

Projektkonzept

„Air-Synthie“

Audio-Video-Programmierung

WiSe20/21

Bachelor Medientechnik/Media Systems

01.12.2021

Hannah Lenz 2339432

Tobias Rawald 2380225

Yannic Kühn 2382131

1. Einleitung
2. Idee/Ziel
3. Anforderungsanalyse
4. Technische Rahmenbedingungen
5. Technisches Konzept
6. Bedienkonzept
7. Zeitplan



Einleitung

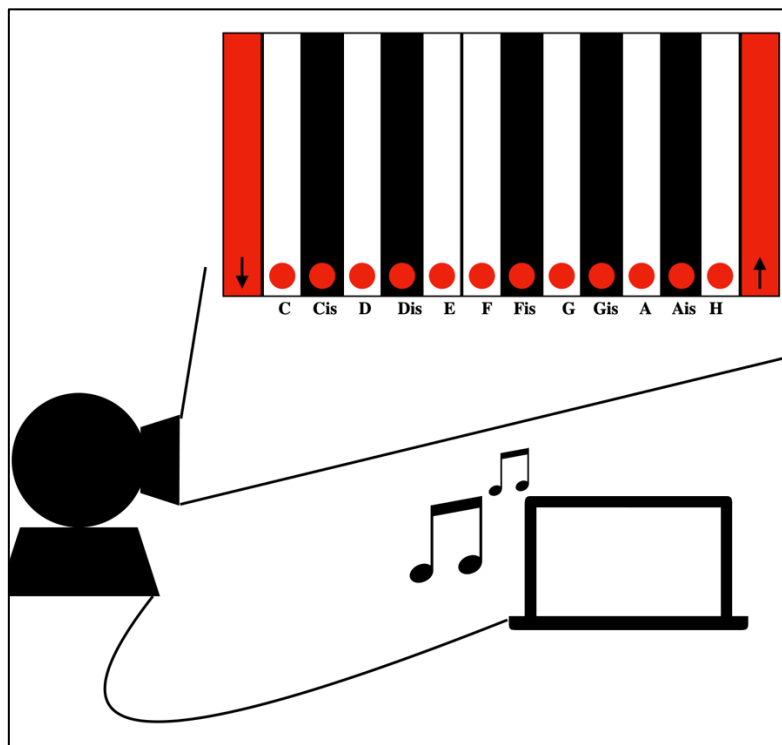
Im Rahmen des studienfachübergreifenden Wahlpflichtkurses „Audio-Video-Programmierung“ für die Studiengänge Medientechnik und Media Systems an der HAW Hamburg soll in Kleingruppen ein Softwareprojekt entwickelt werden. Ziel ist es, die im ersten Teil des Kurses erlernten Inhalte zu den Themen WebAudio (JavaScript) und openCV (Python) zusammen zu bringen und in die Praxis zu übersetzen. Mittels der Auswertung eines Live-Video-Streams soll hierbei ein Live-Video-Stream erzeugt werden.

Abschließend wird das Projekt zum Ende des Semesters in Form einer mündlichen Präsentation und einer technischen Dokumentation inklusive Quelltext abgeschlossen.

Idee/Ziel

Die Software soll ein Live-Video über eine externe Webcam aufnehmen. Diese ist auf ein Blatt Papier gerichtet, auf dem eine volle Oktave einer Klaviertastatur zu sehen ist. Legt man nun einen Finger auf eine Taste, ertönt der passende Ton, welcher durch den Laptop/PC erzeugt wird. Durch zwei weitere Tasten kann die Oktave nach oben und nach unten gewechselt werden, sodass sieben Oktaven eines Klaviers bedient werden können. Außerdem können über die Töne mittels Reglern auf einer Webseite verschiedene Effekte wie Hall, Delay etc. gelegt werden.

Visualisierung



Anforderungsanalyse

Die Software soll in der Lage sein, die Live-Daten mit einer möglichst geringen Latenz zu übertragen und auszuwerten, damit der Anwender ein möglichst natürliches Gefühl beim Spielen bekommt und auch dazu befähigt ist, einfache Melodien wie „Alle meine Entchen“ verzögerungsarm und rhythmisch korrekt umzusetzen. Außerdem sollen die Töne möglichst angenehm und real klingen, das heißt, sie haben eine gewisse An- und Abklingzeit, sodass sie leicht ineinander verschwimmen und so ein fließender Höreindruck entsteht.

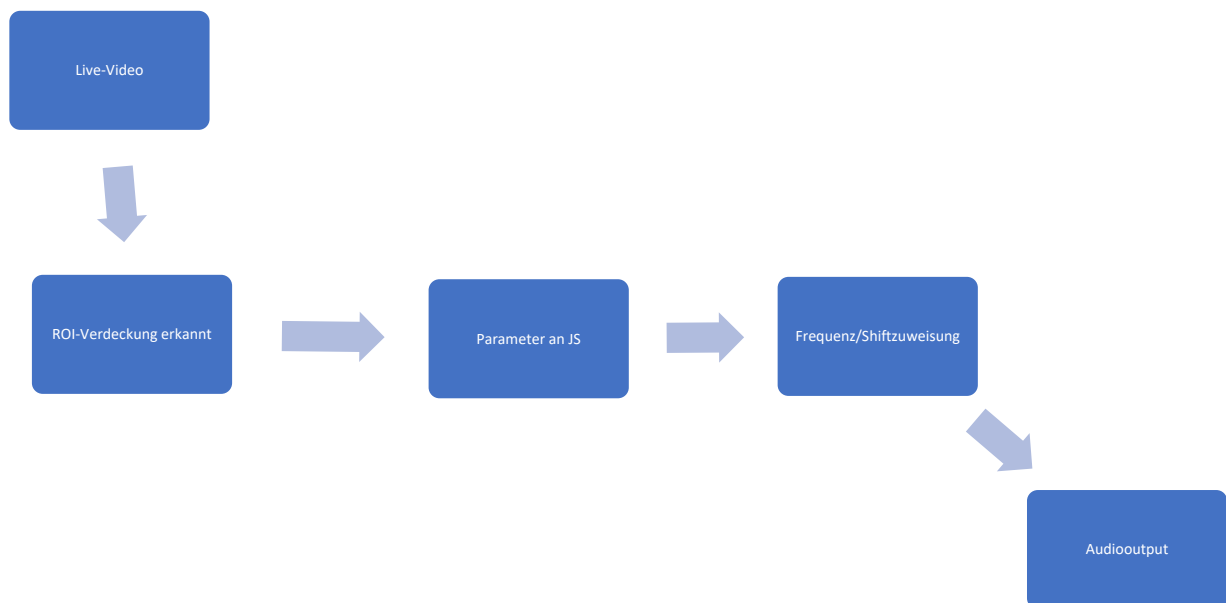
Technische Rahmenbedingungen

Die Programmierung der Software wird audioseitig mittels der **WebAudio API** in der Skriptsprache **JavaScript** und videoseitig mittels der Programmbibliothek **openCV** und der Programmiersprache **Python** erfolgen. Die Programmierung geschieht in **Visual Studio Code**. Die Steuerinformationen für den Audioteil werden über den **MIDI-Standard** geregelt.

Technisches Konzept

Durch die roten Markierungen auf der Tastatur kann das Live-Bild in Regions of Interest eingeteilt werden. Wird nun einer dieser Punkte verdeckt, so wird der entsprechende Parameter an Javascript übergeben. Nun wird der Parameter einer Frequenz aus einem Array zugewiesen und über die Lautsprecher des Computers ausgegeben. Die zwei zusätzlichen Tasten werden mit einer Octave-Shift-Funktion belegt, die dazu führt, dass alle Ton-Tasten eine Oktave hoch bzw, runter gewechselt werden. Auf einer dazu gehörigen Webseite wird es Regler geben, um bestimmte Parameter wie einen Hall, Delay etc. auf die Tasten zu legen.

Flussdiagramm



Bedienkonzept

In einem Beispielvideo wird das Bedienkonzept der Software illustriert. Wird ein Finger auf eine der 12 Ton-Tasten gelegt, so erklingt der passende Ton auf der Tonleiter. Möchte man nun eine andere Oktave bespielen, so gibt es links und rechts der Tastatur eine weitere Bedieneinheit, mittels derer die Tonleiter um eine Oktave tiefer bzw. höher geschaltet werden kann. Die Tonspanne umfasst sieben Oktaven.

Zeitplan/Meilenstein

Datum	Thema
01. Dezember	Konzeptbesprechung
8. Dezember	Projektarbeit
15. Dezember	Projektarbeit/ Meilenstein*
04. Januar	Meilenstein**
05. Januar	Vorstellung Prototyp
19. Januar	Abschlusspräsentation
26. Januar	Projektabgabe

***Meilenstein:** Grundgerüst der Software sollte stehen, also: Ton-Tastaturanschläge sollten zuverlässig erkannt werden.

****Meilenstein:** Das Oktav-Shifting sollte nun funktionieren, die Klangqualität sollte nun angepasst worden sein, sodass bis zur Abschlusspräsentation noch an Stellschrauben wie Latenzen und Design gearbeitet werden kann.

Zeitlicher Aufwand:

20 PS bis zum ersten Meilenstein

15 PS bis zum zweiten Meilenstein/Prototyp

15 PS bis zur Abschlusspräsentation

10 PS für die Nachbereitung und Dokumentation

60 PS Gesamt

Teamplanung

Tobias Rawald: WebAudio, Python

Yannic Kühn: WebAudio, Python

Hannah Lenz: Dokumentation, Javascript/CSS