Google Summer of Code 2025

A text-to-speech (TTS) tool for Pharo

Mentee Name :

Ms. Neerja Doshi

Sardar Patel Institute of Technology, Mumbai

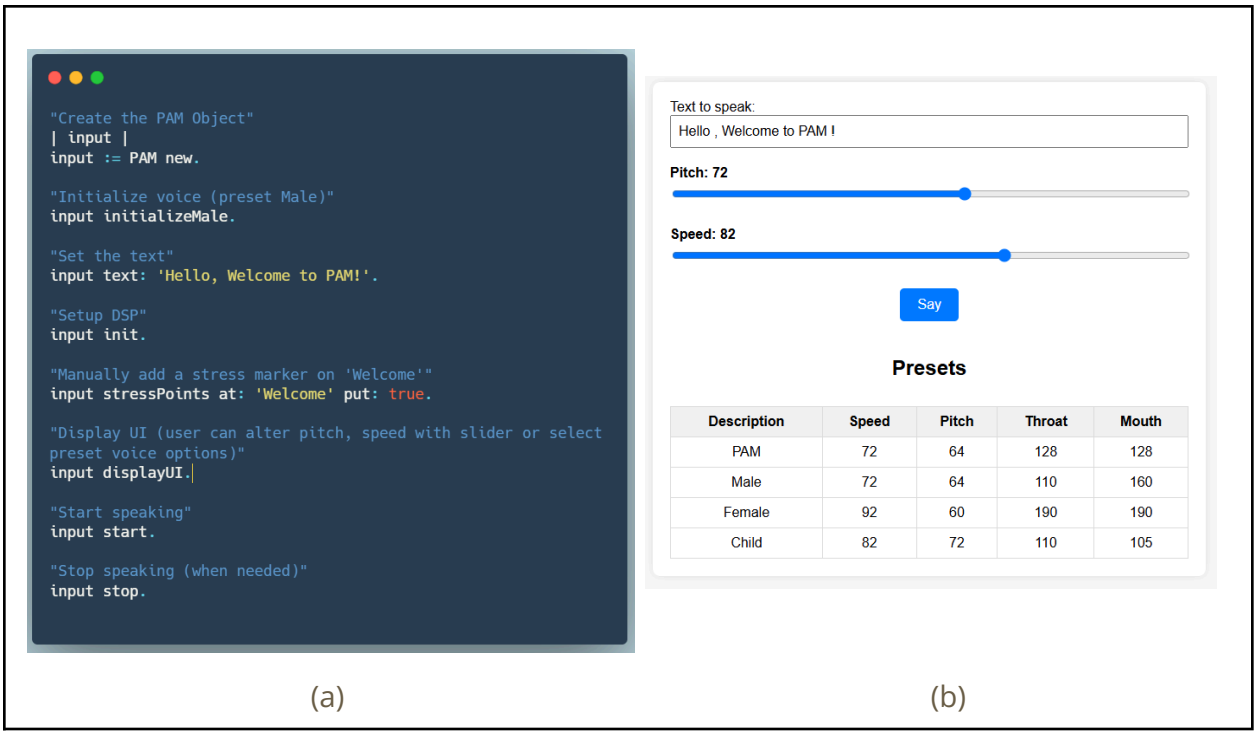
Mentors:

Domenico Cipriani , Nahuel Palumbo

Pharo Consortium

07.04.2025

A Glimpse of what I'll try to achive during GSoC'25



```
"Create the PAM Object"
| input |
input := PAM new.

"Initialize voice (preset Male)"
input initializeMale.

"Set the text"
input text: 'Hello, Welcome to PAM!'.

"Setup DSP"
input init.

"Manually add a stress marker on 'Welcome'"
input stressPoints at: 'Welcome' put: true.

"Display UI (user can alter pitch, speed with slider or select
preset voice options)"
input displayUI.

"Start speaking"
input start.

"Stop speaking (when needed)"
input stop.
```


Description	Speed	Pitch	Throat	Mouth
PAM	72	64	128	128
Male	72	64	110	160
Female	92	60	190	190
Child	82	72	110	105

(a) (b)

(a) Pharo Code snippet for TTS. (b) UI received from displayUI argument

Introduction

The project, **PAM (Pharo Automated Mouth)**, is inspired by the JavaScript/Web Audio adaptation of **SAM (Software Automated Mouth)** originally developed for the Commodore 64. PAM aims at providing a robust and extensible **Text-to-Speech (TTS)** in Pharo environment system with clearer, more intelligible audio compared to SAM.

The tool will be structured into two core components:

- A **Reciter** that processes input text into phonemic and prosodic representations.
- A **Digital Signal Processor (DSP)**, written in **Phausto**, that translates these traits into real-time audio.

My goal is to build a modular, well-documented system that not only lays a **foundation model of TTS** within Pharo but also for extending speech synthesis to other natural languages in the future with better prosody.

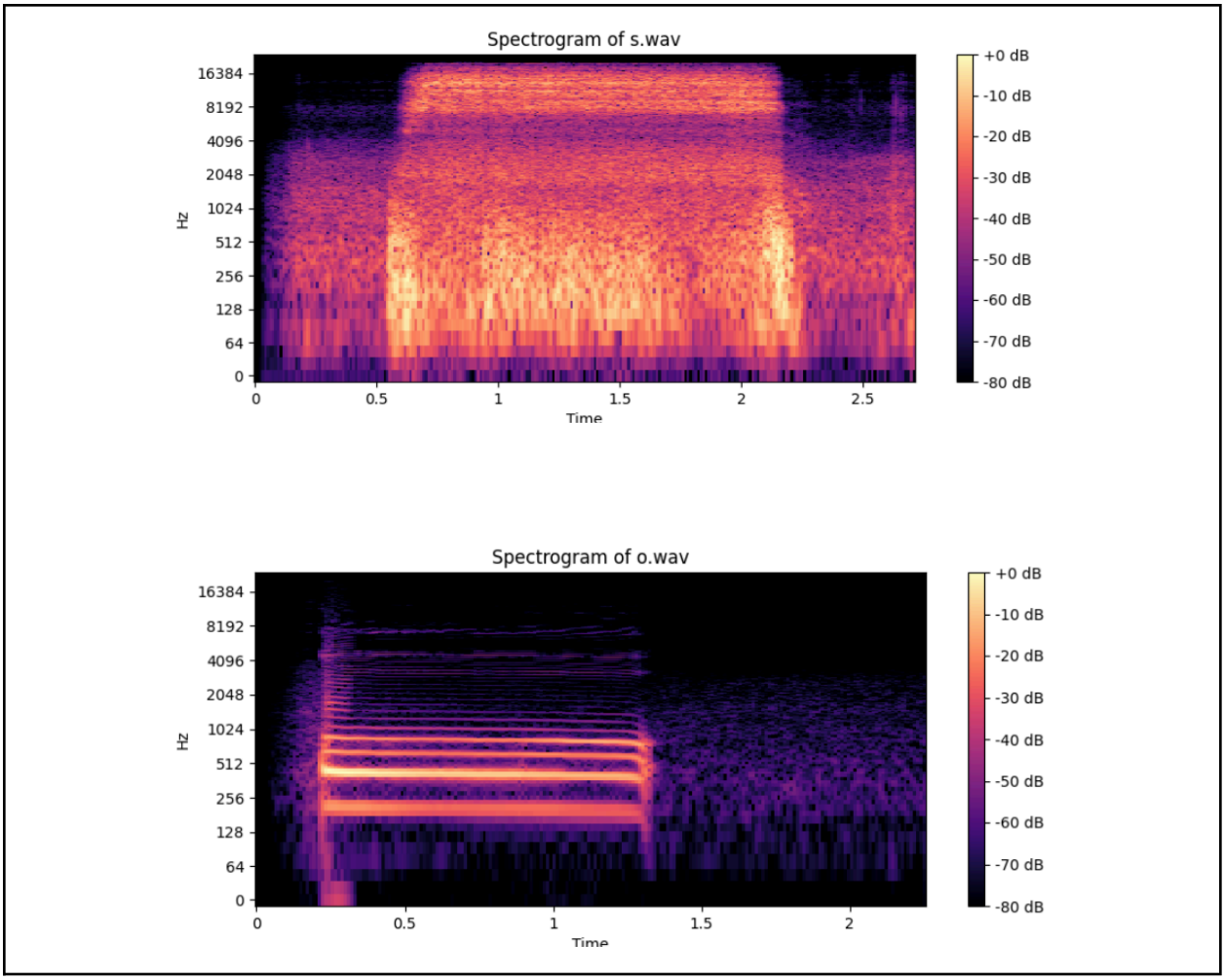
My Initial Work Until Now

Points of Difference in each of the following TTS models	Formant Synthesis (SAM)	Concatenative Synthesis	Parametric Synthesis (HMM based)	Neural TTS (WaveNet, Tacotron, VITS)	Cloud-Based TTS APIs (Google, AWS, Azure, ElevenLabs)
Full Form	-	-	HMM: Hidden Markov Model	WaveNet: DeepMind Neural Speech Model Tacotron: Google Neural Speech Model VITS: Variational Inference Text-to-Speech	AWS: Amazon Web Services API: Application Programming Interface
How It Works	Uses mathematical models to generate human speech by simulating formants (speech frequency components).	Pieces together pre-recorded phonemes or words to form speech.	Uses statistical models (HMMs) to generate speech based on parameters like pitch and duration.	Uses Deep Learning (DL) to generate high-quality, human-like speech.	Sends text to cloud servers for processing using pre-trained AI models.
Quality	❌ Robotic, unnatural	🟡 More natural than Formant Synthesis, but limited	🟡 Better than concatenative, but still lacks naturalness	✅ Best (very realistic, human-like)	✅ Very realistic (close to Neural TTS)
Speed	✅ Very fast	🟡 Medium	✅ Fast	❌ Slow (requires GPU for real-time processing)	✅ Fast (real-time output)
Cost	✅ Free & lightweight	✅ Free	✅ Free	❌ Expensive (computationally intensive)	❌ Paid (cost varies per character/word)
Best For	Retro-style speech, embedded systems, offline apps.	IVRs (Interactive Voice Response), navigation systems.	Older TTS apps, embedded systems.	AI Assistants, Audiobooks, Virtual Avatars.	Commercial applications, customer support bots, TTS in apps.
Pros	✓ Small size, works offline, low CPU usage.	✓ More natural than formant synthesis.	✓ Small model size, flexible speech generation.	✓ Most natural-sounding, supports multiple voices & emotions.	✓ No setup required, supports multiple languages & voices.
Cons	✕ Sounds robotic, lacks emotional tone.	✕ Limited vocabulary, needs large databases.	✕ Still artificial-sounding, needs training data.	✕ Requires high computational power, large datasets.	✕ Requires internet, can be costly for high usage.

A comparative analysis on various parameters (1st Column) of TTS models available today.

This was my **first milestone** I achieved post me researching on the topic. A comparative study of all TTS models. Not only it helped me to judge where we stand now wrt SAM but also what **improvements** can be made when it comes to giving **TTS** a not so robotic voice. Post this I started exploring how each of them can be constructed via code.

My **next milestone** arrived when I understood the relation of **English letters ~ Phonemes ~ Formants and Spectogram**. Its crucial to understand spectogram when talking of phonemes and Formants as a clear differential is seen in those of Vowels and consonants



Spectograms of vowel "o" and consonant "s" .

The vowel formant F1 , F2 is clearly visible in the form of bands , The consonant formants do not have clear bands

Refer for code file to generate the above

Deliverables | Overview

I. Mid-term Evaluation

My first understanding shall be towards building on how **English letters** in a word can be **re-written in phonemes**. A clear **dictionary** shall be build . This shall lay the groundwork towards the upcoming TTS functioning.

The next milestone shall be conversion of **phoneme to audio** , ie extracting the formant parameters from individual phonemes. A more elaborate study of IPA chart and similar topics shall be done from my end

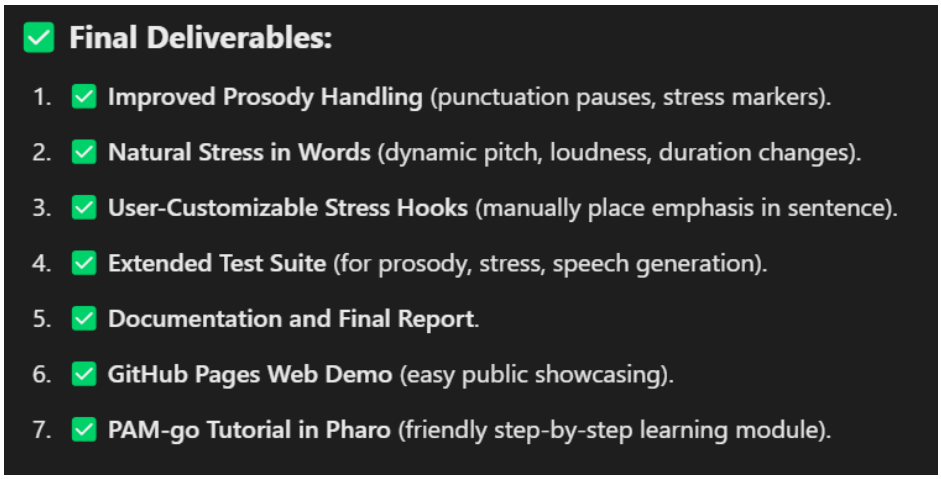
Post my Midterm Evaluation I desire to provide a complete functional **Formant based Synth TTS *(with a very Robotic Voice)*** which is able to process and generate simple keywords. My week-by-week (Week 1 to 7) gives a more detailed vision on this.

Mid-Term Deliverables:

1. **Basic Robotic TTS System** (Letter to Phoneme → Audio Synthesis).
2. **Audio Preset System** (Male, Female, Child).
3. **Simple Display UI** (Sliders + Dropdowns).
4. **Basic Test Suite** (Validation of phoneme mapping, UI elements).
5. **Demo**: Running prototype where user can input simple words and listen.

II. End-term Evaluation

As part of my Final Delivery for the Project **"PAM : TTS tool for Pharo".** I will be delivering a **Format based Syntesizer in Pharo.** Taking inspiration from SAM (which is a Javascript based Formant Synth. A lot of the functionality has been written in Faust Programming language and hence , it will serve as my Guidebook throughout the project. My majority focus shall be to create a complete functional Formant based TTS model. In my Future Plans on this project , I shall be love to extend my work to include Neural based Synthesizers which will use Audio datasets and better prosody genration function. *More on this in the Future Plans section !*



III. Blogging

I always love to document all my learnings that I have done. I try to make my blog **easy to read , quick to grasp** and **not overwhelming** so that new learners can pick it up easily. Some of my documentations are on Pharo Syntax and Phausto tutorial. Check out the linked Github repo's which contain a detailed Readme.md and contain easy to follow code.

For my duration of GSOC, I will be documenting my entire work on Medium. In my week wise plan , I have also mentioned I will making a **PAM go. Tutorial** similar to Profstef.

Deliverables | Week-by-Week Plan

I. Week 1

Starts June 2nd

Devlope a simple rule based converter. English letters to Phonemes.

- For Example **a c before { a , o , u } will be /k/.**
  Hence, cat → /k/ /æ/ /t/ , photo → /f/ /oʊ/ /t/ /oʊ/.
- Having phonemes for ending suffixes such as **-ing → /ɪŋ/ , -ed → /d/.**

II. Week 2

Starts June 9th

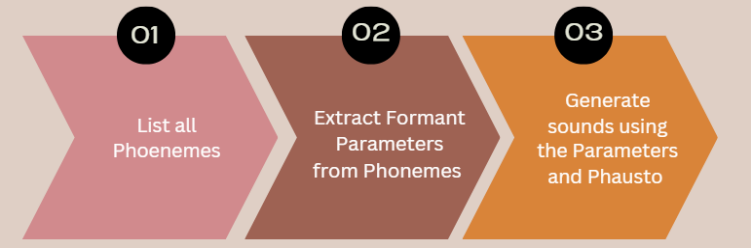
Making a detailed reciter with complex phonemes such as **"ph" → /f/ , "th" → /θ/,** and consonants

III. Week 3

Starts June 16th

Lets hear the sound ! 🔊

Build a DSP with the help of these phonemes. The below illustrates how this shall be done



IV. Week 4

Starts June 30th

Combine consonants and vowels sounds to generate TTS of small english words.
This generates our first ready prototype !! 😀

V. Week 5

Starts July 7th

Create **presets of audio settings** { Male , Female , Child } as in SAM.Hence, a very robotic version of PAM is generated. The Arguements which should support it should be initilazeMan , initilizeFemale , initializeChild.

VI. Week 6

Starts July 14th

Start working on the **DisplayUI of PAM**. The prototype should open in a new window having sliders for Pitch and Speed, a text input and buttons which will set the audio settings to match the predefined Male, Female , Child presets. Refer Page 1 (b)

VII. Week 7

Starts July 21st

Adding **Testcases** for all written code until now. This shall maintain good coding practices. Also a week to complete any pending tasks of previous week.

VIII. Week 8

Starts July 28th

Lets add basic Prosody ! 🎛️

Detection of puntuations and adding stress levels. Some examples are **{ ! , ? }** → **increased stress** towards the end of sentence. **{ . , - }** → **short pauses.**

IX. Week 9 — Starts Aug 4th

Detecting vowels and consonants and increasing / decreasing the **duration of added stress.**

X. Week 10 — Starts Aug 11th

Detecting natural stress in certain words. For example **{ im-POR-tant , bir-TH-DAY }** have more stress in capitalized words . These more stress syllables will have more amplitude , rising pitch and longer duration.

XI. Week 11 — Starts Aug 18th

Add a customisable hook that user can select where to add in the sentence. This gives the user an option to customise their audio by selecting places in senteces where stress needs to be added . On such places this hook can be added .
Refer Pg 1 (a) Method name : **input stressPoints at: 'Welcome' put: true.**

XII. Week 13 — Starts Aug 25th

Final Polishing , Documentation and adding TestCases for code coverage. Additionally make a **Webpage similar to SAM** which makes it easier to give demo on how PAM works . Have decided to host it on Github pages for now. The UI is similar to the UI referenced : Refer Page 1 (b)

XIII. Week 14 — Starts Sept 1st

Adding a **PAM go. Tutorial.** In Pharo**.** Just like Profstef. 👏

Future Plans with TTS

In the current Implementation it can be visible that the **English Letter to Phenome** conversion is Formant , ie Rule based. This as said before, shall make a very **Robotic sound** *c'mon no one likes this ... we need AI chatbots speaking , flirting , consoling us in the most human like voice ... we are secretly wishing for a AI apocalypse ... yipee !!!* 😂

Hence, once a basic Formant based synth is made I plan on **extending this to a TTS** which adds **more prosody**. My TTS comparative table talks about how **HMM** based , **Neural Network** based models can help us achieve that.

Pharo-AI is a comprehensive Github repo which touch bases on Neural Networks usage in Pharo. I believe **Audio datamodels** can be easily fitted in and a in house Neural Network can be created out of it.

As an extension, **Vocoders** can be created and integrated as well for better prosody.

One of my initial ideas to add prosody was adding a **Sentiment Analyser**. So for a Happy Sentence like ; *I am selected for GSOC !!* **vs** a Sad Sentence ; *I missed the Proposal deadline submission* ; the prosody can be adjusted as per the sentiment it matches.

Why I will be the best fit for this Project and Benefits to the Community

A little backstory ....

I am currently in my Fourth year of Bachelors Degree in Computer Engineering. Ever since my Second year I have been coding in OOPs. Java was my first , Javascript , Python followed. Hence, post understanding the initial syntax of Pharo; it was **easy for me to grasp** a lot of terms such as Class side Variables, instance methods etc.

Before I came across this project; I had absolute no knowledge on Computer Music World. But post receiving initial insights from Domenico and Nahuel, my knowledge took a 180 degree turn. My repos on TTS Tutorial, Music ABC's are a testament to this.

I feel like a DJ creating Computer Music 😂

Not many people I presume know about the awesomeness Pharo provides with its minimal syntax and **"Everything is Object"** approach. Hence, what Pharo needs right now is a differentiating factor.

- **Phausto** is one ; it is trying to mitigate the difficulties one faces when developing audio plug- ins.

- **PAM** can be the next. As pointed out previously , Modern TTS with best prosody capabilities come with a lot of additional cost. PAM on the other hand can make it possible to **reduce costs** by using in-built neural networks mentioned in Pharo-AI.

Hence, with all my research and hands on coding I have done in a very limited time, and my desire to give something better to the community ; I believe I have the best of capabilities for this project.

About me

I. Education , Experience , Achievements

- Doing my Bachelors in Computer Engineering from Sardar Patel Institute of Technology. Currently enrolled in my Final year. My research paper publishing is the only thing that needs to complete my Bachelors degree.
- Demonstrated **strong team-work , resilience and coding skills** by winning the Code for Good Hackathon organised by J.P. Morgan Chase.
- Have successfully completed Minor Course in Management from the esteemed SPJIMR. (Selected amongst a batch of 460 students)
- Demonstrated **strong team-work , ownership and leadership skills** by being a member of Placement Team for our batch
- Currently working to meet my family errands and am a Backend developer. I will be devoting around 3 hours daily to the project and will do it 10 hours on weekend. Hence, my weekly devotion to this project shall be a minimum of 25 hours.
- I agree this is less than stipulated but my Inital work done in a minimum time proves that my 3 hours of work ~ a full day work of many. I also do not mind in extending the time frame of the project. I believe in strong ownership qualities and if I get selected for the project , I shall complete all my deliverables as mentioned above

II. Previous Development with Pharo , Phausto , Open Soure

- Checkout my Github Repos : Pharo.
  - Everything I have learned in using the MOOC and Pharo by Example book is documented in Readme.md here.
  - Because i wanted to make this as a **easy to follow** along tutorial ; I have added screenshots in Readme.md along with code snippets.
  - Have made a sample WeatherAPI project which is listed in the same repo
- Checkout my Github Repos : Phausto
  - Here's where i have added my **blog** on the tutorials in MasterLu.
  - Heres a PR in Phausto I had initial thoughts about

- Some other PRs in Open Source :
    - GliderPY
  - Issues Raised
    - BoBOT

III. Previous Work on Music Synthesis and TTS

- The entire repo Music ABCs gives a thorough understanding o**f Sound Synthesis.** The Readme.md file provides a detailed understading on **Unit Genrators , LFOs , ASDRs , Envelopes** and links to some of the resources one can learn them from
- The entire repo Learn TTS with me gives not only a theoretical explanation of early Synth but also provides **code files** to get a glimpse of how it works
  - *PS. audio files are added as well ... its so easy to follow along* 🙂

IV. Contact Details

Name	Ms. Neerja Doshi
Email Address	nirjadoshi2003@gmail.com
College	Sardar Patel Institue of Technology, Mumbai
Degree Program	Bachelors in Computer Engineering
Time Zone	GMT + 5:30
Github URL	Github !
LinkedIn URL	LinkedIn !

The page that describes me , *tune in & enjoy* 😇

Made with Research, Facts , Curiosity and a pinch of Confidence that this shall get me through my First GSOC experience 😀

Thank you for your time to consider the proposal.