Audio-Video-Programmierung

Prof. Dr. Plaß & Sudau

WiSe 17/18

https://github.com/irenabecker/AVPrg\_DIY3DSoundkulisse

SoundSpace

Takes you wherever you want to be.

Carina E. Krafft | 2269579  
Jordanis Lazaridis | 2142772  
Irena Becker | 2238833  
Kevin Hagen | 2270985

17.01.20178  
Hochschule für angewandte Wissenschaften Hamburg  
Fakultät Design, Medien und Information  
Department Medientechnik

Inhalt

[Einleitung 2](#_Toc503972051)

[Back-End Entwicklung 2](#_Toc503972052)

[Form-Erkennung 2](#_Toc503972053)

[Farb-Erkennung 2](#_Toc503972054)

[Positionsbestimmung in 3 Dimensionen 2](#_Toc503972055)

[Front-End Umsetzung 2](#_Toc503972056)

[Interface 2](#_Toc503972057)

[WebAudio 2](#_Toc503972058)

[Schnittstellenprogrammierung 3](#_Toc503972059)

[Herausforderungen 3](#_Toc503972060)

[Reflexion: Umsetzung der User-Stories 3](#_Toc503972061)

# Einleitung

* Projekt in einem Satz zusammenfassen.
* Verschiedene Shapes gleicher Farbe und Form müssen voneinander unterschieden werden
* Konkret sagen welche Shapes and Colors festgelegt wurden

Das Projekt „SoundSpace“ wurde im Rahmen der Vorlesung Audio-Video-Programmierung realisiert. Eine Projektvoraussetzung war daher die Verwendung der, in der Vorlesung behandelten, Programmiersprachen C++, unter Verwendung der Entwicklungsumgebung QT und der Bibliothek openCV, und JavaScript, in Kombination mit der WebAudio API.

Die Entwicklung des Projektes wurde grundlegend in die Bereiche Backend- und Frontend- und Schnittstellen- Entwicklung unterteilt, sodass an allen Bereichen unabhängig voneinander gearbeitet werden konnte.

# Back-End Entwicklung

Das Back-End von „SoundSpace” wurde in C++, unter Verwendung der Entwicklungsumgebung QT und der Bibliothek openCV, umgesetzt.

Ziel des Back-Endes war es ein eingespeistes Kamerabild zu analysieren und dadurch die dreidimensionalen Positionen aller einzelnen Objekte, sowie deren Farbe und Form, erkennen und ausgeben zu können.

## Form-Erkennung

Da es für „SoundSpace“ wichtig war mehrere Objekte gleicher Farbe und möglicherweise

* Formerkennung durch Top-Kamera
* Verwendung von Contour->Anzahl der Seiten + Winkelberechnung
* Filter der Schatten

## Farb-Erkennung

* Farbsuche nur im Bereich der erkannten Formen

## Positionsbestimmung in 3 Dimensionen

* X, Y durch Mittelpunkt der erkannten Konturen
* Z durch Bestimmung des Abstandes zur Kamera
* Umrechnung in relative Raumkoordinaten durch Kalibrierung

UML!! Sequenzdiagramm?!

# Front-End Umsetzung

## Interface

* Intuitive Bedienung
* Auswahl des Soundclips und der Lautstärke jedes Default Soundobjektes
* Anzeige, welche Objekte sich im Raum befinden und anpassen der Objekteeigenschaften
* Anzeige der Kamera

## WebAudio

* Erstellung eines virtuellen Raumes
* Bestimmung der Raumkoordinaten aller Objekte anhand der empfangenen Daten
* Einstellung der einzelnen Klangquellen mit Soundclip und Lautstärke, vorgegeben durch platzierte Objekte.
* Wiedergabe der Klangquellen an den durch die Objekte festgelegten Positionen

UML?

# Schnittstellenprogrammierung

* Normales MIDI sendet nicht genug Daten
* Senden alle 24 Frames, dadurch höhere Latenzzeit
* Versuch eine WebSocket einzurichten
* Verwendung von MIDI Sysex: Datenaufteilung

# Herausforderungen

* Tonlabor Audioanlage --> Features wie z.B. z-Pos wurden umsonst eingebaut
* Verständnis und erfolgreiche Umsetzung von C++ und der Entwicklungsumgebung QT
* Hardware: Farbe der Objekte finden (hell/dunkel), sich ändernde Lichtverhältnisse

# Reflexion: Umsetzung der User-Stories

* Farbe und Größe haben andere Eigenschaften als vorher geplant
* Eine, anstatt zwei, Kameras