



Play

Record

IN (GT-1)

Save/Load

# *Tabgenerator:*

# *Gitarrennoten*

# *Visualisieren*

Ben Kleinschnittger

Code: <https://github.com/SuperPirate-ai/TabGenerator>

# Inhaltsverzeichnis:

## Inhalt

Einleitung .....	3
Problemstellung .....	3
Generelle Idee der Umsetzung .....	3
Aufbau .....	4
Open Source .....	4
Grundidee .....	4
Die Fast Fourier Transformation .....	5
Das Userinterface .....	5
Die Umsetzung .....	6
Die Aufnahme einer Note .....	6
Die Audioanalyse .....	6
Berechnung der Notenfrequenz .....	6
Findung der nächstliegenden Notenfrequenz .....	8
Die Visualisierung .....	8
Aufbau der Gitarre und der Tabs .....	8
Bestimmung der Notenposition im visuellen Bereich .....	9
Berechnung der tatsächlich gespielten Notenposition .....	9
Anzeigen der Noten .....	10
Das Speichern und Laden .....	10

## Einleitung

### Problemstellung

Bei einer Gitarre lernt man wie bei vielen Instrumenten mit Noten. Jedoch gibt es bei der Gitarre ein vereinfachtes System, welches speziell auf sie angepasst ist. Es nennt sich Guitartabs oder kurz einfach nur **Tabs**. Um einen Song in Tabs zu deklarieren, muss man, wie beim normalen Notensystem auch, jede Note **einzel**n eintragen. Das ist in etwa wie, wenn man jeden Tag die Zähne putzt. Etwas das allgemein akzeptiert ist, etwas, was jeder für natürlich hält. Oftmals sogar mit einer nicht elektrischen Zahnbürste, obwohl es doch viel praktikabler und genauer wäre, diesen Prozess zu automatisieren. Und genau darum geht es auch in meinem Projekt. Ich möchte ein schon seit Jahrhunderten bekanntes Verfahren automatisieren, um es praktikabler und einfacher zu gestalten.

### Generelle Idee der Umsetzung

Dieses Projekt beschäftigt sich mit der **automatischen** Visualisierung dieser Tabs mithilfe eines selbstgeschriebenen **Computerprogrammes**. Durch dieses Programm soll das Eintragen der Noten automatisiert werden, ohne dass **spezielles Equipment** benötigt wird. Dieses Projekt dient ebenfalls zur Beschäftigung mit der **Audioanalyse**, da dieses Thema eine wichtige Komponente in der heutigen Industrie darstellt.

Für mich ist dies das erste Projekt, das seinen Fokus auf die Audioanalyse legt und ich hatte beim Start dieses Projektes keine ausschlaggebenden Kenntnisse von diesem Fachgebiet der Informatik.

## Aufbau

Zur Umsetzung dieses Projektes nutze ich die **Spiele-Engine Unity** mit der Programmiersprache **C#**. Ich nutze hier eine Spiele-Engine, da ich sehr vertraut mit ihr bin und ich sie für diesen Zweck als sehr praktisch ansehe.

Der analoge Aufbau besteht aus einer **Gitarre**, einem sog. **Preamplifier** (kurz Preamp) und einem **Audiointerface**. Dieses dient als Schnittstelle zwischen Gitarre und Computer.



## Open Source

Das Projekt wird nach der Entwicklung offen auf GitHub<sup>1</sup> zur Verfügung stehen. Dadurch ist eine volle Transparenz bei der Entwicklung und ein freier Zugriff für jeden gegeben. Um so viele Menschen mit diesem Projekt zu erreichen.

## Grundidee

Die Grundidee besteht daraus mithilfe der **Fast Fourier Transformation** (FFT) das Ausgangssignal, in diesem Fall eine gespielte Note, zu analysieren, und somit die **stärkste Frequenz** in einem bestimmten Zeitintervall auszumachen. Dieser Frequenz, welche der gespielten Notenfrequenz entspricht, wird dann einer **Position im virtuellen Notensystem** zugeordnet und für den Nutzer dargestellt. Es besteht die Möglichkeit die gespielten Noten in einer Datei zu **speichern** und diese dann auch wieder zu **laden**.

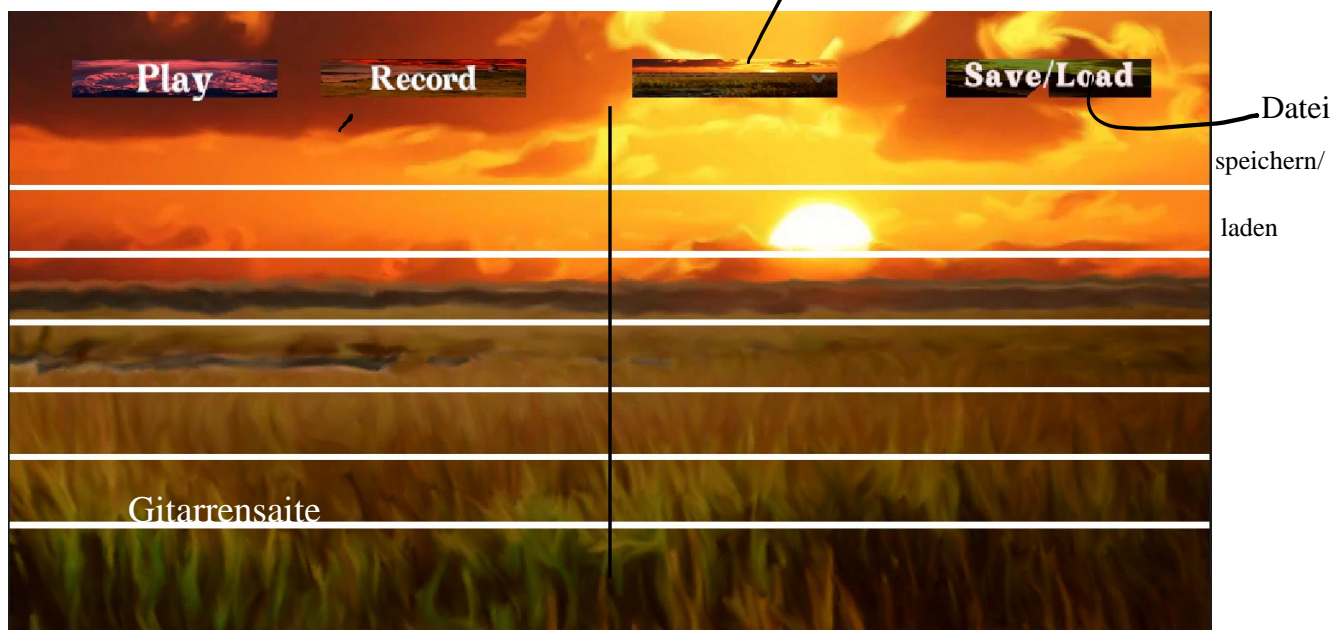
## Die Fast Fourier Transformation

Die **Fast Fourier Transformation**, kurz FFT, ist eine effiziente mathematische Operation zur Berechnung der **Diskreten Fourier-Transformation**. Die FFT ist wesentlich schneller als die herkömmliche Methode, da es mit dem **Teile-und-Hersche** Verfahren arbeitet. Dieses Verfahren teilt im Grunde ein Problem so lange rekursiv in kleine Teile, bis es im Teilproblem lösbar („beherrschbar“) ist. Anschließend wird die Teillösung genutzt, um die Lösung des Gesamtproblems zu rekonstruieren. Im großen Ganzen wandelt die FFT das Ausgangssignal vom **Zeitbereich** in den **Frequenzbereich** um. Wenn man das Signal in einem Graphen zeichnen würde, wäre vor der Kalkulation der FFT die X-Achse die **Zeit** und die Y-Achse die Amplitude, also die Lautstärke, des Signals. Nach der Kalkulation wäre die X-Achse die **Frequenz** und die Y-Achse die Amplitude. In diesem Projekt ist diese Funktion zur Frequenzbestimmung der einzelnen Noten von großem Nutzen. Für die Implementierung dieser Operation besteht die Möglichkeit sie entweder komplett selbst zu implementieren. Da dies aber extrem aufwendig ist, nicht im Sinne dieses Projektes steht und gegen jegliche Programmierkonventionen verstößt, wird in diesem Projekt eine Library genutzt. Genauer gesagt die Accord.Math Library. Die Implementierung der FFT findet man im AudioComponents Skript.

## Das Userinterface

Start/Stopp    Aufnahme starten/stoppen

Auswahl des Audioeingangs



## Die Umsetzung

### Die Aufnahme einer Note

Zur Aufnahme einer Note wird das Signal, welches von der Gitarre „ausgesendet“ wird, zunächst von dem Preamp verstärkt und danach mithilfe des Audiointerfaces auf den Computer übertragen. Da das ankommende Signal endlos weiter geht, also weder einen klar definierten Anfang noch Ende hat, wird es in kleine Samples unterteilt. Diese haben eine Buffer size von 8192 Samples. Bei einer zu kurzen Buffer size besteht die höhere Wahrscheinlichkeit, dass die Frequenz der Note auf zwei Samples aufgespalten ist. Dadurch würde Noise, auf welches später noch einmal genauer eingegangen wird, als eine Note erkannt werden. Die Aufnahme des Signals wird im MicrophoneInput Skript geregelt.

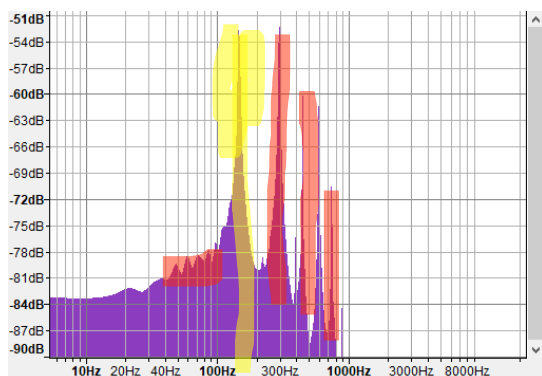
### Die Audioanalyse

Die Audioanalyse findet hauptsächlich im AudioAnalyser und im AudioComponent Skript statt.

#### Berechnung der Notenfrequenz

##### *Berechnung des Frequenzspektrums*

Nachdem das Signal in kleine Samples aufgeteilt wurde, muss nun jedes einzelne Sample analysiert werden, um die Frequenz, der in dem Abschnitt gespielten Note, zu erfassen. Dazu ist es am effizientesten die oben beschriebene Fast Fourier Transformation zu nutzen.



Hier wurde ein D3 mit 145 Hz gespielt. Die gelbe Markierung entspricht der Notenfrequenz

## Noise

Man darf nicht außer Acht lassen, dass die Gitarre kein 100 Prozent reines Signal produziert, weswegen auch viel Noise, das sind eine Art Störfrequenzen, vorhanden ist. Diese sind in der Abbildung rot markiert. Normalerweise kann man das Noise einfach ignorieren, da die Amplitude der Noisefrequenz wesentlich geringer ist als die der gesuchten. Jedoch gibt es Sonderfälle (siehe Abbildung), bei welchen die Noisefrequenz exakt eine Oktave höher ist, und eine ähnliche Amplitude besitzt. Diese Frequenz gilt es zu eliminieren, um eine Falschinterpretation der gesuchten Frequenz zu vermeiden.

## Bestimmung der Hochpunkte

Wie man in der obigen Abbildung erkennt, kann man die gespielte Note anhand ihrer Amplitude (Lautstärke) erkennen. Die Frequenz mit der höchsten Amplitude entspricht meist der Frequenz der gespielten Note. Eine Ausnahme besteht beim oben genannten Sonderfall.

## Saitenanschlagen erkennen

Bis jetzt ist es möglich für jedes Teil die stärkste Frequenz und somit auch die gespielte Frequenz zu bestimmen. Allerdings besteht nun immer noch das Problem, dass **man** eben nicht immer exakt alle 8192 Samples eine neue Note anschlägt. Um diesen Fehler zu beheben, teilen wir jedes Sample nochmal in kleinere Gruppen auf. Für jede Gruppe ermitteln man dann die höchste Amplitude. Nun werden die Amplituden der einzelnen Gruppen verglichen und wenn eine stark heraussteicht, hat dort ein Saitenanschlag stattgefunden. Diese Analyse wird im Zeitbereich durchgeführt, da hier die Amplituden basierend auf ihre Zeit benötigen wird.



## Findung der nächstliegenden Notenfrequenz

Nachdem die Hauptfrequenz des jeweiligen Samples bestimmt und der Saitenanschlag erkannt wurde, muss jetzt die Hauptfrequenz, einer Frequenz auf dem Gitarrenspektrum zugeordnet werden.

Guitar	OPEN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	E	F	F#	G	G#	A	A#	B	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C
	330	349	370	392	415	440	466	494	523	554	587	622	659	698	740	784	831	880	932	988	1047
	B	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C	C#	D	D#	E	F	F#	G
	247	262	277	294	311	330	349	370	392	415	440	466	494	523	554	587	622	659	698	740	784
	G	G#	A	A#	B	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C	C#	D	D#
	196	208	220	233	247	262	277	294	311	330	349	370	392	415	440	466	494	523	554	587	622
	D	D#	E	F	F#	G	G#	A	A#	B	C	C#	D	D#	E	F	F#	G	G#	A	A#
	147	156	165	175	185	196	208	220	233	247	262	277	294	311	330	349	370	392	415	440	466
	A	A#	B	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C	C#	D	D#	E	F
	110	117	123	131	139	147	156	165	175	185	196	208	220	233	247	262	277	294	311	330	349
	E	F	F#	G	G#	A	A#	B	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C
	82	87	92	98	104	110	117	123	131	139	147	156	165	175	185	196	208	220	233	247	262

**Note and frequency in Hertz (approx.) for standard tuning.**

[Bild1]

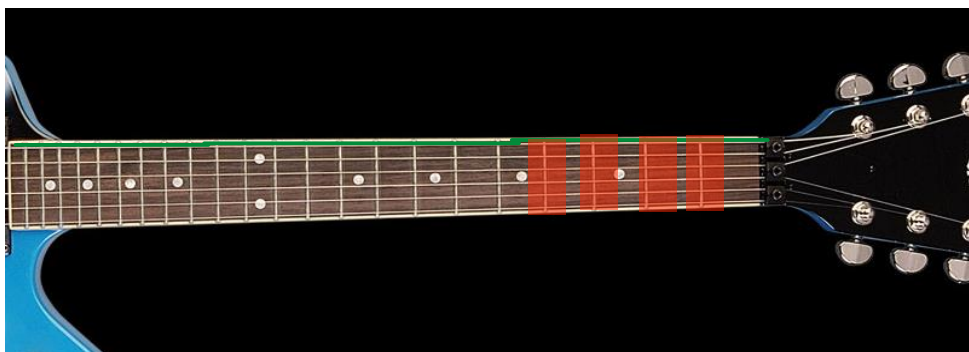
Hierbei wird analysiert, welche Frequenz auf dem Gitarrenspektrum der Hauptfrequenz am nächsten, unter Einberechnung des FFT-Fehlers, liegt. Dieser berechnet sich mit folgender Formel:

$$\text{FFTError} = \text{Samplerate} / \text{Buffersize}$$

Die Samplerate beträgt 44,1 kHz.

## Die Visualisierung

Aufbau der Gitarre und der Tabs

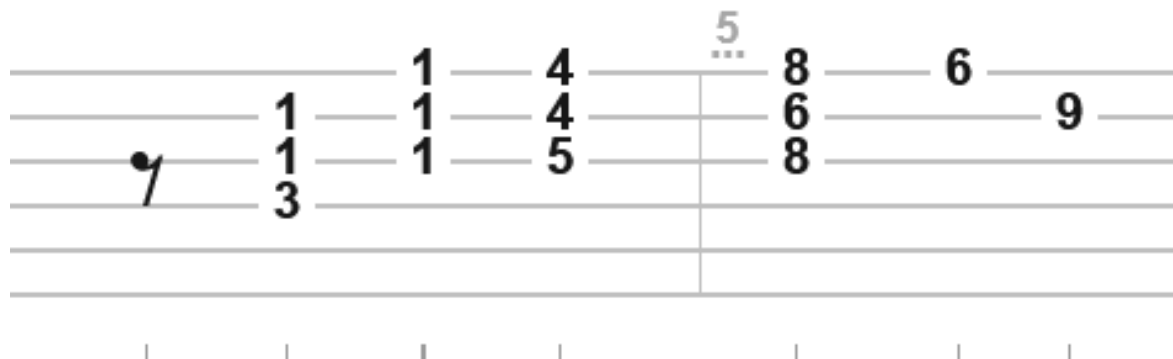


[Bild2]

Der Gitarrenhals ist in sogenannte Bünde aufgeteilt. Diese sind jeweils mit Bundstäbchen (rot) getrennt. Außerdem hat er noch sechs Saiten (grün). Jeder Bund hat einen Halbton



Abstand zum nächstgelegenen Bund. Also auf der obersten Saite, welche leer gespielt einem E entspricht, haben wir dann im ersten Bund ein F und im zweiten ein F# und so weiter.



[Bild3]

Im Tabs-System gibt es ebenfalls diese sechs Saiten, als Striche dargestellt. Allerdings sind diese genau andersherum als auf der Gitarre. Also die höchste Saite ist oben und die tiefste unten. Die Bünde werden als Nummern auf der jeweiligen Saite dargestellt.

#### Bestimmung der Notenposition im visuellen Bereich

Mithilfe der vorher ermittelten Frequenz ist es nun möglich die dazu passenden Positionen auf der Gitarre zu bestimmen. Da die ermittelte Frequenz häufig mehr als einmal vorkommt, werden mehrere mögliche Positionen ermittelt, die auf der Gitarre gespielt worden sein könnten. Zum Beispiel kann man das G4 auf der G-Saite im 12. Bund oder auch auf der D-Saite im 17. Bund gespielt werden.

#### Berechnung der tatsächlich gespielten Notenposition

Nun herauszufinden, wo tatsächlich die Note gespielt worden ist, gestaltet sich etwas schwierig. Denn wir können ohne eine zweite Quelle, wie zum Beispiel eine Kamera, nicht so einfach sagen, wo die Note gespielt worden ist, da jede Notenposition exakt derselben Note und somit derselben Frequenz entspricht. Eine praktikable und einfache Möglichkeit besteht darin, von allen möglichen Notenpositionen diejenige zu wählen, die räumlich gesehen am nächsten zur Vorherigen steht. Allerdings können sich so sehr leicht Fehler einschleichen, da die nächstliegende Position nicht unbedingt immer die am einfachsten zu

spielende Position für den Gitarristen sein muss. Wenn man es sehr genau machen möchte, besteht außerdem die Möglichkeit, saitenspezifische Audiomerkmale, wie zum Beispiel bestimmte Frequenzen, die nur auf einer bestimmten Saite vorkommen, zu identifizieren und anhand derer, die eigentlich identischen Noten voneinander zu differenzieren.

### Anzeigen der Noten

Nun wird die passende Notenposition als Notenobjekt dargestellt und mit einer bestimmten Geschwindigkeit abgespielt. Diese ist abhängig von den Beats per Minute (kurz: BPM).

### Das Speichern und Laden

Zusätzlich besteht die Option, die gespielten Noten in einer Datei zu speichern und diese dann zu einem anderen Zeitpunkt wieder abzurufen. Momentan ist dies nur in einem programmeigenen Format möglich. Dies soll noch so weit ausgeweitet werden, dass die Noten in der gebräuchlichen Form als .gp5 Datei abgespeichert werden können und somit mit den regulären Programmen, wie GuitarPro und Songsterr, nutzbar sind.

## Quellen

[Bild1] <https://i.pinimg.com/originals/6c/3e/7b/6c3e7bb6076c2c9eff8918a28bd6e67e.jpg>

[Bild2] [http://www.edroman.com/guitars/dean/dimebag/dean\\_from\\_hell\\_cfh\\_enlarged.jpg](http://www.edroman.com/guitars/dean/dimebag/dean_from_hell_cfh_enlarged.jpg)

[Bild3] <https://www.songsterr.com/a/wsa/george-benson-song-for-my-father-tab-s417389>