Master of Rock Device and App

- Development and Course Notes -

# 1 | Project Idea: Master of Rock Guitar Hero Device and App

What if there was a device similar to Guitar Hero, which represented the original instrument more accurately? Current devices have five or six buttons and one switch for the strummer bar.

­­­Diagram

Description automatically generatedA picture containing guitar

Description automatically generated

Figure | Guitar Hero Standard Guitars

Guitar Hero offers an excellent opportunity for entertainment; however, it hardly contributes to its users' musical education apart from improving fine motor skills and rhythm sense due to its enormous layout difference compared to the original instrument. Nevertheless, with the proper set of 20 ( or 24) possible finger positions and six-string switches, it might be possible to turn this device into a multi-purpose digital input and build an application around it that is entertaining and seamlessly educational.

This input device could be used for the following:

* Learning the layout and notes of a guitar,
* Follow app instructions to practice scales,
* Play actual music,
* Learn the fundamental chords,
* Learn songs and riffs,
* Learn to read guitar tabs,
* Practice at night,
* Compete with others,
* Practice anywhere without the nuisance of the volume,
* Compose new songs and use monochrome.

The very same time and effort that users spend pressing seemingly arbitrary buttons on a plastic mock guitar could be used to acquire practical knowledge.

Similar devices, apps and inventions:

* Guitar Hero,
* **Digital Guitar** https://www.digitalmusicnews.com/2016/08/01/mi-guitar-easy-to-learn-or-cheating-your-way-to-a-guitar-experience/
* Frets Zealot Fretboard Guitar System,
* Fanville Portable Wooden Pocket Guitar Practice Tool Chord Trainer Finger Exercise Gadget for Beginner Guitar Picks and Finger Strengthener Tool,
* Guitar Riff Pro App,
* RockSmith 2014 Edition https://www.youtube.com/watch?v=mJ8-XzahW1E,
* All available guitar tablature readers.
* Guitar tuners (hardware and software)



Figure | Digital Guitar

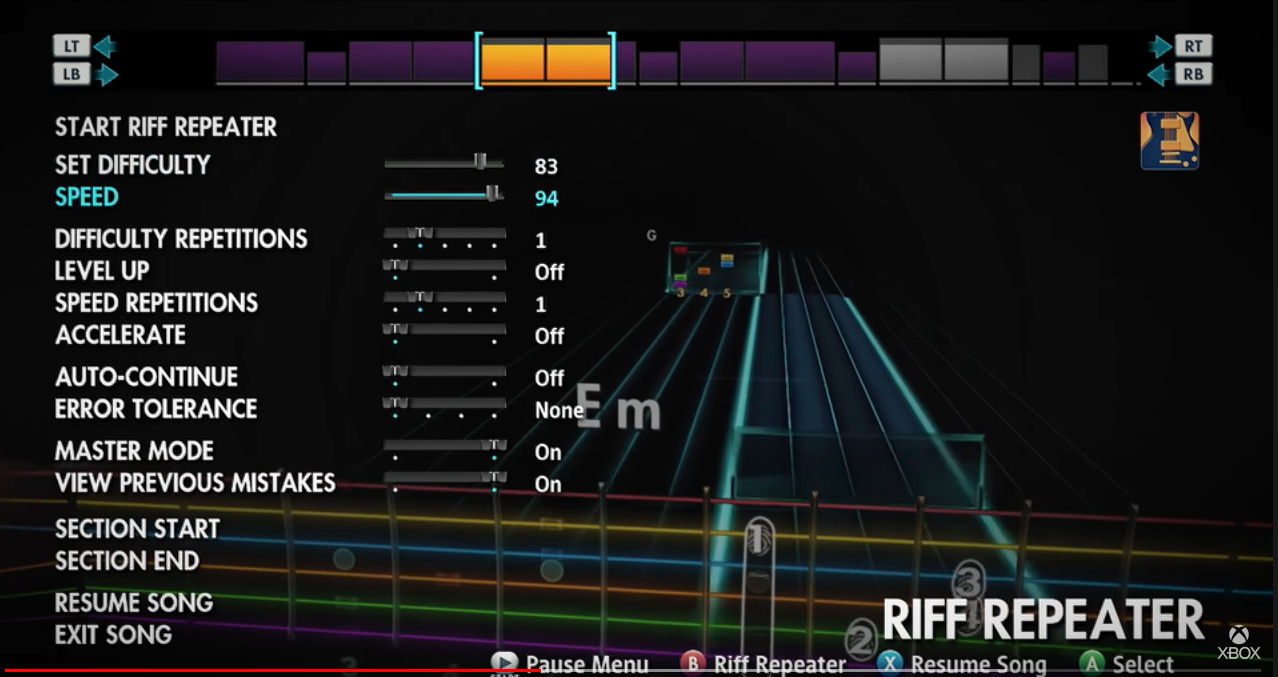


Figure | RockSmith Screenshot

**Challenges**: writing an application that listens to a device's input is not the most significant challenge. However, creating an instrument that handles 20 fret switches and six pieces of strings switches, a USB or Wifi connection that sends user actions to the PC is. Moreover, the device should not only resemble an actual guitar but must be comfortable enough to use. Finally, given the resources and the timeframe, our device and the application would instead serve as proof of concept.

**Approval Form:**

* Title
* Introduction (from general to specific)
  + Topic
  + Background
  + Abstract
* Aim and objectives
* Concept, description,
* Scope, specification, requirements, challenges, benefits, feasibility
* Research and Literature Review
  + Patents
  + Finding Gap
  + Marketability,
  + Feasibility,
  + Cost,
  + Litigations,
  + Limitations and Timeline,
  + Critical Review,
  + Risk Analysis,
* Methodology(Spiral, Waterfall): analyse four and choose one ( two ) (linear – hardware, iterative - software) importance, advantage, disadvantages
* Work Breakdown Structure, Gantt Chart (use ProjectLibre free GantChart project application)

**Strum bar parts:**

https://www.bytearts.com/strumfixplus

# 2 | Concept Diagram

Graphical user interface

Description automatically generated

Figure | Concept

I took the freedom to be creative with the naming, so why not combine the two central aspects of the project, namely, Digital and Guitar? DiGitar.

# 3. | Standard Music and Tabular Notation (Literature Review)

**Introduction 1st version**

Lucky that Sound Exists in Earth -> Ancient Civilisations -> Innate Capability for Musicality -> Music in Everyday life -> Technology Appears -> New Instruments -> Never so Extended Audio Spectrum -> Digital Music -> Gaming

We, Earthly creatures, are very fortunate. Extremely few places in the vast vacuum-filled Universe have a suitable medium that supports audio signals to travel. However, here on Earth, sound vibrations can move through the atmosphere, providing information about our environment. As a result, mammalian evolution adapted to transform soundwaves into electrical signals, engineering us genetically to detect sounds. Hearing sounds increased our survival chances by identifying danger outside our visual zone and extending our communication channels.

Even though humans are not the only species communicating by creating sounds, we invented a way of self-expression that conveyed broader spectrums of emotional range beyond spoken words; music. And from as early as 40000 years ago, music has played an essential part in our everyday life. Our humans' innate musicality drove us to experiment with new ranges of sounds, inventing the primary types of instruments. Ideophones (clapping and bells), membranophones (drums), aerophones (flute), and most importantly, chordophones (harp).

Although the exact origin of the modern guitar is debated, the instrument is already mentioned in the Bible, and it can be traced back to the Greek kithara **κιθάρα** or Arabic qitharah **قيثارة** words. By the 17th century, it became popular among amateurs; and with the advent of the jazz age, the electric guitar had elevated to the instrument of the virtuosos and rock stars. However, this is not the final step for the guitar on the evolutional ladder. The modern digital era opens opportunities to combine the latest technology with musical skills. This project will attempt to bring digital technology, entertainment, art and education under the same roof.

**Inspiration**

I hold in my hand my old buddy, Gabriel's Guitar Hero. Again, I am ready for the next round; this time determined to overdo his performance. Little did I suspect that years of sketchy guitar practices on my side would not score against a seasoned hero like him. After several failed attempts to show off my talent, he concluded that even though I had guitar experience, rhythm sense and some music theory in my pocket, my chances of winning against him were astronomical as a first-timer.

How about him, I asked myself, what type of guitarist would he make, with all those hours of playing the virtual guitar console? The answer came weeks later when he visited me, and I handed him my electric guitar and taught him the intro of a song I knew he liked. Soon enough, he could play a simple piece surprisingly well, though. So, I asked him.

- Why do you waste your time practising an imaginary instrument? You'd become a great guitarist by this time.

- You'll see me playing when they invent guitars for the console. – He answered with a smirk on his face.

Since then, I have been thinking about the wasted talent playing on five plastic buttons and a strum bar. If I could create a lightweight device that resembles an actual guitar, I would be able to develop an online interface that is free, available for everyone, vendor-independent, and educational. I am confident it would be at least as attractive an entertainment option as playing Guitar Hero. Well, the time has come, Gabrel, to wipe off your smirk from your face; you will be the first to play.

**Audio vs Sound (Terminology)**

**The key difference between sound and audio is their form of energy. Sound is mechanical wave energy (longitudinal sound waves) that propagates through a medium causing variations in pressure within the medium. On the other hand, audio is made of electrical energy (analogue or digital signals) that represents sound electrically.**

Shape

Description automatically generated

Figure | Audio vs Sound

**Parts of the Guitar, a Guitar Hero and Master of Rock (Terminology - Diagram)**

**Chord (Terminology)**

"*A chord is three or more notes sounding simultaneously. It can be played on one instrument, like a guitar, or by many instruments at once, like a brassquartet or a choir*" (Ericsson, 2016)

**Riff (Terminology)**

The riff's etymology is unclear; however, it was first used in the 1930-s, probably to shorten the refrain. After that, the term stayed in pop music and now, it refers to short repeated melody patterns. Similarly to other literature, this project and the application will reference these patterns as riffs.

**Monophony (Terminology)**

We address monophony because some of the matrix keyboards are wired, so only one keypress can be recognised at a given time. This limitation would result in a monophonic instrument. However, the guitar is a polyphonic instrument, and strings can be strummed simultaneously. Therefore, to create a complete experience, we need to make the input device polyphonic.

**HOPO (Hammer-on / Pull-off)**

**Marketability (Introduction)**

* Statistics on current trends in music-themed games and applications ()
* Examine gamers who actually would play an instrument
* Examine players who might need a silent device or practice instrument
* Examine composers who might find this device useful (people with a lack of knowledge in tablature or notation)

**Specification list:**

**Input Device:**

* Has the look and feel of a real instrument, only instead of strings, it has momentary press switches and a plastic body,
* Lightweight, preferably plastic solution,
* Switch on/off and LED,
* Has six rows of 20-fret buttons emulating the traditional guitar neck's layout,
* Has six strum bars that can detect upward and downward strums,
* Allow multiple fret button presses simultaneously on different string rows and the strums,
* When the strum bar is activated, it reads the corresponding string row to determine the notes being played,
* The device is compatible with USB ports. The output is easily adaptable; other developers may write applications around it.
* The device must be able to communicate with the computer so that it does not interfere with the laptop's built-in keyboard,
* The hardware must be safe to use and must be in accordance with safety regulations.

**Software:**

* The application can listen to events on the input, such as strum and button press actions,
* The applications can display tablature notations in a user-friendly way,
* Jam option: the user can play the device and listen to the generated music,
* Compose option: the user can record the device, and the produced music is translated into tablature notation, which can be manually edited, saved, or deleted.
* Practice Option: the user can load a tablature, play along with a song, and practice at different speeds. Different sections of the music may be selected for repeated practice.
* Play Option: the user can play a piece of selected music. The application will score the performance according to accurate real-time feedback, considering the player's number of mistakes in note accuracy or rhythm precision.
* Chords Explorer: chords will be clickable throughout the application. When pausing a running session, the player may check chords,
* The software has to be thoroughly tested to employ the best practices and regulations as an utmost priority.

**Limitations**

* A variety of songs, songs should be divided into groups such as genres and eras.
* Multiplayer option,

**Copyrights and Licencing**

Guitar Master is a university project software and device prototype. Check out what licencing needs to be done in music for university and commercial projects.

**Guitar Hero Simulator Arduino Project**

Diagram, schematic

Description automatically generated

Figure | Guitar Hero Simulator Electric Diagram

Source: https://arduino-projects4u.com/guitar-hero/

**Diagram

Description automatically generated**

Figure | Guitar Hero Controller Layout (https://fccid.io/VFIBW95123805/User-Manual/Users-Manual-814804)

**Literature Review: Technical Review and Challenges**

**Exploring Design Options: Strings vs Strumbar**

* **Research on how a string frequency or change of tension pressure can be detected and translated into Arduino digital Porst**
* **Research on string-pulling tension on different strings**
* **Try to build or experiment if possible**
* **Research strum bar switches**
* **Contrast them**
* **Determine the chosen solution and justify**

**Lecture 3**

**Use ProjectLibre for project management instead of Excel. And put milestones by inserting 0 days.**

**Requirements should be atomic and granulated in a list and sublist structure**

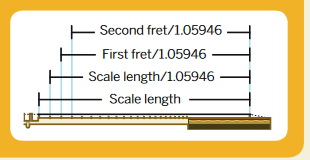


Figure | Fret Distances https://cdn.makezine.com/make/21/make\_v21\_cigar\_box\_guitar.pdf

Text, letter

Description automatically generated

Figure | Calculating Fret Distances (http://www.buildyourguitar.com/resources/tips/fretdist.htm)

**Console Design**

**Calculating Fret Distances on the Console Interface**

As we discussed, the console layout must simulate the guitar experience as closely as possible. Current devices, such as the MI Digital Guitar, do not account for the distances between the frets and button cap sizes. Therefore, we need to calibrate the console neck to a natural proportion where the 12th fret is halfway to the scale length between the nut and the saddle. The fret positions may be calculated by dividing the remainder of a fret-saddle distance by 17.817 recursively. For our initial calculations, we can base our measurements on the length of a Guitar Hero console.

**Text

Description automatically generated**

Figure | Calculate Fret Distances (appendix/fret\_distances.py)

The resulting array has the following values:

Table

Description automatically generated with low confidence

Figure | Fret Distances

These distance values can be used to create the first approximate diagram of the interface's physical layout. However, this layout does not yet consider a wiggle place between the buttons, which should ideally be at least one millimetre.

A picture containing shoji

Description automatically generated

Figure | Fret Distances on The Console Guitar's Neck

**Key Switch Interface Design Alternatives**

To build a successful guitar system based on timing, we need to be able to detect inputs from the user interface accurately and efficiently. From an electrical engineer's perspective, our console can be abstracted down as an intricate keyboard input that consists of key switches, similar to a conventional computer keyboard. "*Depending on how individual switches are connected, mechanical keypads are commonly available in two forms – matrix and common bus*" (Dave, n.d.).

**Keyboard Buses**

In our scenario, we need six times 20 fret buttons and six strum switches, totalling 126 switches. Unfortunately, connecting switches directly to pins would be inelegant even if we could find a microcontroller with 126 digital pins. The fundamental way of sequential wiring design is to use a common bus.

Calendar

Description automatically generated

Figure | Common Bus Consol Interface Wiring Schema (appendix/CommonBus.draw.io)

This solution in the above form is not ideal for our console; however, other linear solutions will be discussed soon.

**Keyboard Matrix**

User interfaces, such as keyboards and keypads, often use a keyboard matrix to consolidate a greater number of input switches to a limited number of microcontroller pins. For example, PC keyboards usually range from 63 to 105 keys, depending on the layout and the existence of a numerical pad. In the same way as conventional keyboards, the guitar interface can be arranged in a matrix. Meshing the switch wires would result in a drastically reduced digital pin requirement.

"*When a key is pressed, a column wire makes contact with a row wire and completes a circuit. The keyboard controller detects this closed circuit and registers it as a key press*" (Dribin, 2000)

Diagram

Description automatically generated

Figure | Keyboard Scanning

**Debounce Mechanism**

"*The physical action of pushing a button might require a half-second or so, so we tend to think in those terms. On the other hand, a digital circuit can react to a million of events in the same time frame*" (Warren, 2015). Because of the switch mechanisms, when a push button is pressed, it may register multiple interactions with the input device in a relatively short interval. In the case of a game console interface, it would result in a disastrous user experience. Various electrical solutions, such as flip-flops and Schmitt triggers, have been used to solve this problem. A "S*chmitt trigger circuit relies on changing the voltage or current threshold levels by means of positive feedback in the analogue loop*" (Kader, 2012), improving the immunity to analogue disturbances.

Diagram, engineering drawing

Description automatically generated

Figure | Schmitt Trigger (https://www.watelectronics.com, 2022)

However, we can solve debouncing using software engineering by measuring oscillation time. For example, Arduino microcontrollers have a millis function that "*returns the number of milliseconds passed since the Arduino board began running the current program. This number will overflow (go back to zero), after approximately 50 days*" (Arduino.cc, 2022). Therefore, we can prevent debouncing by measuring switch state changes. The time of the state changes should be recorded, and an intentional debounce delay should be applied to compensate for the noise. The change should be ignored when low-state changes happen in unreasonably short intervals.

Graphical user interface, text

Description automatically generated

Figure | Compensating Debouncing on Arduino (https://docs.arduino.cc/built-in-examples/digital/Debounce, 2022)

**Ghosting and Masking**

**0. |Reference Suggestions**

* Digitalisation in the music industry and entertainment

https://pubs.aeaweb.org/doi/pdf/10.1257/jep.31.3.195

* Musical Instrument Taxonomy 5 main types

https://www.goshen.edu/academics/music/mary-k-oyer-african-music-archive/instrument-classification/

**Arduino**.cc, 2022., Access: https://docs.arduino.cc/built-in-examples/digital/Debounce,

**Eriksson**, J., 2016. Chord and modality analysis. KTH, School of Computer Science and Communication (CSC), Speech,

Music and Hearing, TMH, Speech Communication and Technology.

**Dribin**, D., 2000. Keyboard matrix help.

**Dave**, V. E., Rajiv B, n.d. Reading Matrix and Common Bus Keypads (PSoC 1)., Cypress AN2034.

**Kader**, W.M., Rashid, H., Mamun, M. and Bhuiyan, M.A.S., 2012. Advancement of CMOS Schmitt trigger circuits. *Modern Applied Science*, *6*(12), p.51.

**Warren**, G., 2015. Exploring the Raspberry Pi 2 with C++. Springer.

**Appendix**



Figure | Notes and Frequencies

Text, arrow

Description automatically generatedFigure | Standard Music Notation

https://www.istockphoto.com/vector/music-notes-and-symbols-set-stock-vector-illustration-gm1291815513-386828576

*A person playing a guitar

Description automatically generated with low confidence*

Figure | Cover Photo

https://www.wallpaperflare.com/boy-playing-the-guitar-on-the-road-music-musician-musical-instrument-wallpaper-wdokr

### 1.4.1 | User Requirements

### 1.4.2. | System Requirements

### 1.4.3. | Scope and Limitations

## 1.5. | Critical Review of Concept

### 1.5.1. | Feasibility

### 1.5.2. | Cost and Marketability and Target Users

### 1.5.3. | Patents and Litigations

### 1.5.4. | Risk Management and Analysis

# 2. | Literature Review

## 2.1. | Understanding Digital Music

### 2.1.1. | Taxonomy and Terminologies

### 2.1.2. | Evolution of Musical Digitalisation

### 2.1.3. | Music in Computer Gaming

### 2.1.4. | Standard Music and Tabular Notation

## 2.2. | Technological Challenges

## 2.1. | Microchips

### 2.2.1. | System, Platform and Browser Compatibility

## 2.3. | Education throughout Gaming

# 3. | Project Management

## 3.1. | Relevant Methodologies

### 3.1.1. | Linear Methodologies

### 3.1.2. | Iterative Methodologies

### 3.1.3. | Critical Comparision of Methodologies

## 3.2. | Justification of Applied Methodology

## 3.3. | Work Breakdown Structure

## 3.4. | Project Phases and Schedule

## 3.5. | Tools

## 3.5.1. | Hardware Technologies

## 3.5.2. | Software Technologies

# 4. | Design

## 4.1. | Hardware: Designing the Instrument

### 4.1.1. | Electric Wiring

### 4.1.2. | Microcontroller Programming

### 4.1.3. | Instrument Design Implementation

## 4.2. | Middleware

## 4.2.1. | Communication and Protocols

### 4.2.2. | Skeleton Prototype

## 4.3. | Software: Designing the Application

### 4.3.1. | Frontend

#### 4.3.1.1. | Webdesign and Wireframes

#### 4.3.1.2. | Components

#### 4.3.1.3. | Visual Design Standards

### 4.3.2. | Backend

#### 4.3.2.1. | Translating the Business Model

#### 4.3.2.2. | Database

#### 4.3.2.3. | Restful Queries

# 5. | Development

## 5.1. | Hardware: Manufacturing the Prototype Instrument

## 5.2. | Middleware: Communication

## 5.3. | Software: Developing the Web Application

### 5.3.1. | Frontend Development

#### 5.3.1.1. | Landing Page

#### 5.3.1.2. | Reusable Functionalities

#### 5.3.1.3. | Gameplay

#### 5.3.1.4. | Composing

#### 5.3.1.5. | Additional Functionalities

### 5.3.2. | Backend Development

# 6. | Quality Assurance

## 6.1. | Feasibility: Testing a Skeleton Device

## 6.2. | Testing Hardware: Electronic Requirements

## 6.3. | Testing the Communication

## 6.4 | Software Testing

## 6.5. | Automated Testing

## 6.6. | Unit Testing

## 6.7. | Functional Testing

## 6.8. | User Acceptance Testing

# 7. | Retrospective: Review of the Final Project

## 7.1. | The End Product

## 7.2. | Fulfillment of the Objectives

## 7.3. | Critical Review and Opportunities for Future Improvements

## 7.4. | Conclusion

**Clarify with Supervisor:**

* Do cover quotes and images require references?
* Do Introduction and Inspiration need references?
* Is the tools section in the Introduction?
* Where does the Terminology section go?
* What is the correct term for academic supervisor?
* What is the word limit on a bachelor's final year dissertation project?
* Do I need to reference the bibliography?
* Where to put existing technologies? Introduction to define our clear objectives or literature review?