

# Projektplan Gruppe B: 3D Audio

Gruppenmitglieder: Felix Pfreundtner, Matthias Lederle, Huaijiang Zhu, Molka Elleuch, Manuela Heiss

## Projektvision:

„Wir wollen ein Konzert simulieren, das ein Hörer auf seinem Kopfhörer wahrnimmt, als würde er „inmitten der Band“ sitzen!“

## Projektbeschreibung:

Was kann die Abgabeverision:

- Binaurale Simulation von bis zu 6 Musikern beziehungsweise eines 5.1 Lautsprechersystems
- horizontale Position des Hörers und der Quellen kontinuierlich im GUI einstellbar, Höhe konstant
- Realtimeanpassung der binauralen Soundausgabe entsprechend aktueller Benutzereinstellungen
- Berücksichtigung der horizontalen Kopfbewegungen durch einen Headtracker
- keine Berücksichtigung der Raumakustik, Abstrahlung der Quellen unter Freifeldbedingungen
- Anordnung der Quellen stets direkt auf den Hörer gerichtet


## Entwicklungsplan/Zeithorizont:

### **Woche 1: bis Dienstag, 19. Mai:**

- Icebreaking / Teambuilding
- Projektplanerstellung
- Einarbeitung in Audiosignalverarbeitung, HRTF (KEMAR), GUI Programmierung, HeadTracker
- Mindestliteratur für alle: Auralization (Vorländer, 2008), Kapitel 6.3, 7, 9, 15.1, 16.1

### **Woche 2 + 3: bis Di, 2. Juni:**

Signalverarbeitungsalgorithmus Team	GUI & Headtracker Team
<b>1. Schritt</b> <ul style="list-style-type: none"><li>- Programmieren Einlesen KEMAR HRTF Dateien</li><li>- Programmieren Einlesen eines mono Wave Files</li><li>- Signalverarbeitung des Eingangssignals zu einem binauralen Ausgabesignal abhängig von den Parametern horizontaler Hörer-Quelle Winkel und Hörer-Quelle Abstand</li><li>- Wiedergabe des generierten</li></ul>	<ul style="list-style-type: none"><li>- Programmierung einer 2D GUI</li><li>- 6 Quellen und 1 Hörer frei durch Mausziehen bewegbar</li><li>- Anzahl an Quellen über +/- Button einstellbar</li><li>- Ausgangsanordnung der Quellen entspricht typischer Heimkinoanordnung, Reset Button stellt Ausgangszustand wieder her</li><li>- Design ähnlich User-Interface Asus Xonar Soundkarte:</li></ul>

<p>Ausgabesignals</p> <p>2. Schritt:</p> <ul style="list-style-type: none"> <li>- Einbeziehen von Veränderungen des Hörer-Quelle Winkels und des Hörer-Quelle Abstands im Zeitintervall <math>\Delta t</math> (Länge eines FFT-Blocks)</li> <li>- Hierfür: Einlesen der (vom GUI abgespeicherten) globalen Variablen horizontaler Hörer-Quelle Winkel und Hörer-Quelle Abstand</li> <li>- Falls geänderter Parameter horizontaler Hörer-Quelle Winkel: Einlesen der passenden HRTF</li> <li>- Berechnung des binauralen Ausgabesignals Block für Block und blockweise Ausgabe über Kopfhörer</li> </ul>	 <ul style="list-style-type: none"> <li>- Raumgröße: Wohnzimmer</li> <li>- Pfad des Eingabesignals jeder Quelle in GUI auswählbar</li> <li>- Quellen und Hörer sind Werte in kartesischen Koordinaten, Ursprung des Koordinatensystems ist der Hörer</li> <li>- Blickrichtung des Hörers und Ausrichtung der Quellen sichtbar machen</li> <li>- Playbutton: Berechnen der Parameter horizontaler Hörer-Quelle Winkel und Hörer-Quelle Abstand für alle Quellen und abspeichern der Parameter als globale Variablen in Polarkoordinaten</li> <li>- Bei Verschiebung der Quellen oder des Hörers Neuberechnung der Parameter: horizontaler Hörer-Quelle Winkel und Hörer-Quelle Abstand, abspeichern der Parameter als globale Variablen in Polarkoordinaten</li> </ul>
---	---

#### Woche 4+5: bis Di, 16. Juni:

Signalverarbeitungsalgorithmus Team	GUI & Headtracker Team
<ul style="list-style-type: none"> <li>- Erweiterung des Algorithmus von einer Quelle auf 6 Quellen</li> <li>- Falls notwendig: Laufzeitoptimierung des Algorithmus</li> <li>- Normalisieren der Lautstärker aller Quellen</li> <li>- Mischen der binauralen Ausgabesignale aller Quellen zu einem einzigen Ausgabesignal unter Berücksichtigung des Abstandes der Quellen zum Hörer</li> </ul>	<ul style="list-style-type: none"> <li>- Integration des Headtrackers in das GUI</li> <li>- Einlesen der relativen horizontalen Winkelveränderungen des Kopfes</li> <li>- Einbeziehen der Winkelveränderungen in die Polarwinkel zu den Quellen</li> <li>- Anzeigen der Kopfausrichtung im GUI</li> </ul>

#### Woche 6: bis Di, 23. Juni:

- Tests + Präsentation erstellen

## Woche 7: bis Di, 30. Juni:

- Zeitpuffer

## Woche 8: bis Di, 7. Juli: Präsentation

### Interne / externe Schnittstellen:

Vom Signalverarbeitungsalgorithmus zum GUI:

- Keine Parameterübergabe

Vom GUI zum Signalverarbeitungsalgorithmus:

- Bei Drücken des Playbuttons und Veränderungen des Hörers oder der Quellen:  
Neuberechnung der Parameter horizontaler Hörer-Quellen Winkel und Hörer-Quellen Abstand und abspeichern der Parameter als globale Variablen in Polarkoordinaten auf welche der Signalverarbeitungsalgorithmus im Zeitintervall  $\Delta t$  zugreift
- Dateipfad zum wave Eingangssignal jeder Quelle

Vom Headtracker zum GUI:

- Horizontale Winkelveränderung des Hörers im Verhältnis zur Ausgangslage  $t=0$  des Kopfes

Vom Benutzer zum GUI:

- Anzahl der Quellen über +/- Button einstellbar
- Bis zu 6 Pfade zu den wave Eingangssignalen der Quellen
- Positionen der Quellen und des Hörers (Verschieben Quellen / Hörer möglich)
- Reset Befehl (stellt Ausgangsanordnung der Quellen wieder her)

Main Funktion:

- Die Main Funktion startet das GUI
- Über das GUI werden vom Benutzer alle notwendigen Eingabeparameter eingegeben
- Danach ruft das GUI den Signalverarbeitungsalgorithmus auf
- Der Signalverarbeitungsalgorithmus generiert für jeden FFT-Block (Länge  $\Delta t$ ) der Quellen-Eingangssignale ein binaurales Ausgangssignal unter Berücksichtigung der vom GUI berechneten Parameter horizontaler Hörer-Quelle Winkel (Auswahl der richtigen HRTF für jede Quelle) und Hörer-Quelle Abstand (Anteil einer Quelle an Klangmischung). Im Anschluss mischt der Algorithmus die binauralen Ausgangssignale aller Quellen zu einem binauralen Ausgangssignal und gibt das Ergebnis mit Blocklänge  $\Delta t$  intervallweise als Soundausgabe aus. Nach jedem FFT-Block prüft der Algorithmus ob sich die vom GUI für jede Quelle global abgespeicherten Variablen horizontaler Hörer-Quelle Winkel und Hörer-Quelle Abstand verändert haben, falls ja wird eine andere HRTF mit dem Quellensignal gefaltet bzw. der Anteil der Quelle an der Klangmischung angepasst.

## **Gruppen Einteilung:**

- Signalverarbeitungsalgorithmus Team: Felix Pfreundtner, Matthias Lederle, Huaijiang Zhu
- GUI & Headtracker Team: Molka Elleuch, Manuela Heiss

## **Fragen an unseren Betreuer:**

Wie kann man die horizontale Winkelausgabe des LDV Headtracker auslesen? Ist hierfür Literatur vorhanden?

Ist das Auswerten von horizontalen Veränderung der Quellen- und Hörerposition für das Projekt ausreichend oder sollten wir zusätzlich eine vertikale Dimension mit einbeziehen?

Kann der Signalverarbeitungsalgorithmus auch nicht objektorientiert programmiert werden?

Ist es nötig für den räumlichen Eindruck Hall zu den Audiosignalen hinzuzufügen?

Ist in Python das Einlesen einzelner Frames eines Audiosignals möglich oder muss stets das gesamte wave Signal eingelesen werden?

Wie können Soundausgabeprobleme (Ruckler) zwischen zwei, mit unterschiedlichen HRTF gefalteten, Fourierblöcken am besten verhindert werden?

Welche Möglichkeiten gibt es den FFT Algorithmus beziehungsweise die Faltung mit den HRTF zu beschleunigen? Ist ein Rückgriff auf diese Möglichkeiten bei Verwendung von 6 Eingangssignalen nötig?