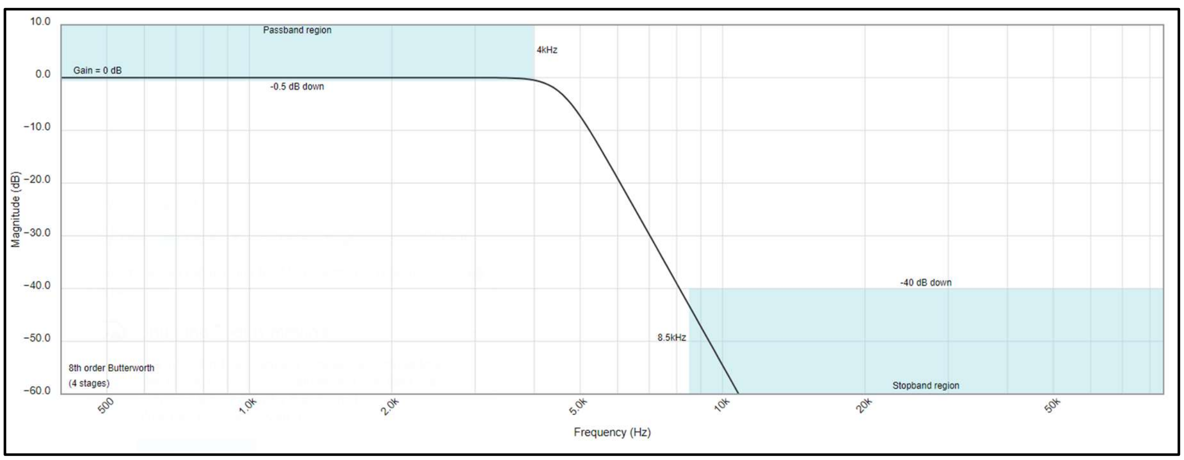

Figure 3. Sallen Key Filter Frequency Response

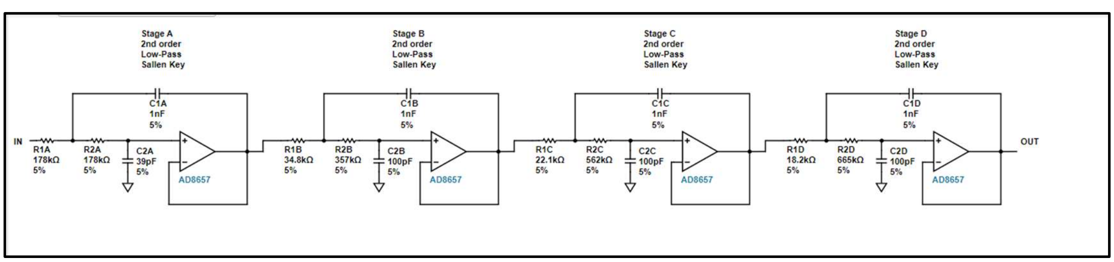

Figure 4. 8th Order Sallen Key Filter Design Wizard

The filter was implemented using 2 MCP 6022 dual Operational Amplifiers [26], chosen for their wide bandwidth. The resulting schematic, using values specified by the filter design wizard are shown below.

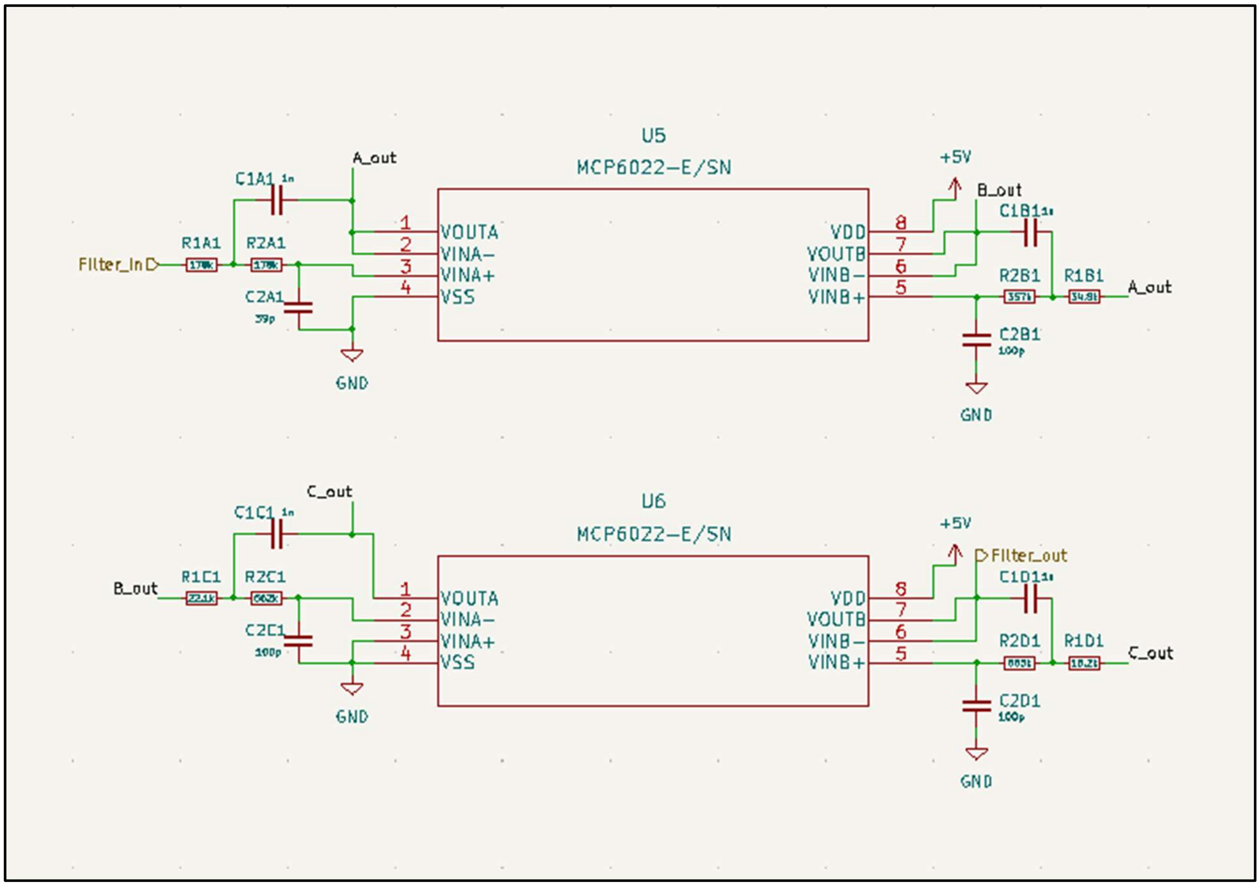

Figure 5. 8th Order Sallen Key Filter Schematic

Analog to Digital Converter

As the Raspberry Pi does not feature any analog input pins, analog to digital conversion must be done before the signal is passed to the Pi. At first, we thought we found an easy solution by potentially utilizing the built in ADC on a readily available and cheap Raspberry Pico. It was able to sample 12 bits at a blazing fast 500 kilosamples/second, but after investigating determined the output was subject to extreme levels of noise. Taking into consideration possible large variances in pitch and volume from users, we desired a chip that was more forgiving. Research led to the MAX11163 ADC [25], which is 16 bit instead of 12 for greater signal resolution, but still sampled at 250 kilosamples/second operation. That's more than enough considering the common industry standard for audio sampling is 44.1 kHz [29]. Additionally, this chip was chosen for its SPI compatibility which will help greatly in interfacing with the Pi.

The ADC was implemented using the typical application circuit specified in its datasheet, with a unity gain buffer implemented using an MCP6022 Operational Amplifier [26]. The resulting schematic is shown below.

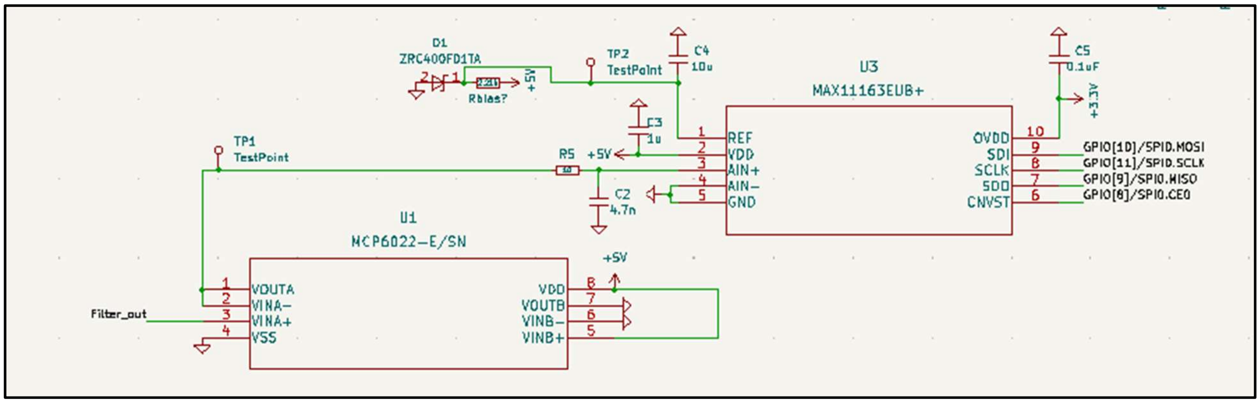

Figure 6. ADC and Unity Gain Buffer Circuitry

Button and Pi Hat Circuitry

Finally, circuitry was required to integrate a button and interface with the raspberry pi. Because the pi featured both pull up and pull down resistors, minimal circuitry was required for the button to be integrated. Therefore, a simple 1000 kOhm resistor was implemented between the GPIO pin on the Pi and ground. The circuitry surrounding the pi hat and button is shown below.

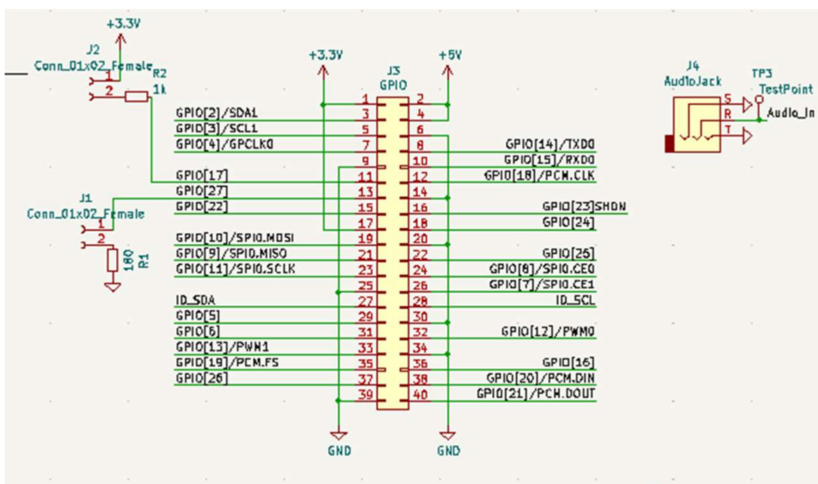

Figure 7. Pi Hat and Button Circuitry

PCB Layout and Assembly

For PCB layout, a Pi Hat shape was chosen to interface easily with the raspberry pi on which the game is run. With that shape constraint in mind, traces were drawn the the convention of copper top traces running vertically, and copper bottom traces running horizontally to minimize unnecessary crossover. One challenge encountered was the small trace size required for many of the integrated circuit chips used. Power traces typically require a higher thickness, but thickness was limited when drawing traces to the integrated circuits. This was addressed by thickening the power traces, but simply drawing thinner traces at the interface with the board. The resulting PCB layout in KiCad is shown below.

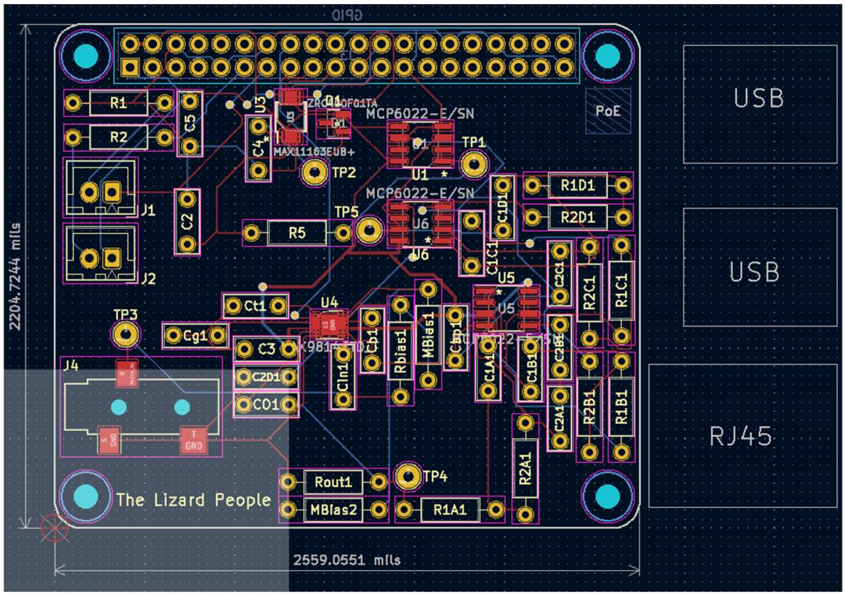

Figure 8. PCB layout

Project Timeline

The plan for our team this semester to complete the project was to split up into a hardware team and a software team. The latter of which was only Isaac who wrote the vast majority of the code for the project with some help from Teddy. Then, the hardware team was made up of Charlie, John, and Teddy and was led by Charlie. Additionally, John was responsible for the housing of our project. These three key components could then be worked on simultaneously as the weeks went on. Figure X below shows the Gantt chart for our projected timeline at the very beginning of this process, which is clear was a very rough outline. The next figure shows the much more detailed and updated timeline which was created after we had been working for a considerable amount of time. The different colors group together the main parts of the project which makes it easier to follow.

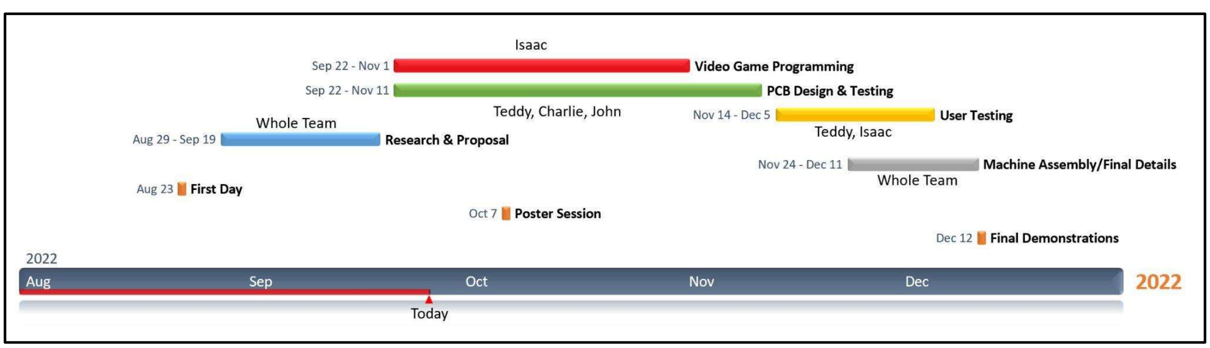

Figure 9. Proposal Timeline

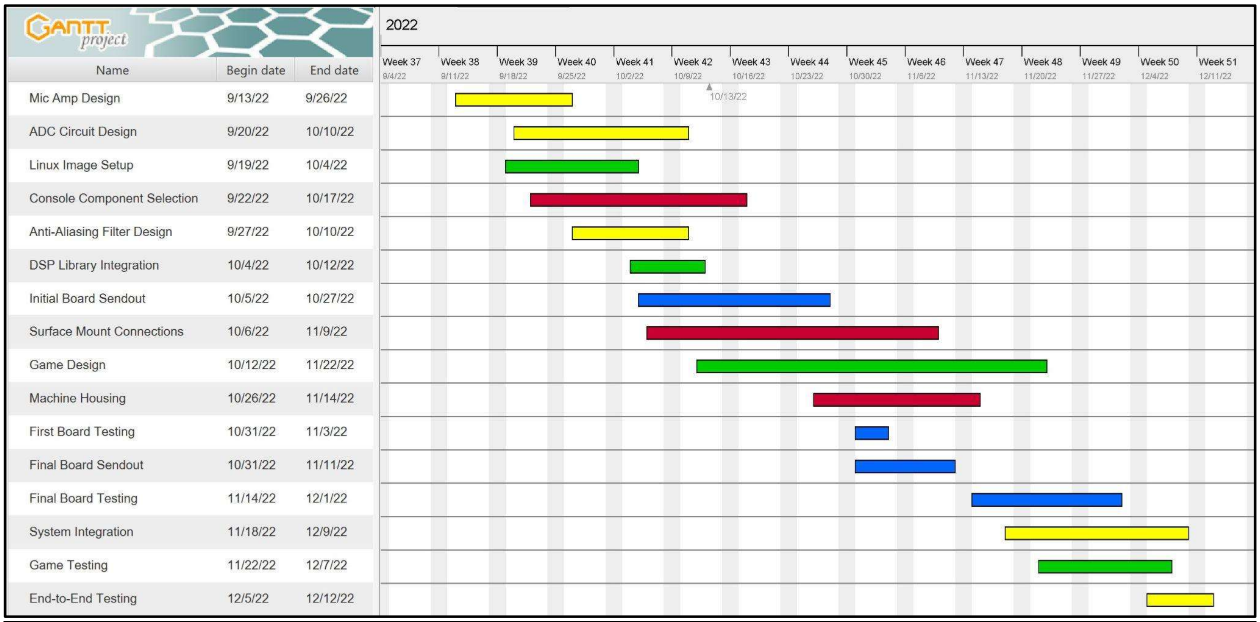

Figure 10. Midterm Timeline

By the end of the semester, our workflow followed pretty closely along with the above chart. The only major things that changed were the construction of the final board and the digital signal processing. The board was only finalized in the last week before the capstone fair. Furthermore, the software that handles the digital signal processing was completed in the final few days available for us to work on it. This was due to the fact that final game testing had revealed too many errors in actually measuring the pitch of the user, so we switched to an entirely different Python package and completely reworked the code. It was in those final few days where instead of working on the different parts in parallel we came together as a team and worked together.

Test Plan

The majority of testing required for the unit as a whole came in the form of software debugging and detailed software testing. To ensure success, the following flowchart was developed to dictate the logical flow of software testing given various outcomes of initial tests.

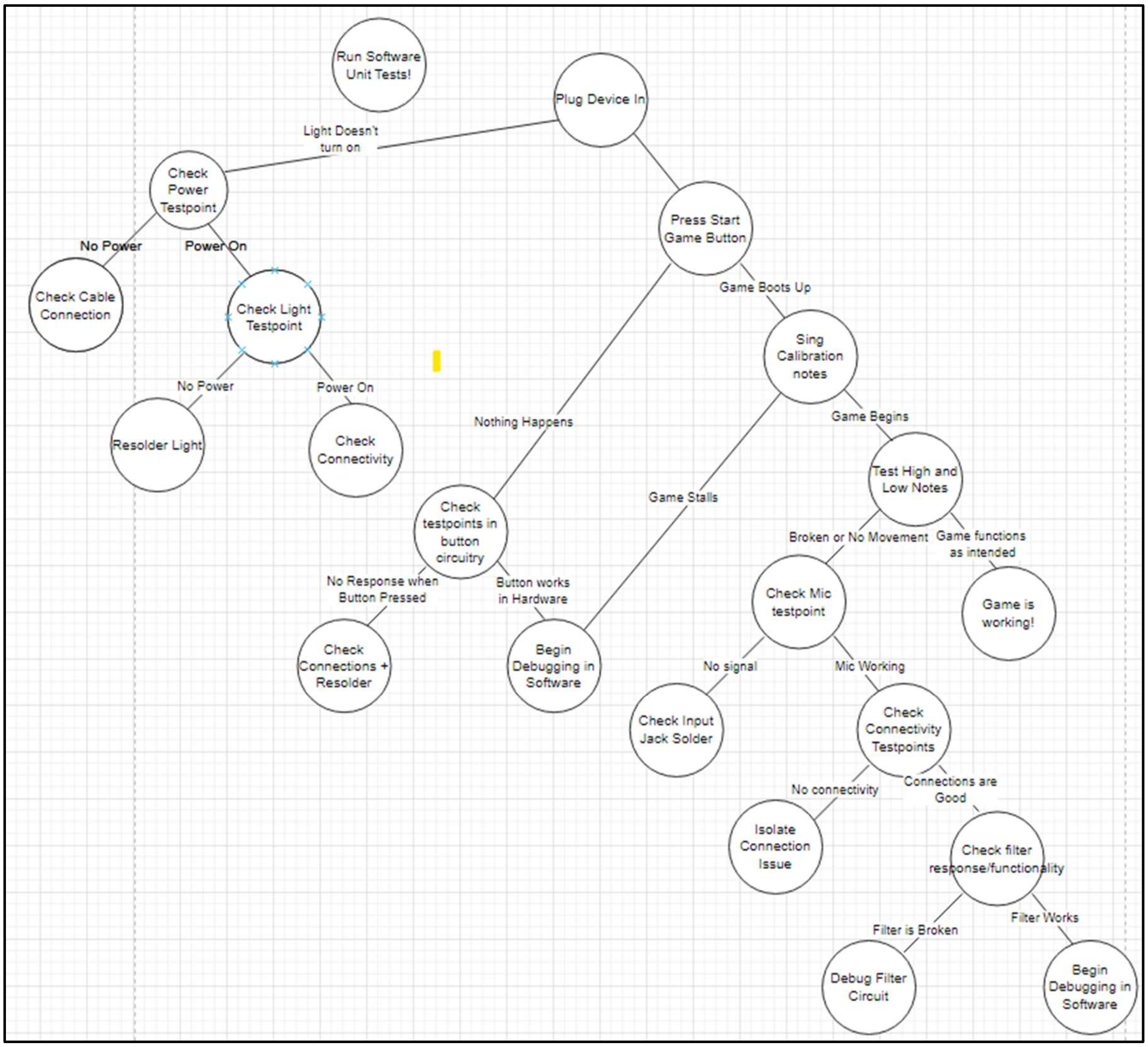

Figure 11. Software Testing Flowchart

In addition to the testing described in the flowchart above, unit tests were added to the game system, pitch recognition software, and state machine implemented in software. A CI/CD pipeline was also added in GitHub to ensure frequent software changes would be supported without danger to game function.

Final Results

We set out to build a game that is an all-in-one experience for a single player. The user needed to be able to plug in the machine to an external USB-C power supply and play Pong against the computer, controlling their paddle using solely their voice. The screen, processor, and PCB needed to be housed in a single portable enclosure that allows anyone to play the game wherever they have access to a power supply. The device should allow for calibration of input pitch in order for players of any voice type to successfully control the input to the game.

More concretely, our success criteria are enumerated below:

- Device can accept and utilize calibration input
- Device can operate a pong game with smooth expected visual output
- Device can provide audio output to complement gameplay
- Device can accept audio input to control user's pong paddle
- Pitch functionality for paddle direction is successfully implemented
- Device is enclosed in an appropriately professional apparatus

Our final project fully realized the goals we set out with at the beginning of the semester. We have successfully met all of the success criteria listed above. Our game of pong is housed in a professional, aluminum enclosure with panel mounted connectors. The user is able to play a pong game with smooth expected visual output with each game started by calibrating a personalized reference pitch. Lastly, a collision sound is played upon contact of the ball with the paddle. In summary, all the criteria we set for ourselves were met with our final product.

Costs

Our development of Voice Controlled Pong used $467.81 of our budget. The largest cost items consisted of the Raspberry Pi, Screen, Console and SD Card. The Raspberry Pi microcontroller is in high demand and is unlikely to be heavily discounted when purchased in volume. The screen purchased for our project was bought on Amazon, at scale these would be sourced from a wholesaler for a significant price deduction. The PCB and Components (Raspberry Pi Hat) for our project were manufactured and assembled separately. For an order of 10,000 boards, the unit price would drop to $0.43 as per an online quote [30]. Combined with the components discounted unit price an order of 10,000 boards would cost $0.72 per unit. Hammond Manufacturing [17], the supplier of the case, is able to custom machine the consoles. This would allow us to easily order large quantities of the console that would come ready to assemble.

Item	Total Cost
Raspberry Pi 4	$167.94
10" IPS Touchscreen Display (Screen)	$75.89
PCB and Components (Raspberry Pi Hat)	$62.04
Hammond Manufacturing Console	$40.72
SD Card (update w/ capacity)	$27.37

Figure 12. Abbreviated Cost Table

Future Work

If we had another semester to develop this project, something we considered was adding a "hard mode" to the game. Currently the game works by determining if the inputted pitch is higher or lower than the user's reference pitch and moves the paddle accordingly. In hard mode, the high and low calibration notes would be used to set upper and lower frequency bounds instead of being averaged to create a reference note. These high and low bound frequencies would correspond to the upper and lower bounds of the screen accordingly, so that a linear scale is made with a certain frequency corresponding to a specific position of the paddle. Therefore, when a user inputs a specific vocal frequency, the paddle moves towards the frequency's corresponding position on the screen. We did not implement this feature because it requires significantly more processing power, microphone performance and user skill.

References

[1] *Maker Faire 2008: Pong and Asteroids Watch*, (Jan. 01, 2010). [Online Video]. Available: https://ieeetv.ieee.org/channels/spectrum/maker-faire-2008-pong-and-asteroids-watch

[2] T. Perry and P. Wallich, "Inventing the Atari 2600," *IEEE Spectrum*, Mar. 1983, [Online]. Available: https://spectrum.ieee.org/atari-2600

[3] R. Rabenja, S. Amine, and J. Davidson, "Speech recognition for game control," in *The 2nd Annual IEEE Northeast Workshop on Circuits and Systems, 2004. NEWCAS 2004.*, 2004, pp. 97–99. doi: 10.1109/NEWCAS.2004.1359028.

[4] "Raspberry pi documentation," *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.com/documentation/. [Accessed: 12-Sep-2022].

[5] "Pygame," *Pygame Front Page - pygame v2.1.1 documentation*. [Online]. Available: https://www.pygame.org/docs/. [Accessed: 12-Sep-2022].

[6] KiCad, "Documentation," *KiCad*. [Online]. Available: https://docs.kicad.org/. [Accessed: 26-Sep-2022].

[7] "DigiKey Electronics - Electronic Components Distributor." https://www.digikey.com/ (accessed Sep. 12, 2022).

[8] Microsoft, "Running visual studio code on raspberry pi," *RSS*, 03-Nov-2021. [Online]. Available: https://code.visualstudio.com/docs/setup/raspberry-pi. [Accessed: 26-Sep-2022].

[9] "Buildroot - making embedded Linux Easy." [Online]. Available: https://buildroot.org/. [Accessed: 27-Sep-2022].

[10] "Printed Circuit Board Recycling Methods," Workshop Materials on WEEE Management in Taiwan, 2012. https://www.epa.gov/sites/default/files/2014-05/documents/handout-10-circuitboards.pdf

[11] "Raspberry Pi boards are now being recycled - at the same factory that made them," *ZDNET*.https://www.zdnet.com/article/raspberry-pi-boards-are-now-being-recycled-at-the-same-factory-that-made-them/ (accessed Sep. 26, 2022).

[12] "Final Report | Recycling of Liquid Crystal Displays for Maximum Resource Recovery | Research Project Database | Grantee Research Project | ORD | US EPA." https://cfpub.epa.gov/ncer_abstracts/index.cfm/fuseaction/display.abstractDetail/abstract_id/9847/report/F (accessed Sep. 26, 2022).

[13] “NFPA 70®: National Electrical Code®.” https://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?code=70 (accessed Sep. 12, 2022).

[14] “NEMA 1 Enclosures - NEMA 1 Electrical Enclosure | Nema Enclosures.” https://www.nemaenclosures.com/enclosure-ratings/nema-rated-enclosures/nema-1-enclosures.html (accessed Sep. 12, 2022).

[15] Webmaster, “Embedded C Coding Standard,” May 26, 2016. https://barrgroup.com/embedded-systems/books/embedded-c-coding-standard (accessed Sep. 12, 2022).

[16] “Electrical - Standards | Occupational Safety and Health Administration.” https://www.osha.gov/electrical/standards (accessed Sep. 12, 2022).

[17] H. Mfg, “Electrical & Electronic Enclosures, Cabinets & Racks, Outlet Strips and Electronic Transformers,” *Hammond Mfg.* https://www.hammfg.com/ (accessed Dec. 13, 2022).

[18] “KiCad EDA.” https://www.kicad.org/ (accessed Dec. 13, 2022).

[19] “Analog Filter Wizard.” https://tools.analog.com/en/filterwizard/

[20] R. P. Sallen and E. L. Key, “A practical method of designing RC active filters,” IRE Transactions on Circuit Theory, vol. 2, no. 1, pp. 74–85, 1955, doi: 10.1109/TCT.1955.6500159.

[21] “PONG Trademark of ATARI INTERACTIVE, INC. - Registration Number 2611782 - Serial Number 76148525 :: Justia Trademarks.” http://trademarks.justia.com/761/48/pong-76148525.html (accessed Dec. 13, 2022)

[22] “Patents Assigned to RASPBERRY PI (TRADING) LIMITED - Justia Patents Search.” https://patents.justia.com/assignee/raspberry-pi-trading-limited (accessed Dec. 13, 2022).

[23] T. Ramabadran, “Method and Apparatus for Speech Reconstruction within a Distributed Speech Recognition,” 2004/0138888A1

[24] “Microphone Amplifier with AGC and Low-Noise Microphone Bias.” Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/max9814.pdf

[25] “MAX11163 16-Bit, 250ksps, +5V Unipolar Input, SAR ADC, in Tiny 10-Pin µMAX.” Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX11163.pdf

[26] “MCP6021/1R/2/3/4 Rail-to-Rail Input/Output, 10 MHz Op Amps.” Microchip. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/20001685E.pdf

[27] “Raspberry Pi trademark rules and brand guidelines,” *Raspberry Pi*. https://www.raspberrypi.com/trademark-rules/ (accessed Dec. 13, 2022).

[28] "Should I match impedances of my microphone to my mixer?". Shure Service & Repair. Shure. May 23, 2022. Retrieved June 27, 2022

[29] R. R. Devi and D. Pugazhenthi, “Ideal Sampling Rate to reduce distortion in Audio Steganography,” in INTERNATIONAL CONFERENCE ON COMPUTATIONAL MODELLING AND SECURITY (CMS 2016), 2016, vol. 85, pp. 418–424. doi: 10.1016/j.procs.2016.05.185.

[30] “PCB Cost Calculator | Online PCB Manufacturing Quote.” https://www.pcbcart.com/quote (accessed Dec. 13, 2022).

Appendix

Item	Total Cost
Raspberry Pi 4	$167.94
SD Card (update w/ capacity)	$27.37
10” IPS Touchscreen Display	$75.89
PCB Manufacture	$33.00
PCB Assembly	$30.00 (estimate)
Hammond Manufacturing Console	$40.72
USB C Round Panel Mount Extension Cable	$24.88
Mini LED Arcade Button - 24mm	$5
Shunt Voltage Reference IC	$3.87
16 Bit Analog to Digital Converter	$11.11
Amplifier IC 1-Channel (Mono) Class AB	$4.36
General Purpose Amplifier	$7.24
Ground Reference (Virtual) Voltage Reference IC	$2.46
Panel Mount Stereo Audio Extension	$3.95
Unidirectional Lapel Microphone	$16.97
Micro USB Right Angle Ribbon Cable	$15.00
Hex Standoff Threaded M2.5x0.45 Steel 0.472"	$2.85
M2.5x0.45 Hex Nut 0.197" (5.00mm) Steel	$0.9
Hex Standoff Threaded M2.5x0.45 Brass 0.197"	$1.45
M2.5x0.45 Flat Head Machine Screw Phillips Drive	$0.75
HDMI Right Angle Ribbon Cable	$8.41
Raspberry Pi Hat Header Pins	$6.03
Silicon Sealant and Adhesive	$7.66

Danger of Generative AI

Introduction

The first implementation of speech recognition technology was created by Bell Laboratories in the early 1950s with their computing system nicknamed “Audrey”. The machine was able to decipher which numerical digit, zero through nine, was spoken by the head engineer with approximately 90% accuracy (Linguae, 2022). Fast forward about a decade to 1962, and IBM debuted the “Shoebox” which could now recognize 16 distinct words. In the 60 years that have passed since then, these technologies have advanced at exponential levels of innovation. Today we have incredibly complex virtual assistants in mobile devices complete with advanced artificial intelligence and machine learning models capable of conducting complete conversations in dozens of languages. Beyond the ability to interpret words and phrases, voice analysis software can detect vocal inflexion to determine the mood and emotion behind speech (McAlpine & Jahnke, 2020). These technologies have gotten so much smarter and more complex they are now extremely dangerous. The new wave of speech technology is using AI to mimic or recreate someone’s voice. A primary example of this is Microsoft’s VALL-E, which can utilize just a three second clip of your voice to replicate it speaking any given text (Blake, 2023). The capstone project undertaken by my group utilized one of the simpler and safer forms of speech recognition, pitch detection, to play a video game. This technology works similar to tuners for musical instruments which analyze the frequency of the audio signal to determine what the pitch is. To apply our technical expertise in this new topic, we integrated pitch detection into the video game, Pong. Pong was the first commercially available video game created in 1972 by Atari in which the player is tasked with moving a bar up and down, representing a paddle, to bounce a ball back and forth with another player, creating a two-dimensional version of tennis (Centre for Computing History, 2019). My team and I recreated this classic game, substituting the joystick controller with a microphone. The player will input a high-pitched sound to move the paddle up, and a low-pitched sound to move the paddle down. There were numerous technical challenges involving both analog and digital signal processing techniques, along with integrating the various hardware and software elements into a cohesive system. Additionally, ensuring everyone is not only capable of playing this game (regardless of any sort of musical ability or unique speech attributes), but can do so without concern over privacy of their voice.

Methodology

The project was chosen to incorporate a wide variety of electrical engineering, computer engineering, computer science, and design skills, while remaining creative, engaging, and fun. As a group, implementing an interactive game that all users could understand and play without background knowledge was a more intriguing task to us than a project in a non-game, or noninteractive format. The plan for this project very much relied on us working as a team utilizing each of our strengths. We split up into a hardware team and a software team. Isaac, the strongest programmer of the group, was the software lead and wrote most of the code for the project while I, John, and Charlie formed the hardware team to develop all the physical components, mainly the custom printed circuit board (PCB). Throughout the semester we were able to work in parallel before coming together as a team during the final weeks.

The heart of our system was a Raspberry Pi 4, a single board computer which functions as both an embedded microcontroller and the primary processor for powering game functionality. It also served to interface with the audio input from our PCB and produce an audio output. As a tool, it was used in testing software, and generating voltage for other parts of the system from its available power pins. Many useful software tools and techniques were used to create the finished product. First, we used BuildRoot, a tool designed for creating a custom operating system (OS) for Raspberry Pi devices. Then, we used a custom build of Visual Studio Code to run on our OS; a software environment for writing and testing code natively on the Raspberry Pi. Several advanced software techniques were utilized including multithreading, locking, and data synchronization to improve performance by allowing distinct parts of the processor to manage different tasks at the same time. On top of that, a finite state machine (FSM) controlled which tasks were to be run at a specific time during operation of the game.

The circuit schematic and the final PCB layout of the components were designed using the KiCad software. In addition, the analytical design work for the PCB was done in part with assistance from the Analog Devices Analog Filter Wizard. The board was comprised of three main systems: the amplifier, the filter, and the analog-to-digital converter (ADC). The amplifier component increases the magnitude of the voltage signal from the microphone to make it easier to analyze. The filter was designed with the Sallen-Key architecture and is responsible for getting rid of as many extraneous high frequencies as possible that were picked up from the microphone that did not come from the user's voice. Finally, the ADC takes this modified signal and converts it into a digital signal as fast as 250,000 times per second which can then be used by the Raspberry Pi.

Technical Work

The total function of our PCB was to accept the microphone signal provided by our implemented lapel microphone, and convert it to a decipherable signal for the Raspberry Pi to use in processing. As stated before, the function was split into three stages: amplification, filtering, and analog to digital conversion.

The amplifier selection was somewhat inhibited by available information for the selected microphone. A lavalier, or lapel microphone was selected for user friendly implementation and flexibility in movement, but a well-documented option was not available. However, the lavalier microphone chosen (like most) was categorically an electret microphone, thus telling us the output impedance would be approximately 2 kOhms, and the output voltage would be approximately 20 mV; the typical ranges for electret microphones used in this context. An amplifier was then selected based upon the automatic gain control (AGC: the ability to amplify a signal at multiple different magnitudes) inclusion, input impedance, and price. The MAX 9814 was then selected, as it includes AGC at desired levels of 40, 50, and 60 dB, has an appropriate input impedance of 100 kOhms, and boasts a price of under $2.20. Standard audio amplification stipulates a minimum amplifier input impedance of 5-10 times output impedance, thus the 45 multiplier that the MAX provides was more than adequate. As a team, we then researched in the supported documentation to design the required surrounding circuity to be placed on the board.

For the filter design, first the performance required of the hardware filter needed to be defined. For audio signals, the industry standard for sampling rate is 44.1 kHz. With this rate in mind, the Nyquist theory dictates that any frequency above 22.05 kHz in our sampled signal would contribute to aliasing. This means that the actual signal frequency can be misidentified, and the reconstruction of that signal during sampling can be distorted in many different ways. Therefore, the primary function of this filter would be to reduce the effect of aliasing, as well as disregard signals that would not have come from the user's voice. It was determined via spectral analysis of a vocal sample that most of the signal frequencies in a human voice will be between 20 Hz and 3.4 kHz. To reduce aliasing, frequencies above 22.05 kHz needed to be reduced as much as possible. Upon initial analysis, our specifications determined above resulted in a fifth order filter—a single filter with the behavioral characteristics of five sequential filters. However, given that the Sallen-Key architecture can produce a second order filter from a single operational amplifier chip, it was determined that the use of two dual op-amp chips could complete construction of an eighth order filter. This eighth order filter provided us with additional attenuation of frequencies that would not contribute to aliasing but would contribute to the noise of the signal (beyond the frequency range of the human voice). The filter did not need to be perfect at this as additional filtering was done in software on the Raspberry Pi.

The final main hardware stage to design was the analog-to-digital conversion. At first, we thought we found an easy solution by potentially utilizing the built in ADC on a readily available and cheap chip the Raspberry Pico. It was able to sample 12 bits at a blazing fast 500,000 samples/second, but after further investigation it was discovered that the output was subject to extreme levels of noise that would make digital processing much more difficult. Taking into consideration possible large variances in pitch and volume from different users, we desired a chip that had better performance and research led to us to the MAX11163. This ADC was 16 bits instead of 12 bits, meaning it was capable of 16 times greater signal resolution, and it still sampled at 250,000 samples/second. Additionally, this chip was chosen for its compatibility with interfacing to the Raspberry Pi. This caused the design of the necessary circuitry on the PCB to be more intuitive.

The entire system was then housed in an aluminum enclosure, complete with a big LCD touchscreen, a push button for starting and resetting the game, and a single power cable to be plugged into any standard wall outlet. Our custom PCB was designed to connect directly to This gave our project a very professional look and gave insight into its potential of being a consumer product.

Ethical Concerns

Broadly speaking, the pitch detection technology used in our project falls under the umbrella of speech recognition. In more advanced and complex applications, there are several ethical questions that come along with that which developers must consider, primarily protecting the right to privacy and preventing algorithmic bias. These questions are extremely relevant considering the prevalence of devices such as Amazon's Echo products with Alexa, which are present in an estimated 25% of US households (Vincent, 2021). Echo devices are controlled by customers using only their own voice, with functions such as playing music, completing internet searches, checking the weather, and much more. The microphones are essentially always on so that they can recognize when there is a new voice command. Therefore, valid concerns arise about how much these devices might be recording and collecting, like private conversations and other sensitive information. Most consumers would much prefer the things they say to not be available to large corporations, so developers must be careful when dealing with something as personal and intimate as someone's own voice.

Now, more than ever before, this is something that needs to be seriously considered and actively fought to prevent. The exponential advancement of artificial intelligence, and more specifically generative artificial intelligence, makes data leaks vastly more dangerous. During the development of this project, Open AI released ChatGPT-3 to the public. This language model is more capable than any other openly available software. It takes in whatever you ask, say, or tell it, and will output a uniquely generated response that can write code, perform math operations, analyze literature, and infinitely more. In fact, ChatGPT helped our team figure out how to immediately launch our game when the device was turned on. The possibilities are truly endless.

As I revealed earlier, VALL-E can replicate a human voice with just three seconds of audio. Imagine the consequences if someone with this technology had access to the data that may or may not be recorded by these smart home devices. As with almost every neural network AI, the larger the data set that it is trained on the more accurate and the better it will be. If VALL-E had access to hours and hours of audio recordings, its mimic of that person's voice would only become more indistinguishable. This has already been proven to be true, with people utilizing a recording artist's entire musical catalog to recreate them singing whatever lyrics they want. With this in mind, our project does not store or even record the actual audio from the microphone, only the frequency of the signal. Therefore, if our device were theoretically hacked, the possible damage that could be done would be miniscule in comparison to other audio devices.

Algorithmic bias also comes into play here, as it does with most modern algorithms whose behavior is determined by the type of input information. It is well known now how prejudice can be programmed into modern software, whether unintentional or not (Heilweil, 2020). Simply put, if there is a skew or bias in the data being fed to the neural network or machine learning algorithm, the output functionality will be affected in a similar fashion. When this arises in vocal technologies, such as pitch detection used in our project, it could potentially give way for certain biases such as sexism or racism. The sound of one's voice is not often initially associated with various stereotypes, but it has an enormous impact. There are undeniable societal norms that try to dictate how a certain person should sound (Simonite, 2018). For example, it is commonly thought that men have higher pitched voices than women. However, not only is that an extremely inconsistent generalization, but it also ostracizes how transgender individuals may sound. The potential of a product either misgendering a user or wrongly operating based on some assumed gender is extremely dangerous. These stereotypes come along with many other aspects of an individual, and no one should ever be categorized based on the sound of their voice.

The way in which our group is tackling a miniscule portion of this issue is by setting up a calibration sequence at the start of each game. The player will start by making a sound into the microphone at their personal standard tone, and that will function as a cutoff for future inputs. Any sound at a relative higher pitch will be what causes the paddle to move up, and any sound at a relative lower pitch will move the paddle down. With a project on the scale of ours, some of these surrounding fears and questions are not necessarily valid for our scope, as there is no actual recording of the microphone output being done. Nevertheless, when working in a certain field, it is important to understand the bigger picture to gain a more wholistic perspective on the subject.

Conclusion

In our final product, the game works by determining if the input pitch is higher or lower than the user's reference pitch and moves the paddle accordingly. For future work on this topic, one could implement a much more difficult game mode that could help with vocal training for singers. Instead of simply recognizing a high note and a low note relative to a reference pitch, a linear scale could be created so that a certain frequency corresponds to a specific position of the paddle in the game. Therefore, when a user inputs a specific vocal pitch, the paddle moves towards that frequency's corresponding position on the screen. The player must be perfectly accurate with what pitch they are singing on, and so this could help singers practice hit the right note every time.

Throughout the development of our capstone project, I learned a lot about the skills and techniques required to design and create a complex project from start to finish. The technical work challenged me in ways I had not previously seen in my courses and increased my ability to work in a team while simultaneously requiring me to solve problems with greater independence. Additionally, I was able to discover nuances about some of the dangers of modern software with my STS research. As someone who does not identify with a minority group, diving into the field of algorithmic bias not only enhanced my understanding of how intelligent algorithms are made, but deeply widened my perspective on the impacts of these technologies to the people around me. As my graduation is fast approaching and I am about to enter the work force, gaining as much knowledge and experience from an opportunity like this was extremely beneficial. Artificial intelligence is not going away anytime soon. By the second, it only becomes more powerful. It is up to engineers like us to determine whether that power will be used for good or for evil.

References

Blake, A. (2023, January 10). *This AI can spoof your voice after just three seconds*. Digital Trends. Retrieved April 7, 2023, from https://www.digitaltrends.com/computing/microsoft-vall-e-ai-can-replicate-your-voice/

Centre For Computing History. (2019). *Atari PONG*. Centre For Computing History. Retrieved October 31, 2022, from http://www.computinghistory.org.uk/det/4007/Atari-PONG/

Heilweil, R. (2020, February 18). *Why algorithms can be racist and sexist*. Vox. Retrieved October 31, 2022, from https://www.vox.com/recode/2020/2/18/21121286/algorithms-bias-discrimination-facial-recognition-transparency

McAlpine, K. J., & Jahnke, A. (2020, September 17). *Why policing the way voices sound has to stop*. The Brink. Retrieved October 31, 2022, from https://www.bu.edu/articles/2020/why-policing-the-way-voices-sound-has-to-stop/

Simonite, T. (2018, March 19). *This call may be monitored for tone and emotion*. Wired. Retrieved October 31, 2022, from https://www.wired.com/story/this-call-may-be-monitored-for-tone-and-emotion/

Summa Linguae. (2022, March 7). *Speech recognition software: History, present, and future*. Summa Linguae. Retrieved October 31, 2022, from https://summalinguae.com/language-technology/speech-recognition-software-history-future/

Vincent, J. (2021, December 23). *Alexa is nagging you more because Amazon knows you don't care about its new features*. The Verge. Retrieved October 31, 2022, from https://www.theverge.com/2021/12/23/22851451/amazon-alexa-by-the-way-use-case-functionality-plateaued

Prospectus

Introduction:

The first implementation of speech recognition technology was created by Bell Laboratories in the early 1950's with their computing system nicknamed "Audrey". The machine was able to decipher which numerical digit, zero through nine, was spoken by a user (Linguae, 2022). In the 70 years since then, these technologies have advanced at exponential levels of innovation. Today we have incredibly complex virtual assistants in our phones complete with advanced artificial intelligence and machine learning models capable of carrying out complete conversations in dozens of languages. Beyond the ability to interpret words and phrases, voice analysis software can actually detect vocal inflexion to determine the mood and emotion behind the words (McAlpine & Jahnke, 2020). The capstone project undertaken by my group will utilize one of the simpler forms of speech recognition, pitch detection, in order to play a video game. This technology works similar to musical instrument tuners which analyze the frequency of the audio signal to determine what the pitch is. The application we are integrating this into is the classic game, Pong. Pong was the first commercially available video game created in 1972 by Atari in which the player is tasked with moving a bar up and down, representing a paddle, to bounce a ball back and forth with another player, creating a two-dimensional version of tennis (Centre for Computing History, 2019). My team and I will recreate this classic game, substituting the joystick controller with a microphone. The player will input a high-pitched sound to move the paddle up, and a low-pitched sound to move the paddle down. There will be numerous technical challenges involving both analog and digital signal processing techniques required for a successful project, along with integrating the various hardware and software elements into a cohesive system. Additionally, it will be vital to ensure everyone is capable of playing this game, regardless of any sort of musical ability or unique speech attributes.

Technical Topic:

The essential problem that challenges our group in this project is converting an analog audio signal, a human voice, into a digital signal that a computer can read and understand to use as the input for the game. As mentioned before, this task will involve both hardware and software elements responsible for analog and digital signal processing, respectively. The former will be accomplished by designing a custom printed circuit board, or PCB, using the KiCad software. The board will be comprised of three main systems: the amplifier, the filter, and the analog-to-digital converter (ADC). The amplifier component will increase the magnitude of the voltage signal from the microphone to make it easier to analyze. The filter will be designed with the Sallen-Key architecture, responsible for getting rid of as many extraneous high frequencies as possible that were picked up from the microphone that did not come from the user's voice. Finally, the ADC will take this modified signal and convert it into a digital signal as fast as 250,000 times per second which can then be used by the computer.

This computer will be a Raspberry Pi device that will then be responsible for performing all necessary software operations, including running the actual video game itself. The game will have to be programmed from scratch with the assistance of libraries to help with animations such as PyGame. In order to do so, we will need to change the operating system on board as well. This will involve modifying a version of Linux, allowing us to obtain the ability to run in real-time accessing the inputs on the Raspberry Pi. Furthermore, the desktop operating system will expand our debugging capabilities when we encounter obstacles, as well as open the door to implement other external libraries like Liquid DSP to aid in digitally processing the signal from the PCB.

The goal of a project like this is twofold; to take the knowledge and skills developed throughout the curriculum and apply it to something real, and also to create something that allows others to be involved and engaged with this technical work. The electrical and computer engineering fundamentals course sequence centers around PCB design projects which gave us a great base of information to build from. It was important for us to utilize this opportunity to make something fun – to expand beyond typical assignments and design outside the box. Choosing the game Pong to focus on was very intentional in that it is the one video game in which almost everyone knows. Older users can recollect on memories of the original and be reinspired by our unique refresh, and younger players can appreciate the greater level of user engagement compared to the average video game. Since moving the paddle only requires a relatively high or low pitch, no singing talent is needed and anyone can play the game. In the future, the blueprint of our system could be used to make another version where an exact note determines the position of the paddle in the game. A product like this might be a tool for singers to help train and control their pitch. Overall, our project has a fairly small number of physical components and simpler assembly compared to other projects. This, along with a small budget, creates potential for certain versions of the system to be a marketable product. There are numerous aspects of this endeavor that could warrant further research and development.

STS Topic:

Broadly speaking, the pitch detection technology used in our project falls under the aforementioned umbrella of speech recognition. In more advanced and complex applications, there are several ethical questions that come along with that which developers must consider, primarily protecting the right to privacy and preventing algorithmic bias. These questions are extremely relevant considering the prevalence of devices such as Amazon's Echo products with Alexa, which are present in an estimated 25% of US households (Vincent, 2021). Echo devices are controlled by customers using only their own voice, with functions such as playing music, completing internet searches, checking the weather, and much more. The microphones are essentially always on so that they can recognize when there is a new voice command. Therefore, valid concerns arise about how much these devices might be recording and collecting, like private conversations and other sensitive information. The vast majority of consumers would much prefer the things they say to not be available to large corporations, so developers must be careful when dealing with something as personal and intimate as someone's own voice.

Algorithmic bias also comes into play here, as it does with most modern algorithms whose behavior is determined by the type of input information. It is well known now how prejudice can be programmed into modern software, whether unintentional or not (Heilweil, 2020). Simply but, if there is a skew or bias in the data being fed to the neural network or machine learning algorithm, the output functionality will be affected in a similar fashion. When this arises in vocal technologies, such as pitch detection used in our project, it could potentially give way for certain biases such as sexism or racism. The sound of one's voice is not often initially associated with various stereotypes, but it definitely has a large impact. There are undeniable societal norms that try to dictate how a certain person should sound (Simonite, 2018). For example, it is commonly thought that men have higher pitched voices than women. However, not only is that an extremely inconsistent generalization, but it also ostracizes how transgender men and women may sound. The potential of a product either misgendering a user or wrongly operating based on some assumed gender is very dangerous. These stereotypes come along with many other aspects of an individual, and no one should ever be categorized based on the sound of their voice.

The way in which our group is tackling a miniscule portion of this issue is by setting up a calibration sequence at the start of each game. The player will start by making a sound into the microphone at their personal standard tone, and that will act as a cutoff for future inputs. Any sound at a relative higher pitch will be what causes the paddle to move up, and any sound at a relative lower pitch will move the paddle down. With a project on the scale of ours, some of these surrounding fears and questions are not necessarily valid for our scope, as there is no actual recording of the microphone output being done. Nevertheless, when working in a certain field, it is important to understand the bigger picture to gain a more wholistic perspective on the subject.

Conclusion:

Throughout the development of my capstone project, I look forward to learning a lot about the skills and techniques required to design and create a complex project from start to finish. The technical work will challenge me in ways I have not seen in just my courses and will increase my ability to work in a team while simultaneously requiring me to solve problems with greater independence. Additionally, I will be able to discover nuances about some of the dangers of modern software with my STS research. As someone who does not identify with a minority group, diving into the field of algorithmic bias will not only enhance my understanding of how intelligent algorithms are made, but deeply widen my perspective on the impacts of these technologies to the people around me. As my graduation is fast approaching and I am about to enter the work force, gaining as much knowledge and experience from an opportunity like this is vital.

References

Centre For Computing History. (2019). *Atari PONG*. Centre For Computing History. Retrieved October 31, 2022, from http://www.computinghistory.org.uk/det/4007/Atari-PONG/

Heilweil, R. (2020, February 18). *Why algorithms can be racist and sexist*. Vox. Retrieved October 31, 2022, from https://www.vox.com/recode/2020/2/18/21121286/algorithms-bias-discrimination-facial-recognition-transparency

McAlpine, K. J., & Jahnke, A. (2020, September 17). *Why policing the way voices sound has to stop*. The Brink. Retrieved October 31, 2022, from https://www.bu.edu/articles/2020/why-policing-the-way-voices-sound-has-to-stop/

Simonite, T. (2018, March 19). *This call may be monitored for tone and emotion*. Wired. Retrieved October 31, 2022, from https://www.wired.com/story/this-call-may-be-monitored-for-tone-and-emotion/

Summa Linguae. (2022, March 7). *Speech recognition software: History, present, and future*. Summa Linguae. Retrieved October 31, 2022, from https://summalinguae.com/language-technology/speech-recognition-software-history-future/

Vincent, J. (2021, December 23). *Alexa is nagging you more because Amazon knows you don't care about its new features*. The Verge. Retrieved October 31, 2022, from https://www.theverge.com/2021/12/23/22851451/amazon-alexa-by-the-way-use-case-functionality-plateaued