

Audio-Codierung

Praktikum

In diesem Praktikum werden Sie die Audiocodierung und das Erstellen eines «Wave-Files» genauer untersuchen. Zur Verifikation verwenden wir dafür das Open-Source Audio Tool «Audacity» (<https://www.audacityteam.org/>), welches auf Windows, Mac und Linux lauffähig ist. Laden und installieren Sie dieses Tool auf Ihrem Rechner.

Laden Sie das Java-Template «AudioTest.java» für dieses Praktikums aus Moodle in ein Verzeichnis. Sie brauchen nur an den bezeichneten Stellen in der Vorlage den Code zu ergänzen respektive zu ändern.

Aufgaben

1. **Header des Wave-Files:** Studieren Sie das Template, insbesondere die Stellen, wo der Header des Wave-Files generiert wird. Vervollständigen Sie die fehlenden Stellen, sodass ein Wave-File Header generiert wird für
 - a) Mono Audio
 - b) Abtastrate 44.1kHz
 - c) 16 Bit pro Sample

Konsultieren Sie dafür die entsprechenden Vorlesungsfolien und die Kommentarzeilen im Code. Wie schon in den ersten Code-Zeilen ersichtlich verwenden Sie dazu die Java-Funktionen `raw.writeShort()`, `raw.writeInt()` und `raw.writeBytes()` um jeweils 2, 4 oder eine Sequenz von Bytes in das File zu schreiben.

Achtung: Verwenden Sie zum Schreiben von «Short» und «Int» ebenfalls die Funktion «`reverseBytes`», um die Daten in Little-Endian Format abzuspeichern (Siehe Code-Zeilen im Template, die dies bereits verwenden).

Tipp: Verwenden Sie die bereits definierten Variablen `sampleRate`, `bitsPerSample`, `nChannels` und `duration` in der `main()` Funktion.

2. **Samples eines einfachen Sinustons:** In der folgenden «for»-Schleife sollen nun die Samples eines Sinustones berechnet und ins File geschrieben werden. In der Schleife ist bereits die Funktion `writeSample` vorhanden, welche als Parameter einen *double* Wert erwartet, den Sie berechnen sollen. Die Berechnung der einzelnen Samples S_i hängt von der Sampling-Rate R , der Frequenz f des gewünschten Tones und natürlich von $2 * \pi$ (in Java: `2.0 * Math.PI`) ab:

$$S_i = K * \sin\left(\frac{i * 2\pi * f}{R}\right)$$

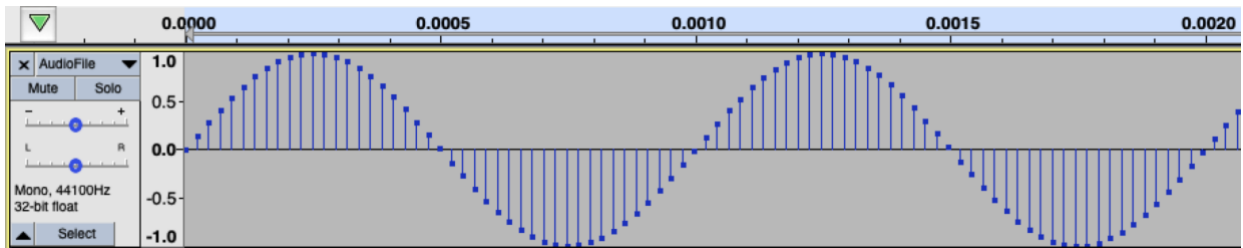
Der Skalierungsfaktor K soll so gewählt werden, dass die 16Bit pro Sample möglichst vollständig ausgenutzt werden. D.h. der resultierende Sinus soll zwischen dem negativsten und dem positivsten Wert der 16 Bit signed Zahl oszillieren.


Wie gross muss der Skalierungsfaktor K gewählt werden? _____

3. Generieren Sie nun in der «for» Schleife einen Ton von 1kHz bei einer Sampling-Rate R von 44.1kHz. Compilieren und starten Sie das Programm. Compilieren und starten Sie das Programm.

Anschliessend können Sie das generierte File «AudioFile.wav» mit Audacity (oder einem anderen Mediaplayer) anhören. Am einfachsten, Sie ziehen das Wave-File mit **Drag&Drop** in Audacity.

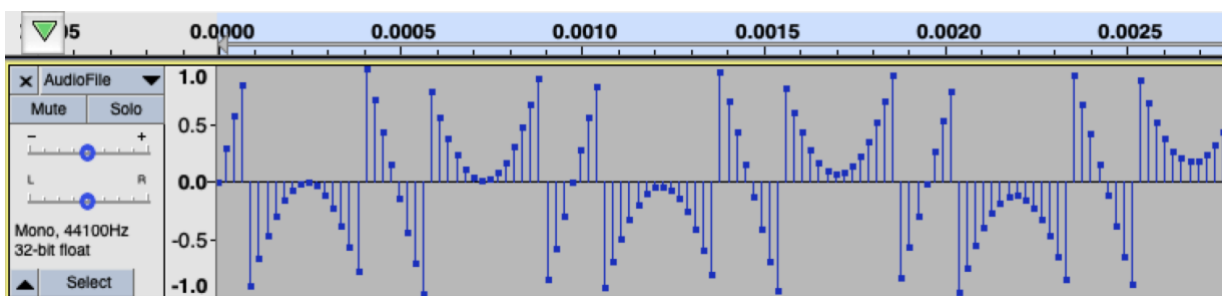
Falls Sie keinen Ton hören: Gehen Sie nochmals genau durch die einzelnen Parameter im Header, die Sie programmiert haben. Sehen Sie Pegel in der Tonspur? Wenn Sie in der Zeitachse Zoomen, werden Sie folgenden Verlauf sehen können:



4. **Untersuchung:** Mit dem «Zoom» Tool  können Sie in Audacity so weit in die Signalform zoomen bis Sie schliesslich die einzelnen PCM Werte sehen. Wie viele Stützstellen hat Ihr 1kHz Signal über eine Periode? _____
Wie gross ist die Dynamik? (Wertebereich peak-to-peak) _____
Wie hätten Sie die Anzahl Stützstellen berechnen können? _____
5. **2-Töne:** Kopieren Sie jetzt Ihr Programm nach Audio2f.java. Erweitern Sie dieses zum Generieren von zwei Tönen gleichzeitig mit den Frequenzen $f_1 = 1000\text{Hz}$ und $f_2 = 1050\text{Hz}$ indem Sie die obige Formel erweitern:

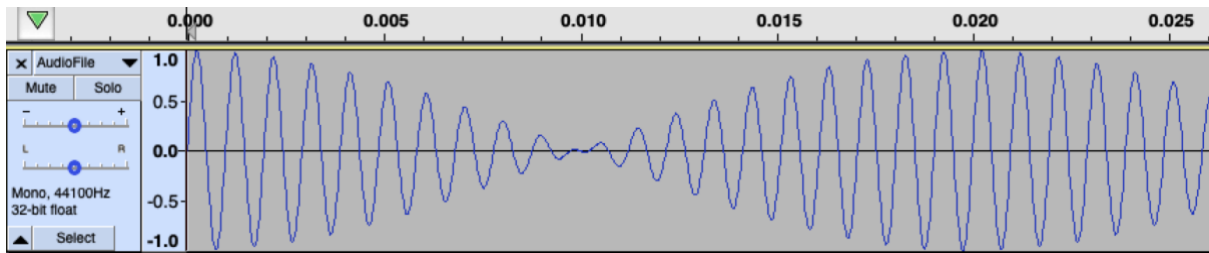
$$S_i = K * (sig_1 + sig_2) = K * \left(\sin\left(\frac{i * 2\pi * f_1}{R}\right) + \sin\left(\frac{i * 2\pi * f_2}{R}\right) \right)$$

Schauen Sie sich das Signal in Audacity an und hören Sie sich das Wave-File an. Wenn Sie den Faktor K nicht verändert haben, wird es ungefähr so aussehen:



Was hat zu dieser Signalform geführt? _____

Passen Sie nun die Erzeugung der Audiosamples so an, dass Sie in Audacity einen Voll-Ausschlag, aber kein Clipping sehen. Das Resultat müsste dann so aussehen:

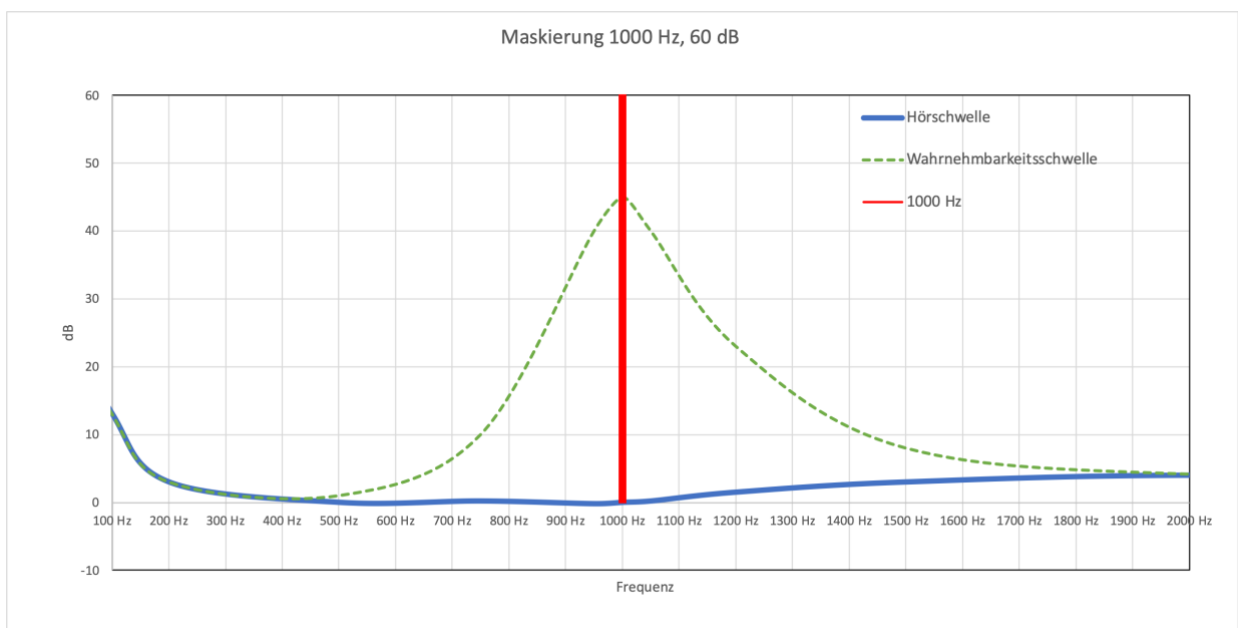


Wie erreichen Sie das? _____

6. **Maskierung:** Nun wollen wir untersuchen, bei welchem Pegel der zweite Sinus sig_2 mit $f_2 = 1050 \text{ Hz}$ **nicht mehr hörbar ist**, wenn gleichzeitig sig_1 mit einem normalen Pegel gespielt wird. Dazu dividieren wir die Samples von sig_2 durch eine 2-er Potenz (2, 4, 8, 16, ...). Erhöhen Sie die 2-er Potenz, bis Sie sig_2 nicht mehr hören.

a) Bei welcher 2-er Potenz-Division ist sig_2 unhörbar, wenn gleichzeitig sig_1 abgespielt wird?	
b) Wieviel Dezibel entspricht jetzt der Signal-Abstand zwischen sig_1 und sig_2 ?	
c) Hören Sie sich das soeben abgeschwächte sig_2 nochmals allein, ohne sig_1 an. Ist es nun hörbar?	
d) Funktioniert die Maskierung auch, wenn sig_2 auf 1900Hz gestellt ist? Bei wie viel Dezibel Signal-Abstand?	

Auf der folgenden Graphik ist bildlich sig_1 (1kHz) in rot, die Hörschwelle in blau und eine typische Wahrnehmungsschwelle gezeigt. sig_1 hat hier einen Pegel von 60dB. Tragen Sie nun sig_2 für die Frequenzen 1050Hz und 1900Hz mit der Abschwächung gegenüber sig_1 , die Sie oben in 6b) herausgefunden haben:



7. **Spiegelung:** Nun untersuchen wir das Phänomen der Spiegelung (Siehe **Vorlesung Folien 5-7**). Generieren Sie wieder nur einen Ton mit 1kHz Frequenz, aber ändern Sie diesmal die Sampling-Rate **R** auf **8kHz**. Die Sampling-Rate muss auch im Header geändert werden.

(Nach dem Ändern der Sampling-Rate muss das AudioFile neu in Audacity geladen werden).

Wieviele Stützstellen haben Sie nun pro Periode? _____

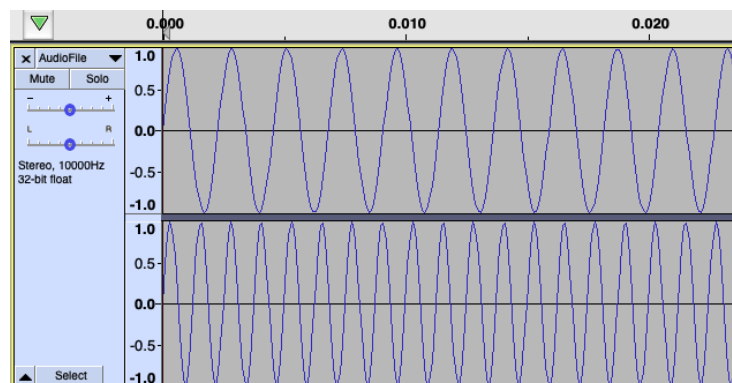
Ändern Sie nun die Frequenz des generierten Tones **f** auf **7kHz**.

Hat sich der Ton verändert? Wieviele Stützwerte sehen Sie? _____

Ändern Sie nun nur die Sampling-Rate wieder zurück auf **44.1kHz** und hören Sie das File jetzt an (Neuladen in Audacity!).

Welche Frequenz hören Sie nun? _____

8. **Stereo:** Schliesslich erweitern wir den Versuch, indem wir ein Stereo Wave-File erzeugen. Ändern Sie dazu die Anzahl Kanäle auf 2 und erzeugen Sie einen 440Hz Ton für den linken Kanal und einen 880Hz Ton für den rechten Kanal. (Neuladen in Audacity!). Wenn Sie alles richtig gemacht haben, werden in Audacity die Spuren für den linken und rechten Kanal separat angezeigt:



Hören Sie den **einen Ton nur links** und den **anderen nur rechts** im Kopfhörer?