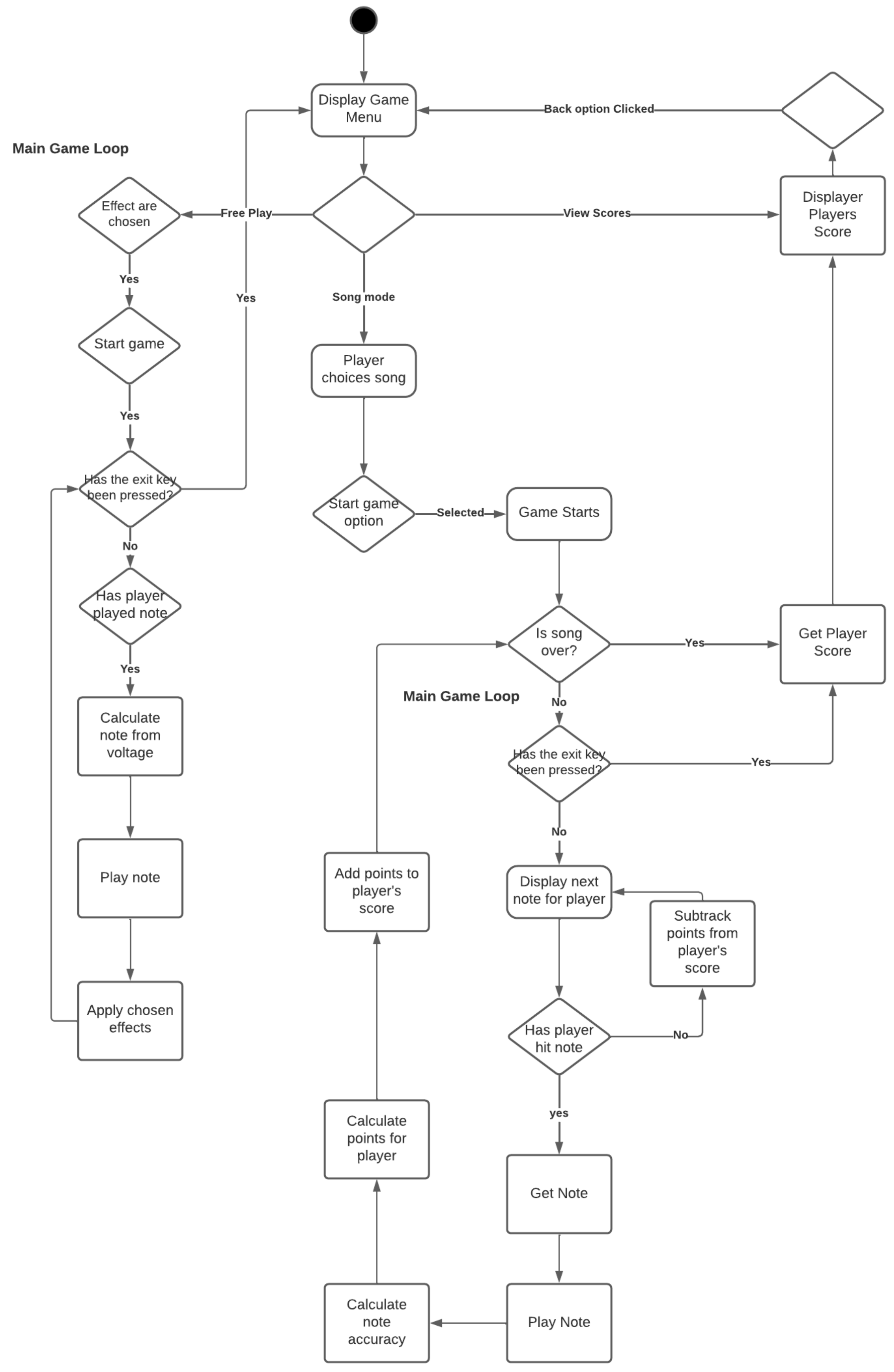
IRL Guitar Hero Midi Trainer

Tom Chambers

Games Academy, Falmouth University

Introduction

The project is an interactive physical game where you train you guitar skills and compete for higher and higher scores in a guitar hero inspired game. A guitar will form the controller and when played, will record the notes from the fretting hand. An interface will show the guitar neck and light up colour on the neck for the user to play. LED's on the neck will also light up, testing the players reaction speed and note accuracy. The game will be challenging but will also improve the musical ability and dexterity of the player. There has also been really promising research using VBT [1] into how games could help train people in certain skills or help them physically and I'm intrigued if the controller could help people learn or improve their skills at guitar or their improve hand dexterity[2].



Main Game Loop

Display Game Menu

Free Play

View Scores

Song mode

Back option Clicked

Effect are chosen

Yes

Start game

Yes

Has the exit key been pressed?

Yes

No

Has player played note

Yes

Calculate note from voltage

Play note

Apply chosen effects

Player choices song

Start game option

Selected

Game Starts

Displayer Players Score

Is song over?

Yes

Get Player Score

Main Game Loop

No

Has the exit key been pressed?

Yes

No

Display next note for player

Subtrack points from player's score

Has player hit note

No

yes

Get Note

Play Note

Calculate note accuracy

Calculate points for player

Add points to player's score

Figure 1:UML Diagram of Games Flow

Note Calculation

For each pin/string there are six possible notes. The number of required resistors is $R_{Totalnum} = Fret_{Totalnum} + 1$. So for 6 notes, thats 6 frets thus 7 resistors. The Adrduino needs a way to differentiate between each note. The potential divider circuit does this. Between each resistor, where the note is played, the voltage is stepped down. The lower frets have the highest potential and the higher frets have the lower potential, such as $fret_{num} \propto v_{out}{}^{-1}$. The V_{out} for each fret, taking into account only the two surrounding resistors is calculated by $V_{out} = \frac{R_1}{R_1+R_2}V_{in}$. Where in the full circuit its calculated by,

$$V_{out} = \frac{Sum\ of\ Resistors\ after\ V_{out}}{Sum\ of\ all\ Resistors} * Voltage\ In$$

Which can be simplified to,

$$V_{out} = \frac{Fret_{num}}{R_{numTotal}}V_{in}$$

Once the Arduino reads this voltage from the input analogue pin, it converts the analogue input voltage into a digital output value. The input voltage is given the closest value between 0-1023 (2^{10} values). This output reading is then the product of the input voltage and the maximum voltage value divided by the range of voltage values.

$$VoltageReading = \frac{5}{1023} * InputReading$$

The difference in V_{out} between each fret is equal to the product of the input voltage and the value of one resistor, divided by the sum of all resistors, such as

$$\frac{nr+r}{c}V_{in} - \frac{nr}{c}V_{in}$$

$$\frac{nr+r-nr}{c}V_{in}$$

$$\frac{r(n-n)+r}{c}V_{in}$$

$$\frac{r}{c}V_{in}$$

where n is the number of resistors and c is the total resistance and all the resistors are equal. This difference is also equal to the V_{out} of the highest most fret.

Method Hardware

The guitar will have 6 strings and 6 functional frets. A future iteration could have more. Each analogue input pin on the Arduino (A0-A5) will act as the inputs for each string. Along each string, on the neck of the guitar, will be a potential divider circuit. Each fret will be divided by a fixed resistor of 1000Ohms, with each string having a potential of 5v. Each fret will act as a V_{out}. At rest the circuit is incomplete. When the player plays a note, the circuit is closed at the desired fret, and the voltage can leave from between those two resistors. The following schematic represents a prototype of one string, except each pin, A0...A5 represents each fret's V_{out}, rather than each string, as if the player had played every fret at once.

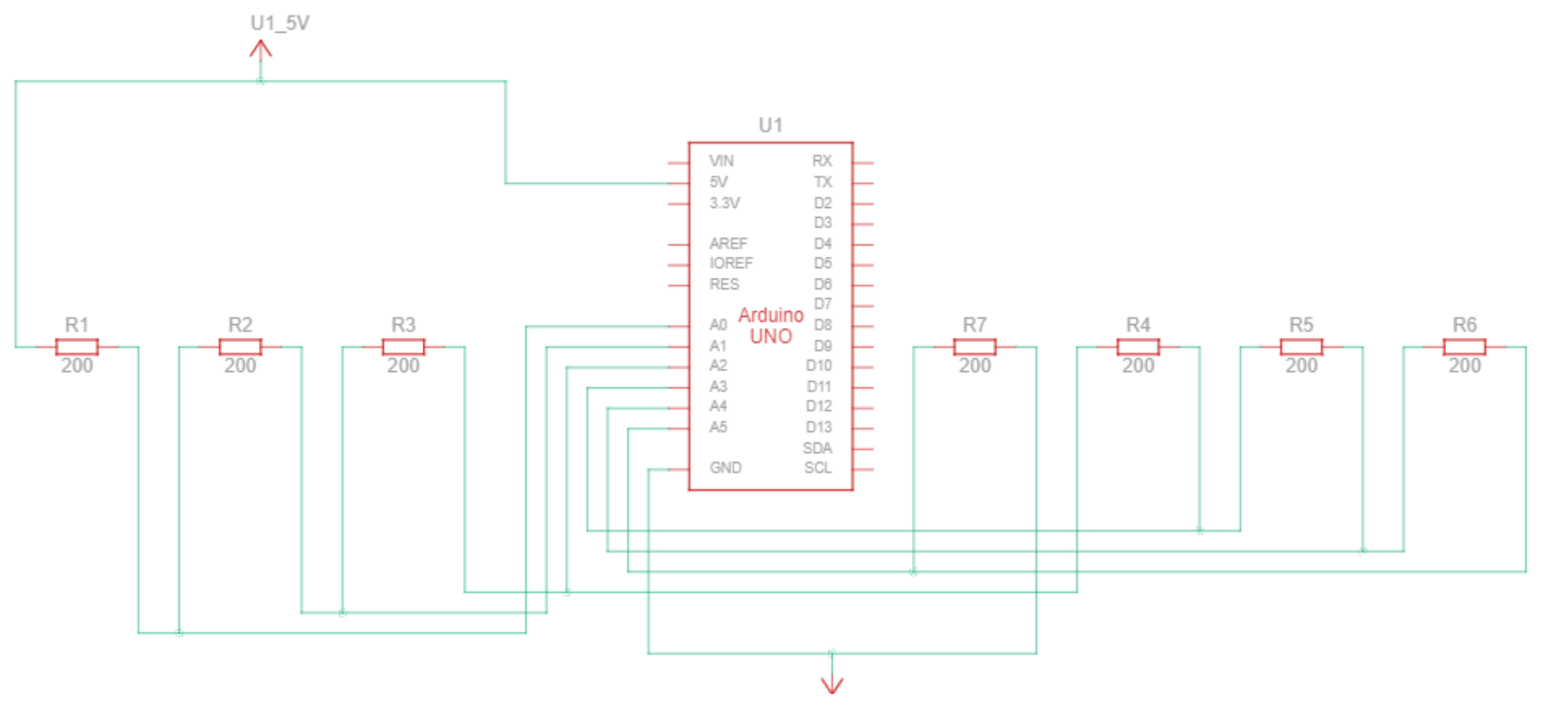


Figure 2:Schematic of a prototype for one string, with a V_{out} for each pin at each resistor

Through testing, nickle guitar strings and human contact do not change the outputted voltage. Nor does metal fret contact on a guitar.

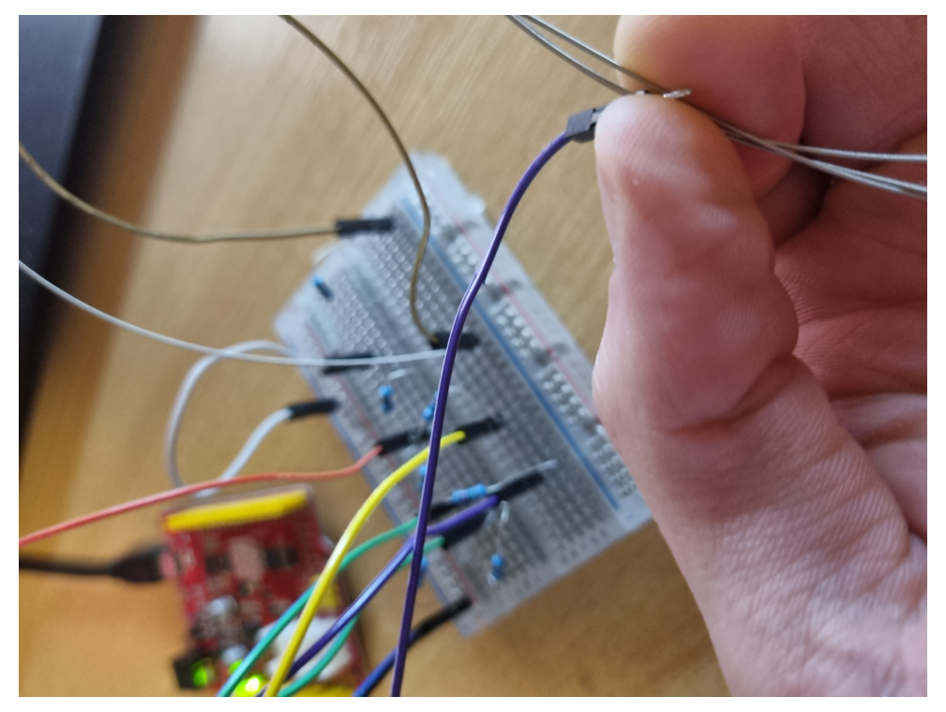


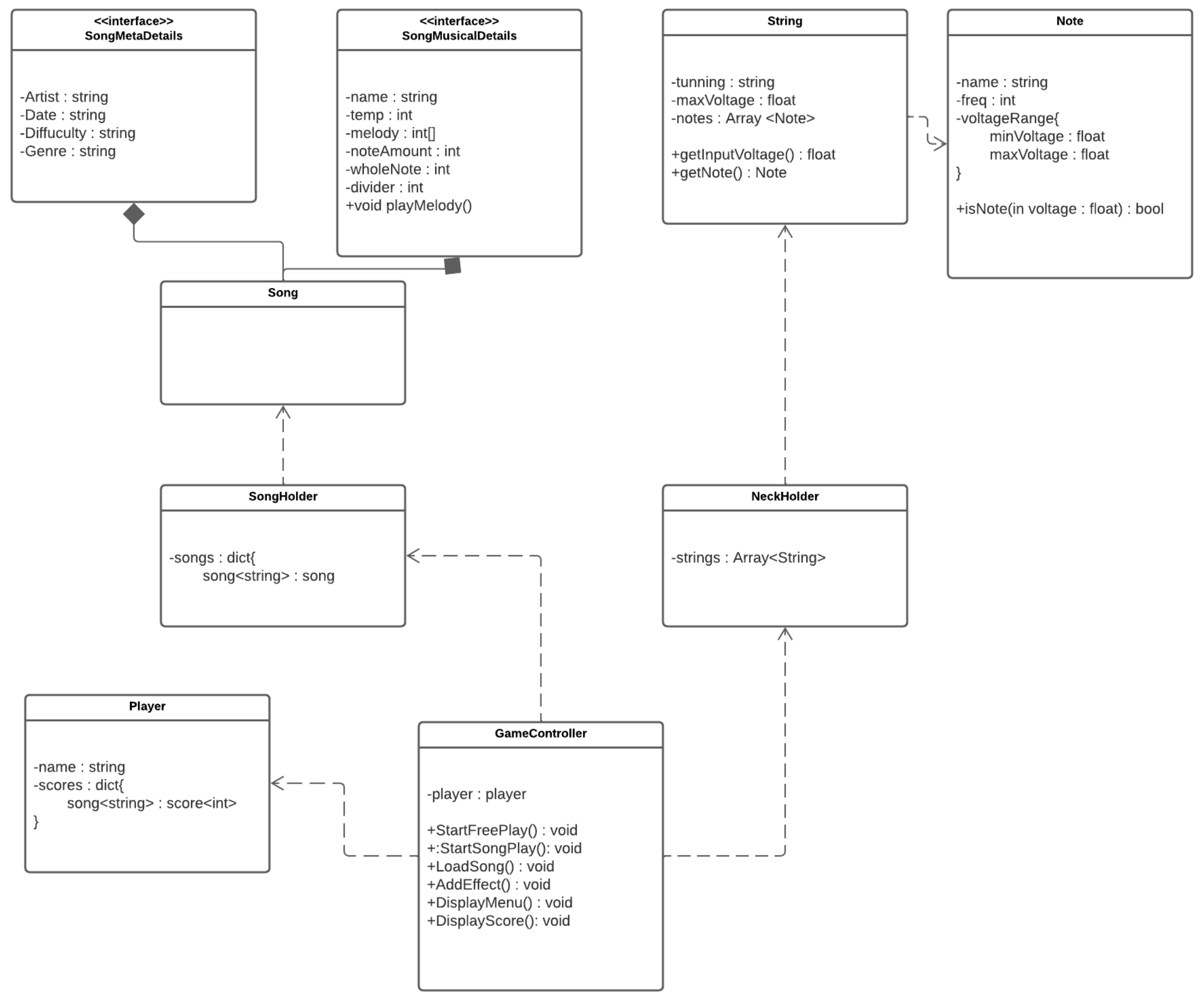
Figure 3:Test with a guitar string in a physical version of the above schematic

Key Components

- 42 100Ohm Resistors
- 6 Nickle Guitar Strings
- Guitar
- Arduino
- Speakers

Software

The software design will take a object-oriented approach will a particular focus on SOLID and the open closed principle. Each pre-made song will be an object instantiated from the song class. They can play through a Piezo or normal speaker. The game will register the notes and require the player to play them in time for points. As a stretch goal, there will also be a free play mode which does not have a scoring system, and is there for effects on the guitar. Songs could also be uploaded which a audio-track to play.



<<interface>>
SongMetaDetails

-Artist : string
-Date : string
-Diffuculty : string
-Genre : string

<<interface>>
SongMusicalDetails

-name : string
-temp : int
-melody : int[]
-noteAmount : int
-wholeNote : int
-divider : int
+void playMelody()

String

-tunning : string
-maxVoltage : float
-notes : Array <Note>

+getInputVoltage() : float
+getNote() : Note

Note

-name : string
-freq : int
-voltageRange{
minVoltage : float
maxVoltage : float
}

+isNote(in voltage : float) : bool

Song

SongHolder

-songs : dict{
song<string> : song

NeckHolder

-strings : Array<String>

Player

-name : string
-scores : dict{
song<string> : score<int>
}

GameController

-player : player

+StartFreePlay() : void
+:StartSongPlay(): void
+LoadSong() : void
+AddEffect() : void
+DisplayMenu() : void
+DisplayScore(): void

Figure 4:Class diagram of the software implementation

References

[1] Tim Vanbellingen, Suzanne J Filius, Thomas Nyffeler, and Erwin E H van Wegen.
Usability of videogame-based dexterity training in the early rehabilitation phase of stroke patients: A pilot study, Dec 2017.

[2] Bettina Conradi, Martin Hommer, and Robert Kowalski.
From digital to physical: Learning physical computing on interactive surfaces.
In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, page 249–250, New York, NY, USA, 2010. Association for Computing Machinery.