

*Tapgenerator: Gitarrennoten in visueller Brillanz*

Ben Kleinschnittger

Website: <https://thetapgenerator.000webhostapp.com>

**Inhaltsverzeichnis:**

1. Einleitung……………………………………………………3
2. Aufbau
3. Open Source
4. Grundidee
5. Die Fast Fourier Transform
6. Kleiner Überblick des Userinterfaces
7. Die Umsetzung
   1. Die Aufnahme einer Note
   2. Die Audioanalyse
      1. Berechnung der Notenfrequenz
         1. Berechnung des Frequenzspektrums
         2. Noise
         3. Bestimmung der Peaks im FFT
      2. Findung der nächstliegenden Notenfrequenz
   3. Die Visualisierung
      1. Bestimmung der Position im visuellen Bereich
      2. Anzeigen der Noten
   4. Das Speichern und Laden
8. **Einleitung**

Problemstellung:

Bei der Gitarre lernt man wie bei vielen Instrumenten mit Noten. Jedoch gibt es bei ihr ein vereinfachtes System, welches speziell auf die Gitarre angepasst ist. Es nennt sich Guitartaps oder kurz einfach nur Taps. Um einen Song in Taps zu deklarieren, muss man, wie beim normalen Notensystem auch, jede Note **einzeln** eintragen.

Generelle Idee der Umsetzung

Dieses Projekt beschäftigt sich mit der **automatischen** Visualisierung dieser Taps mithilfe eines selbstgeschriebenen **Computerprogrammes**. Mit diesem Programm soll das Eintragen der Noten automatisiert werden. Dieses Projekt dient ebenfalls dazu, sich mit der **Audioanalyse** zu beschäftigen, da dieses Thema eine wichtige Komponente in der heutigen Industrie darstellt.

Für mich ist dies das erste Projekt, welches sich mit der Audioanalyse beschäftigt und ich hatte beim Start dieses Projektes keine ausschlaggebenden Kenntnisse von diesem Fachgebiet der Informatik.

1. **Aufbau**

Zur Umsetzung dieses Projektes nutze ich die **Spiel-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.







1. ***Open Source***

*Das Projekt wird nach der Entwicklung offen auf GitHub1 zur Verfügung stehen. Dadurch ist* eine volle Transparenz bei der Entwicklung und ein freier Zugriff für jeden gegeben.

1. **Grundidee**

Die Grundidee besteht daraus mithilfe der sogenannten **Fast Fourier Transform** (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 Frequenz der gespielten Note entspricht, wird dann einer **Position im virtuellen Notensystem** zugeordnet und für den Nutzer virtuell dargestellt. Es besteht ebenfalls die Möglichkeit die gespielten Noten in einer Datei zu **speichern** und diese dann auch wieder zu **laden**.

1. **Die Fast Fourier Transform**

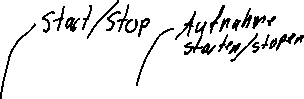
Die **Fast Fourier Transform**, kurz FFT, ist eine effizient mathematische Operation zur Berechnung der **Diskreten Fourier-Transformation**. Die FFT ist wesentlich schneller als die herkömmliche Methode, da es mit dem **Teile-und-Hersche** Verfahren arbeitet. Dieses Verfahren teilt im Grunde ein Problem 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 Libary genutzt. Genauer gesagt die Accord.Math Libary. Die Implementierung der FFT findet man im AudioComponents Skript.

1. **Kleiner Überblick des Userinterface**

## 

Ein Bild, das Screenshot, Wolke, draußen, Gras enthält.

Automatisch generierte Beschreibung



1. **Die Umsetzung**
   1. *Die Aufnahme einer Note*

Zur Aufnahme einer Note wird das Signal, welches von der Gitarre kommt, zunächst von dem Preamp verstärkt und danach mithilfe des Audiointerfaces auf den Computer übertragen. Dort wird es dann als Mikrophone dargestellt. Da das ankommende Signal endlos weiter geht also weder einen klar definierten Anfang noch Ende hat, wird es in kleine Teile von ca. 0,5 Sekunden unterteilt. Diese Zeit ist aber variabel. Denn je kürze sie ist, desto schneller können Noten hintereinander abgespielt werden. Da gilt: BPM = 60/t. Wobei t die Zeit eines Teiles in Sekunden entspricht und BPM für Schläge pro Minute (engl. beats per minute) steht. Allerdings besteht bei einer zu kurzen Aufnahmezeit die größere Wahrscheinlichkeit, dass die Frequenz der Note auf zwei Teile aufgeteilt 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.

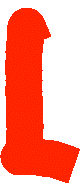
* 1. *Die Audioanalyse*
     1. Berechnung der Notenfrequenz

7.2.1.1 Berechnung des Frequenzspektrums

Nachdem das Signal in kleine Teile aufgeteilt wurde, muss nun jedes einzelne Teil analysieren werden, um die Frequenz der in dem Abschnitt gespielten Note zu erfassen. Dazu ist es am effizientesten die oben beschriebenen Fast Fourier Transform zu nutzen.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung



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

7.2.1.2 Noise

Man darf nicht außeracht lassen, dass die Gitarre kein 100 Prozent reines Signal produziert, weswegen auch viel Noise, das sind eine Art Störfrequenzen, vorhanden sind. Diese sind in der Abbildung rot markiert.

7.2.1.3 Bestimmung der Peaks

Wie man in der obigen Abbildung erkennt, kann man die gespielte Note anhand ihrer Amplitude (Lautstärke) erkennen. Die Frequenz mit der höchsten Amplitude entspricht eigentlich der Frequenz der gespielten Note. Da wir aber sehr oft viel Noise haben, welche das Signal verfälscht, müssen wir nicht nur die Amplitude, sondern auch die Frequenz in Betracht ziehen. Das Noise hat größtenteils eine höhere Frequenz als die gespielte Note. Weswegen wir den niedrigsten Hochpunkt (engl. Peak) finden müssen. Dafür iterieren wir durch jeden Wert in der FFT und prüfen, ob der jeweils nächste Wert links oder rechts eine höhere Amplitude besitzt als der Wert selbst. Ist dem nicht der Fall handelt es sich um einen lokalen Peak. Nun sortieren wir alle Peaks aus, die unter dem Durchschnitt liegen, um das Ergebnis nicht zu verfälschen. Danach nehmen wir den Peak mit der niedrigsten Frequenz. Dieser ist der Peak unserer gespielten Note.

* + 1. Findung der nächstliegenden Notenfrequenz