DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

ARCHITECTURAL DESIGN SPECIFICATIONS
CSE 4316: SENIOR DESIGN I
FALL 2023

WORKING IN PROGRESS
THEREMELO

SCOTT FRAZIER
RON NGUYEN
FARHAN SADIQ
SAZZADUL ISLAM
CALEB MYATT

Revision History

Revision	Date	Author(s)	Description
0.1	10.25.2023	CM	document creation
0.2	11.01.2023	RN	additions
0.3	11.02.2023	SI	draft editing
0.4	11.03.2023	FS, SF, RN, SI	completed draft
0.5	11.04.2023	RN	revision

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

The ThereMelo application will take the user's hands as input to convert into a sound. This will allow the user to play it as an instrument, based on the theremin. Our application is designed to be simple to use, and easy to learn since we intend to make the application available for the public to use. This section will outline the ThereMelo and its' layers that build the application. This application utilizes a computer, an IR camera, and the user's hands. As input, we will utilize the user's hand positions relative to the IR camera. The position of the user's dominant hand will be the application's cursor, allowing the user to interact with our Graphical User Interface (GUI). The GUI will allow the user to tweak a few options, like the preferred dominant hand, volume slider, exit, and lessons/playground environments. The user's computer will be displaying the environment on their screen, as well as outputting sound for the user.

2 System Overview

The ThereMelo is currently made out of 4 layers, each has its interactions with the other. Those layers are called "External", "Application", "Data", and "User Interface". Our External are the devices that will take input information from the user and project the results as well. The Application Layer will process all of the information given by the user, to generate a result to output into both External and User Interface. The User Interface will be able to provide extra options for the user to choose from, and the data layer will store the necessary information.

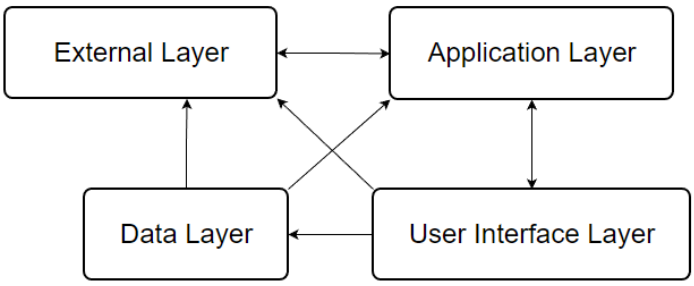

Figure 1: ThereMelo's Architectural Layers

2.1 External Layer Description

The external layer consists of external devices. The user's hands, the IR Camera, and the system the application is running on are found all in this layer. This layer takes all of the inputs the user gives it and sends that information over to the Application Layers. All inputs include the user's hand movements and gestures. This layer also displays results as output, requiring the user's computer to handle most of the work. Outputs include all sound transcripts from hand movements and visuals for the computer screen.

2.2 Application Layer Description

Our Application Layer is the main layer of the ThereMelo. Using the input information given in the External Layer, the application should be able to recognize a set of hands by utilizing Gemini and be able to calculate the hand magnitude from the virtual antenna, as well as the user's hand gestures. From the position of the user's dominant hand, they can control the Pitch. The volume will be controlled by the position of the user's non-dominant hand. Both of this information will be sent to our sound engine, FMod, to produce the user's desired sound wave and send it off to their speakers.

2.3 User Interface Description

The User Interface will handle options the user may want to set to their desires. This will include the preferred hand, which will determine how our application handles the sound, the scene, and the overall audio output. Scenes will handle how the player can interact with our virtual theremin. A sandbox environment will allow the user to play the instrument without many restrictions. Lessons scenes will allow the user to learn how to use and play the theremin.

2.4 Data Layer Description

The Data layer will keep track of important data that we will be using for the progress and options for the player. We will mostly call to check what is the dominant hand and keep an updated track of which lessons have been completed.

3 Subsystem Definitions & Data Flow

Here is the ThereMelo's system overview, which will include the details of each layers.

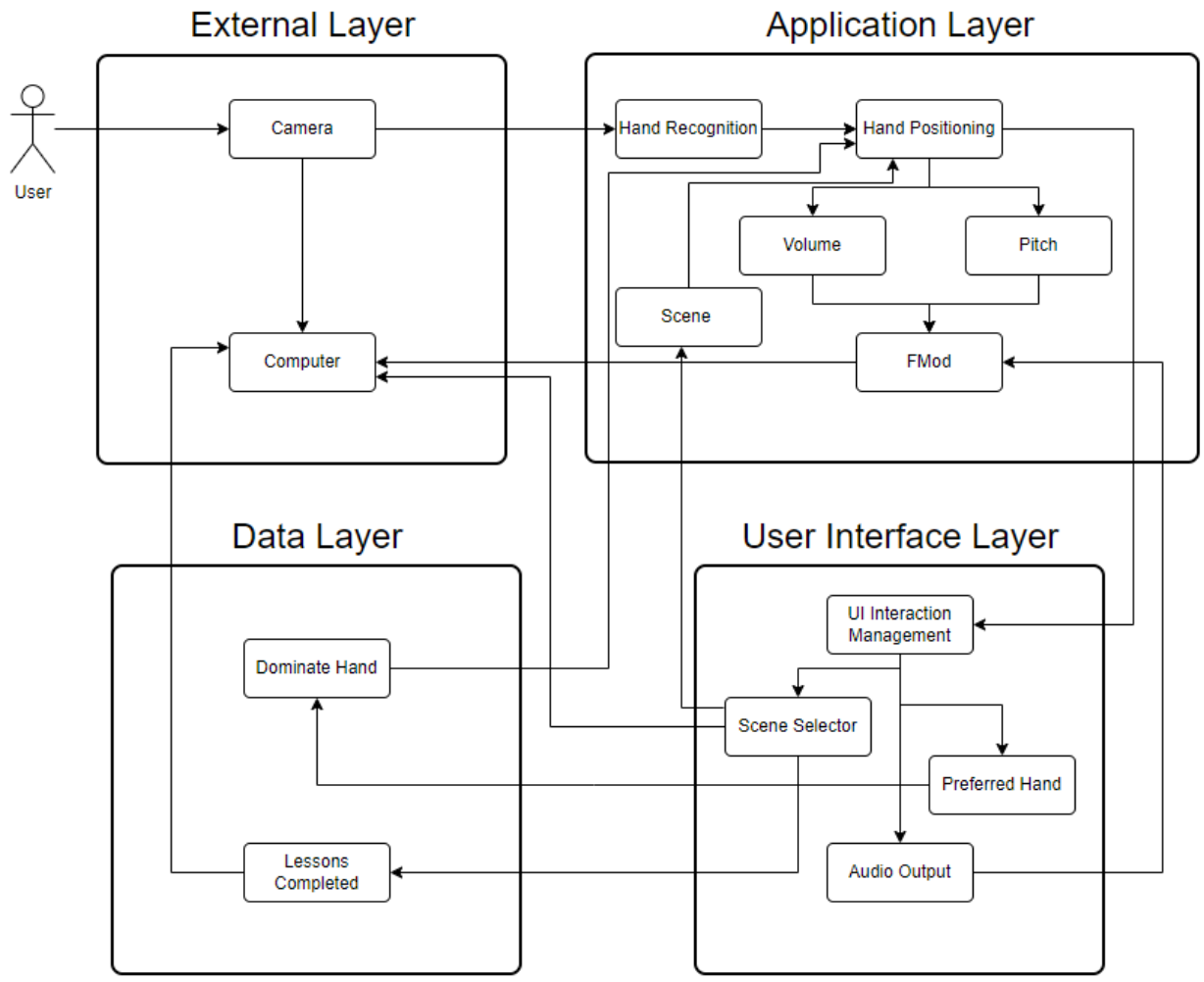

Figure 2: Data Flow for our ThereMelo application.

4 External Layer Subsystems

Once again, the external layer handles the user's inputs and our application's output. Its main goal is to be the input and output of our entire system

4.1 Camera

This subsystem is IR Camera, allowing our application to "see" the real world and allows the hand recognizer to do its job. The camera should be the Leap Motion Controller 2, and utilize the Gemini Hand Tracker. The Camera subsystem should communicate with the computer, allowing it to send data over the system, as well as data to the Application Layer.

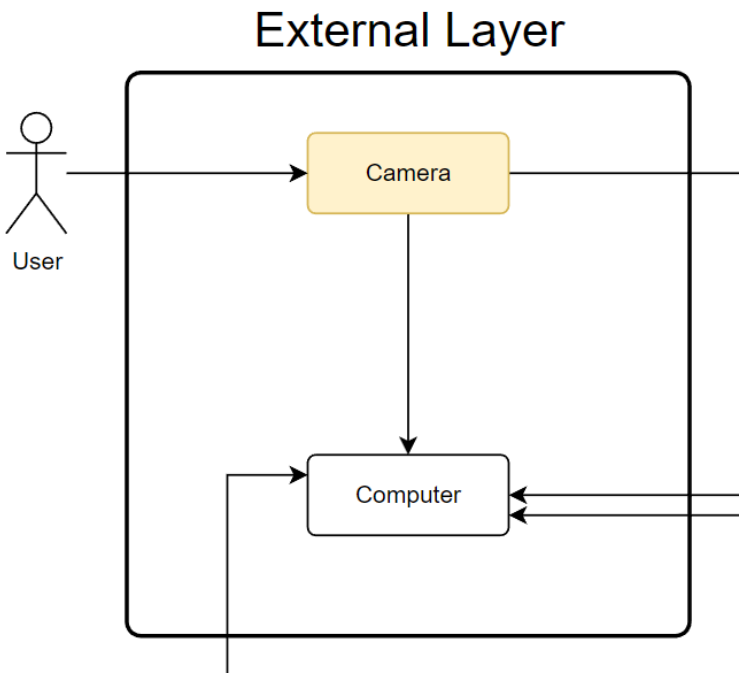

Figure 3: Camera subsystem diagram

4.1.1 Assumptions

It's assumed that the current user has two hands, both with five fingers, and can maneuver them. It is also assumed they are the only users, as the camera cannot handle more than 2 hands at a time. All required installations should be already installed as well.

4.1.2 Responsibilities

What this subsystem does is constantly monitor for any hands within its view. This subsystem will NOT do any recognizing, only sending critical information to the hand recognizer.

4.1.3 Subsystem Interfaces

Table 2: Camera interfaces

ID	Description	Inputs	Outputs
#1	IR Camera	User Hands	IR Data

4.2 Computer

The computer is the user's preferred device to run the ThereMelo on. It is crucial to have a computer that follows our system requirements. The computer will run our application, and utilize the camera for information sending/receiving.

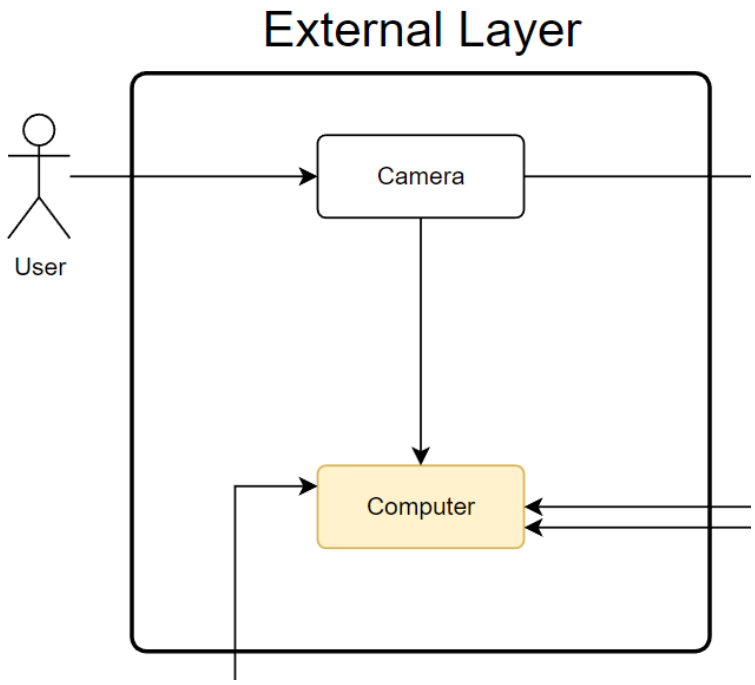

Figure 4: Computer subsystem diagram

4.2.1 Assumptions

The computer needs to meet system requirements and has all the necessary packages installed, and any other software needed. We will assume the computer has a screen to display visuals, and a speaker to output audio.

4.2.2 Responsibilities

This subsystem is responsible for running our application, and allows the camera to send information over to the application. The computer also handles the outputs, from any visual changes to audio outputs to the speakers.

4.2.3 SUBSYSTEM INTERFACES

Table 3: Computer interfaces

ID	Description	Inputs	Outputs
#1	Screen	N/A	Visual Display
#2	Speakers	N/A	Audio from App

5 Application Layer Subsystems

The application layer will manage the information from our external layer to process. The information goes through the hand tracker, which determines which hands are on the left or the right. Then the information is sent to our hand positioning, which calculates the magnitude of both hands and the positioning. Those will be used for our UI layer, as well as controlling the volume and pitch of our sound engine.

5.1 Hand Recognition

Hand Recognition is a separate application installed on the user's device. Gemini is Ultraleap's hand tracker, used by the Camera to determine the hands' orientation and to send positioning data over to our hand positioning subsystem.

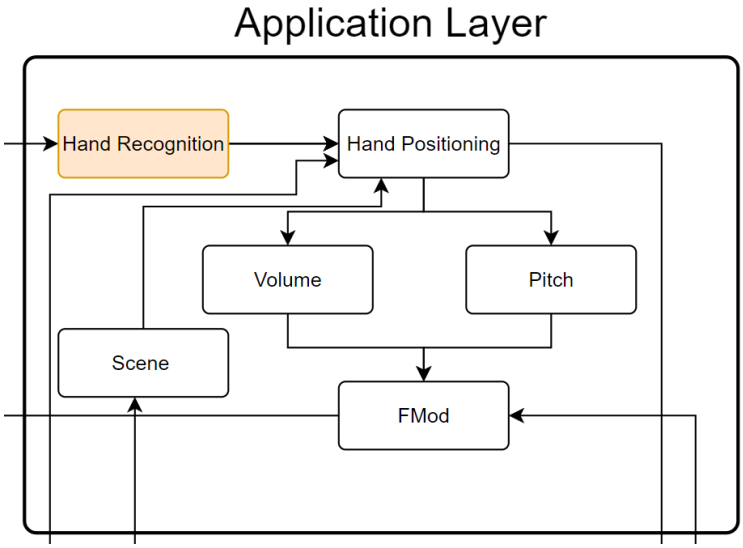

Figure 5: Hand Recognition subsystem diagram

5.1.1 Assumptions

The Gemini Hand Tracker is fully installed, and up to date. The tracker also recognizes the Leap Motion Controller, and the camera, and can "see" and recognize the user's hands. We assume there should be 1 pair of hands, as the tracker cannot handle more than one pair of hands.

5.1.2 Responsibilities

Hand recognition determines which hand is the left, and which hand is the right. It will send orientation data and positional data to our hand positioning subsystem to calculate the position and visualize virtual hands in our application.

5.1.3 Subsystem Interfaces

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Gemini Software	IR Data	Orientation Positions Hand Labels

5.2 Hand Positioning

Our hand positioning will calculate the given data to position the hands in the virtual environment, as well as get the magnitude between the virtual antenna with the user's dominant hand for the pitch. The magnitude of the user's non-dominant hand will be calculated as well to determine the instrument's volume. Our hand positioning will also take account of the user's hand gesturing.

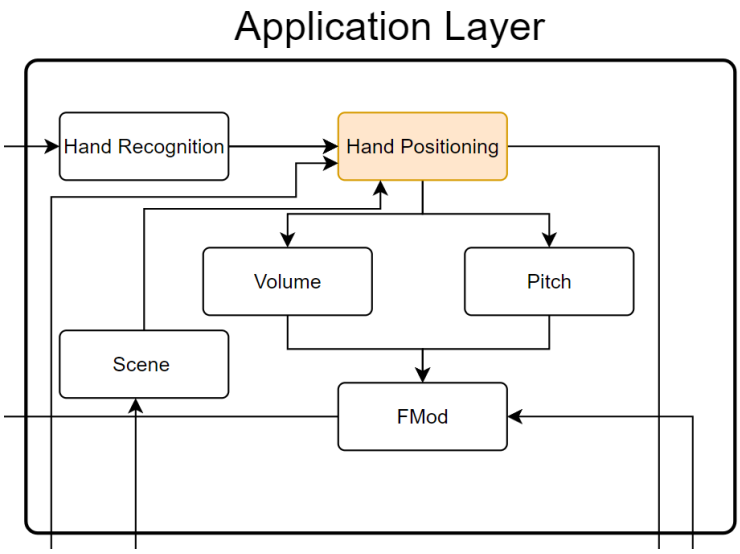

Figure 6: Hand Positioning subsystem diagram

5.2.1 Assumptions

The user has one pair of hands, with opposing thumbs, and both are neither the same orientation. This means the user will be required to have only one left hand and one right hand. We are assuming the hands are within the camera's field of view and are stable right above the camera.

5.2.2 Responsibilities

Our hand positioning calculates the distance between the hand and each antenna. If the user selects their right hand as their dominant, the distance between the right hand and the virtual antenna will determine the instrument's pitch. The magnitude between their left hand and the horizontal antenna will determine the volume of the note.

5.2.3 Subsystem Interfaces

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Horizontal Magnitude	Positions Hand Labels	Volume Value
#2	Vertical Magnitude	Positions Hand Labels	Pitch Value

5.3 Volume

The volume control in this virtual environment mimics the behavior of a traditional theremin, where the proximity of the hand determines the loudness of the sound. By adjusting the hand's vertical position on the camera, the user can modulate the volume of the instrument.

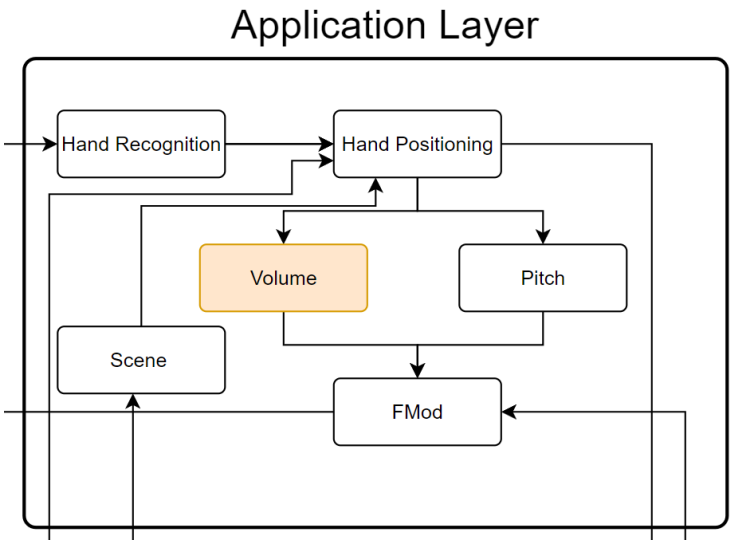

Figure 7: Volume subsystem diagram

5.3.1 Assumptions

Assumed that the camera can accurately detect the vertical positioning of the user's hand and distinguish between varying distances. there is a linear relationship between hand distance from the camera and the volume level. This means as the hand moves further away, the volume increases at a consistent rate, and as it moves closer, the volume decreases at a similar rate. The system assumes there are no significant external factors that might interfere with the camera's detection capability, such as obstructions between the hand and the camera.

5.3.2 Responsibilities

The system is responsible for translating the vertical distance of the hand from the camera into corresponding volume levels. Closer distances mean softer volumes, while greater distances mean louder volumes. The system should provide real-time feedback to the user, allowing them to understand the current volume level and adjust their hand position accordingly. To ensure accuracy and user comfort, the system might need a calibration mode. This will allow users to set their desired position.

5.3.3 Subsystem Interfaces

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Volume Parameter	Volume Value	Note Amplitude

5.4 Pitch

Pitch is controlled in this system by the user's hand positioning. The horizontal proximity of the hand determines the frequency of the sound. By adjusting the hand's horizontal position on the virtual rod, the user can alter the pitch.

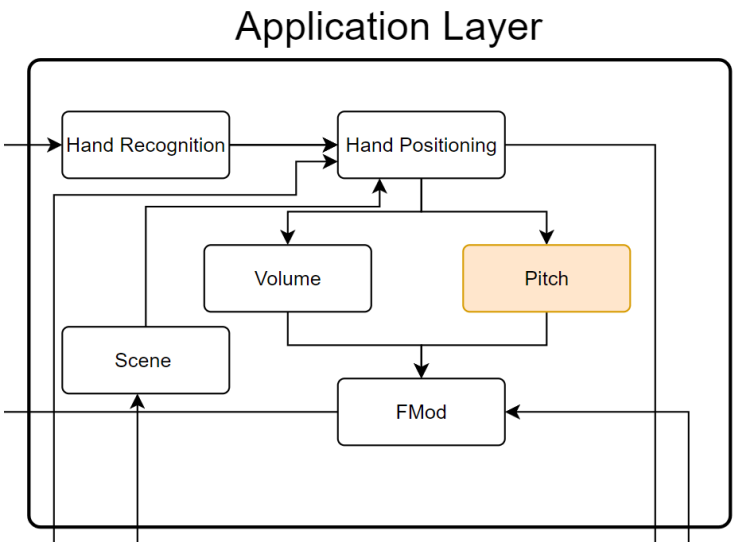

Figure 8: Pitch subsystem diagram

5.4.1 Assumptions

The system can accurately detect the horizontal movement and positioning of the user's hand to distinguish between varying distances and directions. consistent and predictable response in the pitch change as the hand moves horizontally. As the hand moves closer/away from the antenna, the pitch will decrease/increase, respectively, at a steady rate. The system assumes there are no significant obstructions or interference between the camera and the user's hand, which might impact the pitch detection. The application assumes that the hand's shape is distinct enough for accurate pitch modulation, irrespective of the background.

5.4.2 Responsibilities

The system should convert the horizontal positioning of the hand into corresponding pitch levels or frequencies. The further/closer the hand moves, the lower or higher the pitch, respectively. it must provide immediate auditory feedback to the user, ensuring they can hear the change in pitch and adjust their hand positioning to achieve the desired note. the system may need a calibration mode to allow users to set their "lowest pitch" and "highest pitch" hand positions, ensuring user comfort and customized.

5.4.3 Subsystem Interfaces

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Pitch Parameter	Pitch Value	Tone Frequency

5.5 FMod

Our sound engine will be taking all of the parameter values and utilizing them to change a specific sine wave to generate a theremin-like noise. FMod allows for easy integration with our Unity scenes and allows easy editing with the parameters.

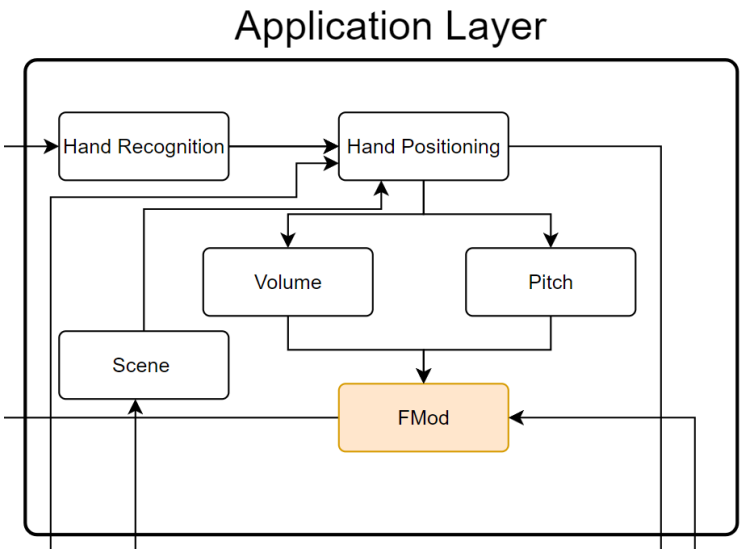

Figure 9: FMod subsystem diagram

5.5.1 Assumptions

FMod requirements are already installed on the user-preferred system, and fully up to date as well.

5.5.2 Responsibilities

FMod is responsible for all of the audio the user can hear when interacting with anything in the scenes. From the theremin noises, User Interface SFX, and other ambient noises.

5.5.3 Subsystem Interfaces

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Audio Output	Pitch Value Volume Value Master Output	Sound Wave

5.6 Scene

Our scene is selected by our user in the User Interface Layer from the Scene selector. This will put the user in the selected lesson to the user.

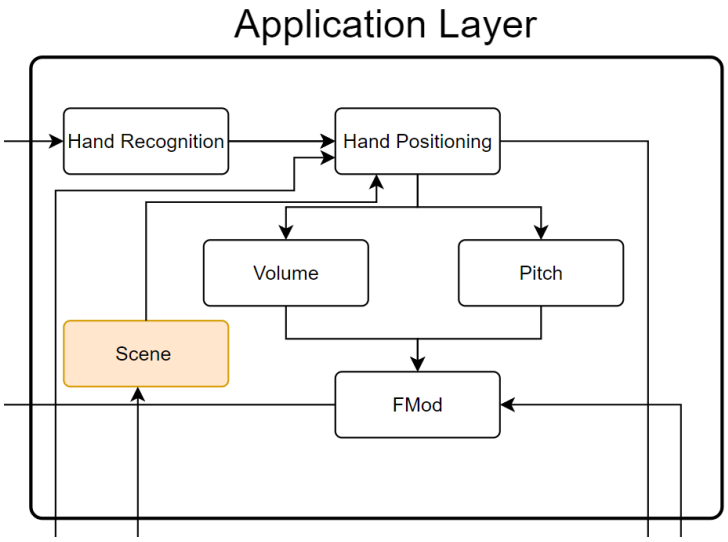

Figure 10: FMod subsystem diagram

5.6.1 Assumptions

We will have a preset lesson plan and the user will keep progressing. We should be able to go back on previously done lessons. The application was correctly installed within the user's preferred system.

5.6.2 Responsibilities

The scene is responsible for the environment, including how the theremin will function. Lesson-based scenes will require the user to follow instructions to understand how to play the instrument, while a sandbox-based scene has no restrictions.

5.6.3 Subsystem Interfaces

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Scene	Selected Scene	Changes scene

6 User Interface Layer Subsystems

The User Interface Layer is the main layer that will handle how the User and the application will interact with one another. This layer will take input from the Application Layer and display this data to the user through interaction with the External Layer. This information being displayed will be a combination of visual and audio outputs that reflect the data that was processed by the Application Layer.

The UI Interaction Management will determine what dominant hand is being displayed to the scene as well as what scene is currently being shown to the External Layer. This User Interface Layer also controls the audio's Master Output which can be changed by the user.

6.1 UI Interaction Management

The general purpose of this sub-layer is to process the hand positioning and other data from the Application Layer and serves as a medium for the user to alter certain aspects of the current application such as the scene and master output for audio.

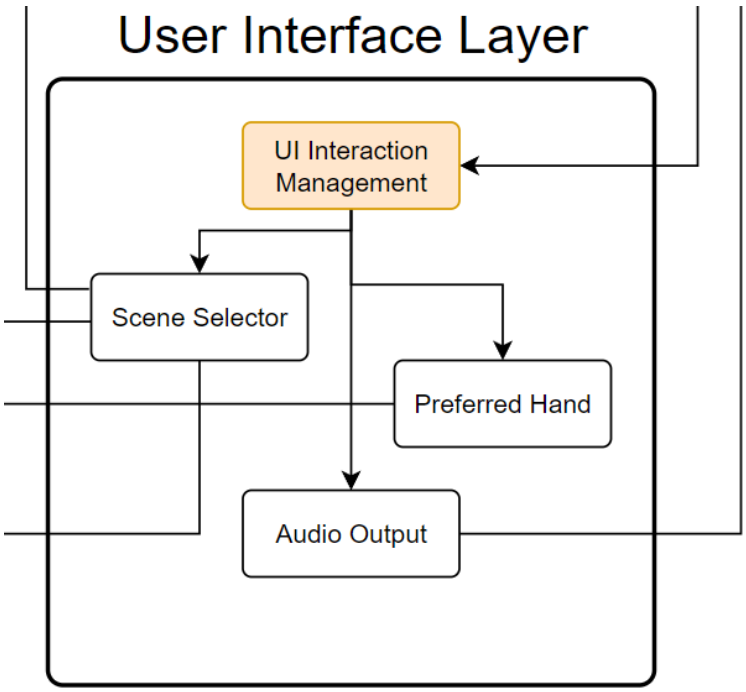

Figure 11: UI Interaction Management subsystem diagram

6.1.1 Assumptions

This sub-layer assumes that the input from the previous layer is correct and calculated accurately with the previous assumptions stated.

6.1.2 Responsibilities

UI Interaction Management is responsible for taking the the input that is given by the user and processed by the application layer then taking that data information and reflecting the changes that were inputted within the Application Layer and the External Layer.

6.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labeled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements that will pass through this interface.

Table 10: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	UI Interaction Management	Hand Positioning	Scene Master Output Preferred Hand

6.2 Scene Selector

This sub-layer will be in charge of selecting the current scene that is being displayed and displaying the hand in the correct orientation based on the data that was processed by previous layers.

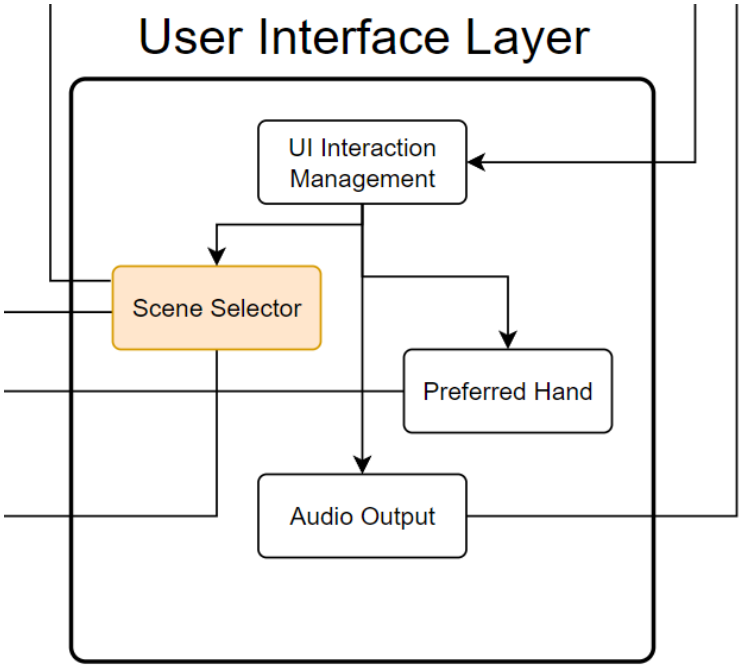

Figure 12: Scene subsystem diagram

6.2.1 Assumptions

This sub-layer assumes that the input from the previous layer is correct and calculated accurately with the previous assumptions stated.

6.2.2 Responsibilities

This sub-layer is responsible for sending the correct orientation of the hands from the scene to the Computer in the External Layer. This is important to reflect that the changing hand orientation from the user is being received from the application.

6.2.3 Subsystem Interfaces

Table 11: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Scene	UI Interaction Management	Computer

6.3 Preferred Hand

Preferred Hand is the subsystem in the User Interface Layer. This is responsible for the selection of the user's preferred hand. This is connected with the UI interaction management and allows the user to choose their preferred hand. The data gets stored as the dominant hand of the user in the data layer.

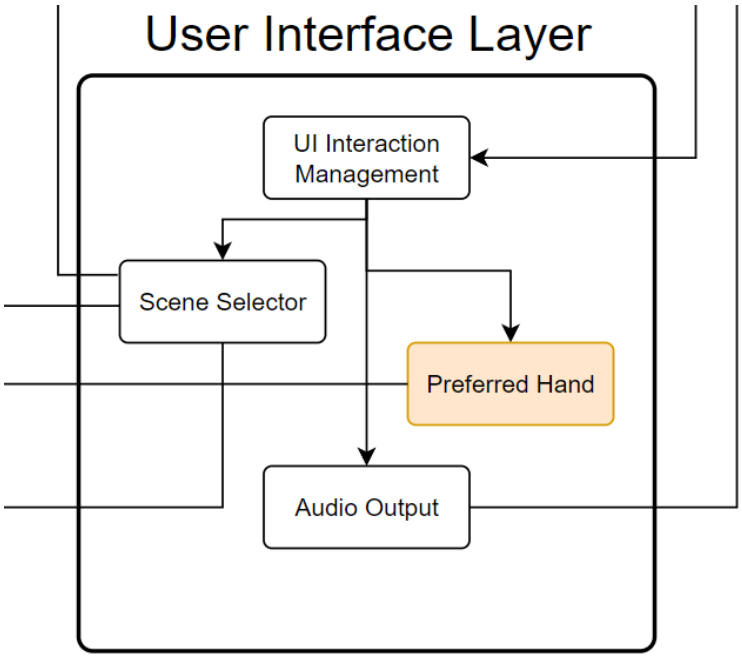

Figure 13: Preferred Hand subsystem diagram

6.3.1 Assumptions

The user knows which is their preferred hand.

6.3.2 Responsibilities

This subsystem relies on a user profile that includes hand preference as part of the user's settings. The UI detects the preferred hand of the user according to their choice. This also determines the data that is being stored for the dominant hand on the data layer.

6.3.3 Subsystem Interfaces

Table 12: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Choice of the preferred Hand	Right/Left Hand	Selected Hand

6.4 Audio Output

Audio Output is the subsystem in the User Interface Layer. It is connected to the FMod sound engine in the application layer. This allows an easy way to control the overall volume of the theremin.

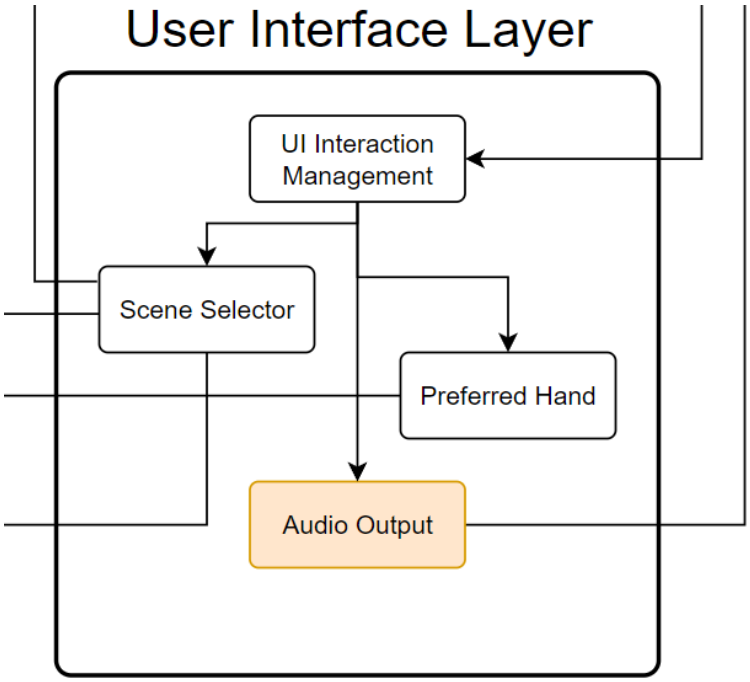

Figure 14: Audio output subsystem diagram

6.4.1 Assumptions

FMod is installed on the user's preferred device and is up to date. The user's preferred device has a way to output sounds and knows the decibel level they wish to set the theremin on.

6.4.2 Responsibilities

This subsystem receives signals from UI interaction management and sends the data to FMod to generate the audio output.

6.4.3 Subsystem Interfaces

Table 13: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Decibel Level	Hand Positioning	dB Level

7 Data Layer Subsystems

7.1 Dominate Hand

This is part of the data layer. The UI would allow the user to choose their preferred hand, and this data would be stored as the user's dominant hand. The data for the dominant hand will also determine the hand positioning.

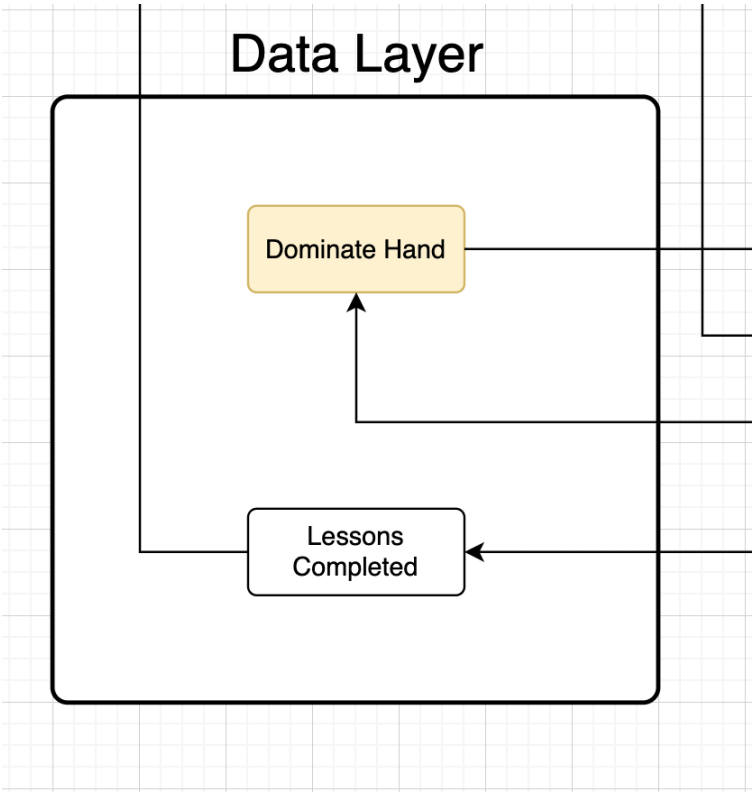

Figure 15: Dominate Hand subsystem diagram

7.1.1 Assumptions

The user has two hands (left and right) and they know which is their dominant hand.

7.1.2 Responsibilities

The dominant hand is the subsystem that is responsible for the data regarding the preferred hand of the user. The UI detects the dominant hand and it is related to the hand positioning/gestures, in the application layer.

7.1.3 Subsystem Interfaces

Table 14: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	User's Preferred Hand	Boolean	Hand Option

7.2 Lessons Completed

This is part of the data layer. The UI would allow the users to view the lessons they have completed. This will be displayed on the screen. The data regarding the user's lesson completion will also be stored in this subsystem.

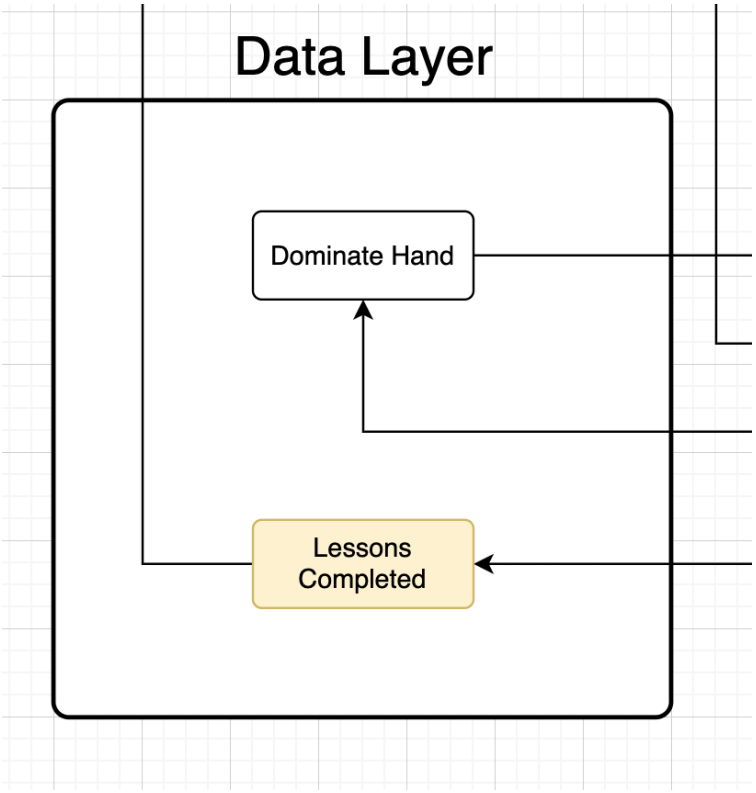

Figure 16: lessons completed subsystem diagram

7.2.1 Assumptions

The user acknowledges that lessons will be either completed or uncompleted. The user knows lessons are repeatable.

7.2.2 Responsibilities

Lessons completed is the subsystem that is responsible for the data regarding the lessons that the users have attempted to complete or completed. It is also connected to the external layer as the data gets displayed on the computer screen when the user wants to view it.

7.2.3 Subsystem Interfaces

Table 15: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Lesson Parameters	Boolean	Visual Feedback