

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit

**Ein Tool zur live und retrospektiven
Analyse von persönlich
aufgenommenen MIDI-Gitarren-Daten**

Dimitrios Komninakis

Studiengang: Softwaretechnik

Prüfer/in: Prof. Michael Sedlmair

Betreuer/in: Dr. rer. nat. Marcel Hlawatsch

Beginn am: 9. Januar 2019

Beendet am: 9. Juli 2019

Kurzfassung

Die Gitarre ist ein sehr beliebtes Laieninstrument. Jedoch ist das Erlernen der Gitarre ein sehr aufwändiger, schwieriger und langer Prozess, an dem viele scheitern. Während einige versuchen, es sich selbst beizubringen, nehmen andere Gitarrenstunden bei einem Lehrer oder sehen sich Videokurse an. Eine speziell zu diesem Zweck geschriebene Software kann dem Nutzer im Selbststudium grundlegende Griffe beibringen, oder einem Lehrer Daten über Genauigkeit und Fehleranzahl liefern. Ausschlaggebend für den Lernerfolg sind dabei verständliche Visualisierungen von musikalischen Informationen.

Diese Bachelorarbeit beschreibt den Bedarf, die Konzipierung, die Entwicklung und das Testen von MIDITabs, einer Software zum Gitarre spielen lernen und üben. Sie umfasst eine Einführung in theoretische Grundlagen der Musik und MIDI, der Technologie auf welcher MIDITabs basiert und bietet eine detaillierte Beschreibung der Software. Zuletzt wird in einer Evaluation die Nutzbarkeit und das Potenzial von MIDITabs erläutert.

Jeder Gitarrist ist unterschiedlich geübt und talentiert im Umgang mit der Gitarre. Einige Gitarristen spielen seit mehreren Jahrzehnten, während andere erst seit wenigen Monaten oder Wochen mit einer Gitarre arbeiten. MIDITabs wurde sowohl für Anfänger, als auch für Fortgeschrittene und Profis geschrieben. Die Software verfügt über drei Modi, welche dem Nutzer zur Verfügung stehen. Abhängig von dessen Erfahrungsgrad, kann er den Practice-Modus nutzen um einen Song oder erste Griffe in seinem eigenem Lerntempo kennen zu lernen oder direkt im Performance-Modus einen Song nachspielen und Gespieltes, zur späteren Analyse im Analysis-Modus, aufnehmen. Der Analysis-Modus bietet dem Nutzer eine detaillierte Übersicht über Statistiken und Problemherde, mithilfe dessen er seine Fertigkeiten im Umgang mit der Gitarre verbessern kann. Wie eine Evaluierung von MIDITabs mit Gitarristen ergab, konnten bereits nach sehr kurzer Nutzungszeit Lernerfolge registriert werden.

Inhaltsverzeichnis

1	Einleitung	11
1.1	Motivation	11
1.2	Aufgabenbeschreibung	11
2	Verwandte Arbeiten	15
3	Theoretische Grundlagen	17
3.1	Musiktheorie	17
3.2	Gitarre	19
3.3	MIDI	22
3.4	MIDI-Gitarre	24
4	Entwicklung	27
4.1	Konzipierung	27
4.2	Wahl der Technologien	33
4.3	Webentwicklung mit HTML, CSS, JavaScript und Node.js	35
4.4	Implementierung	36
5	Ergebnis	41
5.1	MIDITabs	41
5.2	Evaluation	51
5.3	Schlussfolgerung	54
5.4	Ausblick	54
	Literaturverzeichnis	57

Abbildungsverzeichnis

3.1	Notenschreibweise	19
3.2	Klassische Akustik Gitarre [CC 06a]	20
3.3	Gitarrenhals mit Frets	20
3.4	Griffbrettdiagramm für A-Moll Akkord [CC 06b]	21
3.5	FISHMAN Triple Play MIDI-Tonabnehmer [FIS19]	25
3.6	Booten im Hardware Modus [FIS13]	25
4.1	MIDITabs Drei-Schritte-Lernprozess	28
4.2	Prototyp Practice-Modus	28
4.3	Prototyp Performance-Modus	30
4.4	Prototyp Analysis-Modus	31
4.5	Softwarearchitektur - Live Tabs	37
4.6	Softwarearchitektur - Root Tabs	38
4.7	Aufnahme von „Road Trippin‘“ mit MIDI-Gitarre [Fru17]	39
4.8	Bereinigte Aufnahme von „Road Trippin‘“ [Fru17]	39
5.1	MIDITabs Sitemap	41
5.2	MIDITabs Startseite	42
5.3	MIDITabs Practice-Modus	43
5.4	Live-Ansicht mit gespieltem E-Moll Akkord	43
5.5	Practice-Ansicht	44
5.6	Richtig gespielter Tab in Practice-Ansicht	44
5.7	Falsch gespielter Tab in Practice-Ansicht	44
5.8	Metronomsteuerung des Practice-Modus	45
5.9	Statistiken im Practice-Modus	45
5.10	MIDITabs Performance-Modus	46
5.11	Tabs in Performance-Ansicht	46
5.12	Richtig gespielter Tab in Performance-Ansicht	47
5.13	Verpasster und falsch gespielter Tab in Performance-Ansicht	47
5.14	Ungenau gespielter Tab in Performance-Ansicht	47
5.15	Songsteuerung des Performance-Modus	48
5.16	MIDITabs Analysis-Modus	48
5.17	Session Liste mit aufgenommenen Sessions	49
5.18	Root-Tabs im Analysis-Mode (Song „Road Trippin‘“ [Fru17])	49
5.19	Aufnahmen und Statistiken unterhalb der Root-Tabs	50
5.20	Heatmap des Analysis-Mode (Song „Road Trippin‘“ [Fru17])	51
5.21	Diagramm (Fehleranzahl) im Analysis-Mode	51
5.22	Kapodaster am zweiten Fret einer Gitarre	54

Tabellenverzeichnis

3.1	Noten und Frequenzen der Oktaven 1-6 [Joe06]	18
3.2	Beispiel eines Note-On MIDI-Ereignisses der Note C4 bei voller Anschlagstärke	22
3.3	Übersicht MIDI-Befehle mit Status-Bytes	23
4.1	Prototyp Vergleich	35
4.2	Input Kanäle MIDI-Tonabnehmer	37
5.1	Proband 1 Practice-Modus Ergebnisse	52
5.2	Proband 1 Performance-Modus Ergebnisse	53
5.3	Proband 2 Performance-Modus Ergebnisse	53

1 Einleitung

Diese Bachelorarbeit beschreibt „MIDITabs“, eine Software zum Gitarre spielen lernen, welche MIDI-Daten eines Gitarren MIDI-Tonabnehmers aufnimmt und visuell aufbereitet, um dem Anwender ein effektives Lernerlebnis zu ermöglichen. Sie gliedert sich in eine Einleitung, welche die Motivation und Aufgabenbeschreibung erläutert, gefolgt von einer Auflistung verwandter Arbeiten und einer Einführung in theoretische Grundlagen der Musik, MIDI und Gitarre. Darauf folgt eine Beschreibung der Konzipierung, die Wahl der Programmiersprachen sowie Entwicklungsumgebungen und eine Erklärung einiger Implementierungsdetails der Software. Zuletzt wird die MIDITabs Software detailliert beschrieben und in einer Evaluation die Eignung und Nutzbarkeit der Software ermittelt und ein Ergebnis mit Ausblick präsentiert.

1.1 Motivation

Gitarre spielen lernen ist, wie beim Erlernen jedes beliebigen Musikinstrumentes ein anstrengender, zeitintensiver und schwieriger Prozess. Unabhängig davon, ob dabei ein Gitarrenanfänger bei einem Lehrer Gitarrenstunden nimmt oder es sich alleine, über Videos, Anleitungen, Akkorddiagramme oder Ähnliches, beibringt, bedarf es in der Regel viel Zeit und Geduld um das Instrument zu beherrschen. Lernsoftware ist ein effektiver Weg, um diesen Lernprozess zu unterstützen. Dabei könnte die Software im Selbststudium oder begleitend mit einem Musiklehrer genutzt werden.

Eine speziell angefertigte Software könnte MIDI-Signale, digitale Musikinformationen, von einer MIDI-Gitarre aufnehmen, analysieren und verarbeiten, um einem Nutzer Griffe beizubringen, diese auf Richtigkeit zu prüfen, Feedback zu geben und Fortschritte zu dokumentieren. Jeder Nutzer hat unterschiedlich viel Erfahrung im Gitarre spielen. Anfänger müssen erst lernen wie eine Gitarre zu spielen ist, während Fortgeschrittene und Profis ihre Fertigkeiten verbessern wollen. Die Software müsste also entweder für eine bestimmte Zielgruppe entwickelt werden oder Funktionen anbieten, welche allen Zielgruppen helfen.

Die grundsätzliche Fragestellung ist jedoch, ob eine Software, welche MIDI-Daten visualisiert, Gitarristen helfen kann Gitarre zu lernen, zu üben oder sich zu verbessern.

1.2 Aufgabenbeschreibung

Die Software soll eine Vielzahl an Anforderungen erfüllen. Der Anwender soll einen Song zum Nachspielen angezeigt bekommen. Dabei soll er in Echtzeit die von ihm gespielten Noten sehen und sie mit einer vorgefertigten „Root“- oder Soll-MIDI-Notenspur vergleichen können. In einem zweiten Schritt soll er nach dem Spielen seine Aufnahme gründlich analysieren können und sie mit der Soll-MIDI-Notenspur, sowie vorherigen gespielten Aufnahmen, vergleichen können, um

Informationen über Fortschritte, Fehler und Verbesserungsvorschläge zu ermitteln. Die Software soll für verschiedene Zielgruppen geeignet sein, wie beispielsweise Anfänger, welche damit ihre ersten Schritte machen, erfahrenere Gitarristen, die ihre Fähigkeiten verbessern wollen, bis hin zu professionellen Gitarristen, welche ein kompliziertes Solo für eine Studioaufnahme vorbereiten müssen.

1.2.1 Ziel

Ziel dieser Arbeit ist das Design und die Implementierung eines Tools, welches die einfache und schnelle Aufnahme von MIDI-Signalen erlaubt, von einer mit einem MIDI-Tonabnehmer ausgestatteten Gitarre. Diese Daten werden dann visualisiert, um einen Nutzer in dessen Lernprozess zu unterstützen.

1.2.2 Softwareteile

Die Software soll sich aus drei Teilen zusammensetzen. Einer Play-Live-View zum Nachspielen von Songs, einem MIDI-Track-Rekorder zur Aufnahme der nachgespielten Songs und einem MIDI-Track-Analyser zur Analyse der Aufnahmen und Dokumentation des Fortschritts.

1.2.2.1 Play-Live-View

Die Nutzeroberfläche soll eine „Root“-MIDI-Notenspur in Tab-Darstellung (mehr unter Abschnitt 3.2.3) zum Nachspielen darstellen, sowie eine Tab-basierte, Echtzeit-Anzeige der vom Nutzer gespielten Noten. Der Modus soll ein Metronom anbieten, an dem sich der Nutzer akustisch orientieren kann. Des Weiteren soll dargestellt werden, wie weit der Nutzer im Song vorangeschritten ist oder an welcher Stelle des Songs sich dieser gerade befindet.

1.2.2.2 MIDI-Track-Rekorder

Die Software bietet ein Recording Interface zur Aufnahme von mindestens 100 Aufnahmen eines Songs über eine große Zeitspanne, sowie einen Algorithmus zur automatischen Bereinigung der MIDI-Aufnahmen.

1.2.2.3 MIDI-Track-Analyser

Die Aufnahmen sollen von der Software über eine größere Zeitspanne gespeichert, verwaltet und analysiert werden können. Dabei soll der Analysemodus eine Tab-basierte Darstellung haben. Ein Vergleich von mindestens 100 Aufnahmen eines Songs mithilfe einer Heatmap (erläutert in Abschnitt 4.1.3.4) soll durch die Software ermöglicht werden. Sie soll des Weiteren eine Visualisierung der Verbesserungsfortschritte über eine gewählte Zeitspanne, z.B. mithilfe eines interaktiven Sliders, anbieten, mit visuellen Statistiken und Zusammenfassungen über Fehleranzahl, Ungenauigkeiten, etc. Stellen die regelmäßig falsch gespielt werden, sollen automatisch ermittelt und hervorgehoben werden. Da die Anzahl der Aufnahmen sehr hoch werden kann, soll dynamisches Filtern aus der

Liste der Aufnahmen ermöglicht werden. Die Wiedergabe und das Nachspielen von ausgewählten Stellen soll angeboten werden, um dem Nutzer die Möglichkeit zu bieten die Problemstellen der Songs so lange zu üben, bis dieser den Song beherrscht.

2 Verwandte Arbeiten

Für einen umfassenden und verständlichen Einblick in MIDI wird der Titel „MIDI: A Comprehensive Introduction“ von Joseph Rothstein empfohlen. Rothsteins Buch bietet eine Übersicht über die MIDI-Spezifikation und Funktionsweise und erläutert darüber hinaus musikalische Grundlagen, Musikproduktionstechniken für Computer und MIDI-Programmierung [Rot95].

Das Buch „Teaching Music with Technology“ von Thomas E. Rudolph bietet eine umfangreiche Übersicht über verschiedene technologisch unterstützte Lehrmethoden zum Unterrichten. Rudolph beschreibt zuerst was Musik ist und in welchem Verhältnis sie zur heutigen Technologie steht. Er erläutert grundlegende Sachverhalte über das Internet, Computer, MIDI, Keyboards, MIDI-Controller und andere elektronische Instrumente. Später wird beschrieben wie Software zur Notation von Musik, zur Anleitung, zur kreativen Musikproduktion oder zum Üben und Verbessern musikalischer Fähigkeiten verwendet werden kann. Zusätzlich gibt er eine Orientierung zur Einrichtung eines eigenen Studios und Multimedia, sowie Internet Applikationen [Rud04].

Fred Kersten veröffentlichte 2004 im Music Educators Journal einen Artikel mit dem Titel „Using MIDI Accompaniments for Music Learning at School and at Home“, in welchem er beschreibt, wie online oder auf Datenträgern zur Verfügung gestellte MIDI-Dateien zur akustischen Begleitung einen Musikschüler bei Übungen unterstützen können und bietet Vorschläge, wie sich solche Ressourcen leichter in den Lehr- und Übungsalltag integrieren lassen. Diese Methoden bieten einem Schüler jedoch nur akustische Unterstützung an, an welcher sich dieser orientieren und mitsingen oder ein Instrument mitspielen kann. Die MIDI-Dateien werden nicht verwendet um dem Schüler neue Techniken beizubringen [Ker04].

Analysen von Gitarrendaten wurden in Ricardo Scholz' und Geber Ramalhos Arbeit „COCHONUT: Recognizing Complex Chords from MIDI Guitar Sequences“ vorgestellt. Sie beschreibt, wie Akkorde innerhalb von MIDI-Aufnahmen einer Gitarre mithilfe von „COCHONUT“ (Complex Chords Nutting) erfolgreich erkannt werden können. Diese vielversprechende Technologie ermöglicht Akkorderkennung durch eine Kombination aus regelbasierter, musterbasierter und harmonisch kontextueller Erkennung [Ric08].

In „Visualization of concurrent tones in music with colours“ von Peter Ciuha, Bojan Klemenc und Franc Solina wurde getestet, wie Töne in Zusammenklängen farblich visualisiert werden können. Dabei wurden Tönen, je nach Klang, Farben zugeordnet, welche dann beim spielen angezeigt werden [Pet10].

Die Eignung von Web-Browsern als MIDI- und Musikapplikation wurde von Lonce Wyse und Srikumar Subramanian untersucht. Ihre wissenschaftliche Arbeit „The Viability of the Web Browser as a Computer Music Platform“ beschreibt das große Potenzial der Web MIDI API und deren Nutzbarkeit. Die Resultate dieser Untersuchung ergaben, dass ein Web-Browser durchaus viele Features

zur Musikproduktion bietet, jedoch aus Aspekten der Soundqualität noch nicht PC-Applikationen ablösen kann, welche hohe Sample-Raten, also Häufigkeit der Abtastung von Signalen in einer bestimmten Zeitspanne, unterstützen [WS14].

Es existieren bereits Softwarelösungen in welchen Gitarrendaten visualisiert werden. Einige dieser Lösungen sind ausschließlich Entertainment orientiert, wie beispielsweise die Videospiele „Guitar-Hero“ von Neversoft oder „Rockband“ von Harmonix Music Systems, bei denen ein Nutzer auf gitarrenförmigen Controllern Tastenkombinationen eingeben muss, um beim Spiel zu gewinnen. Das Videospiele „Rocksmith“ von Ubisoft geht sogar einen Schritt weiter und bringt dem Nutzer interaktiv Gitarre bei, indem dieser eine echte, elektrische Gitarre über ein Kabel an eine Spielekonsole anschließt, welche dann gespielte Audiosignale aufnimmt, verarbeitet und auswertet [Act18][Sys18][Ubi13].

Softwarelösungen wie Guitar-Pro von Arobas Music erlauben dem Nutzer aus MIDI-Dateien Notenblätter oder Tabs für Gitarre zu erstellen und zu editieren. Dies ermöglicht die Visualisierung und Notation von Musik [Mus19].

3 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen erläutert, die MIDITabs zugrunde liegen. Es werden einige fundamentale Konzepte der Musik erklärt, wie unter anderem Töne, Noten oder Akkorde. Des Weiteren werden Tabs, Gitarren und deren Funktionsweise sowie der MIDI-Standard näher beleuchtet.

3.1 Musiktheorie

Musiktheorie ist die systematische Erfassung und Darstellung musikalischer Sachverhalte, wie z.B. Töne, Tonsysteme, Noten, Akkorde oder Harmonien. [Bro11e].

3.1.1 Töne

Töne sind akustische Ereignisse in Form von Schallwellen mit harmonischem Schwingungsverlauf und einer einzigen Frequenz. Sie werden mit akustischen Materialien produziert. Die Frequenz eines Tons macht die Tonhöhe aus. Hier gilt, je höher die Frequenz, desto höher ist der Ton. Die Lautstärke des Tons wird von der Amplitude der Schwingung bestimmt. Je höher die Amplitude ist, desto lauter ist der Ton.

Töne werden in Tonsysteme eingeteilt. Das am weitesten verbreitete ist das diatonische Tonsystem. Es umfasst sieben Tonstufen mit insgesamt zwölf Tönen (sieben Ganztöne und fünf Halbtöne). Sie werden mit den Buchstaben „A“ bis „G“ versehen, wobei Halbtöne mit einem „#“ dargestellt werden. Das diatonische Tonsystem beginnt bei C und endet mit B, bevor es erneut mit C und einer höheren Oktavenzahl beginnt. Die Töne von C1 bis B6 werden in der Tabelle 3.1 dargestellt [Bro11b][Bro11c][Bro11e][Bro11f].

3.1.2 Intervalle

Die Beziehungen von Tönen zueinander, hinsichtlich der Schwingungsfrequenz, entsprechen Verhältnissen, welche in der Musiktheorie Intervalle genannt werden. Ein Schwingungsverhältnis von drei zu zwei stellt beispielsweise eine Quinte dar, eine Tondifferenz von fünf Tonstufen. Eine Oktave, also eine Tondifferenz von acht Tonstufen, ist ein Schwingungsverhältnis von zwei zu eins, also eine doppelt so hohe Frequenz [Bro11c][Bro11e][Bro11b].

Ton	MIDI-Note	Frequenz in Hz	Ton	MIDI-Note	Frequenz in Hz
C1	24	32.703	C4	60	261.63
C#1	25	34.648	C#4	61	277.18
D1	26	36.708	D4	62	293.67
D#1	27	38.891	D#4	63	311.13
E1	28	41.203	E4	64	329.63
F1	29	43.654	F4	65	349.23
F#1	30	46.249	F#4	66	369.99
G1	31	48.999	G4	67	392.00
G#1	32	51.913	G#4	68	415.30
A1	33	55.000	A4	69	440.00
A#1	34	58.270	A#4	70	466.16
B1	35	61.735	B4	71	493.88
C2	36	65.406	C5	72	523.25
C#2	37	69.296	C#5	73	554.37
D2	38	73.416	D5	74	587.33
D#2	39	77.782	D#5	75	622.25
E2	40	82.407	E5	76	659.26
F2	41	87.307	F5	77	698.46
F#2	42	92.499	F#5	78	739.99
G2	43	97.999	G5	79	783.99
G#2	44	103.83	G#5	80	830.61
A2	45	110.00	A5	81	880.00
A#2	46	116.54	A#5	82	932.33
B2	47	123.47	B5	83	987.77
C3	48	130.81	C6	84	1046.5
C#3	49	138.59	C#6	85	1108.7
D3	50	146.83	D6	86	1174.7
D#3	51	155.56	D#6	87	1244.5
E3	52	164.81	E6	88	1318.5
F3	53	174.61	F6	89	1396.9
F#3	54	185.00	F#6	90	1480.0
G3	55	196.00	G6	91	1568.0
G#3	56	207.65	G#6	92	1661.2
A3	57	220.00	A6	93	1760.0
A#3	58	233.08	A#6	94	1864.7
B3	59	246.94	B6	95	1975.5

Tabelle 3.1: Noten und Frequenzen der Oktaven 1-6 [Joe06]

3.1.3 Noten

Noten werden schriftlich mit kreisförmigen Symbolen, auch Notenköpfe genannt, auf einem Liniennraster dargestellt. Die Abbildung 3.1a zeigt eine Tonleiter, welche links bei C beginnt und über eine Oktave zum nächsthöheren C geht. Die Darstellung der Noten, welche entweder als alleinstehende Notenköpfe oder als Notenköpfe mit Notenhälsen und Fähnchen auftreten können, gibt Aufschluss über die Notendauer. Die Abbildung 3.1b zeigt Noten unterschiedlicher Dauer innerhalb des Zeitraums eines Taktes. In der ersten Zeile, auf der linken Seite, ist eine ganze Note abgebildet. Darunter sind zwei Halbe und vier Viertelnoten dargestellt. Auf der rechten Seite sind acht Achtelnoten, 16 $\frac{1}{16}$ - und 32 $\frac{1}{32}$ -Noten abgebildet [Bro11f].

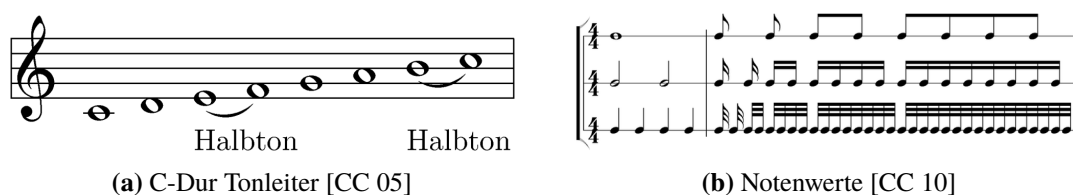


Abbildung 3.1: Notenschreibweise

3.1.4 Akkorde

Ein Akkord ist eine „sinnvolle Verbindung mehrerer Töne zu einem Zusammenklang“ [Bro11a]. Akkorde sind dementsprechend mehrstimmig. Je nach dem aus wie vielen Tönen sie sich zusammensetzen, sind sie zwei-stimmig, drei-stimmig, vier-stimmig, usw. Des Weiteren haben Akkorde ein Tongeschlecht nach Moll und Dur, definiert durch die Intervalle der Noten des Akkordes [Bro11a]. Beispielsweise setzt sich ein drei-stimmiger C-Dur-Akkord aus den Noten C, E und G zusammen, während ein C-Moll-Akkord hingegen aus C, D# und G besteht.

3.2 Gitarre

Gitarren sind Zupfinstrumente mit einem hohlen Korpus, in welchem ein Schallloch eingelassen ist. Von einem Wirbelkasten aus, werden sechs Saiten am Gitarrenhals entlang und über das Schallloch gespannt und an einem Querriegel, welcher auch Steg genannt wird, befestigt. Eine klassische Akustik Gitarre wird in Abbildung 3.2 dargestellt. Wenn die Saiten gezupft werden, versetzt sie das in Schwingung. Die durch die Schwingungen verursachten Schallwellen werden durch den Korpus verstärkt und über das Schallloch ausgegeben.

Wird eine Saite abgegriffen und auf den Gitarrenhals gedrückt, verkürzt dies die Länge der Saite, die über dem Schallloch schwingt. Dies erhöht die Schwingfrequenz und damit die Tonhöhe des gespielten Tons. Am Gitarrenhals sind Querstreben, auch Bündle oder „Frets“ genannt, angebracht, die zum Spannen der Saite dienen und zur Orientierung auf welcher Höhe die Saiten abgegriffen werden müssen. Die Saiten werden meist auf E2, A2, D3, G3, B3 und E4 gestimmt, wobei auch andere Stimmungen möglich sind. Eine klassische Gitarre hat 19 Frets und erreicht somit Töne von E2 bis B5, siehe Tabelle 3.1.



Abbildung 3.2: Klassische Akustik Gitarre [CC 06a]



Abbildung 3.3: Gitarrenhals mit Frets

Gitarren existieren auch in elektrischer Form. Anstatt eines Schalloches haben diese jedoch elektrische Tonabnehmer, welche die Schwingungen aufnehmen, elektrisch verarbeiten und an einen Verstärker senden. Die meisten E-Gitarren haben 22 Frets und erreichen Töne bis D6. [Bro11d][CC 19].

3.2.1 Griffdarstellungen

Die Gitarre ist ein sehr beliebtes Laieninstrument. Viele Gitarristen spielen sie, ohne dabei Noten lesen zu können. Stattdessen bedienen sie sich grafischer Griffdarstellungen, wie Griffbrettdiagrammen oder Tabs, um Akkorde und Melodien zu spielen. Diese Darstellungen visualisieren die Griffweisen der Akkorde, Rhythmen, oder Notenfolgen auf eine, auch für Laien, verständliche Weise.

3.2.2 Griffbrettdiagramme

Eine mögliche Darstellungsform ist die Griffbrettdiagramm-Darstellung. Griffbrettdiagramme, oder auch Akkorddiagramme, sind eine visuelle Repräsentation eines Gitarrengriffs. Es zeigt auf welcher Höhe eine oder mehrere Saiten abgegriffen und gespielt werden müssen, um eine gewünschte Note oder Akkord zu erzeugen. Die Abbildung 3.4 zeigt ein Akkorddiagramm für einen A-Moll Akkord. Es stellt dar, wie beide E-Saiten und die A-Saite normal ohne Fret, die D- und G-Saiten am zweiten Fret und die B-Saite am ersten Fret angespielt werden müssen, um einen A-Moll Akkord zu erzeugen.

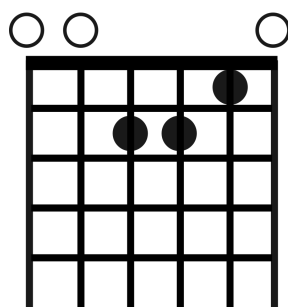
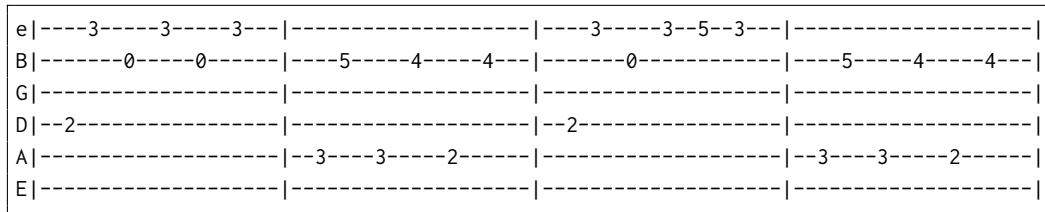


Abbildung 3.4: Griffbrettdiagramm für A-Moll Akkord [CC 06b]

3.2.3 Tab-Darstellung

Griffbrettdiagramme sind eine einfache Methode, um Akkorde nachzuspielen. Jedoch können diese nicht darstellen wie Melodien, also Notenfolgen, zu spielen sind, weswegen eine andere Griffdarstellung benötigt wird. Die am häufigsten verwendete Griffdarstellung ist die Tab-Darstellung. Tabs stellen grafisch sechs Saiten dar, oben angefangen mit der Hohen E Saite, auf denen mit Zahlen notiert ist, auf welchem Fret die jeweilige Saite abgegriffen und angespielt werden muss. Das Listing 3.1 zeigt Tabs einer Melodie. Die Reihenfolge der zu spielenden Noten ist dabei von links nach rechts zu lesen. So wird bei den Tabs in der Abbildung beschrieben, dass zuerst die D-Saite am zweiten Fret gegriffen und gespielt wird. Danach wird auf der hohen E-Saite der dritte Fret gespielt, woraufhin die B-Saite ohne greifen gespielt wird und so weiter.



Listing 3.1: Tab-Darstellung von Road Trippin’ der Red Hot Chili Peppers [Fru17]

3.3 MIDI

„Musical instrument digital interface“, (abgekürzt mit „MIDI“) zu Deutsch „Schnittstelle für digitale Musikinstrumente“, ist ein von Dave Smith erfundener, seit 1982 international verwendeter Standard zur Verarbeitung von Klang- und Steuersignalen einzelner digitaler Musikinstrumente. Darunter fallen Synthesizer, Effektgeräte, Drumcomputer, Soundsampler, digitale Keyboards und viele Weitere. Dieser Abschnitt beschreibt die Funktionalität des MIDI-Protokolls, gibt einen kurzen Einblick in die technischen Hintergründe und spezifiziert das MIDI-Dateiformat [Bro11g].

3.3.1 Funktionalität

Über das MIDI-Protokoll werden Nachrichten von MIDI-Geräten gesendet und empfangen. Diese MIDI-Nachrichten bestehen, je nach Befehlstyp, aus zwei bis drei Bytes, geführt von dem Status-Byte, welches Befehlstyp und einen Kanal spezifiziert. Dieses Byte setzt sich aus zwei Vier-Bit-Datensätzen zusammen, welche „nibbles“ genannt werden. Das „most significant nibble“ (die höchstwertigsten vier Bits) beinhaltet einen Wert in hexadezimal, welcher den MIDI-Befehlstyp spezifiziert. Das „least significant nibble“ beinhaltet ebenfalls einen hexadezimalen Wert, welcher den Kanal festlegt, was die maximale Kanalanzahl auf 16 begrenzt. MIDI-Befehle, wie Notenbefehle beispielsweise, setzen sich aus einem Drei-Byte-Datensatz zusammen, dem Notenbefehl und Kanal, der Note und zuletzt der Anschlagstärke [MMA19]. Die Tabelle 3.2 zeigt ein Beispiel eines solchen Notenbefehls, für das Anschlagen der Note C4. Hexadezimale Werte werden mit „16“ indiziert und dezimale Werte mit „10“.

Byte	Inhalt	Beispielwert (in Hex)	Beschreibung
0	Note-On und Kanal	91	Note-On (9) auf Kanal (1)
1	Note	3C	3C ₁₆ = 60 ₁₀ ist Note C4 nach 3.1
2	Anschlagstärke	7F	7F ₁₆ = 127 ₁₀ ist max. Anschlagstärke

Tabelle 3.2: Beispiel eines Note-On MIDI-Ereignisses der Note C4 bei voller Anschlagstärke

Neben Note-On Befehlen gibt es noch sieben weitere Befehle. Die Tabelle 3.3 beinhaltet alle Befehls- oder „Status“-Bytes mit Befehl und Kanal (hier mit Variable X gekennzeichnet) und deren Datenbytes.

Befehl	Status-Byte	1. Datenbyte	2. Datenbyte
Note-Off	8X	Note	Entlastungsstärke
Note-On	9X	Note	Anschlagstärke
Polyphonic Aftertouch	AX	Note	Tastendruck
Control Change	BX	Controller	Wert
Program Change	CX	Instrument	-
Monophonic / Channel Aftertouch	DX	Tastendruck	-
Pitch Bending	EX	Wert 1	Wert 2
System (exclusive) Message	FX	Steuermeldung	-

Tabelle 3.3: Übersicht MIDI-Befehle mit Status-Bytes

Note-Off Beenden des Spielens von einer Note. Das erste Datenbyte enthält die Note und das Zweite die Stärke, mit der die Taste losgelassen wird.

Note-On Beginnen des Spielens einer Note. Das erste Datenbyte enthält die Note und das Zweite die Anschlagstärke.

Polyphonic Aftertouch Änderung von Tastendruck, während eine Note bereits gespielt wird. Die Datenbytes enthalten die betreffende Note und den neuen Wert des Tastendrucks.

Control Change Änderung eines Controller Werts. Datenbytes enthalten Controller und den Wert, um welchen dieser verändert wird.

Program Change Festlegung eines Instrumentes auf einem Kanal. Dieser Befehl hat nur ein Datenbyte, mit dem Instrument.

Monophonic / Channel Aftertouch Änderung von Tastendruck aller gedrückten Tasten eines Kanals. Hier wird ebenfalls nur ein Datenbyte, mit dem zu ändernden Wert des Tastendrucks, benötigt.

Pitch Bending Viele Keyboards besitzen ein Pitchbend-Wheel, welches die gespielten Noten um einen gewünschten Wert, fließend höher, bzw. tiefer beugt, oder „transponiert“. Das MIDI-Protokoll unterstützt Pitchbending als Befehl und beinhaltet in zwei Datenbytes Werte, um die eine Note gebeugt werden kann.

System (exclusive) Message Beinhaltet Steuermeldungen und Werte in einem Datenbyte.

3.3.2 Technische Übersicht

Über MIDI lassen sich sequenziell Signale unidirektional, mit einer Übertragungsgeschwindigkeit von 31250 Bit/s, versenden. Da das Protokoll nur unidirektional funktioniert, wurden drei verschiedene Arten von Anschlüssen eingeführt [MMA19]:

- **MIDI-In** - Signale an den MIDI-In Anschluss werden vom Gerät aufgenommen und verarbeitet.
- **MIDI-Out** - Signale vom MIDI-Gerät werden über dem MIDI-Out Anschluss versendet.
- **MIDI-Thru** - Signale die im MIDI-In Anschluss eingehen werden unverarbeitet aus dem MIDI-Thru Anschluss versendet.

3.3.3 MIDI Dateiformat

MIDI-Dateien (Dateien mit der Dateierdung „.mid“) ermöglichen das Speichern von Sequenzen von MIDI-Ereignissen und Befehlen in sogenannten „Spuren“. Standard-MIDI-Files (SMF) existieren in drei Formaten [MMA19]:

- **SMF0** - Alle MIDI-Kanäle werden zusammengefasst auf eine MIDI-Spur geschrieben.
- **SMF1** - Jeder MIDI-Kanal erhält eine eigene MIDI-Spur.
- **SMF2** - Jede MIDI-Kanal kann auf jeder Spur Signale verteilen.

3.4 MIDI-Gitarre

Gitarren sind keine digitalen Instrumente wie Keyboards oder E-Schlagzeuge. Sie haben keine MIDI-Interfaces und können keine gespielten Noten als MIDI-Signal übertragen. Um von einer Gitarre gespielte Noten in MIDI-Signale zu konvertieren, müssen die analogen Schwingungen der Saiten digitalisiert werden. Hierzu gibt es verschiedene Ansätze. Eine Möglichkeit wäre ganze Audioaufnahmen einer Gitarre, mithilfe einer Software nachträglich zu analysieren und in eine MIDI-Spur zu konvertieren. Der größte Nachteil dieser Möglichkeit ist jedoch, dass eine Echtzeitverarbeitung der Daten nicht möglich wäre. Dies ließe sich mit einem Audio-zu-MIDI-Konverter umgehen, welcher in Echtzeit gespielte Noten in MIDI-Signale konvertieren, jedoch immer noch nicht unterscheiden kann, welche Noten auf welchen Saiten gespielt wurden oder aus welchen Noten ein Akkord aufgebaut ist. Diese Probleme lassen sich nur umgehen, wenn die analogen Schwingungssignale direkt an jeder einzelnen Saite aufgenommen und in MIDI-Noten konvertiert werden. So wird auch die Aufnahme von Akkordinformationen mehrerer Saiten ermöglicht. Hierzu werden MIDI-Tonabnehmer benutzt. Diese speziellen Tonabnehmer haben sechs Sensoren, welche unter den sechs Gitarrensaiten angebracht werden und dort deren Schwingungen aufnehmen. Ein MIDI-Tonabnehmer konvertiert jedoch nicht zuverlässig, fehlerfrei die analogen Schwingungen der angeschlagenen Saiten in digitale Signale. Die Sensoren des Tonabnehmers nehmen unter Umständen nicht nur die darüber gespannten Saiten wahr, sondern auch die Eigenresonanz der Gitarre oder Schwingungen anderer Saiten. Mit einer Software lassen sich solche Ungenauigkeiten korrigieren.

3.4.1 Fishman Triple Play

Für die Entwicklung und das Testen von MIDITabs wird der „Fishman Triple Play“-MIDI-Tonabnehmer der Firma „FISHMAN“ verwendet. Das Gerät setzt sich aus zwei Einheiten zusammen, dem Tonabnehmer mit integrierter Sendeeinheit und einer USB-Empfangseinheit, links dargestellt in Abbildung 3.5. Wie in derselben Grafik rechts abgebildet, wird der Tonabnehmer am unteren Ende der Gitarre angebracht. Die über ein Kabel verbundenen Sensoren werden unter den Saiten, oberhalb des Stegs, angebracht. Die Empfangseinheit wird dann an eine USB-Schnittstelle in einen PC eingesteckt und ist bereit zur Verbindung.



Abbildung 3.5: FISHMAN Triple Play MIDI-Tonabnehmer [FIS19]

Das Triple Play befindet sich beim Einschalten in einem einfachen MIDI-Modus. Alle Saitensensoren senden auf dem ersten MIDI Kanal. Die mitgelieferte Software „TriplePlay“ erlaubt dem Nutzer Hardwareeinstellungen vorzunehmen, wie Sensorempfindlichkeit oder Kanalauswahl und diese Konfigurationen in „Patches“ zu speichern. Die Einrichtung solcher Patches sind notwendig, um für jede Saite einen eigenen MIDI-Kanal zu konfigurieren. Um in einen dieser Patches zu wechseln, kann der Nutzer die TriplePlay Software verwenden oder alternativ, durch das Gedrückthalten der „C. Up“-Taste, wie in Abbildung 3.6 gezeigt, beim Einschalten in den Hardware-Modus booten. Mit den Pfeiltasten lässt sich nun der gewünschte Patch auswählen.



Abbildung 3.6: Booten im Hardware Modus [FIS13]

4 Entwicklung

Dieses Kapitel beschreibt die Entwicklung von MIDITabs und erläutert zuerst die grundlegende Konzipierung der Software, welche aus der Aufgabenbeschreibung hergeleitet wird. Dabei wird auch der Lernprozess des Nutzers von MIDITabs näher beleuchtet. Nach der Konzipierung folgt der Vergleich und die Wahl der Programmiersprache, bzw. Programmiersprachen, der Entwicklungsumgebung und grundlegenden Technologien. Zuletzt werden einige Implementierungsdetails und die Softwarearchitektur beschrieben.

4.1 Konzipierung

Dieser Abschnitt beschreibt ein erstes Konzept der Software, welches die Anforderungen aus Abschnitt 1.2 erfüllt. Alle hier gezeigten Prototypen sind Mock-Ups.

Die in Abschnitt 1.2.2.1 beschriebenen Features werden in zwei Modi realisiert. Dem „Practice-Modus“ und dem „Performance-Modus“. Der Practice-Modus richtet sich an Nutzer, welche zuerst die Tabs eines ausgewählten Songs erlernen wollen. In diesem Modus werden die einzelnen Akkorde und Tonfolgen, auch Riffs genannt, in Tab-Darstellung abgebildet. Der Nutzer kann nun die Tabs in eigenem Tempo nachspielen und erhält Feedback zu Fehlern und Fortschritten in Echtzeit. Sobald ein Nutzer die Akkorde und Riffs eines Songs im Practice-Modus verinnerlicht hat, kann dieser im Performance-Modus versuchen, den gesamten Song in einem Zug nachzuspielen. In diesem Modus werden auch die MIDI-Track-Recording Features aus Abschnitt 1.2.2.2 implementiert. Die Features aus Abschnitt 1.2.2.3 werden in einem dritten und letzten Modus realisiert, dem „Analysis-Mode“. Darin werden die Aufnahmen, welche im Performance-Modus aufgenommen wurden, aufgelistet und zur Analyse bereitgestellt. Die Software setzt sich somit aus folgenden drei Modi zusammen:

- Practice-Modus
- Performance-Modus
- Analysis-Modus

Die Software bietet somit ein Drei-Schritte-Lernprogramm, illustriert in Abbildung 4.1. Auf das erste Kennenlernen und Üben des Songs, wird das Geübte aufgenommen und schlussendlich im dritten Schritt evaluiert. Schritt zwei und drei werden wiederholt bis der Anwender mit seiner Leistung zufrieden ist. Ein Anfänger beginnt mit dem Practice-Modus. Einige fortgeschrittene Spieler können diesen Schritt sogar überspringen und direkt mit dem Performance-Modus beginnen, da sie geübter darin sind Tabs zu lesen und bereits viele Griffe beherrschen.

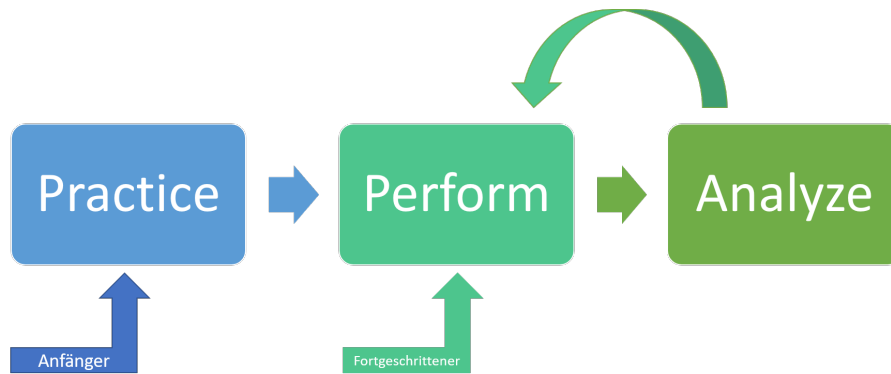


Abbildung 4.1: MIDITabs Drei-Schritte-Lernprozess

4.1.1 Practice-Modus

Der Practice-Modus erlaubt dem Nutzer einen ersten Blick auf die Tabs eines Songs zu werfen und zu versuchen diese nachzuspielen. Die Oberfläche des Practice-Modus setzt sich aus vier Bereichen zusammen: Der Live-Ansicht, welche die derzeit vom Nutzer gespielten Noten anzeigt, der Practice-Ansicht, welche die zu spielenden Tabs anzeigt, einer Metronomsteuerung und einer Statistikanzeige. Die Abbildung 4.2 ist einen Prototyp des Practice-Modus, mit den Tabs aus Abschnitt 3.2.3.

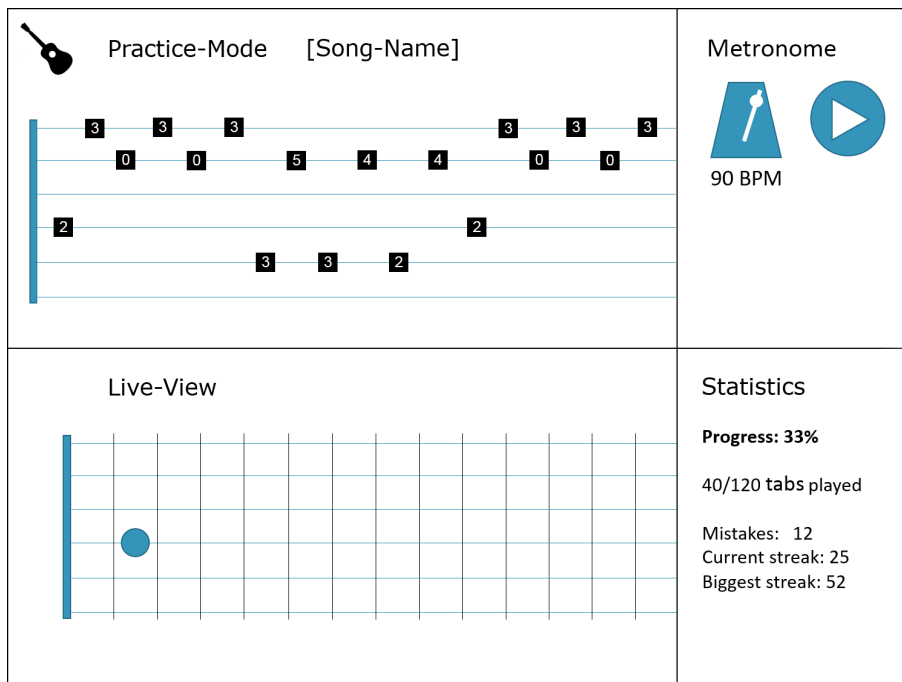


Abbildung 4.2: Prototyp Practice-Modus

4.1.1.1 Live-Ansicht

Die Live-Ansicht bietet eine Echtzeit Darstellung der gespielten Seiten. Es wird ein 6x22 Raster abgebildet, für 22 Frets auf je einer von sechs Saiten. Die oberste Saite ist, wie bei der Tab-Darstellung üblich, die hohe E Saite. Darunter folgen B, G, D, A und die tiefe E Saite. Spielt ein Nutzer nun die D Saite auf dem zweiten Fret, so wird ein Punkt auf der dritten Linie in der zweiten Spalte angezeigt, ähnlich wie in der Prototyp Abbildung 4.2.

4.1.1.2 Practice-Ansicht

Die Practice-Ansicht stellt in Tab-Darstellung den zu spielenden Song dar. Es werden hier, wie in der Live-Ansicht, durch sechs Linien, die Saiten angezeigt, jedoch ohne vertikale Linien, die die Frets darstellen. Stattdessen werden Quadrate auf den Linien abgebildet, welche, wie bei Tabs, Zahlen beinhalten und indizieren, an welchem Fret die betreffende Saite abgegriffen und angeschlagen werden soll. Wird die korrekte Saite richtig gespielt und damit die gewünschte Note erzeugt und erkannt, so wird dem Nutzer visuelles Feedback gegeben, dass die Note richtig gespielt wurde und das Quadrat mit der Fretzahl verschwindet. Die übrigen Tab-Quadrate rücken auf, indem sie nach links wandern und der Nutzer kann nun die nächste Note spielen. Spielt der Nutzer eine falsche Note, so wird ihm das erneut, über visuelles Feedback, angezeigt. Das Tab-Quadrat wird nicht entfernt und bleibt so lange stehen, bis die Note korrekt gespielt wird.

4.1.1.3 Metronom

Die Metronomsteuerung soll die Konfiguration eines Metronoms erlauben, einem akustischen Feature zur Vorgabe eines regelmäßigen Taktes, welches dem Nutzer zur Orientierung dienen soll. Das Tempo des Taktes soll frei wählbar sein.

4.1.1.4 Statistik-Ansicht

In dem Statistikbereich werden für den Practice-Modus relevante Statistiken angezeigt, wie beispielsweise die Anzahl der Fehler, des Songfortschrittes oder der Anzahl korrekt hintereinander gespielter Noten.

4.1.2 Performance-Modus

Sobald der Nutzer die Tabs eines Songs kennen gelernt hat, kann er versuchen sie im Performance-Modus mit vorgegebenem Tempo nachzuspielen. Dieser Modus ist ähnlich aufgebaut wie der Practice-Modus. Er verfügt ebenfalls über eine Live-Ansicht und einen Statistikbereich. Die Performance-Ansicht unterscheidet sich optisch nicht von von der Practice-Ansicht, hat jedoch eine andere Funktionsweise. Im Gegensatz zur Practice-Ansicht, wandern die Tabs kontinuierlich von rechts nach links. Der Nutzer muss die Tabs in Echtzeit nachspielen. Die Metronomsteuerung ist in eine Song-Steuerung integriert in welcher auch der MIDI-Rekorder gesteuert wird. Abbildung 4.3 zeigt einen Prototypen des Performance-Modus.

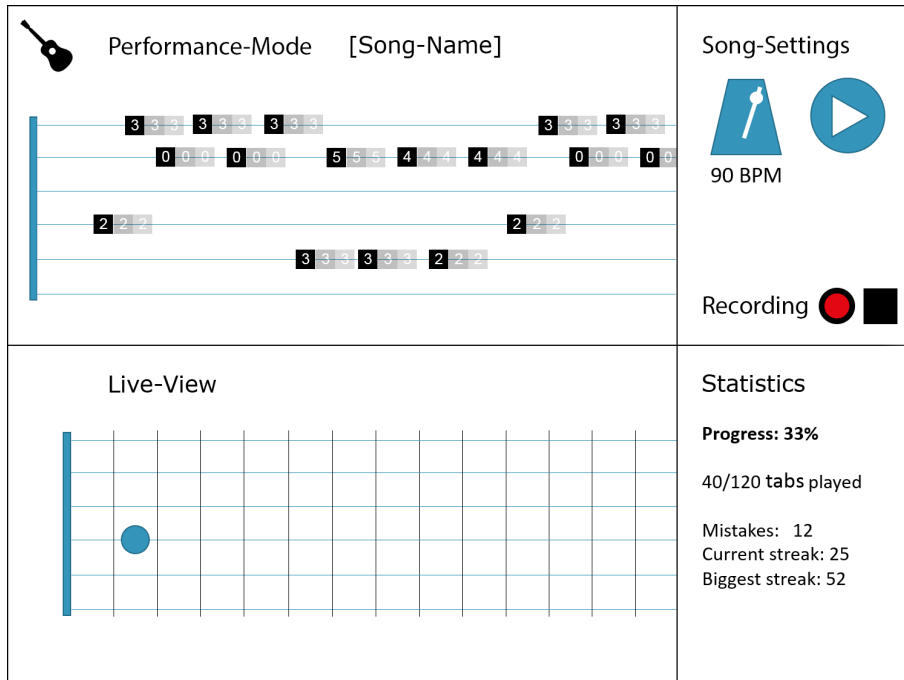


Abbildung 4.3: Prototyp Performance-Modus

4.1.2.1 Live-Ansicht

Die Live-Ansicht des Performance-Modus ist identisch zu der des Practice-Modus. Das Design und die Funktionsweise der Live-Ansicht wird in Abschnitt 4.1.1.1 spezifiziert.

4.1.2.2 Performance-Ansicht

Die Performance-Ansicht ist, ebenso wie die Practice-Ansicht in Abschnitt 4.1.1.2 beschrieben, eine Darstellung der vom Nutzer zu spielenden Tabs. Nach dem Laden des Songs und starten des Performance-Modus, sind die Tab-Quadrate noch ganz rechts auf den Saitenlinien in der Performance-Ansicht und bewegen sich nach links bis zur Spielmarkierung. Der Nutzer muss, sobald das Tab-Quadrat die Markierung erreicht, die richtige Note spielen. Spielt er diese richtig, bekommt er visuelles Feedback. Spielt er eine falsche Note, erhält der Nutzer erneut visuelles Feedback.

4.1.2.3 Songsteuerung

Die Songsteuerung erlaubt dem Nutzer den Song zu pausieren, das Songtempo zu konfigurieren, ein zu dem Song synchron laufendes Metronom ein- oder auszuschalten (siehe Abschnitt 4.1.1.3) und eine MIDI-Aufnahme zu starten oder abzubrechen (siehe Abschnitt 4.1.2.4).

4.1.2.4 MIDI-Rekorder

Der MIDI-Rekorder ist eine interne Softwarekomponente, welche die Performance-Sitzung, oder „Session“, also eine Spielinstanz eines Songs, aufnimmt und zur späteren Analyse speichert. Die Aufnahme soll automatisch mit dem Starten des Performance-Modus beginnen und mit Beenden des Songs aufhören. Über die Songsteuerungsoberfläche sollen jedoch einige weitere MIDI-Rekorder Funktionen angeboten werden, wie das vorzeitige Beenden und Löschen, oder Speichern der Aufnahme, oder das Neustarten des Songs und der Aufnahme.

4.1.2.5 Statistik-Ansicht

Der Statistikbereich des Performance-Modus ist identisch zu dem des Practice-Modus, welcher in Abschnitt 4.1.1.4 spezifiziert ist.

4.1.3 Analysis-Modus

Der Analysis-Modus erlaubt dem Nutzer eine detaillierte Analyse der aufgenommenen Instanzen („Sessions“) eines Songs. Die Oberfläche besteht aus einer Root-Tabs-Ansicht, welche die Root-Tabs des Songs anzeigt, einer Recording-Ansicht, mit den Tabs einer aufgenommenen Session, einer Session-Liste aller aufgenommenen Sessions, diversen Analysetools und Statistiken, sowie einer Funktion für das gezielte Üben von Songstellen. Die Abbildung 4.4 zeigt einen Prototyp des Analysis-Modus.

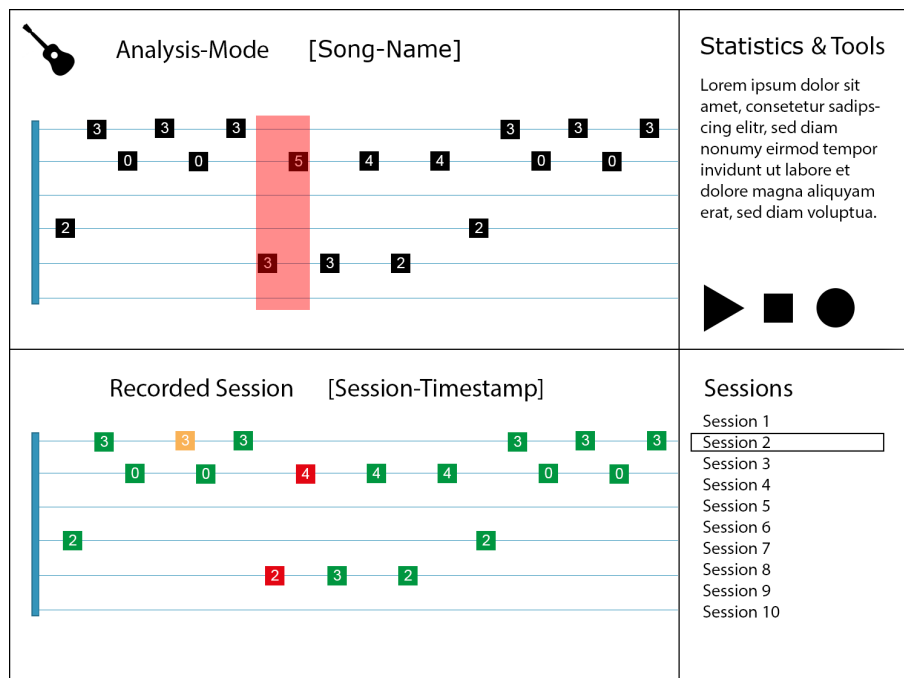


Abbildung 4.4: Prototyp Analysis-Modus

4.1.3.1 Root-Tabs-Ansicht

Die Root-Tabs-Ansicht zeigt, wie die Practice- und Performance-Ansicht aus Abschnitt 4.1.1.2 und Abschnitt 4.1.2.2, die Soll- oder „Root“-Tabs eines ausgewählten Songs. Im Analysis-Modus, sollen die Tab-Quadrate jedoch nicht nachgespielt werden, sondern lediglich dem visuellen Abgleich mit einer aufgenommenen Session dienen. Horizontales Scrollen durch die Tabs ist möglich, da die gesamte Aufnahme nicht auf ein Fenster passt.

4.1.3.2 Recording-Ansicht

Die Recording-Ansicht stellt die Tabs einer aufgenommenen Session dar. Sie indiziert die Korrektheit und Genauigkeit gespielter Tab-Quadrate durch eine farbliche Hervorhebung. Falsche oder fehlende Tabs werden rot markiert, während richtige, aber zeitlich versetzte, Tabs gelb hervorgehoben werden. Korrekt gespielte Tabs werden grün gefärbt. Wie in dem Prototyp in Abbildung 4.4 dargestellt, sind Root-Tabs-Ansicht und Recording-Ansicht übereinander abgebildet. Ebenso wie bei der Root-Tabs-Ansicht, sind nicht die gesamten Tabs der Session zu sehen, weshalb auch durch diese horizontal gescrollt werden kann. Die beiden Tabs sollen jedoch nicht unabhängig voneinander scrollbar sein. Wenn man den zu sehenden Abschnitt der Root-Tabs beispielsweise verschiebt, passt sich auch die Ansicht der Session-Tabs an, sodass immer die Session-Tabs mit den Root-Tabs verglichen werden können.

4.1.3.3 Session-Liste

Alle aufgenommenen und gespeicherten Sessions eines Songs werden mit Zeitstempel in einer durchsuchbaren Liste angezeigt. Wird die gewünschte, anzuzeigende Session ausgewählt, so werden die Session-Tabs in der Recording-Ansicht abgebildet. Über diese Liste können des Weiteren einzelne Sessions gelöscht oder umbenannt werden.

4.1.3.4 Analyse-Tools

Analyse-Tools erlauben dem Nutzer Informationen aus den aufgenommenen Sessions zu gewinnen. Dabei kann dieser einzelne, oder ggf. alle Sessions auswählen und, beispielsweise mithilfe einer Heatmap, die Stellen im Song hervorheben, welche am häufigsten falsch gespielt werden. Durch Markieren gewünschter Sessions, werden Statistiken generiert. Neben den Analyse-Tools wird dem Nutzer die Möglichkeit angeboten, Abschnitte aus den Root-Tabs zu extrahieren und als neue Songs in die Songliste im Hauptmenü einzutragen. So können beispielsweise häufig falsch gespielte Abschnitte gezielt in Practice- und Performance-Modus wiederholt und geübt werden, ohne dabei den ganzen Song nachzuspielen.

4.1.3.5 Statistik-Ansicht

Im Statistikbereich können Statistiken zu den einzelnen Sessions aufgerufen werden. Sind keine spezifischen Sessions in der Session-Liste ausgewählt, so werden globale Statistiken über den gewählten Song angezeigt. Sind eine oder mehrere Sessions ausgewählt, werden in den Statistiken Durchschnittswerte und Fortschritte vermerkt. Die Statistiken umfassen Daten, wie durchschnittliche Fehleranzahl oder Anzahl korrekt hintereinander gespielter Noten, gewählter Sessions.

4.1.4 Zusammenfassung

MIDITabs gliedert sich somit in drei Modi, zugänglich über ein Hauptmenü mit Songauswahl. Der Nutzer kann dort aus einer Liste vorgefertigter Songs einen auswählen oder alternativ einen Song hochladen und dann wählen in welchem Modus er fortfahren will.

4.2 Wahl der Technologien

Die Wahl der richtigen Programmiersprache, der Bibliotheken und der Entwicklungsumgebung, ist ausschlaggebend für den Erfolg des Projektes. Im Bachelorstudium Softwaretechnik werden erste Programmiererfahrungen mit objektorientierten Programmiersprachen gelehrt. Da deswegen bereits Erfahrungen mit C++ vorhanden waren, hatte sich die Realisierung der Software als eine C++ Desktop Applikation angeboten. Als Alternative wurde von Professor Sedlmair eine Webanwendung vorgeschlagen, da dieser bereits einige Abschlussarbeiten betreute, welche die Verarbeitung von MIDI-Daten mithilfe der Web MIDI API behandeln.

4.2.1 Erstellung von Prototypen

Um die beste Option zu wählen, wurden zwei Prototypen erstellt, eine Desktop Applikation und eine Webanwendung. Beide sollten nur sehr rudimentäre Anforderungen erfüllen. Sie sollten die Input-Signale eines generischen MIDI-Controllers lesen und deren Informationen in Echtzeit ausgeben, wie Note oder Control Change, Anschlagstärke oder Kanal. Mithilfe der Prototypen soll dann, nach folgenden Kriterien, die Eignung der Technologie bewertet werden:

- Funktion des Prototyps
- Unterstützte Plattformen
- Entwicklungsumgebung
- Bibliotheken
- Potenzial

4.2.2 Desktop Applikation

Die Desktop Applikation wurde in C++ realisiert. Im ersten Schritt wurde eine Entwicklungsumgebung installiert und eine MIDI-Bibliothek gesucht, welche MIDI-Signale verarbeiten kann. Dafür wurde Visual Studio Community 2017 verwendet und die RtMidi Bibliothek von Gary P. Scavone [Sca19].

Der Prototyp konnte die Anforderungen aus Abschnitt 4.2.1 erfüllen und gibt in einer Konsole die MIDI-Inputs eines generischen MIDI-Controllers wieder. Der Prototyp ist nicht plattformunabhängig. C++ Desktop Applikationen können zwar für Windows, Linux, Unix und MacOS entwickelt werden, jedoch müssen gegebenenfalls einige Justierungen an Code und Entwicklungsumgebung durchgeführt werden, um die geschriebene Software auf ein anderes Betriebssystem zu portieren, als das, für welches die Software ursprünglich geschrieben wurde. Eine funktionierende Windows Desktop Applikation beispielsweise, kann einige Anpassungen erfordern um für Linux oder MacOS exportiert werden zu können. Das Installieren und Einrichten von Visual Studio ist einfach und benutzerfreundlich, jedoch ist der Import und die richtige Einrichtung der RtMidi Bibliothek zeitaufwändig und bedarf Recherche. Visual Studio bietet viele Vorteile, darunter Debugging zur Laufzeit und Syntax Highlighting oder automatische Vervollständigung.

4.2.3 Webanwendung

Die Webapplikation wurde in HTML und JavaScript realisiert. Eine Entwicklungsumgebung war hierzu nicht nötig, da ein einfacher Texteditor ausreicht. Die geschriebenen Dateien werden dann über einen Webbrowser aufgerufen und ausgeführt. Als MIDI-Bibliothek wurde die Web MIDI API verwendet. Der Google Chrome Browser implementiert die Web MIDI API nativ und in vollem Umfang, während andere Browser nur Teile davon und manche nur mithilfe von Add-Ons diese unterstützen [Web19].

Auch dieser Prototyp erfüllt alle Anforderungen aus Abschnitt 4.2.1. Da er über das Web als Plattform läuft und der Chrome Browser, welcher für die Web MIDI API benutzt wird, für Windows, Linux, Unix und MacOS verfügbar ist, wird der Prototyp von allen gängigen Betriebssystemen unterstützt. Die Einrichtung einer Entwicklungsumgebung ist zwar, wie bereits in diesem Abschnitt erläutert, nicht notwendig, jedoch wird empfohlen einen Editor zu verwenden, der Syntax Highlighting und automatische Vervollständigung anbietet. Zur Entwicklung dieses Prototypen wurde Visual Studio Code verwendet, ein vielseitig einsetzbarer Code Editor mit einem integrierten Marktplatz für Erweiterungen, Module, wie unter anderem Compilern, Syntax Highlighting für Sprachen oder UI-Designs. Es wurden lediglich ein HTML und ein JavaScript Syntax Highlighter installiert und die Entwicklungsumgebung war einsatzbereit. Eine Installation einer MIDI-Bibliothek war nicht notwendig, da die Web MIDI API vom Google Chrome Browser bereits nativ unterstützt wird.

4.2.4 Schlussfolgerung

Nach der Entwicklung und Evaluation der Prototypen, fiel die Entscheidung auf eine JavaScript Webanwendung für die Entwicklung von MIDITabs. Beide Prototypen erfüllen zwar die Anforderungen ausreichend, jedoch unterscheiden sich die Vor- und Nachteile der Technologien stark voneinander. Die Tabelle 4.1 bietet eine Übersicht über die Unterschiede.

Kriterium	C++	JavaScript
Funktion des Prototyps	Vollständig	Vollständig
Unterstützte Plattformen	Win, Linux, Unix oder Mac	Web
Entwicklungsumgebung	Compiler und Bibliotheken benötigt	Editor ausreichend
Bibliotheken	Installation notwendig	Installation nicht nötig
Potenzial	Zahlreiche Bibliotheken	Als Website hostbar, D3
Lines of Code (LOC)	54	29

Tabelle 4.1: Prototyp Vergleich

Während die C++ Anwendung für die gewünschte Plattform entwickelt werden kann oder unter Umständen für eine andere Plattform angepasst und portiert werden kann, ist die JavaScript Webanwendung nicht von irgendwelchen Betriebssystemen abhängig. Es wird keine Entwicklungsumgebung für JavaScript benötigt, jedoch wird empfohlen einen umfangreichen Editor dafür zu installieren. Nötig ist lediglich ein Interpreter, wie der des Google Chrome Browsers, um den Code auszuführen. Für die Entwicklung in C++ wird ein Compiler benötigt, der in den meisten C++ IDEs enthalten ist. Des Weiteren ist die Einbindung von Bibliotheken sehr viel umständlicher bei der C++ Anwendung. Während bei JavaScript lediglich eine Zeile Programmcode benötigt wird, muss bei C++ die Bibliothek heruntergeladen, in die Entwicklungsumgebung importiert, konfiguriert und schließlich im Code eingebunden werden. Am auffälligsten ist der Unterschied in den Zeilen Code (LOC), den „lines of code“. Die C++ Anwendung umfasst 54 Zeilen, während der JavaScript Prototyp lediglich 29 benötigt, inklusive des HTML Codes in welchen er eingebettet ist. Der JavaScript Code alleine umfasst lediglich 20 Zeilen, was auf die Effizienz der JavaScript Syntax und der Web MIDI API zurückzuführen ist.

Der größte Vorteil bei einer Webanwendung ist aber, dass sie sich im Internet bereitstellen lässt und damit geräteunabhängig für Nutzer zur Verfügung steht. Dies erhöht die Nutzbarkeit der Software enorm. Ein Musikschüler kann MIDITabs auf diese Weise überall verwenden, wie beispielsweise in einer Musikschule an einem PC oder zu Hause an einem Tablet. Sollte ein Musiklehrer MIDITabs in seine Lehre einbinden, hätte dieser ebenfalls online Einblick in die Lernerfolge des Schülers. MIDITabs hat dadurch als Webanwendung das größte Potenzial, weswegen die Wahl auf diese Technologie fiel.

4.3 Webentwicklung mit HTML, CSS, JavaScript und Node.js

Zur Entwicklung von MIDITabs als Webanwendung wurden vier Webtechnologien verwendet, HTML, CSS, JavaScript und Node.js. HTML steht für Hypertext Markup Language und ist eine Sprache zum strukturellen Aufbau von Webseiten. CSS (Cascading Style Sheets) bestimmt die Darstellung der mit HTML aufgebauten Webseite. Damit lassen sich Inhalte formatieren und einheitlich gestalten. JavaScript ist eine Skriptsprache, die in einem Webbrowser läuft. Mithilfe dieser Sprache lassen sich Funktionen implementieren und in HTML-Dateien einbinden. D3 (Data-Driven Documents) ist eine JavaScript Bibliothek, mit welcher sich Dokumente wie HTML-Dateien datenbasiert verändern und visualisieren lassen. Die vierte und letzte Technologie ist Node.js, eine serverseitige Laufzeitumgebung für JavaScript, welche auf Googles Chrome V8 JavaScript Engine basiert. Sie wurde speziell für die Entwicklung von skalierbaren Netzwerkanwendungen implementiert und ist

aufgrund ihrer asynchronen Ereignisbehandlung sehr effizient. Diese erlaubt nämlich die Verarbeitung vieler Verbindungen zur gleichen Zeit, während andere Technologien die sonst weit verbreitete Ereignisbehandlung mit Threads verwenden [sel19b][sel19a][Eic19][Bos11][Fou19a].

Neben diesen Webtechnologien wurde eine Reihe weiterer, externer Bibliotheken und Ressourcen für die Entwicklung von MIDITabs genutzt. „moment.js“ wurde zur Datums- und Zeitformatierung verwendet. „jQuery“ wurde zur Manipulation und Interaktion mit dem DOM-Tree benutzt. Neben D3 wurde noch „Bootstrap“ zur Visualisierung verwendet. Des Weiteren wurden die externen Node-Module „express“, „body-praser“, „md5-hex“, „midi-parser-js“ und „multer“ verwendet [Luc19][Fou19b][Tea19][Str19][Wil19][Sor19][Col19][Lin18].

4.4 Implementierung

Die Implementierung von MIDITabs beinhaltete fünf Herausforderungen. Das bilden einer Server-Client Architektur zur Datenübertragung, die Erstellung der Softwarearchitektur, die Visualisierung der Tabs sowie deren Analyse und zuletzt der Filterung der Gitarrendaten.

4.4.1 Server-Client Architektur

Da MIDITabs eine Webanwendung ist und der Nutzer Dateien hochladen und verwalten soll, musste eine Server-Client Architektur eingerichtet werden. Darunter versteht man ein Konzept der Aufgabenverteilung innerhalb eines Netzwerks. Der Client, in diesem Fall die MIDITabs Webseite, stellt Anfragen und Aufgaben an einen Server, wie beispielsweise die Abfrage der verfügbaren Songs und der Server bearbeitet und beantwortet diese Anfrage.

Für MIDITabs wurde ein Node.js-Script erstellt, welches HTTP-Requests auf der Root URI auf den „public“-Ordner abbildet und so den Zugriff auf die dortigen HTML-Websites und JavaScripts erlaubt. Unter der Sub-URI /api stellt das Node.js-Script eine API zur Verfügung, über die neue Songs und Aufnahmen hochgeladen und abgefragt werden können. Das JavaScript kann über eine Konsole mit dem Befehl „node MIDITabs.js“ ausgeführt werden, womit dann ein Node-Webserver gestartet wird, welcher unter der lokalen Adresse 127.0.0.1 zu erreichen ist. Wird diese Adresse in einem Webbrowser aufgerufen, so wird die MIDITabs Startseite, welche in Abbildung 5.2 dargestellt ist, angezeigt.

4.4.2 Softwarearchitektur

Die Softwarearchitektur von MIDITabs lässt sich in zwei Teile unterteilen. Der erste Teil behandelt die Verarbeitung der Root-Tabs, während der zweite Teil den Live-Input verarbeitet. Abbildung 4.5 stellt dar, wie MIDITabs die Root-Tabs erstellt.

Die Noten eines Songs werden in eine SMF1-Datei geschrieben (siehe Abschnitt 3.3.3). Es dürfen keine SMF0-Dateien verwendet werden, da diese nur einen Kanal unterstützen und alle MIDI-Nachrichten darauf abbilden. SMF1-Dateien hingegen, bilden die Nachrichten aller Spuren auf

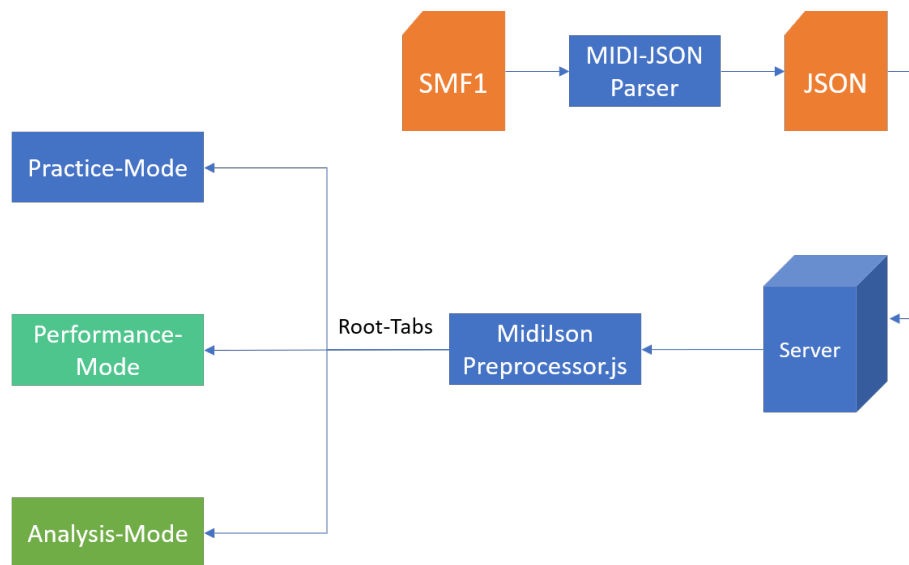


Abbildung 4.5: Softwarearchitektur - Live Tabs

separate Kanäle ab. Bei der Erstellung der SMF1-Dateien müssen alle Noten nach gespielten Saiten separiert werden und auf separaten Kanälen senden. Die Tabelle 4.2 zeigt die zugeordneten Kanäle zu den jeweiligen Saiten.

Saite	Kanal	Least significant nibble des Statusbyte (Hex)
e	CH1	0
B	CH2	1
G	CH3	2
D	CH4	3
A	CH5	4
E	CH6	5

Tabelle 4.2: Input Kanäle MIDI-Tonabnehmer

Beim hochladen in MIDITabs konvertiert ein MIDI-JSON Parser die SMF1-Dateien in JSON-Dateien, welche dann auf dem Server in die Songbank abgelegt werden. Beim Laden eines Songs in einen beliebigen Modus wird die JSON-Datei mithilfe eines MIDI-JSON-Präprozessors zu den Root-Tabs aufbereitet und an die betreffenden Seiten weitergegeben.

Die Verarbeitung des Live-Inputs wird in Abbildung 4.6 veranschaulicht. Der Live-Input wird aufgenommen und die NoteOn-Ereignisse werden an den Filter übergeben. Dieser filtert dann, wie in Abschnitt 4.4.5 beschrieben, die Ereignisse. Die bereinigten NoteOn- und die NoteOff-Ereignisse werden dann in Tabs konvertiert und entweder direkt an den Practice-Modus übermittelt oder, wenn der Performance-Modus verwendet wird, zuvor vom MIDI-Rekorder aufgenommen, bevor dieser die Daten an den Performance-Modus übergibt. Der Performance-Modus gleicht nun die gespielten Tabs mit den vorhandenen Root-Tabs ab und übergibt Informationen an den Rekorder, ob ein Tab korrekt, ungenau, falsch oder gar nicht gespielt wurde. Der Rekorder speichert nun die

Aufnahme mitsamt den Metadaten über die Tabs auf dem Server in Aufnahme-dateien. Diese werden dann mit den Root-Tabs im Analysis-Modus abgerufen. Innerhalb des Analysis-Modus können Songausschnitte generiert, an den Server übermittelt und als separate Songs gespeichert werden.

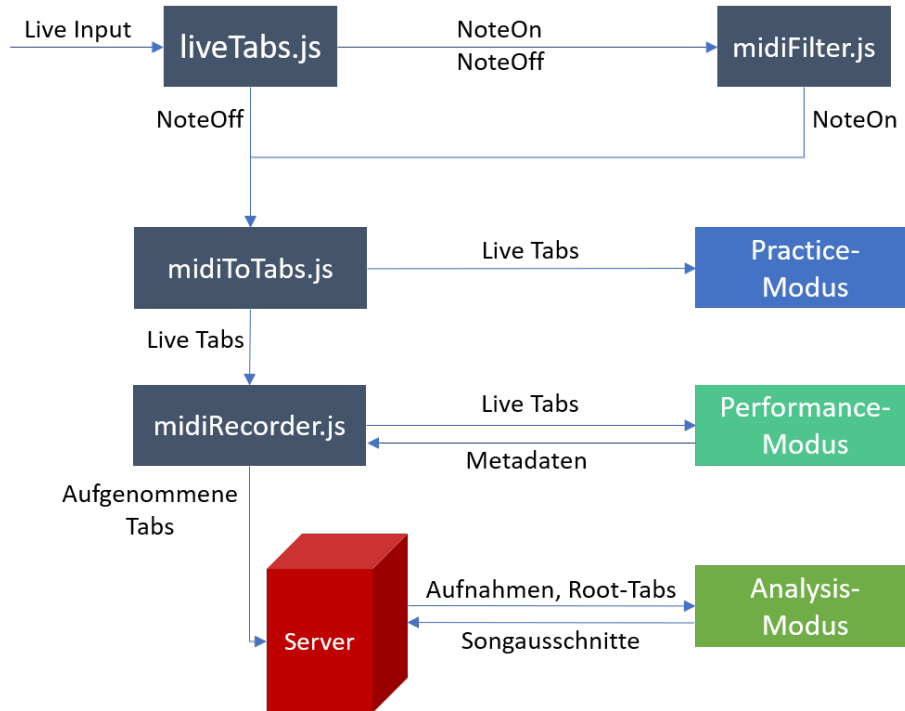


Abbildung 4.6: Softwarearchitektur - Root Tabs

4.4.3 Tab-Visualisierung

Aus den darzustellenden Daten, unabhängig ob bei den Root-Tabs oder bei Aufnahmen, werden diejenigen herausgefiltert welche sichtbar sein sollen. Über D3 wird eine Datenbindung an eine Selektion aus HTML-Elementen erzeugt. Mithilfe dieser kann D3 die Tabs darstellen.

4.4.4 Tab-Analyse

MIDITabs verarbeitet Tabs auf verschiedene Arten, abhängig vom gewählten Modus. Im Practice-Modus wird die Abspielzeit immer dann angehalten, wenn ein Tab zu spielen ist. Wird ein Tab gespielt wird verglichen, ob gerade ein Tab auf dieser Saite zu spielen ist. Wenn einer existiert und die Tabs zueinander passen, fährt der Modus fort und hält die Zeit beim nächsten Tab an. Andernfalls wird dies als Fehler gewertet.

Wenn der Performance-Modus einen gespielten Tab empfängt, vergleicht er diese Note mit der nächsten zu spielenden Note auf derselben Saite. Liegt diese zu weit in der Zukunft, hat man eine unnötige Note gespielt, was als Fehler gewertet wird. Liegt diese im spielbaren Zeitbereich, ist aber eine andere Note als die gespielte, gilt diese Note als falsch gespielt. Stimmen die beiden Noten

allerdings überein, wird entschieden ob sie korrekt oder unpräzise gespielt wurde. Der Performance-Modus erkennt darüber hinaus wenn eine zu spielende Note nicht gespielt wird. Das Ergebnis dieser Analyse wird direkt zum MIDI-Rekorder rückgekoppelt, sodass dieser die Verarbeitung mit in die Aufnahme schreibt.

Der Analyse-Modus kann die Resultate einzeln gespielter Noten direkt aus der Aufnahme auslesen. Dies erlaubt ein korrektes Einfärben der Tabs und eine Neuberechnung der Statistiken für jede Aufnahme. Dies liefert auch die Basis für die Diagramme und die Heatmap.

4.4.5 Filterung der Gitarrendaten

Wie bereits in Abschnitt 3.4 erläutert sind die Daten, welche der MIDI-Tonabnehmer aufnimmt, ungenau. Die Abbildung 4.7 zeigt eine unbereinigte Aufnahme des Tonabnehmers, während Abbildung 4.8 veranschaulicht, wie die Aufnahme eigentlich aussehen sollte. Falsch aufgenommene Noten wurden in der Abbildung rot gefärbt.

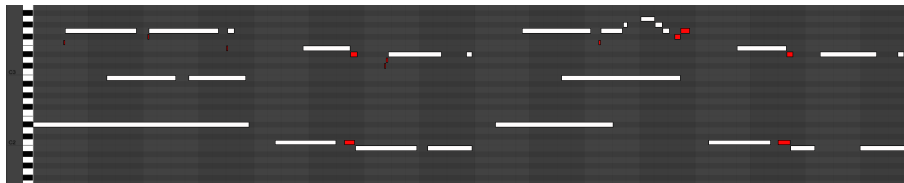


Abbildung 4.7: Aufnahme von „Road Trippin“ mit MIDI-Gitarre [Fru17]

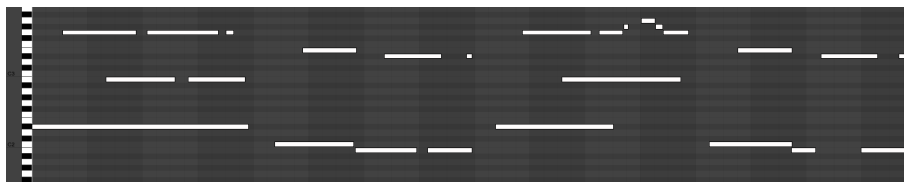


Abbildung 4.8: Bereinigte Aufnahme von „Road Trippin“ [Fru17]

Um diese Ungenauigkeiten herauszukorrigieren, wurde ein Filter implementiert der drei Bereinigungsschritte durchführt.

Velocity-Filterung Wenn eine Note neu angespielt wird, muss das NoteOn-Ereignis eine Mindest-Anschlagstärke haben, ansonsten lässt der Filter das Ereignis fallen.

Zeitentprellung Passiert ein NoteOn-Ereignis den Velocity-Filter, überwacht der Filter den Status der Note. Wenn sie eine gewisse Mindestzeit aktiv ist, sendet der Filter ein NoteOn-Ereignis in die Anwendung weiter. Außerdem werden dann alle weiteren Ereignisse zu dieser Note unverzüglich durchgeschleift bis die Note wieder deaktiviert wird. So können ungewollte Noten die aus einem Rauschen oder Ungenauigkeiten des Abnehmers entstehen und meistens sehr kurz sind, gefiltert werden.

Plausibilisierung Eine gespielte Note muss plausibel spielbar sein. Es gibt einen möglichen Bereich pro Saite, der sich aus der Grundstimmung und der Fret-Anzahl ergibt. Liegt die Note außerhalb dieses Bereiches, wird sie fallen gelassen. E-Gitarren haben in der Regel 22 Frets, weswegen Noten, die tiefer als die Grundstimmung der Saite oder mehr als 22 Halbtöne höher sind, gefiltert werden. Die sich daraus ergebenden Notenspektren lassen sich aus Tabelle 3.1 ableiten.

5 Ergebnis

Nach dem Abschluss der Implementierung von MIDITabs wurde die Software inspiziert und von Gitarristen getestet. In diesem Kapitel werden die Ergebnisse dieser Implementierung und der Tests präsentiert. Zunächst folgt eine Beschreibung der Software. Es werden alle Features erklärt und in Screenshots gezeigt. Danach werden die Testergebnisse der Gitarristen präsentiert und daraus wird eine Schlussfolgerung über die Nutzbarkeit von MIDITabs gezogen. Zuletzt wird in einem Ausblick das Potenzial von MIDITabs erläutert und mögliche Features vorgeschlagen, welche die Software ergänzen würden.

5.1 MIDITabs

Die MIDITabs Webapplikation setzt sich aus vier Webseiten zusammen, der Startseite mit Song- und Modus-Auswahl und je eine Seite für jeden der drei Modi. Daraus ergibt sich die in Abbildung 5.1 dargestellte Sitemap.

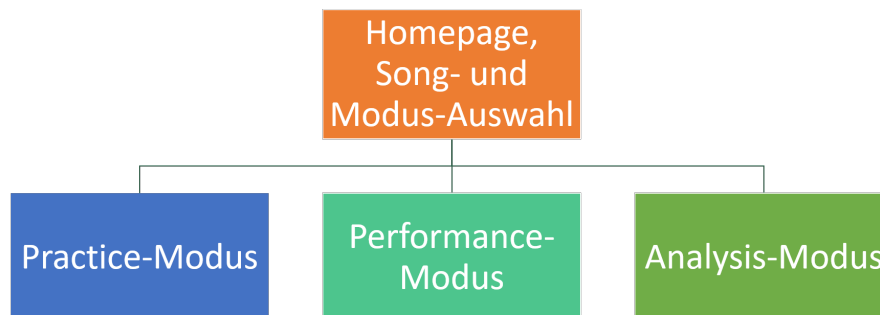


Abbildung 5.1: MIDITabs Sitemap

Um MIDITabs zu starten, muss ein node.js Server aufgesetzt und die Adresse des Servers in einem Chrome Webbrowser unter Port 3000 aufgerufen werden.

5.1.1 Startseite

Die Startseite, hier gezeigt in Abbildung 5.2, ist das Hauptmenü von MIDITabs und setzt sich aus vier Bereichen zusammen. Ganz oben ist der Titel und eine kurze Beschreibung zu MIDITabs. Darunter befindet sich die Songbank, eine scrollbare Auflistung aller verfügbaren Songs. Unter der Song-Bank ist der Upload-Bereich, in dem man eigene Songs in die Song-Bank hochladen kann und darunter ist der vierte und letzte Bereich, die Modus-Auswahl.

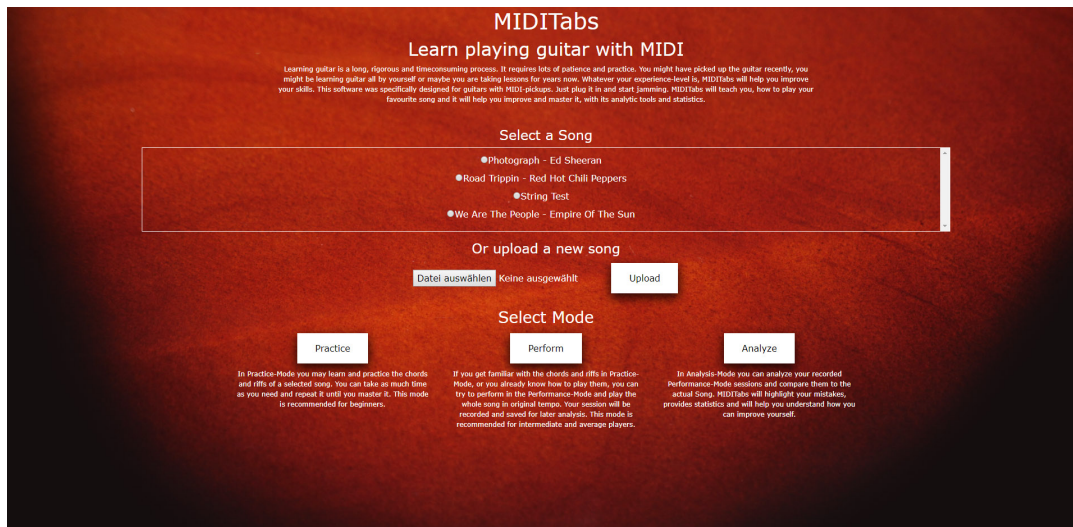


Abbildung 5.2: MIDITabs Startseite

5.1.1.1 Songbank

Zum Testen der Software wurden in die Songbank drei Songs eingefügt, „Photograph“ von Ed Sheeran [She17], „Road Trippin“ von den Red Hot Chili Peppers [Fru17] und „We Are The People“ von Empire Of The Sun [Sun17]. Zusätzlich wurde ein Song mit dem Titel „String Test“ erstellt, welcher zum Testen aller Saiten geeignet ist. Er besteht nur aus sechs Tabs, einen für jede Saite der Gitarre. Alle Songs in der Songbank sind mit einem Radiobutton versehen. Der Nutzer kann durch Klicken auf diesen Button einen Song zum Spielen auswählen.

5.1.1.2 Song-Upload

Der Nutzer kann auch selbst einen Song in die Songbank importieren und spielen, indem er die Upload Funktion im Bereich direkt unter der Songbank verwendet. Hierzu muss dieser erst eine MIDI-Datei auswählen, in dem er auf den Button „Datei auswählen“ klickt. Danach öffnet sich der Datei-Explorer und es kann zum Pfad und der MIDI-Datei navigiert werden. Nach der Auswahl und Bestätigung schließt sich der Explorer und rechts vom „Datei auswählen“-Button erscheint nun der Name der zuvor ausgewählten Datei. Mit einem Klick auf den „Upload“-Button wird die MIDI-Datei aufbereitet und in die Songbank eingefügt. Der Nutzer kann nun den Song darin auswählen und fortfahren.

5.1.1.3 Modus-Auswahl

Der vierte und letzte Seitenbereich beinhaltet die Modus-Auswahl. Ist ein Song in der Songbank über dessen Radiobutton ausgewählt worden, kann nun der gewünschte Modus gewählt werden. Dabei stehen „Practice“, „Performance“ oder „Analysis“ zur Verfügung. Für alle drei Modi existiert jeweils ein Button und eine kurze Beschreibung des jeweiligen Modus. Werden die Modi aufgerufen ohne

davor einen Song ausgewählt zu haben, öffnet sich zwar die Seite, jedoch ohne zu spielende oder analysierende Tabs. Lediglich die Live-Ansicht, sofern in dem gewählten Modus eine vorhanden ist, funktioniert.

5.1.2 Practice-Modus

Abbildung 5.3 zeigt den Practice-Modus mit geladenem Song. Er ist, wie in Abschnitt 4.1.1 definiert, aufgeteilt in vier Sektionen, einer Live-Ansicht, einer Practice-Ansicht, einer Metronomsteuerung und einer Statistik-Ansicht.

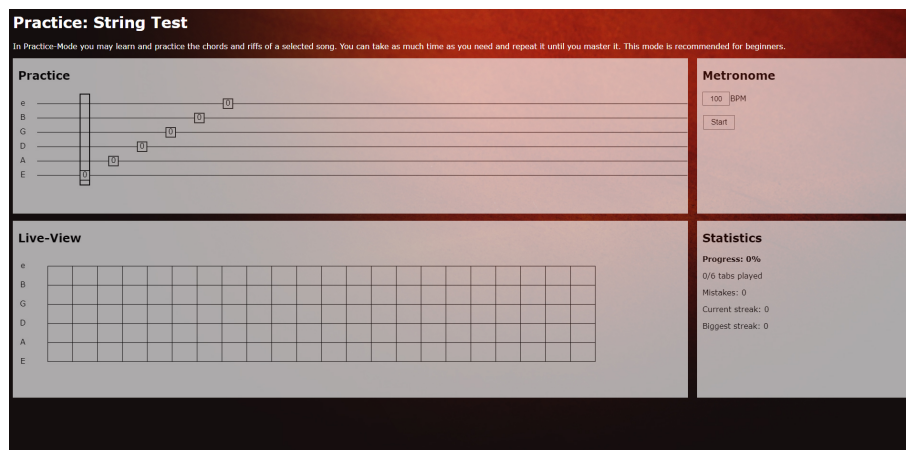


Abbildung 5.3: MIDITabs Practice-Modus

5.1.2.1 Live-Ansicht

Die Live-Ansicht in der linken, unteren Sektion, dargestellt in Abbildung 5.4, gibt in Echtzeit gespielte Gitarrendaten wieder. Es wird ein 6x23 Raster abgebildet, welches einen Gitarrenhals mit sechs Saiten und 23 Frets darstellt. Links vom Raster werden die Grundstimmungen der einzelnen Saiten abgebildet, oben angefangen mit dem hohen E (hier klein „e“ geschrieben), über B, G, D und A, bis zum tiefen E. Wird eine Saite angespielt so erscheint ein kreisförmiger Punkt auf der betreffenden Saite und dem Fret. Wird eine Saite ohne Fret angespielt, so erscheint der Kreis links vor dem Raster, auf Höhe der gespielten Saite.



Abbildung 5.4: Live-Ansicht mit gespieltem E-Moll Akkord

5.1.2.2 Practice-Ansicht

Über der Live-Ansicht befindet sich die Practice-Ansicht, dargestellt in Abbildung 5.5, welche die Tabs des gewählten Songs anzeigt. Es werden sechs horizontale Linien abgebildet, welche Gitarrensaiten darstellen. Links davon sind, wie auch in der Live-Ansicht, die Grundstimmungen der einzelnen Saiten untereinander abgebildet, oben angefangen mit dem hohen, bis zum tiefen E. Alle Tabs sind in kleine Quadrate eingerahmt und nach Spielfolge eingereiht. Links auf den Saitenlinien ist ein schwarzes, hohes Rechteck abgebildet, welches sich vertikal über alle Saiten erstreckt. Dies ist die Spielmarkierung. Wenn ein Song geladen ist, so befinden sich die aktuell zu spielenden Tabs innerhalb dieses Rechteckes.

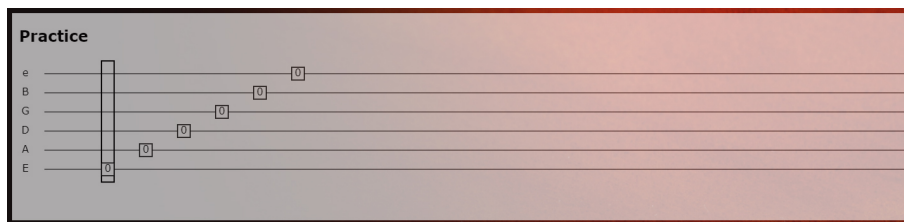


Abbildung 5.5: Practice-Ansicht

Wird nun die richtige Note, welche der Tab darstellt, gespielt, leuchtet die Spielmarkierung, wie in Abbildung 5.6 gezeigt, grün auf und das Tab-Quadrat verschwindet. Die Tabs rücken dann alle automatisch nach links, bis der nächste zu spielende Tab in der Spielmarkierung ist.

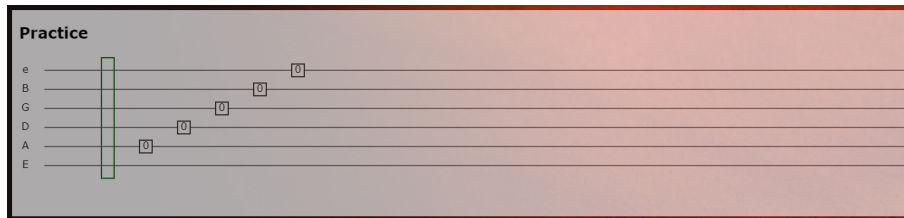


Abbildung 5.6: Richtig gespielter Tab in Practice-Ansicht

Wird die Note, welche der Tab darstellt, nicht korrekt gespielt, leuchtet die Spielmarkierung, wie in Abbildung 5.7 abgebildet, rot auf und das Tab-Quadrat bleibt in der Spielmarkierung. Der Fehler wird in der Statistik (siehe Abbildung 5.9) vermerkt.

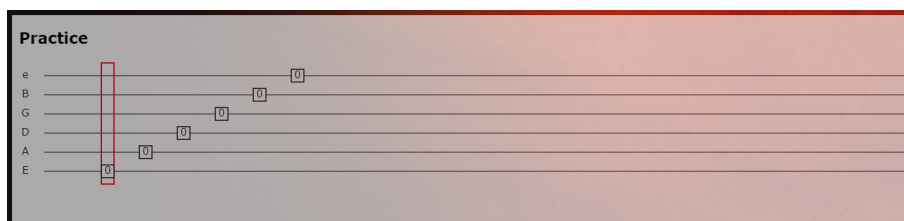


Abbildung 5.7: Falsch gespielter Tab in Practice-Ansicht

5.1.2.3 Metronomsteuerung

Die Sektion rechts oben beinhaltet die Metronomsteuerung, abgebildet in Abbildung 5.8. Über den Start-Button lässt es sich aktivieren und gibt dann, mithilfe eines regelmäßigen, akustischen Signals, einen Takt vor. Standardmäßig läuft das Metronom im Originaltempo des Songs. Da der Practice-Modus jedoch zum üben des Songs und für Anfänger gedacht ist, kann das Metronomtempo über einen Klick in die Textbox und eine Eingabe eines neuen Wertes verändert und so verlangsamt oder beschleunigt werden. Die Eingabe muss durch tippen der „Enter“-Taste bestätigt werden.

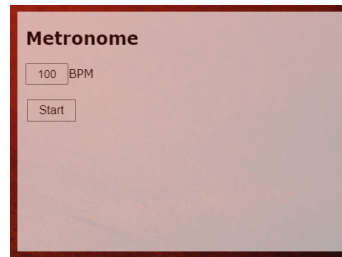


Abbildung 5.8: Metronomsteuerung des Practice-Modus

5.1.2.4 Statistik-Ansicht

Die Statistik-Ansicht, gezeigt in Abbildung 5.9, erlaubt dem Nutzer Informationen über seine Lernfortschritte zu gewinnen. Ganz oben wird der Songfortschritt, also wie viel vom Song bereits gespielt wurde, in Prozent angegeben. Eine Zeile darunter wird gezeigt, wie viele Tabs insgesamt zu spielen sind und wie viele bereits gespielt wurden. Darunter werden alle Fehler aufgezählt und registriert wie viele Noten hintereinander richtig gespielt wurden. Die größte Folge richtig gespielter Noten wird in der letzten Zeile festgehalten. Die Statistiken im Practice-Modus sind nicht persistent. Sobald MIDITabs geschlossen oder die Seite aktualisiert wird gehen diese Daten verloren.

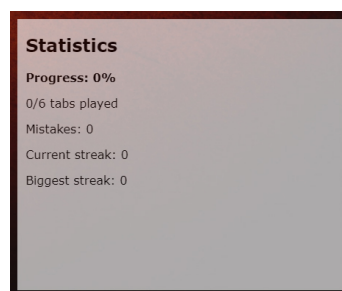


Abbildung 5.9: Statistiken im Practice-Modus

5.1.3 Performance-Modus

Abbildung 5.3 zeigt den Practice-Modus mit geladenem Song. Er ist, wie in Abschnitt 4.1.1 definiert, aufgeteilt in vier Sektionen, einer Live-Ansicht, einer Practice-Ansicht, einer Metronomsteuerung und einer Statistik-Ansicht.

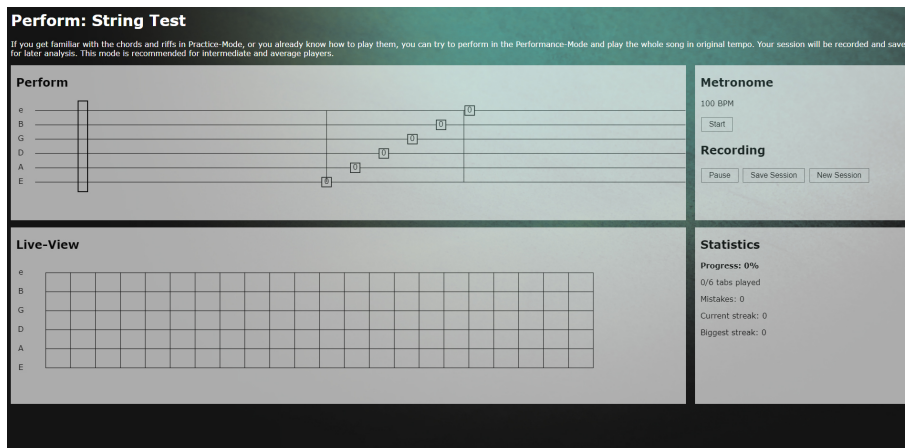


Abbildung 5.10: MIDITabs Performance-Modus

5.1.3.1 Live-Ansicht

Die Live-Ansicht des Performance-Modus ist identisch zu der des Practice-Modus und wird in Abschnitt 5.1.2.1 beschrieben.

5.1.3.2 Performance-Ansicht

Wie im Practice-Modus, befindet sich die Performance-Ansicht über der Live-Ansicht, dargestellt in Abbildung 5.11. Auch diese Ansicht zeigt die Tabs des gewählten Songs auf sechs horizontalen Saitenlinien an. Beim Aufrufen des Practice-Modus sind die Tabs des zuvor gewählten Songs anfangs weit rechts von der Spielmarkierung abgebildet. Sie wandern jedoch kontinuierlich von rechts nach links auf die Spielmarkierung zu. Abbildung 5.11 demonstriert diese Wanderung.

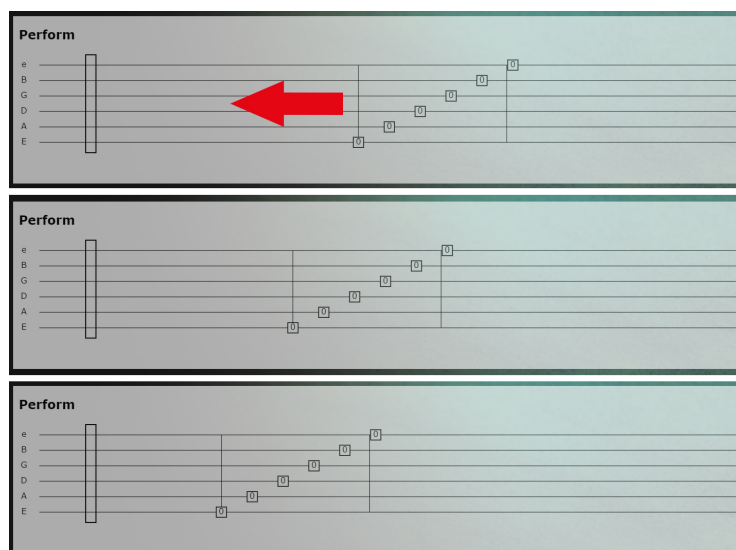


Abbildung 5.11: Tabs in Performance-Ansicht

Sobald ein Tab innerhalb der Spielmarkierung ist, soll dieser gespielt werden. Wird die richtige Note, welche der Tab darstellt, zur richtigen Zeit gespielt, leuchtet, wie in Abbildung 5.12 dargestellt, die Spielmarkierung grün auf und das Tab-Quadrat wird grün gefärbt. Der grün gefärbte Tab wandert dann weiter, bis er ganz links, kurz vor den Saitenstimmungen, verschwindet.

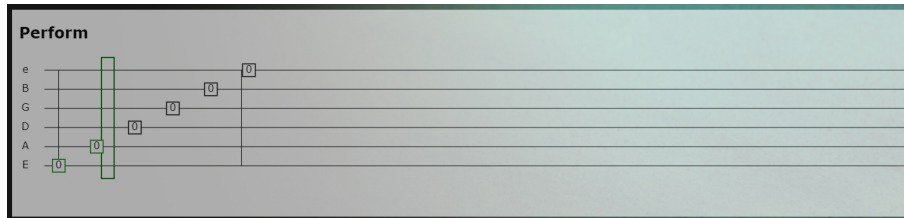


Abbildung 5.12: Richtig gespielter Tab in Performance-Ansicht

Wird eine falsche Note gespielt, leuchtet die Spielmarkierung, wie in Abbildung 5.13 abgebildet, rot auf. Wird ein zu spielender Tab verpasst, färbt sich das betreffende Tab Quadrat rot.

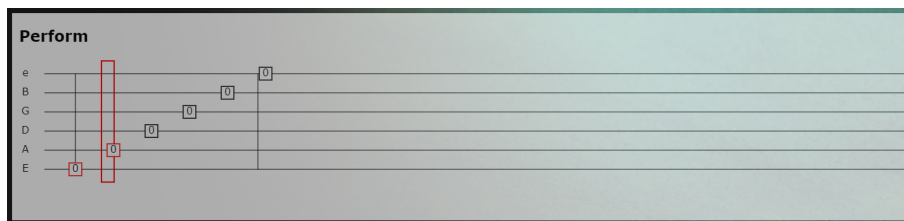


Abbildung 5.13: Verpasster und falsch gespielter Tab in Performance-Ansicht

Wird eine Note minimal zu früh oder zu spät gespielt, blinkt die Spielmarkierung gelb auf und weist den Nutzer darauf hin, dass dieser zeitlich nicht optimal spielt. Das Tab-Quadrat wird jedoch trotzdem grün gefärbt, wie in Abbildung 5.14 gezeigt, da die Note zwar ungenau aber dennoch getroffen wurde.



Abbildung 5.14: Ungenau gespielter Tab in Performance-Ansicht

5.1.3.3 Songsteuerung

Rechts oben befindet sich die Songsteuerung, abgebildet in Abbildung 5.15. Sie ist in zwei Bereiche aufgeteilt. Oben befindet sich die Metronomsteuerung. Darüber lässt sich ein Metronom zuschalten, welches einen an den Song synchronisierten Takt im Originaltempo des Songs vorgibt. Unter der Metronomsteuerung befindet sich die Steuerung des Rekorders. Wenn der Performance-Modus gestartet wird, wird auch automatisch eine Aufnahme gestartet und endet sobald der letzte Tab hinter

der Spielmarkierung verschwindet. Über einen „Play/Pause“-Button lässt sich das abspielen der Song-Tabs, sowie die Aufnahme, pausieren. Durch einen erneuten Klick, laufen die Tabs weiter und die Aufnahme wird fortgesetzt. Über einen „Save Session“-Button kann die Aufnahme vorzeitig beendet und gespeichert werden, woraufhin der Song gestoppt und zurück auf Anfang gesetzt wird. Durch betätigen des „Play/Pause“-Buttons wird der Song und eine neue Aufnahme gestartet. Mithilfe des „New Session“-Buttons kann eine Aufnahme abgebrochen und der Song neu gestartet werden.

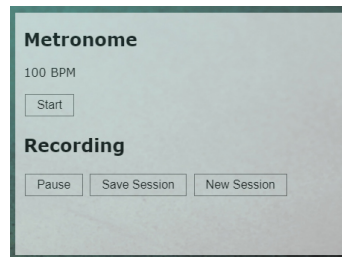


Abbildung 5.15: Songsteuerung des Performance-Modus

5.1.3.4 Statistik-Ansicht

Die Statistik-Ansicht des Performance-Modus ist identisch zu der Statistik-Ansicht des Practice-Modus, welche in Abschnitt 5.1.2.4 beschrieben wird. Die Statistiken des Performance-Modus sind jedoch persistent und werden mit den Aufnahmen abgespeichert.

5.1.4 Analysis-Modus

Die Analysis-Modus, dargestellt in Abbildung 5.16, ist der funktionsumfangreichste aller Modi. Er ist beim Aufrufen aufgeteilt in nur zwei Sektionen, von denen aus alle Funktionen erreichbar sind. In einer Sektion sind die Root-Tabs eines gewählten Songs abgebildet und drei Buttons für die Funktionen des Analysis-Modus. Die zweite Sektion ist die Session-Liste, welche alle Aufnahmen des Songs aus dem Performance-Modus beinhaltet. Die meisten Funktionen des Analysis-Modus können nur genutzt werden, wenn mindestens eine Aufnahme in der Session-Liste ausgewählt wurde.

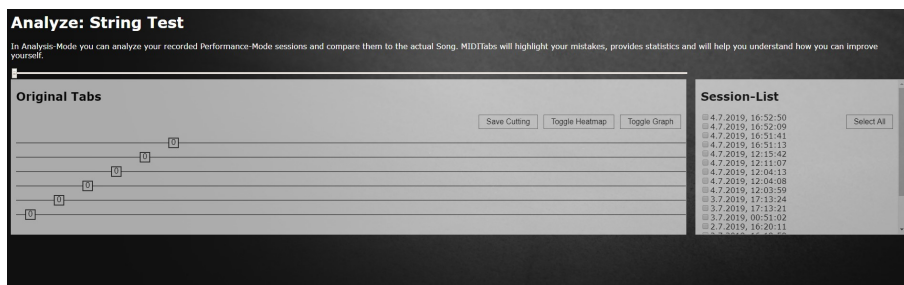


Abbildung 5.16: MIDITabs Analysis-Modus

5.1.4.1 Session-Liste

Die Session-Liste, abgebildet in Abbildung 5.17, beinhaltet alle aufgenommenen Sessions eines Songs. Jede Session verfügt über eine quadratische Check-Box, welche durch einen Klick an- und abgewählt werden kann. Wird eine Session angewählt, so tauchen unterhalb der Root-Tab-Ansicht und Session-Liste die Tabs der Aufnahme und die dazugehörigen Statistiken auf. Sollten mehrere Sessions angewählt werden, dann werden diese alle untereinander aufgelistet. Mit einem Klick auf den „Select All“-Button werden alle Sessions angewählt.

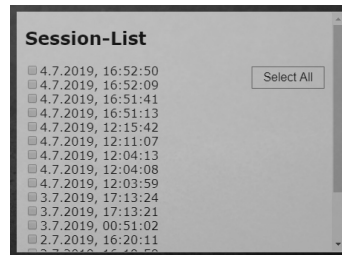


Abbildung 5.17: Session Liste mit aufgenommenen Sessions

5.1.4.2 Root-Tabs-Ansicht

Die Root-Tabs-Ansicht, welche in Abbildung 5.18 dargestellt ist, zeigt die Tabs des zuvor ausgewählten Songs. Mithilfe eines Sliders oberhalb der Tabs lässt sich der sichtbare Ausschnitt des Songs verschieben.

Zwischen Tabs und Slider befinden sich drei Buttons mit den Funktionen des Analysis-Modus'. Mithilfe des „Toggle Heatmap“-Buttons kann der Nutzer sich die Stellen im Song hervorheben lassen, welche am häufigsten von ihm in den zuvor gewählten Sessions falsch gespielt werden. Nähere Informationen zur Heatmap werden in Abschnitt 5.1.4.4 erläutert. Mit dem „Toggle Graph“-Button kann sich der Nutzer Statistiken zu den ausgewählten Sessions in Diagrammen anzeigen lassen. Diese Diagramme werden in Abschnitt 5.1.4.5 beschrieben. Der „Save Cutting“-Button erlaubt dem Nutzer fünfzehn Sekunden des Songs, ab dem sichtbaren Songteil, zu extrahieren und als separaten Song in die Songbank aufzunehmen. So können gefundene Problemstellen im Song gezielt im Practice- und Performance-Modus geübt werden.

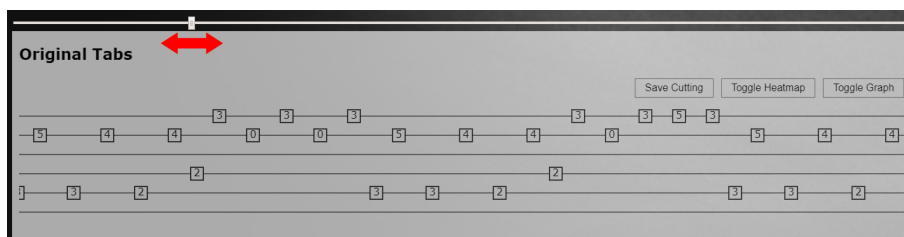


Abbildung 5.18: Root-Tabs im Analysis-Mode (Song „Road Trippin’“ [Fru17])

5.1.4.3 Session- und Statistik-Ansicht

Wurden eine oder mehrere Sessions in der Session-Liste ausgewählt, so tauchen diese unterhalb der Root-Tab-Ansicht und Session-Ansicht untereinander aufgelistet auf. Wie in Abbildung 5.19 dargestellt, stehen auf der linken Seite die gespielten Tabs in der Session-Ansicht und rechts davon einige Statistiken zu der betreffenden Session. Jede Session lässt sich in der Session-Ansicht über den „Delete“-Button löschen. Die Statistiken umfassen Daten wie Fehleranzahl, größte Folge korrekt gespielter Tabs und Anzahl getroffener Tabs. Zur besseren Übersicht verfügt jede Session-Ansicht über den Zeitstempel der betreffenden Session.

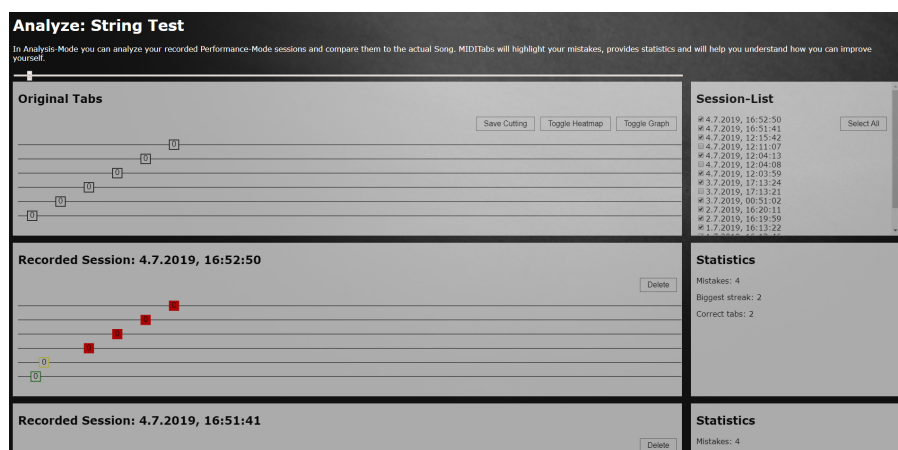


Abbildung 5.19: Aufnahmen und Statistiken unterhalb der Root-Tabs

5.1.4.4 Vergleich von Root-Tabs und Sessions

Da die Root-Tabs und die Session Tabs untereinander abgebildet sind, lassen sich die Aufnahmen anschaulich mit den Tabs des Songs vergleichen. Die Aufgenommenen Tabs der Sessions sind zu diesem Zweck farblich kodiert. Grüne Tabs, wurden korrekt gespielt. Gelbe Tabs gelten als getroffen, aber zeitlich ungenau. Rot umrahmte Tabs sind falsch gespielte Noten und rot gefüllte Tabs stellen fehlende, also nicht gespielte Noten dar. Wird der Slider oberhalb der Root-Tab-Ansicht verschoben, verschieben sich sowohl der sichtbare Song-Ausschnitt der Root-Tabs, als auch die der Aufnahme gleichermaßen. So stehen die korrespondierenden Tabs immer untereinander, egal welcher Song-Ausschnitt angezeigt wird. Wird in der Root-Tab-Ansicht der „Toggle Heatmap“-Button angeklickt, werden die Stellen in den Root-Tabs rötlich hervorgehoben, welche in den Angewählten aufnahmen falsch gespielt wurden. Je rötlicher ein Bereich ist, desto häufiger wurde er falsch gespielt. Diese sogenannte Heatmap und die Farbkodierung der Tabs wird in Abbildung 5.20 veranschaulicht.

5.1.4.5 Diagramm-Ansicht

Durch Klicken des „Show Graph“-Buttons in der Root-Tab-Ansicht wird ein Schaubild oberhalb der Root-Tabs eingeblendet, wie in Abbildung 5.21 dargestellt. Es besteht aus einer X- und einer Y-Achse. Wurden Sessions in der Session Liste ausgewählt, werden deren Statistiken in das Schaubild

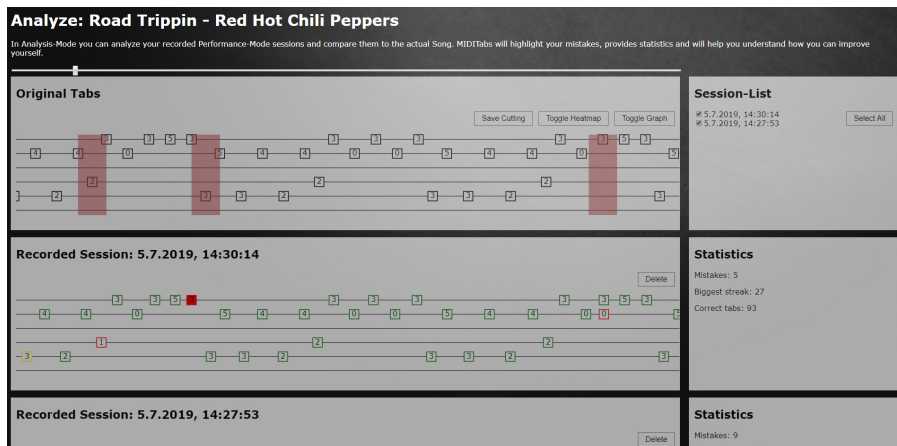


Abbildung 5.20: Heatmap des Analysis-Mode (Song „Road Trippin“ [Fru17])

übertragen und daraus ein Kurvendiagramm gebildet. Über dem Schaubild befinden sich drei Buttons zur Anzeige der Fehler, größten Folgen getroffener Tabs und Gesamtanzahl getroffener Tabs. Wird einer der Buttons geklickt, verändert sich die Kurve und gibt die entsprechenden Werte wieder.



Abbildung 5.21: Diagramm (Fehleranzahl) im Analysis-Mode

5.2 Evaluation

Zur Evaluation wurden alle Features von MIDITabs von zwei Gitarristen getestet. Beide sollten einen Song auswählen und diesen zunächst im Practice-Modus üben, im Performance-Modus spielen und dann im Analysis-Modus die Sessions vergleichen.

Es wurde ein Fragebogen erstellt, um Informationen über die Probanden zu sammeln und deren Beurteilung von MIDITabs nach dem Testen zu erfassen. Der Fragebogen umfasste Fragen darüber, wie lange schon Gitarre gespielt wurde, ob Gitarrenstunden genommen wurden und ob die Probanden Tabs verwenden. Des Weiteren wurden Sie nach ihrer Erfahrung mit MIDITabs befragt. Gefragt

wurde nach Bedienbarkeit, Lieblingsfeatures, Bewertung der Tab-Visualisierung und ob sie die Software privat verwenden würden oder zusammen mit einem Lehrer. Die relevanteste Frage jedoch war, ob die Probanden einen Lernfortschritt registrieren konnten.

5.2.1 Proband 1: Anfängergitarrist

Proband 1 spielte Gitarre seit etwa fünf Monaten und nahm seit vier Monaten Gitarrenunterricht. Er benutzte selten Tabs, da er eher zunächst nur Grundlegende Akkorde lernte und dafür Akkorddiagramme verwendete.

Dem Proband wurde die MIDITabs Homepage gezeigt, woraufhin er selbstständig alle Beschreibungen las und nach eigener Aussage, gut verstanden hat. Er wählte den Song „Photograph“ von Ed Sheeran, welchen er noch nie gespielt hatte und wählte den Practice-Modus. Der Proband hat sofort verstanden, welche Noten zu spielen waren und spielte den Song fünf mal, mit folgenden in Tabelle 5.1 dargestellten Ergebnissen, welche der Statistik-Ansicht entnommen wurden.

Practice-Session	Fehler	Größte Folge korrekter Noten
1	157	36
2	133	51
3	81	74
4	72	93
5	69	95

Tabelle 5.1: Proband 1 Practice-Modus Ergebnisse

An den Ergebnissen lässt sich ablesen, dass mit jedem Durchlauf der Proband weniger Fehler gemacht hatte und die Folgen korrekt gespielter Noten wurden mit jedem Mal größer. Was an den Ergebnissen jedoch nicht abzulesen ist, ist dass er im ersten Durchlauf den Song kaum spielen konnte und durch ständige Unterbrechungen akustisch kein Song erkennbar war. Nach dem fünften Durchlauf jedoch konnte der Proband den Song, selbstständig und ohne auf die Tabs zu achten in kontinuierlichem Tempo spielen.

Im Performance-Modus testete Proband 1 denselben Song und nahm insgesamt fünf Sessions auf. Während der ersten Session hat er Proband Schwierigkeiten festgestellt mit dem Tempo an einer bestimmten Songstelle mitzuhalten, weshalb dieser die betreffende Stelle im Analysis-Modus identifiziert und dann als separaten Song extrahiert hat. Unter Verwendung des Practice-Modus hat dieser die Stelle drei Mal geübt und hat dann vier Sessions im Performance-Modus aufgenommen. Anschließend analysierte dieser die Aufnahmen mit folgenden Ergebnissen, dargestellt in Tabelle 5.2

Der Proband wurde mit jeder Session besser, hat immer mehr Noten getroffen und weniger Fehler gemacht. Nach Aussage des Probanden, hat dieser einen eindeutigen Lernfortschritt feststellen können, da er einen neuen Song beherrschte. Ihm gefiel das Nutzererlebnis und er empfand die Bedienbarkeit von MIDITabs sehr gut. Das Feature zur Extraktion bestimmter Songstellen empfand dieser am hilfreichsten. Obwohl der Proband nur selten Tabs verwendete, kam er mit der Visualisierung sehr gut zurecht. Er würde MIDITabs sowohl im Selbststudium als auch mit dessen Lehrer wiederverwenden. Als Kritik wurde die fehlerhafte Notenerkennung genannt, welche beim spielen von Akkorden auftritt.

Performance-Session	Fehler	Größte Folge korrekter Noten	Insgesamt getroffene Noten
1	558	24	451
2	532	13	437
3	365	15	570
4	291	19	628
5	270	27	654

Tabelle 5.2: Proband 1 Performance-Modus Ergebnisse

5.2.2 Proband 2: Fortgeschrittener Gitarrist

Proband 2 spielte Gitarre seit über vierzehn Jahren und nahm elf Jahre Gitarrenunterricht. Er benutzte nie Tabs, da er als klassischer Gitarrenspieler schon von Anfang ausschließlich mit Noten arbeitete, kann jedoch ohne Probleme Tabs lesen.

Dieser Proband wählte ebenfalls den Song „Photograph“ von Ed Sheeran, welcher auch ihm zuvor unbekannt war und wählte den Practice-Modus. Er hat im Gegensatz zu Proband 1 nur zwei Durchgänge in diesem Modus gebraucht und hat sich von 127 Fehlern in der ersten Session auf 79 Fehler in der Zweiten verbessert. Nach dem zweiten Durchlauf konnte der Proband den Song selbstständig und ohne auf die Tabs zu achten in kontinuierlichem Tempo spielen, woraufhin er in den Performance-Modus wechselte.

Im Performance-Modus spielte Proband 2 insgesamt vier Aufnahmen ein. Anschließend analysierte er die Sessions mit folgenden Ergebnissen, dargestellt in Tabelle 5.3

Performance-Session	Fehler	Größte Folge korrekter Noten	Insgesamt getroffene Noten
1	358	15	601
2	271	28	679
3	266	30	696
4	245	26	705

Tabelle 5.3: Proband 2 Performance-Modus Ergebnisse

Auch Proband 2 wurde mit jeder Session besser, machte immer weniger Fehler und traf mehr Noten. Lediglich in der vierten Session hat die größte Folge korrekt gespielter Noten abgenommen, was aber dem gesamten Verbesserungsfortschritt nicht widerspricht. Nach dessen Aussage, hat auch er einen eindeutigen Lernfortschritt feststellen können, da er einen ihm zuvor unbekanntem Song nun spielen konnte. Er empfand die Nutzeroberfläche von MIDITabs außerordentlich bedienbar und wählte die Heatmap-Funktion der Analyse als sein Lieblingsfeature. Der Proband kam sehr gut mit der Tab-Visualisierung zurecht, obwohl dieser sonst ausschließlich mit Noten arbeitete. Er würde MIDITabs jedoch aus diesem Grund nicht immer verwenden, sondern nur bei sehr schwer zu spielenden Songs, deren Noten sehr kompliziert sind. Würde die Software neben der Tab-Darstellung auch Noten-Darstellung unterstützen, würde er die Software öfter nutzen. Als Wunsch wurde die Einstellbarkeit des Tempos im Performance-Modus gewünscht.

5.3 Schlussfolgerung

Die grundlegende Fragestellung, welche MIDITabs beantworten sollte, war: „Kann die Visualisierung von MIDI-Daten beim Erlernen oder Üben der Gitarre helfen?“ Diese Frage lässt sich mit einem „Ja“ beantworten. Die Probanden haben beide nach nur wenigen Sessions einen neuen Song zu spielen gelernt und sich nachweislich verbessert. Beide Probanden hatten Spaß bei der Benutzung der Software und würden diese weiterempfehlen. Dabei spielte es keine Rolle, ob der Gitarrist erst seit wenigen Monaten oder bereits seit vielen Jahren spielt. MIDITabs ist eine funktionierende Softwarelösung zum Gitarre lernen mithilfe einer Tab-Visualisierung.

5.4 Ausblick

MIDITabs lieferte innerhalb der Evaluierung vielversprechende Ergebnisse. Die Software verfügt jedoch über sehr viel mehr Potenzial. Weitere Softwarekomponenten könnten MIDITabs zu einer sehr umfangreichen Lernsoftware ergänzen.

Derzeit unterstützt MIDITabs nur eine Gitarrenstimmung (E, A, D, G, B, e). Mithilfe einer Erweiterung ließen sich andere, weit verbreitete Stimmungen, wie die Drop-D Stimmung (D, A, D, G, B, e) unterstützen.

Viele Songs wurden mit einem Kapodaster, auch „Capo“ genannt, geschrieben. Dabei handelt es sich um eine Klammer, wie in Abbildung 5.22, welche an einem Gitarrenhals angebracht wird und alle Saiten an einem beliebigen Fret festhält. Dies transponiert den Grundton aller Saiten nach oben, ohne dass dabei die Gitarre umgestimmt werden muss. Mithilfe einer Softwarekomponente ließen sich Songs, welche ein Capo benötigen, unterstützen. Dabei müssen die Tabs in der Practice-, Performance- und Analysis-Ansicht, sowie die Live-Ansicht angepasst und der Fret, an welchem sich der Kapodaster befindet, als neuen Grundton aller Saiten eingestellt werden.



Abbildung 5.22: Kapodaster am zweiten Fret einer Gitarre

MIDITabs unterstützt bisher nur Songs mit Viervierteltakt. Mit einer Erweiterung, könnten auch andere Taktarten unterstützt werden, wie der weit verbreitete Dreivierteltakt.

Linkshänder spielen anders Gitarre als Rechtshänder. MIDITabs unterstützt bisher nur Gitarren- und Tab-Ansichten für Nutzer die mit Rechts spielen. Mithilfe einer Linkshänderfunktion, ließen sich Tabs und Live-Ansicht spiegeln, sodass auch Linkshänder mithilfe von MIDITabs Gitarre lernen können.

Mit zunehmender Songanzahl in der Songbank, wird es immer schwieriger für den Nutzer einen gewünschten Song zu finden. Eine Suchfunktion für die Songbank würde diesen Vorgang beschleunigen und die Software nutzerfreundlicher gestalten. Ebenso wie die Songbank, kann auch die Session-Liste im Analysis-Modus irgendwann sehr lang werden. Auch hier könnte eine Suchfunktion oder dynamische Filterung der Aufnahmen von Vorteil sein. Die dynamische Filterung von Aufnahmen war sogar eine Anforderung aus der Aufgabenbeschreibung in Abschnitt 1.2, welche jedoch aus zeitlichen Gründen nicht umgesetzt werden konnte.

Das Metronom in dem Practice- und Performance-Modus hilft dem Nutzer sich zu orientieren und im Takt zu spielen. Eine noch effektivere Orientierung wäre die akustische Wiedergabe eines Songs mithilfe eines Synthesizers. So könnte der Nutzer begleitend zur Musik mitspielen. Selbiger Synthesizer könnte verwendet werden, um aufgenommene Sessions abzuspielen und Fehler nicht nur visuell, sondern auch akustisch zu identifizieren.

Der Proband aus Abschnitt 5.2.2 erwähnte bei seiner Evaluation, dass er sich im Performance-Modus gewünscht hätte den Song langsamer spielen zu können, weil ihm das kontinuierliche Wandern der Tabs besser gefiel, als das stückchenweise Abarbeiten der Tabs im Practice-Modus. Es ließe sich eine variable Tempoeinstellung im Performance-Modus implementieren, wobei die Idee des Modus dann verfehlt wäre, weil dessen Zweck es ist den Song im Originaltempo nachzuspielen und die Aufnahmen davon später zu analysieren. Es würde die Ergebnisse der Analyse verfälschen, wenn man in einer Session den Song in Originaltempo und ihn dann, in der folgenden Session, sehr viel langsamer spielt und so die Noten leichter trifft. Die bessere Alternative ist den Practice-Modus um eine Funktion zu erweitern, welche mit klicken eines Buttons das Verhalten der Practice-Ansicht so verändert, dass die Tabs genauso wie im Performance-Modus von rechts nach links wandern. Selbiger Proband hatte sich auch eine traditionelle Noten-Visualisierung gewünscht, da dieser nur mit diesen arbeitete. Die Implementierung einer Noten-Visualisierung würde mehr potenzielle Nutzer für MIDITabs bedeuten.

Es ließen sich auch, ähnlich wie bei Videospiele wie Guitar Hero, Punktesysteme und Highscore-Tabellen implementieren, welche einen Entertainment-Aspekt einbringen würden. Gitarristen könnten kompetitiv um die höchsten Punktzahlen spielen und dabei ihre Fertigkeiten verbessern.

MIDITabs ließe sich auch im Gitarrenunterricht mit einem Lehrer einbinden, welcher die Griffe und Techniken erklärt und demonstriert. Der Schüler kann dann die Songs nachspielen und die Ergebnisse mit dem Lehrer analysieren und besprechen. Da MIDITabs eine Webanwendung ist, könnte ein Lehrer sogar online die Ergebnisse seines Schülers einsehen und über einen längeren Zeitraum seine Fortschritte überprüfen.

Zusätzlich ließe sich MIDITabs für mobile Android-Endgeräte optimieren, da Android die Web MIDI API unterstützt [Web19].

Das wohl größte Potenzial dieser Technologie liegt jedoch in der Vielzahl anderer möglicher Instrumente. MIDITabs wurde nur für MIDI-Gitarren konzipiert, aber dieses Konzept lässt sich auf jedes elektronische Instrument, welches MIDI unterstützt, übertragen. Eine Softwareerweiterung für Keyboards könnte Klavierschülern auf die selbe Weise helfen, wie MIDITabs Gitarrenschülern helfen kann. Ebenso lässt sich die Software für E-Schlagzeuge und generische MIDI-Controller erweitern.

MIDITabs hat sehr großes Potenzial und könnte vielen Nutzern das Spielen von verschiedenen Musikinstrumenten beibringen. Ausschlaggebend für den Lernerfolg sind intuitiv verständliche und übersichtliche Visualisierungen von Daten. Mit diesen können Nutzer ihre Fehler verstehen und sich mithilfe der Software verbessern.

Literaturverzeichnis

- [Act18] Activision. *Guitar Hero*. 2018. URL: <https://www.guitarhero.com/de/game> (zitiert auf S. 16).
- [Bos11] M. Bostock. *D3 Data-Driven Documents*. 2011. URL: <https://d3js.org/> (zitiert auf S. 36).
- [Bro11a] Brockhaus. *Brockhaus A-Z Wissen in 12 Bänden, Band 1*. F. A. Brockhaus, 2011, S. 131. ISBN: 978-3-7653-9385-3 (zitiert auf S. 19).
- [Bro11b] Brockhaus. *Brockhaus A-Z Wissen in 12 Bänden, Band 10*. F. A. Brockhaus, 2011, S. 487–491. ISBN: 978-3-7653-9385-3 (zitiert auf S. 17).
- [Bro11c] Brockhaus. *Brockhaus A-Z Wissen in 12 Bänden, Band 3*. F. A. Brockhaus, 2011, S. 738. ISBN: 978-3-7653-9385-3 (zitiert auf S. 17).
- [Bro11d] Brockhaus. *Brockhaus A-Z Wissen in 12 Bänden, Band 4*. F. A. Brockhaus, 2011, S. 265. ISBN: 978-3-7653-9385-3 (zitiert auf S. 20).
- [Bro11e] Brockhaus. *Brockhaus A-Z Wissen in 12 Bänden, Band 7*. F. A. Brockhaus, 2011, S. 298–301. ISBN: 978-3-7653-9385-3 (zitiert auf S. 17).
- [Bro11f] Brockhaus. *Brockhaus A-Z Wissen in 12 Bänden, Band 7*. F. A. Brockhaus, 2011, S. 559–562. ISBN: 978-3-7653-9385-3 (zitiert auf S. 17, 19).
- [Bro11g] Brockhaus. *Brockhaus A-Z Wissen in 12 Bänden, Band 7*. F. A. Brockhaus, 2011, S. 105. ISBN: 978-3-7653-9385-3 (zitiert auf S. 22).
- [CC 05] P.-E. Z. CC BY-SA 2.0 de. *C-Dur Tonleiter*. 15. Juni 2005. URL: <https://commons.wikimedia.org/w/index.php?curid=188120> (zitiert auf S. 19).
- [CC 06a] M. M. CC BY-SA 2.0 de. *Classical Guitar two views2.png*. 2006. URL: https://de.wikipedia.org/wiki/Gitarre#/media/File:Classical_Guitar_two_views2.png (zitiert auf S. 20).
- [CC 06b] C. D. N. C. CC BY-SA 3.0. *Griffbrettdiagramm A-Moll*. 2006. URL: <https://commons.wikimedia.org/w/index.php?curid=1365914> (zitiert auf S. 21).
- [CC 10] j. CC BY-SA 3.0 de. *Notenwerte*. 2010. URL: <https://de.wikipedia.org/w/index.php?curid=5268605> (zitiert auf S. 19).
- [CC 19] W. CC BY-SA 2.0 de. *Gitarre*. Hrsg. von G. Hügler. 7. Jan. 2019. URL: <https://de.wikipedia.org/wiki/Gitarre> (zitiert auf S. 20).
- [Col19] Colxi. *MIDI-Parser-js*. 2019. URL: <https://www.npmjs.com/package/midi-parser-js> (zitiert auf S. 36).
- [Eic19] B. Eich. *Wie Sie JavaScript in Ihrem Browser aktivieren*. 2019. URL: <https://www.enable-javascript.com/de/> (zitiert auf S. 36).
- [FIS13] FISHMAN. *TriplePlay1.4 User Guide*. 2013 (zitiert auf S. 25).

- [FIS19] FISHMAN. *TriplePlay Wireless MIDI Guitar Controller*. 2019. URL: <https://www.fishman.com/fishman-equipped/tripleplay-wireless-guitar-controller-oem/> (zitiert auf S. 25).
- [Fou19a] N. Foundation. *Über Node.js*. Hrsg. von Joyent. 2019. URL: <https://nodejs.org/de/about/> (zitiert auf S. 36).
- [Fou19b] T. jQuery Foundation. *jQuery*. 2019. URL: <https://jquery.com/> (zitiert auf S. 36).
- [Fru17] J. Frusciante. *Road Trippin - Red Hot Chili Peppers (Tabs)*. Hrsg. von K. 4_W4lt3r. 2017. URL: https://tabs.ultimate-guitar.com/tab/red_hot_chili_peppers/road_tripplin_tabs_1184296 (zitiert auf S. 22, 39, 42, 49, 51).
- [Joe06] U. S. Joe Wolfe. *Note names, MIDI numbers and frequencies*. 2006. URL: <https://newt.phys.unsw.edu.au/jw/notes.html> (zitiert auf S. 18).
- [Ker04] F. Kersten. „Using MIDI Accompaniments for Music Learning at School and at Home“. In: *Music Educators Journal* (2004). URL: <https://doi.org/10.2307/3399998> (zitiert auf S. 15).
- [Lin18] J. P. Linus U. Hage Yaapa. *Multer*. 2018. URL: <https://www.npmjs.com/package/multer> (zitiert auf S. 36).
- [Luc19] M. P. Lucas Sanders Isaac Cambron. *Moment.js*. 2019. URL: <https://momentjs.com/> (zitiert auf S. 36).
- [MMA19] M. M. A. (MMA). *The Complete MIDI 1.0 Detailed Specification, Version 96.1*. 2019 (zitiert auf S. 22, 24).
- [Mus19] A. Music. *Guitar-Pro*. 2019. URL: <https://www.guitar-pro.com/en/index.php> (zitiert auf S. 16).
- [Pet10] F. S. Peter Ciuha Bojan Klemenc. „Visualization of concurrent tones in music with colours“. In: *ACMM 10* (2010). URL: <https://dl.acm.org/citation.cfm?id=1874320> (zitiert auf S. 15).
- [Ric08] G. R. Ricardo Scholz. „COCHONUT: Recognizing Complex Chords from MIDI Guitar Sequences“. In: *ISMIR 2008 – Session 1a – Harmony*. 2008 (zitiert auf S. 15).
- [Rot95] J. Rothstein. *MIDI: A Comprehensive Introduction*. 1995. ISBN: 0-89579-309-1 (zitiert auf S. 15).
- [Rud04] T. E. Rudolph. *Teaching Music with Technology*. GIA Publications, 2004. ISBN: 1-57999-313-3 (zitiert auf S. 15).
- [Sca19] G. P. Scavone. *rtmidi*. 11. Jan. 2019. URL: <https://github.com/thestk/rtmidi> (zitiert auf S. 34).
- [sel19a] selfhtml. *CSS*. 2019. URL: <https://wiki.selfhtml.org/wiki/CSS> (zitiert auf S. 36).
- [sel19b] selfhtml. *HTML*. 2019. URL: <https://wiki.selfhtml.org/wiki/HTML> (zitiert auf S. 36).
- [She17] E. Sheeran. *Photograph - Ed Sheeran (Tabs)*. Hrsg. von nick.mcmahon.71. 2017. URL: https://tabs.ultimate-guitar.com/tab/ed_sheeran/photograph_tabs_1743880 (zitiert auf S. 42).
- [Sor19] S. Sorhus. *md5-hex*. 2019. URL: <https://www.npmjs.com/package/md5-hex> (zitiert auf S. 36).
- [Str19] I. StrongLoop. *Express*. 2019. URL: <https://expressjs.com/de/> (zitiert auf S. 36).

- [Sun17] E. O. T. Sun. *We Are The People - Empire Of The Sun (Tabs)*. Hrsg. von AnthonyMazzo. 2017. URL: https://tabs.ultimate-guitar.com/tab/empire_of_the_sun/we_are_the_people_chords_799957 (zitiert auf S. 42).
- [Sys18] H. M. Systems. *Rock Band*. 2018. URL: <http://www.rockband4.com/> (zitiert auf S. 16).
- [Tea19] B. Team. *Bootstrap*. 2019. URL: <https://getbootstrap.com/> (zitiert auf S. 36).
- [Ubi13] Ubisoft. *Rocksmith*. 2013. URL: <https://rocksmith.ubisoft.com/rocksmith/en-us/home/> (zitiert auf S. 16).
- [Web19] M. WebDocs. *Browser compatibility Web MIDI API*. Hrsg. von mdnwebdocsbot. 2019. URL: https://developer.mozilla.org/en-US/docs/Web/API/MIDIAccess#Browser_compatibility (zitiert auf S. 34, 55).
- [Wil19] D. Wilson. *Body-Parser*. 2019. URL: <https://www.npmjs.com/package/body-parser> (zitiert auf S. 36).
- [WS14] L. Wyse, S. Subramanian. „The Viability of the Web Browser as a Computer Music Platform“. In: *Computer Music Journal* (2014) (zitiert auf S. 16).

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift