ThereminFreaks – Theremin-based Rhythm Game

Team 34
Esteban Looser-Rojas – looserr2
Yhoas Olivas Hernandez – olivash2
Michael Recklein – recklei2
TA: Amr Martini
ECE 445 – Spring 2019

Introduction

Objective and Background

There are many different kinds of rhythm games on the market, stretching back all the way to the 90s and early 2000s with games like DanceDanceRevolution and Pump It Up, two dance games where one steps on panels; beatmania IIDX, which is a "DJ simulator" and uses a turntable and 7 rectangular buttons; and DrumMania which uses an electric drum set to simulate playing a drum. They typically involve the player pressing a button or a touch screen and/or simulating a conventional rock instrument like a guitar or a drum set. They also typically only allow discrete notes and when there is an analog or continuous control, it's usually not that precise, like the knobs in a game like Sound Voltex. The closest thing to what we plan on making is the singing mode in Rock Band, which only takes the user's voice pitch into account.

We are planning on creating a theremin-based rhythm game for the PC platform. The hardware component is a PC peripheral theremin where the capacitance is controlled by moving one's hands closer or farther from the antenna-like plates on the theremin controller. We plan on using this capacitance to affect an oscillator's frequency and capture this frequency as a variable on the PC program using USB. Thus the theremin circuit will be connected to a USB controller which will connect to the PC. On the software side, we will make a simple driver for this controller and a video game written in C++ using OpenGL. The way the game works is there is a stream of notes coming at the player. The vertical axis represents the volume to correspond with the loop antenna on the theremin. The horizontal axis represents pitch to correspond with the straight antenna.

What makes this project unique is therefore the fact that it uses an unconventional instrument where there is no contact between the player and the instrument. And the pitch and volume are continuous rather than being discrete like pressing a piano. A game like Rock Band has a singing mode (actually based off a previous game called Karaoke Revolution) that is somewhat similar to what we are doing, but it only takes pitch into account and not volume [1].

Rhythm games with realistic controllers like the aforementioned DrumMania also have the advantage of teaching people how the instruments they simulate are played. In the words of Ben Weinmann and Chris Pennie from The Dillinger Escape Plan [2], "Being big fans of Konami games, we are extremely happy to be a part of the newest versions of GUITARFREAKS and DRUMMANIA. We can remember our first trip to Japan and having the pleasure of playing earlier versions of these games in the Arcade. These Games are not only fun but a great way for young people to start looking at music in a more intricate and educational way. Thanks Konami!"

High-level requirements

- The user interface and hardware be intuitive enough that a new player can learn how a theremin works. They should then be able to play the theremin with some sort of skill outside of the game.
- Playtesters should be able to consider the game responsive and bug-free, along with being enjoyable and possessing engaging mechanics as a game.
- The sound engine should work properly such that there is no pitch distortion and provide a good mapping between the distance of the hands from the antennas and the controls.

Design

Block Diagram

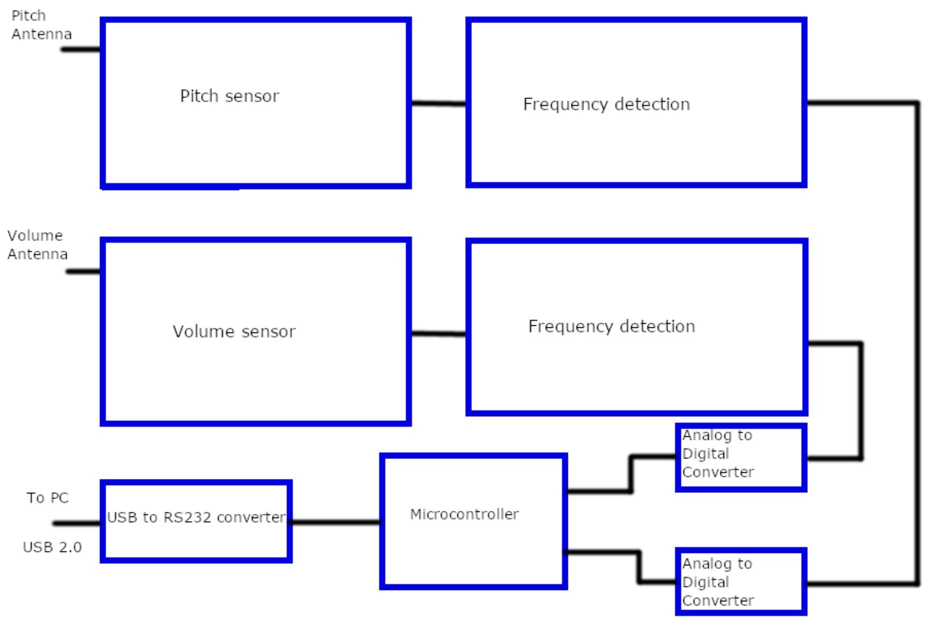

Figure 1: Hardware block diagram for the theremin

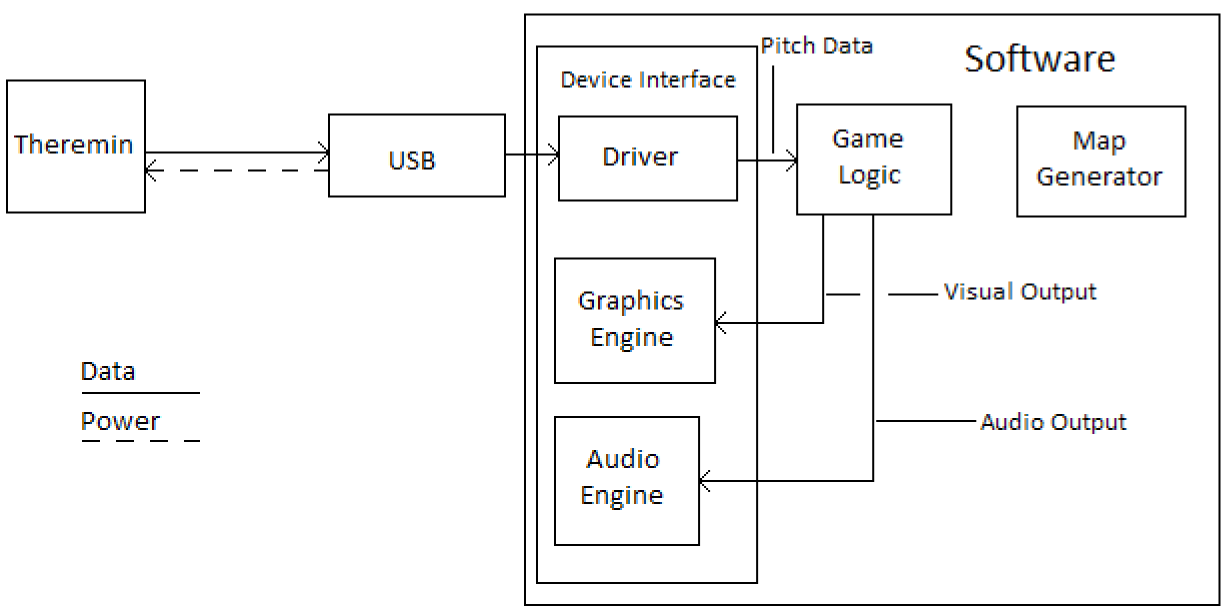

Figure 2: Software block diagram for the video game

The theremin controller is implemented as close as possible to a real theremin in order to fulfill the first requirement, though we need to use ADCs and a USB interface rather than having an audio amplifier. With the amount of control we get from the software audio engine and game logic, we should be able to finely tweak how the scaling of the theremin is configured and how it sounds while it is being played.

Functional Overview and Block Requirements

Hardware:

Pitch sensor – The pitch sensor is a capacitive sensor that senses the position of the players hands and outputs a frequency. Requirements: This sensor must respond quickly to the players movement (about 40 updates per second) and must give us a one to one mapping of distance to frequency such that we can solve for the distance between the players hands and the antenna by counting the frequency. Care should be taken to ensure that this module can operate from a single 5 volt supply, so that it can be powered directly from the USB port.

Volume sensor – The volume sensor controls the volume for the game. It has the same requirements as the pitch sensor.

Frequency detection – This module will be used to convert the output of the sensors into an analog signal that can be sampled. The circuit must have a one to one mapping of frequency to voltage, such that we can determine the frequency output of the sensors by sampling the analog output. Requirements: This module must have a high enough bandwidth to respond to the output of the sensors (up to 30 KHz), and must operate from a single 5 volt supply.

Analog to digital converter – This module is used to digitize the signal from the frequency detector. We will probably use an off-the-shelf 16 bit ADC for this. Requirements: The sample rate does not need to be very high (less than 1000 samples per second). This module should be able to operate from a single 5 volt supply. A parallel interface would be preferable for the sake of simplicity, but if the microcontroller we chose does not have enough pins an I2C or SPI interface can be used.

Microcontroller – The microcontroller must read data from the analog to digital converters and send it to the PC. This does not have to be a USB-capable microcontroller, since a separate module will handle USB communication. Requirements: The data must be sent to the USB interface using an RS232 internal interface. The task handled by this microcontroller is simple, so most general purpose PIC or AVR microcontrollers are suitable. The microcontroller must operate from a single 5 volt supply, and must have 5 volt IO.

USB to RS232 converter – In order to simplify the the software design a virtual RS232 interface will be used. To see an example of a device that uses this type of interface, look at a RedBoard (an Arduino clone). Requirement: The USB to RS232 converter will use a USB interface to emulate an RS232 port, which will be connected to the microcontroller. An off-the-shelf FTDI chip will be used for this task.

Software:

Graphics engine and game logic – This is what will take in the data from the theremin and the map generator and show the notes to the player, along with judging the player based off of how close their pitch and volume is to the note map. Requirement: should be able to run at least 60 fps in order to maintain a smooth playing experience.

Audio engine – Takes in pitch and volume data from the theremin, along with background music files and plays the background music along with a synthesized theremin track manipulated by the player.

Requirement: should be able to synthesize a theremin-like sound in real time at 44.1 kHz to simulate theremin-playing convincingly.

Driver – Moves pitch and volume data from theremin USB input to the game. Requirement: should provide a steady stream of 16-bit samples from the theremin at a high enough bandwidth to minimize delay below 50 ms in order to prevent audio lag.

Map generator – Takes a song and, using DSP, creates a note map for the game. Requirement: the note map should make sense and be possible to play.

Risk Analysis

The oscillators pose the greatest risk for the success of our project. For one, we are using transformers for our oscillators, so we need to order them and then wire them properly. They might use more voltage and power provided by the USB. Once we do that we also have to tune them precisely so that the user's movements are tracked accurately and faithfully to how a real theremin operates. If we are careless with how we approach this part of the theremin circuit, we could have a very inaccurate frequency. For example, the oscillators work at megahertz frequencies. If it's off by a kilohertz, then we'll have a tone of a kilohertz out of nowhere, throwing everything off.

Ethics and Safety

The ACM code of ethics talks about reducing harm, including physical and mental injury and unjustified harm to the environment [3]. We won't be working with any lethal voltages, so the risk of harming someone with our theremin is very low, comparable to the risk of someone having to plug in any household appliance. And our theremin doesn't use radio in order to communicate, so the risks associated with interference are negligible if we shield the components well enough. Our video game is simply a free musical instrument simulator without any micro-transactions, and the music we are composing for it won't have any explicit lyrics. So anyone of any age should be able to play it.

The IEEE and ACM codes of ethics also mention honesty and trustworthiness, along with abiding by copyright law [3][4]. We will disclose all the components we are using for the theremin and will keep track of all our source code with appropriate licenses. That includes code used to control the theremin's microcontroller and the code for the video game. We will also create our own circuit schematics and will only use software and libraries pursuant to their disclosed licenses. For example, OpenGL uses a free license that allows us to use it for our project. As for the music used in the video game, we will avoid infringing on copyright law by using either our own arrangements of public domain music or music we have created ourselves.

The ACM code of ethics discusses user privacy and confidentiality [3]. We are not designing the theremin or video game to communicate any information through the Internet or to store any kind of user information. So users' privacy rights are not at risk either.

References

[1] “Karaoke Revolution,” *Wikipedia*, May 17 2018. [Online]. Available: https://en.wikipedia.org/wiki/Karaoke_Revolution. [Accessed Feb 7, 2019].

[2] “Panasonic Youth,” *RemyWiki.com*, Dec. 27 2018. [Online]. Available: https://remywiki.com/Panasonic_Youth#Song_Production_Information. [Accessed Feb 1, 2019].

[3] Association for Computing Machinery, “ACM Code of Ethics and Professional Conduct,” *Association for Computing Machinery.* [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: Feb. 1 2019].

[4] IEEE, “IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: Feb. 1, 2019].