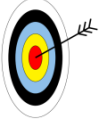


Sounds

Sounds bearbeiten



Nicht nur, dass ihr jetzt mit eurem Wissen über Arrays Bilder bearbeiten könnt, Arrays sind in sehr vielen Kontexten nützlich. Zum Beispiel zum Bearbeiten von Sounds. Noch besser: diese könnt ihr dann für euer Video verwenden. Sounds können in Audacity aufgenommen oder z.B. auf www.freesound.org heruntergeladen werden.

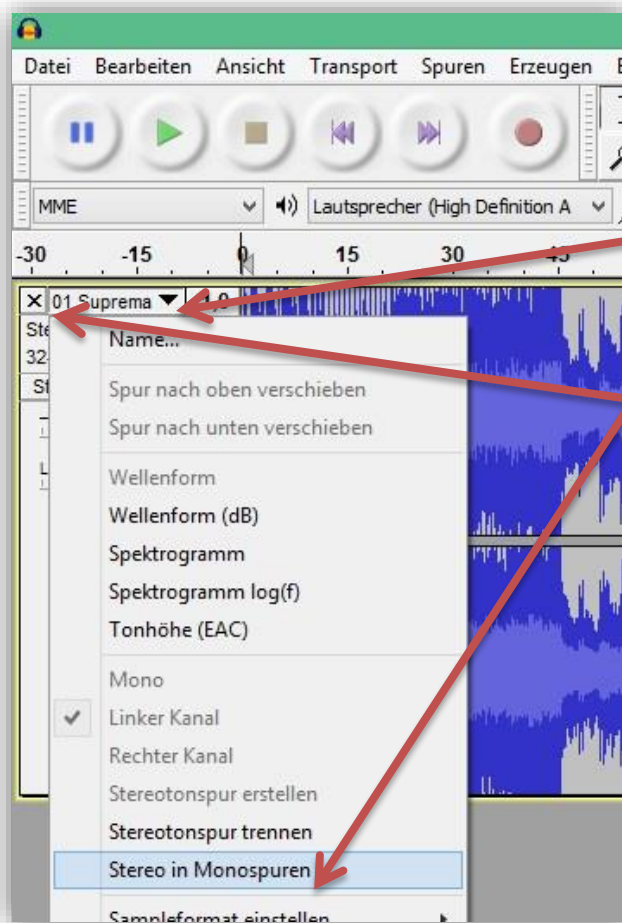


Vorsicht! Ihr könnt hier nur mit **Mono-Sounds**, also Sounds mit nur einem Kanal, und nur mit **.wav Dateien** arbeiten.

Mit Audacity lassen sich beliebige Sounds in dieses Format umwandeln.

- Mp3-Dateien: Mit Audacity laden und dann als .wav exportieren (Datei→Exportieren).
- Stereo Dateien: Siehe Bild unten

Umwandeln von Stereo in Mono-Dateien mit Audacity:



Sorgt am besten jetzt schon dafür, dass euch mindestens 2 Sounds zum Testen eurer Methoden zur Verfügung stehen.

1. Auf den Namen der Tonspur klicken.
2. „Stereo in Monospuren“ auswählen.
3. Eine Tonspur schließen.
4. Datei → exportieren als .wav

Sounds

Rückgabe eurer Methoden:

Sounds können als **double-Arrays** gespeichert werden. In jeder der Methoden werden double-Arrays übergeben, deren Werte entsprechend verändert **im jeweiligen double-Array „ausgabesound“ zurückgegeben** werden sollen. Das Array „ausgabesound“ müsst ihr selbst erzeugen.

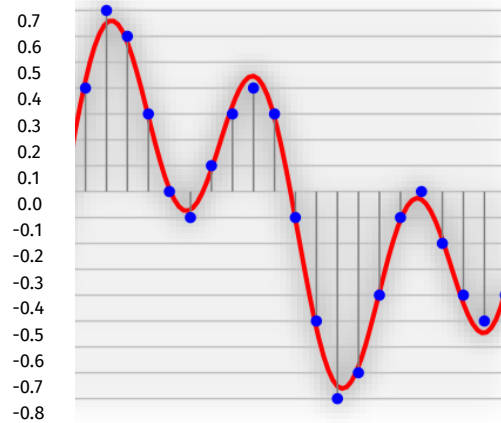
	double[] eingabesound					
Frame	0	1	2	3	4	5
Wert:	-0.4	0.0	0.4	0.7	0.6	0.3

	double[] ausgabesound		
Frame	0	1	2
Wert:	0.4	0.7	0.6

(Red dashed arrows point from the 'ausgabesound' table back to the 'eingabesound' table, indicating return values.)

Zusatzinfo: Bedeutung der Array-Einträge

Die Einträge der double-Arrays nennt man in diesem Kontext „**Frames**“. Die Qualität einer Sounddatei wird durch ihre „**Samplerate**“ begrenzt. Diese legt fest, wie viele Frames für eine Sekunde Sound verwendet werden (ähnlich der Auflösung). Sounds sind **digitalisierte Schallwellen**. Auf dem Bild rechts seht ihr eine Schallwelle (rot) und die digitalen Integer-Zahlenwerte oder Frames, die diese repräsentieren (blau). In unserem Fall handelt es sich um Double-Frames, die im Bereich von -1.0 bis 1.0 liegen dürfen.



Quelle: Wikipedia

Dieses Arbeitsblatt enthält drei Aufgabenstellungen. Diese sollen ebenfalls im Team verteilt werden.



Um die Tonspur für euer Video zu erzeugen, braucht ihr ein paar grundlegende Methoden, mit denen ihr dann aus verschiedenen Sounds eure Tonspur zusammenbauen könnt. Diese grundlegenden Methoden sollen in dieser Aufgabe implementiert werden. Ladet diese, wenn sie funktionieren auch in den Dropbox-Ordner.

1. Implementiert die Methode „**rueckwaerts**“ in der Klasse Soundoperationen. Diese soll einen Sound umdrehen. Zurückgeben müsst ihr ebenfalls ein double-Array.
2. Die Methode „**aneinanderhaengen**“ soll in der Klasse Soundoperationen implementiert werden. Hier werden zwei Sounds, einer im „**eingabesound1**“, der andere im „**eingabesound2**“ übergeben. Der „**ausgabesound**“ soll eine **Aneinanderreihung** der beiden sein. Nach dem Ausführen sollte sich die Anzahl der Frames, die in der GUI angezeigt wird, verändern.

Sounds

3. Implementiert die Methoden „**abschneidenvorne**“ und „**abschneidenhinten**“ in der Klasse Soundoperationen.java. Die Anzahl an Frames, die abgeschnitten werden soll, wird ebenfalls übergeben. **Die Länge des „ausgabesound“ muss entsprechend angepasst werden.** Nach dem Ausführen sollte sich die Anzahl der Frames, die in der GUI angezeigt wird, verändern.

Hier könnt ihr euch etwas notieren oder Skizzen davon machen, wie die Frames umsortiert werden sollen.

Ihr habt etwa 60 Minuten Zeit mit Sounds zu experimentieren und mit dem nächsten AB ein Video zu erstellen.