AIST2010 Project Report

Name: XING Jinbo SID: 1155091876

Abstract

This project is to build an application on the iOS platform for sound synthesis and visualization. The name of this app is "SoundThings",  whose basic framework for sound is AudioKit and user interface adopts the latest SwiftUI released in Apple Worldwide Developers Conference (WWDC) 2019. There are 9 compelling functions in SoundThings, including Draw the Sound, See the Sound, etc. It can help users to explore the interest of sound synthesis and find the beauty of sound.

1. Background

1.1. Correlation

In this course, we learned a lot about computer music, including basic knowledge of sound, analysis, synthesis, retrieval, etc. I like sound synthesis very much and apply what I have learned during the lecture and tutorials on this project. For sound synthesis, this app contains additive synthesis, basic oscillators, and so on. Along with the synthesis, there are sound effects and sound visualization in some functions.

1.2. Inspiration and Related Works

I extremely liked music games before and want to develop an application for music. But it's hard for me to build a novel music game and I find it's possible to build an application to play with sound based on knowledge from AIST2010 lecture. How about drawing to synthesize the sound with interesting human-computer interaction? It's also meaningful to turn the theoretical knowledge into practice on the iOS platform. There exist several applications in the App Store with similar ideas. I take two of them as examples here.

The first one is called MobMuPlat[1], which has been introduced during our lecture. It is a platform, built on libpd[2,3], for running Pure Data (Pd) patches on mobile devices[4]. Instead of using text coding, it has a convenient graphical user interface to help users handle the sound. I have tried MobMuPlat on the iOS platform; besides, it also has other versions on Android, OSX, and cross-platform (Java Swing). It has some interesting built-in functions, such as Touch

Arpeggiator and Networking Pad, which can synthesize sounds in some novel ways. What's more, it can be used with the combination of Pd on the laptop for more complex patches you choose.

The second one is called Kazu. Users can make music with motion with this application [5]. It uses the camera on the mobile phone to capture the motion of users and applies the features of motion on the music parameters. That means users can conjure music out of the thin air such as playing air instruments. It's very creative and compelling for lovers of music.

2. Methodologies

2.1. Development

The programming language used in this project is Swift 5.1, which was released in September this year. The development tool/software used for developing the iOS app is Xcode 11.2.1 and the version of aiming application is iOS 13. Besides, the basic framework for sound is AudioKit[6], which is a powerful and leading open-source audio framework on iOS, macOS, tvOS, and watchOS platforms. It has amounts of built-in functions for audio synthesis, processing, and analysis. And SwiftUI is adopted for the framework of user interface, which was released in WWDC 2019 and for declarative UI structure design across all Apple platforms.

2.2. System Structure

The basic system structure of this project is Model-View-Controller (MVC), which separates an application into three main components: the model, the view, and the controller (Fig. 1).

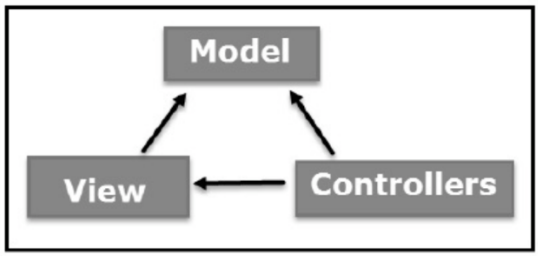

Figure. 1 MVC architectural pattern

The overall system structure can also refer to file structure (Fig. 3). Each function is based on a simple MVC pattern. And the model is directly related to AudioKit functions. Each model belongs to the ObservableObject class and it should be added into SceneDelegate as an environmentObject. Besides, AudioKit sound models should be initialized and set in each

DataModel file and the basic process can be shown in Fig. 2. Views and controllers will be further introduced in section 2.4.

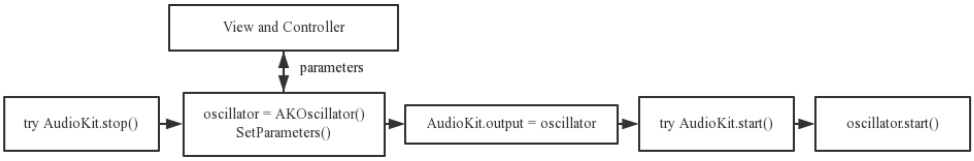

Figure. 2 Process on using AudioKit in DataModel

2.3. File Structure

There are 42 code files in this project, including some necessary files such as AppDelegate.swift, SceneDelegate.swift, LaunchScreen.storyboard, Info.plist, and other functions .swift files. Fig. 3 shows the overall file structure in this project.

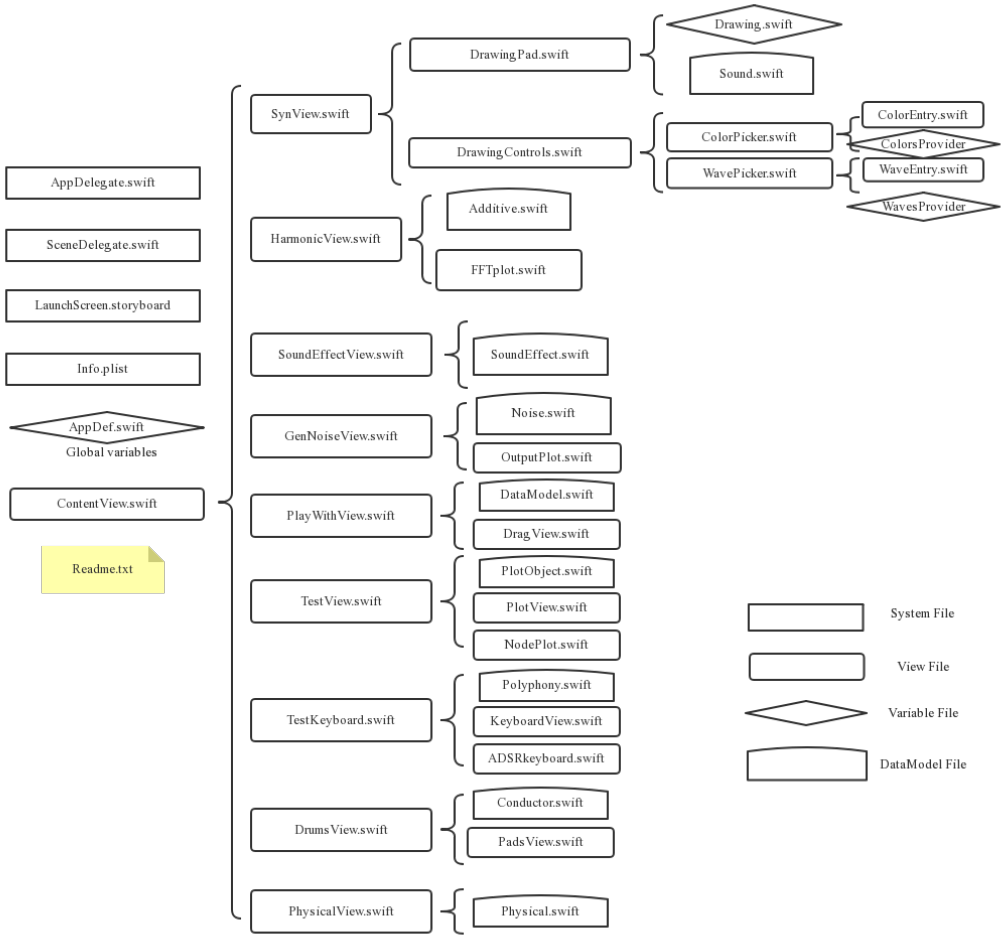

Figure. 3 Overall file structure

2.4. User Interface

SwiftUI is a new framework for UI design on the Apple software platform. I combine the controller and view into a single view .swift file and let those parameters of sound models can

be directly modified by users through the UI. The example can be shown in Fig. 4.

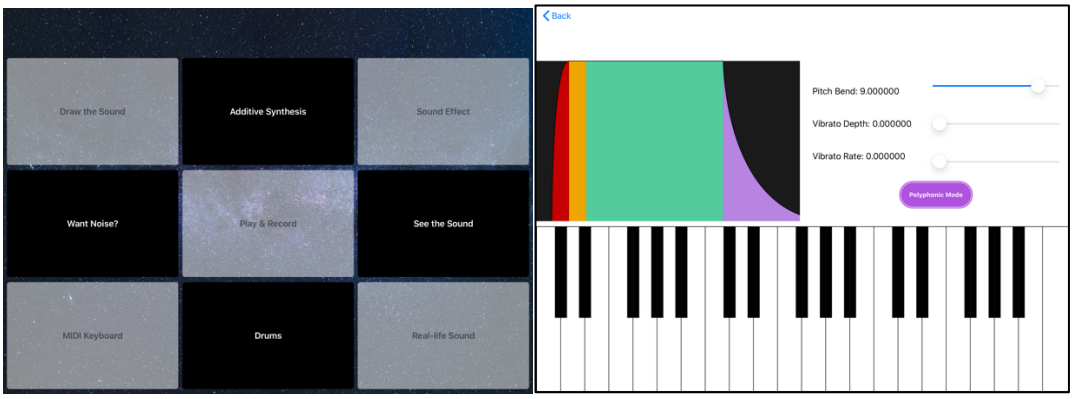

Figure. 4 UI of the main menu (left) and MIDI Keyboard (right)

2.5. Functions Implemented

I. Draw the Sound

Users can synthesize sounds by drawing on the pad with the upper part corresponding to the lower frequency and the lower part corresponding to the higher frequency. The oscillator in this function is FMOscillator, so the carrier multiplier, modulation multiplier, and modulation index can be modified by slider below the pad. Besides, users can change the width and color of pen and waveform of oscillators to sine, triangle, square, and sawtooth.

II. Additive Synthesis

Users can synthesize sounds by the 10 harmonics. The basic frequency can be modified through the text input and amplitude of them can be changed by slider. The upper right part is FFT plot, which can show the components of sound frequencies intuitively.

III. Sound Effect

Wah wah and equalizer filter sound effect can be applied on the music through modifying the value of wah wah and filter by sliders. Music picker is used to choose the music users like. There is a sound tracker that can display the frequency and amplitude of this music in the upper part.

IV. Want Noise

Users can synthesize the noise by mixing brownian, pink, and white noise. The amplitude of these components can be changed by the sliders. There is an output waveform plot in the upper part of this view.

V. Play & Record

It can record the operation of users and play them like a sequencer.

VI. See the Sound

It can plot waveform and node based on different sound processing operations. The left upper part is sound sampling with bit depth and sampling rate as sliders. The right upper part is delay sound effect and users can change time and feedback of this effect.

VII. MIDI Keyboard

Users can use its MIDI Keyboard to generate music. Besides, the ADSR can be changed through touching and dragging the ADSR board. Users can choose to play using polyphonic mode or monophonic mode of keyboard. Three parameters of this keyboard can be changed, including pitch bend, vibrato depth, and vibrato rate.

VIII. Drums

By using the drums pad, users can finish the drum solo. It contains bass drum, closed/open hi-hat, high-tom, low-tom, and mid-tom.

IX. Real-life Sound

Here, I try to synthesize the sounds in real life by models and functions provided by AudioKit, including sound of water drip, flute, vocal tract, and tubular bells.

3. Conclusion

There are 3037 lines of code in this application and I finished it in almost one month, including learning development on iOS platform, learning the usage of AudioKit, design, developing, debugging, and testing. And this project/application contains 9 main functions, including Draw the Sound, Additive Synthesis, Sound Effect, Want Noise, Play & Record, See the Sound, MIDI Keyboard, Drums, and Real-life Sound. Some functions are about sound synthesis, some are analysis and visualization. To summarise, I have implemented all the functions in my plan and quality of these functions is not bad.

However, there are still some drawbacks of this project. For the code part, it should have good enough comments and clear structure. For the UI part, some views are complex and not easy to control, and their style is too simple. The stability of this app is not bad, and the app is never crashed during the test.

The further direction I have explored is sound **synthesis** and **visualization** with the help of AudioKit. In this process, I also try to apply some sound effects I like. I try to synthesize sounds in some new ways and with interesting interaction, for example shaking the device to left or right to modify the stereo channel and drawing to generate sound.

There are a lot of powerful functions in AudioKit package, which helps me do sound synthesis, visualization, and analysis. I read the document once and know some other things in sound synthesis and effects, such as different built-in oscillators and Dynamics Processing. I try to implement part of them and experience a lot. I learned a lot from the process of developing this application, including the knowledge of sound synthesis and development of this iOS application.

It's hard for me to build a good enough app on iOS platform which is totally new to me. It's harder than those software and languages we learned during lectures and tutorials, which focus on sound stuff and are professional. And I find the result is not good enough when synthesizing real-life sounds using AudioKit functions, I may try to improve AudioKit functions in this part and upload them to their repository on github in the future.

4. References

[1] "MobMuPlat - Mobile Music Platform". Accessed on: Dec. 14, 2019. [Online]. Available: www.mobmuplat.com

[2] "Libpd". Accessed on: Dec. 14, 2019. [Online]. Available: www.libpd.cc

[3] Brinkmann, Peter, et al. "Embedding Pure Data with libpd." Proceedings of the Pure Data Convention. 2011.

[4] Iglesia et al. "The mobility is the message: The development and uses of MobMuPlat." in Pure Data Conference (PdCon16). New York. 2016.

[5] "Kazu Music From Motion and Real-Time Effects". Accessed on: Dec. 14, 2019. [Online]. Available: https://www.kazu.rocks

[6] "||AudioKit - Powerful audio synthesis, processing, and analysis, without the steep learning curve". Accessed on: Dec. 14, 2019. [Online]. Available: https://audiokit.io