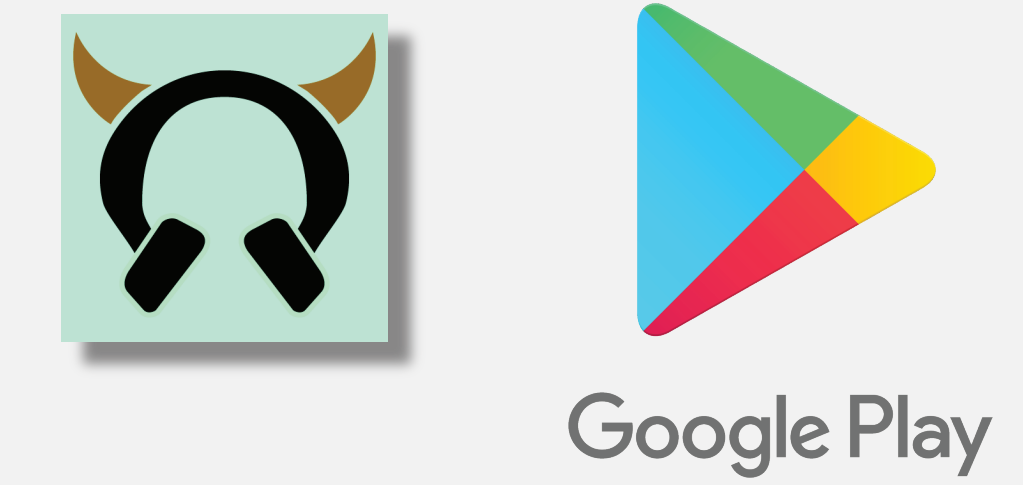


the Roleplaying Audio Mixer

a mobile app
for Android devices



A dark dungeon, a buzzing marketplace, a dense jungle... Whatever the place may be, create an audio atmosphere for it with the Roleplaying Audio Mixer, a mobile application for Android devices.

Introduction

I made the first version of this app in the class “Mobile Application Development” in 2017. The app gives the user a simple interface with 4 channels that could each play an audio file. The user can play these sounds synchronously to create an audio ambiance. The first version of the app is too simple, making it more of a toy than a tool that dungeon masters of roleplaying games might use. To further bring this app to life, I had several ideas for improvements:

- saving and loading mixes
- add/remove audio tracks
- categorized sound list

feature

1. Save & Load Mixes

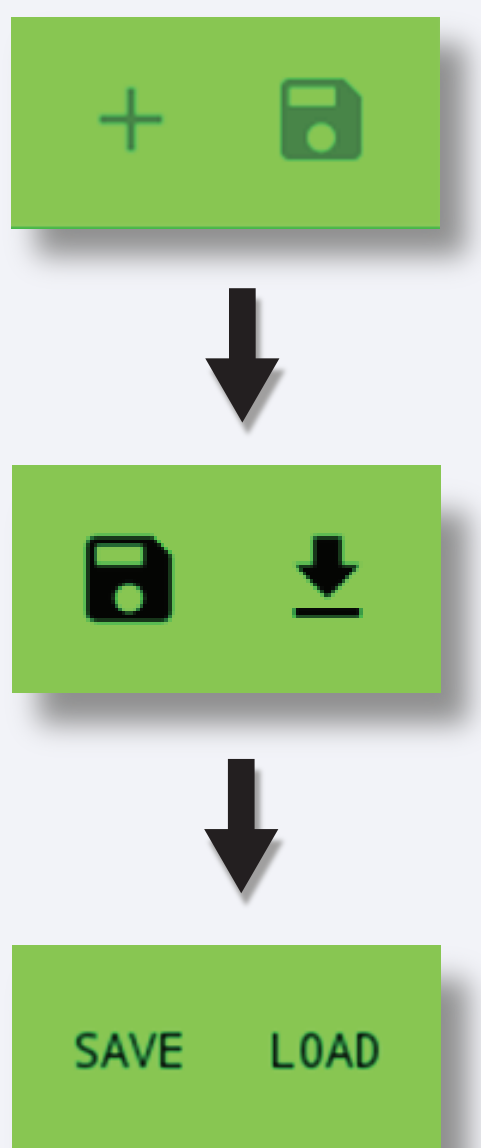
The first feature that I worked on was being able to save and load a “mix”. A mix is just a collection of sounds and volumes representing an ambiance. While the implementation of this feature was fairly straightforward, I first had to setup a database to store the data in.

Database

Having never setup an intuitive functioning database before this was challenging for me. To make the database, I used a class called Room that Android developers have access to. Room is a layer of abstraction on the SQLite database. Setting up Room took several days to complete, but once it was running, it was very easy to store and retrieve data.

Design Process

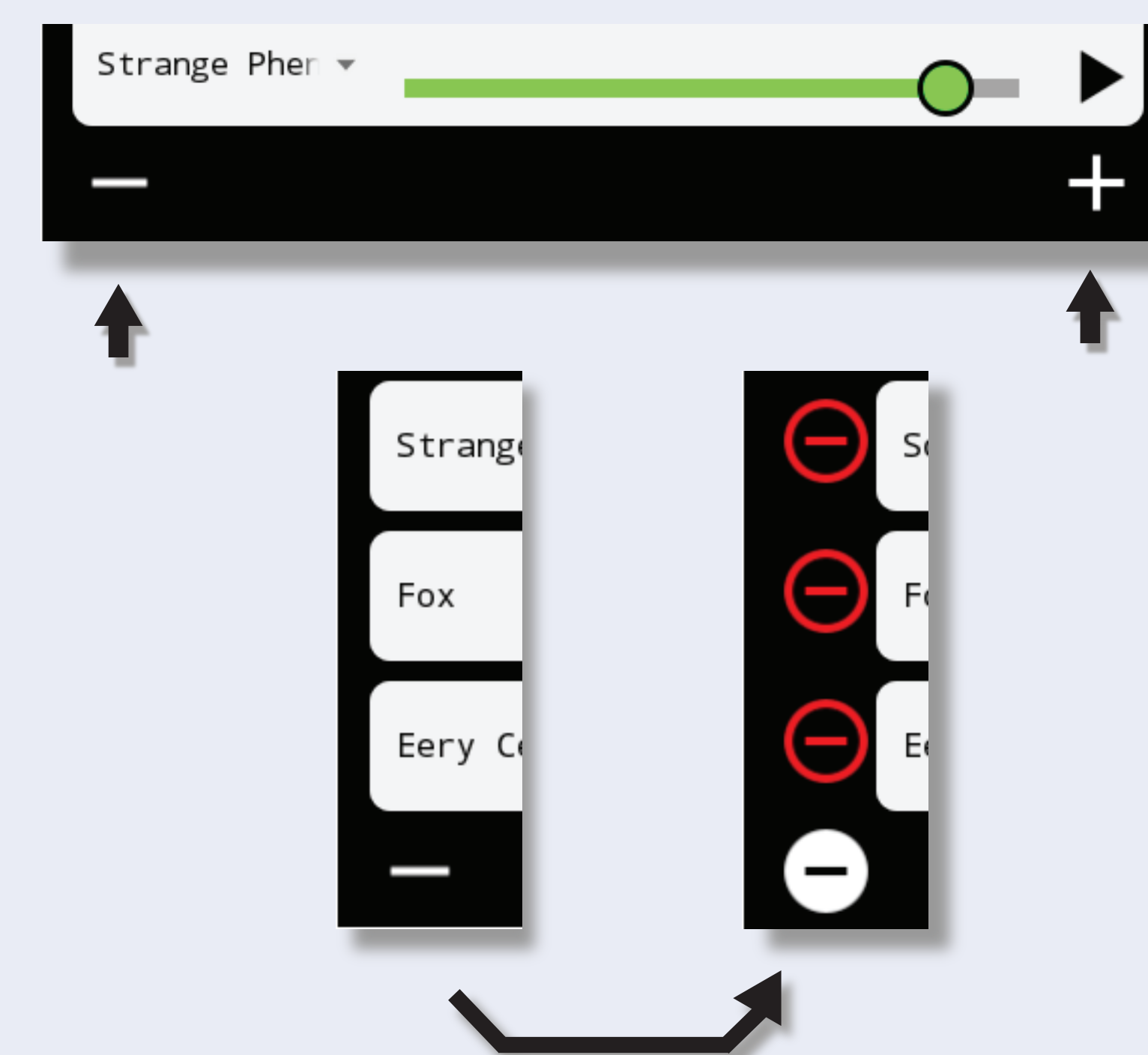
Through several user tests, the saving and loading buttons changed until I felt they properly communicated their function. Words ended up being the most intuitive.



feature

2. Add/Remove Tracks

The second feature that I implemented was being able to delete or add a new audio track. My initial release of the app gave the user only 4 tracks that they could play audio from at a time. With this feature, they can have as many sounds playing at once as needed.



clicking ‘-’ opens up a panel for deleting tracks

Better Code

My current code was not suited for this feature. I had each audio track hard-coded. I rewrote many parts of the code by making classes that were specific in their function. I made sure to follow the rule of “separation of concerns” or keeping classes confined in their functions. It was good to see that I have improved my programming skills since I first made this application.

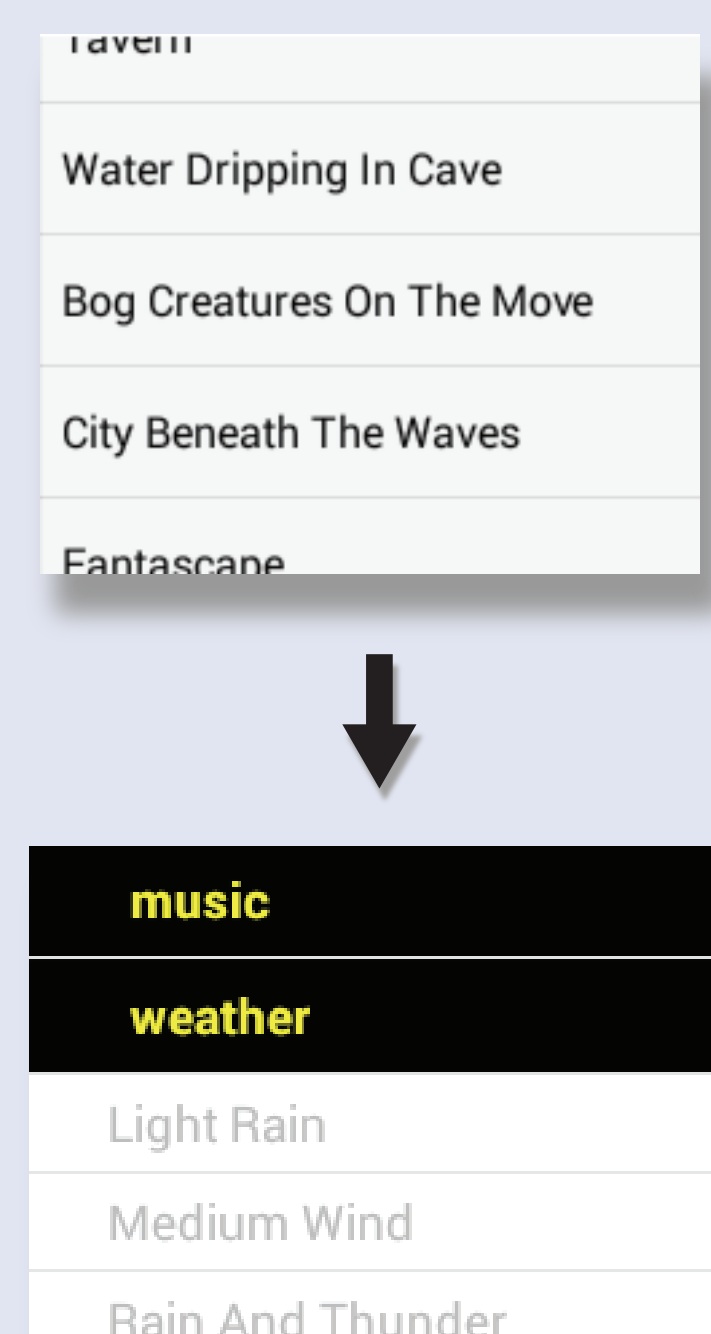
User Testing

User testing was not an afterthought for my project. From feature 1 through the end, I was often asking people to try out my app. Sometimes I merely gave them an overview of the app then handed it to them and observed. Other times I gave a specific task to complete, such as “create an audio atmosphere of a scary forest”. These tests helped me to craft a product that is more intuitive, useful, and pretty than it would have been.

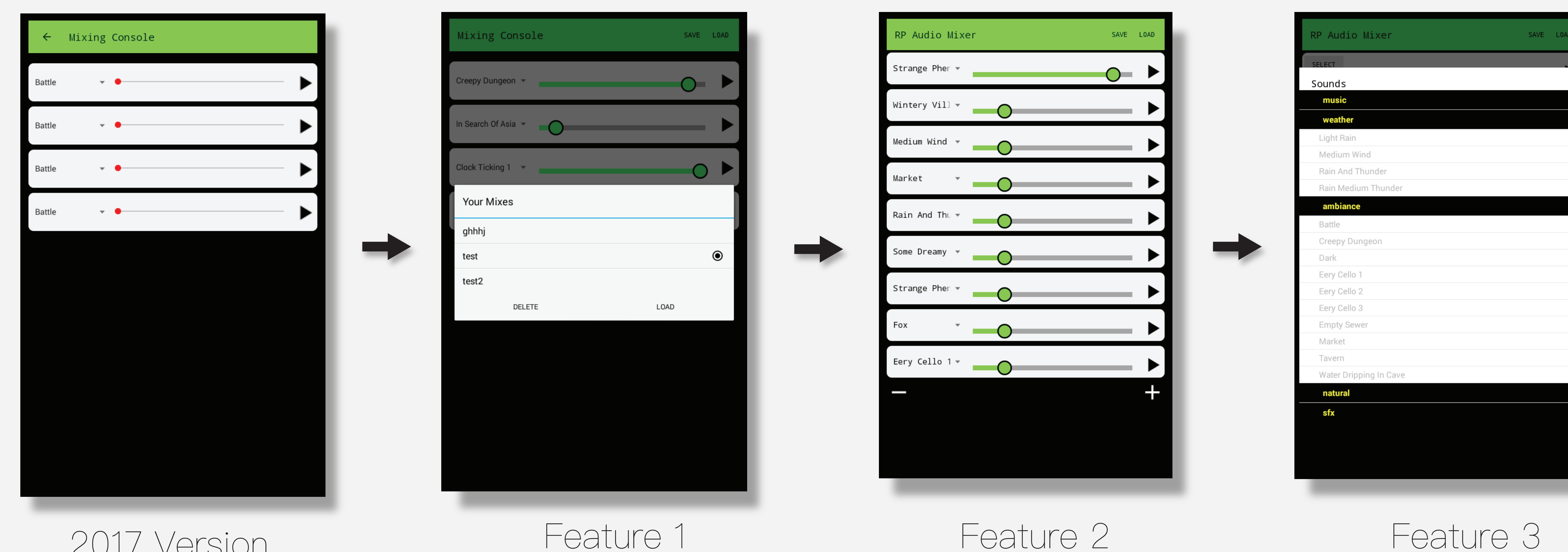
feature

3. Sound Categorization

When selecting a sound to play, the user is presented with a long list of all the available sounds. This is an annoyance as it may take a long time to find the sound they want. To solve this, I updated the code to display sounds within categories.



Iterations



Future Development

This project has been going on for almost 2 years and has gained a small but active user base of around 200 people. I plan to continue development indefinitely. I want to create a better experience for my current users and entice others to use this tool. My future goals are many, but include:

- Add a master control track
- deepen the sound library
- visual improvements
- port to iOS

developed by
Micah Stewart

To the following people, thanks for the help and love!

Alec Ellsworth, Gabe Helmuth, Dong Hoon Ha, Jon Denning, Dannie Stanley, Jonathan Geisler, Zach from the Bridge, Mike Kotlarski, William Slauson, Tim Lehrian, Maki Ohashi, Shawn Stewart, Lisa Stewart, Jillian Stewart and Thad Stewart

Taylor University
CSE Department