

Project Name: The Funk Forge Synth
Name: Marvin Leon
Email: marvleon@pdx.edu
Github: github.com/marvleon/FunkForgeSynth

Vision	1
Application Overview	1
Core Components	2
Graphical User Interface (GUI)	2
Application Interface	2
Synthesizer Logic	2
User Interaction Flow	2
Issues of Concern	3
Sound Complexity	3
Performance	3
Flexibility	3

Vision

The vision of this project is to make sound synthesis accessible to users who have never used a synthesizer. By providing a tool that combines simple design with sound manipulation features, the Funk Forge Synth will encourage creativity and experimentation. I want this application to help the user with sound creation without the need for deep technical knowledge. I'm inspired by the synth sounds of Parliament - Mothership Connection and Ohio Players - Funky Worm.

Application Overview

This project aims to create a usable sound synthesizer application using Python, providing users with a user-friendly interface to manipulate sound parameters and generate custom audio output. The Funk Forge Synth will enable users to create and manipulate sounds through a GUI. This synthesizer features the capability to adjust sound parameters such as waveforms, ADSR (Attack, Decay, Sustain, Release) envelope, frequency, reverb, and cutoff, through sliders. Additionally, there will be some frequency modulation synthesis to aid in unique sound creation.

Core Components

Graphical User Interface (GUI)

- **Functionality:** Offers a platform for users to interact with the synthesizer, adjusting sound parameters through sliders and buttons.
- **Components:**
 - Oscillator sliders for waveform selection (sine, square, triangle, sawtooth)
 - ADSR envelope sliders to manipulate the sound's attack, decay, sustain, and release phases
 - Frequency slider to control the pitch of the sound
 - Sliders for reverb and cutoff to adjust the sound's effects and filtering

Application Interface

- **Role:** Acts as an intermediary, storing user adjustments from the GUI as input parameters.
- **Functionality:** Connects the GUI and the Synthesizer Logic, ensuring that user settings are accurately transmitted for sound synthesis.

Synthesizer Logic

- **Role:** The core engine that interprets user input from the Application Interface to generate sound
- **Functionality:**
 - Reads stored settings from the Application Interface
 - Applies digital signal processing techniques to synthesize sound based on user-defined parameters
 - Outputs the sound when the user clicks play, based on the adjustments made through the GUI

User Interaction Flow

Initialization: The user launches the application, presented with the GUI containing all adjustable sliders and controls.

Sound Design:

- The user adjusts the oscillator sliders to choose the base waveforms for the sound.

- The ADSR envelope, frequency, reverb, and cutoff settings are modified to shape the sound further.
- All adjustments are continuously saved to the Application Interface.

Playback:

- Upon clicking the play button, the Synthesizer Logic reads the user-defined settings from the Application Interface.
- The application synthesizes the sound based on these parameters and plays back the result to the user.

Issues of Concern

Sound Complexity

Right now there is limited waveform variety. The synthesizer has basic waveforms (sine, square, triangle, sawtooth) but that can limit the range of sounds that it can produce. Additionally, there are limited modulation and shaping options. This is something I'm not super familiar with but I want to explore frequency modulation.

Performance

I don't have an expectation for optimizing my application for performance. I'm more concerned with functionality and making sure that sounds can be made and manipulated. I want this synthesizer to eventually work in real time to encourage sound experimentation so performance should be prioritized. I am also deprioritizing visual feedback (representing waveforms, envelopes, etc.) for the time being but this would be useful for sound creation.

Flexibility

There is no support for external plugins or sound libraries. I would like to introduce a feature that allows the user to upload their own .wav file so they can manipulate it. There is also no support for MIDI or VST plugins so this application right now is very self-contained and limited.