

## Konzept - Klangkarte Hamburg

### Einleitung

Das Projekt Klangkarte Hamburg wird als Prüfungsleistung im Rahmen der Vorlesung Audio-Video-Programmierung erstellt. In dem Kurs soll das Verständnis und die Anwendung von Algorithmen erlernt werden, die für Audio- und Videobearbeitung ihren Einsatz finden, sowie die Verknüpfung beider Elemente in einem System. Das Arbeitsthema für die Prüfungsleistung lautet „Tangible Sound Control“, das heißt eine greifbare Klangerzeugung und -steuerung. Die Klangkarten-Station verbindet entsprechend zu den Kursinhalten und dem Arbeitsthema Videoerkennung von bestimmten greifbaren Gegenständen, die wiederum unterschiedliche Klänge erzeugt, je nachdem welche Gegenstände wo auf der Karte platziert werden. Zum Abschluss wird das Projekt auf einer Ausstellung als Demonstration mit begleitendem Plakat präsentiert.

### Projektziel

Ziel des Projekts ist es eine Station zu konstruieren, welche mittels einfachen Interaktionen Informationen vermittelt. An dieser Station wird ein Kartenausschnitt von Hamburg zu sehen sein und es stehen 4 farblich unterschiedliche Würfel zur Verfügung. Diese Würfel symbolisieren verschiedene Interaktionen mit der Karte.

#### **Grün: Verkehrswürfel**

- Liefert Informationen zu naheliegenden Anschlussmöglichkeiten

#### **Rot: Geräusche-/Kinderwürfel**

- Liefert Informationen in Form von einfachen Geräuschen

#### **Gelb: Sehenswürdigkeitenwürfel**

- Liefert Informationen zu dem betreffenden Gebäude etc.

#### **Blau: Geschichtswürfel**

- Liefert Informationen zu Historischen Ereignissen

Für ein besseres Verständnis ist es wichtig zu verstehen, dass nur ein Würfel zur Zeit eine Interaktion aktivieren kann. Die Aufgabe der Software ist es hier, die Position von diesen Würfeln auf der Karte zu erkennen, der Farbe zuzuordnen und dementsprechend Audio wiederzugeben. Ein Beispiel zum besseren Verständnis wäre hier: Ein Proband nimmt den roten Würfel und legt diesen auf ein Schiff am Hafen. Es wird erkannt, dass es sich um den roten Geräuschwürfel handelt und dass sich dieser auf dem Schiff befindet. Als Reaktion wird über die Lautsprecher ein Schiffshorn ertönen und dem Probanden somit in Echtzeit Feedback gegeben.

## Anforderungsanalyse

### Anforderungen:

- Um eine gute Benutzererfahrung in Zusammenhang mit Ein- und Ausgabe/Feedback zu gewährleisten, ist eine Reaktionszeit von unterhalb einer Sekunde angestrebt.
- Um die Audioausgabe verständlich zu halten, wird zu jedem Zeitpunkt nur eine Tonspur abgespielt (das heißt im Umkehrschluss, dass es nie zu einer Tonspur-Überlappung kommt).
- Um die Audioausgabe verständlich zu halten, wird eine Tonspur ohne erneute Eingabe nur einmal abgespielt.
- Um eine einheitliche und konstante Bedienung zu gewährleisten, wird der zuletzt aktiv auf ein Interaktionsfeld bewegte Würfel als der aktuelle Würfel interpretiert (dies gilt unabhängig von Farbe und Position des Würfels).
- Um eine einheitliche und konstante Bedienung zu gewährleisten, wird bei übereinander liegenden Würfeln der Würfel registriert, welcher näher am Scanner/Kamera liegt.
- Um eine gute Benutzererfahrung im Bezug auf die Interaktionsfelder zu gewährleisten, werden diese durch bekannte Symbole bzw. erkennbare Silhouetten dargestellt.
- Um eine einheitliche und konstante Bedienung zu gewährleisten, werden die Würfel einfarbig gehalten und somit garantiert, dass eine Rotation auf jeder Achse keinen Einfluss auf das Ergebnis hat.
- Um die Bedienung zu erleichtern und um die Karte übersichtlich zu halten, ist eine Ablagefläche für die Würfel vorgesehen.
- Um die Audioausgabe verständlich zu halten, werden qualitativ hochwertige Sounds und eigens aufgenommene Tonspuren verwendet.
- Um die Audioausgabe einheitlich und verständlich zu halten, werden die Audioausgabe sowie das Feedback nur in deutscher Sprache vorhanden sein.
- Um die Benutzerzufriedenheit zu erhöhen, werden nur auf Korrektheit überprüfte Informationen genutzt.
- Um eine gute Benutzererfahrung im Bezug auf die Bedienbarkeit zu gewährleisten, wird keine bzw. wenig Technik von außen sichtbar sein (das betrifft zb die Kamera).
- Um die Benutzerzufriedenheit in Bezug auf die Optik zu erhöhen, wird die Karte in einem Holzsockel liegen und mit bemalten Holzwürfeln bedient.
- Um eine gute Benutzererfahrung in Zusammenhang mit der Eingabe zu gewährleisten, wird eine realistisch passende Würfelgröße verwendet.
- Um eine gute Bedienbarkeit zu ermöglichen, wird die Zuordnungsbarkeit der Würfelfarben zu den Themen anhand der Station eindeutig erkennbar sein.
- Um eine gute Bedienbarkeit zu ermöglichen, werden Falscheingaben abgefangen und ignoriert (zb. Finger/Hand auf der Karte).
- Extra: Erweiterbarkeit für alternative Anwendungen zum Beispiel eine andere Karte, eine Tierpark-Attraktion für Kinder oder ein Synthesizer.

## Technische Rahmenbedingung

### Hardware:

Wir arbeiten mit einer Kamera/Webcam, welche unterhalb einer durchsichtigen Platte die Hamburg-Karte aufnimmt. Die Kamera erlaubt es uns die Würfelpositionen und -farben zu tracken. Lautsprecher werden genutzt um die Audiodateien abzuspielen. Die gesamte Anwendung soll auf einem Computer (Desktop/Laptop) laufen.

### Software:

Den Videostream verarbeiten wir mittels openCV/C++, wie wir es in der Vorlesung behandelt haben. Wir bestimmen ob die Würfelfarbe innerhalb der festgesetzten Farbspektren ist und berechnen den Schwerpunkt der Würfelposition. Anhand der Position des Schwerpunktes bestimmen wir, wo sich der Würfel in unserem Gatter befindet. Anschließend werden MIDI Signale an unseren JavaScript/HTML Teil übertragen. In JavaScript wird mithilfe der Audio API, die wir in der Vorlesung mit Herr Sudau behandelt haben, die korrespondierende Audiodatei gesucht und abgespielt.

Um das Projekt für alle Teammitglieder abrufbar zu halten werden wir mit der Versionsverwaltung Git arbeiten.

Eine alternative Softwarelösung wäre das Ersetzen von openCV/C++ durch JavaScript oder MatLab.

## Technisches Konzept

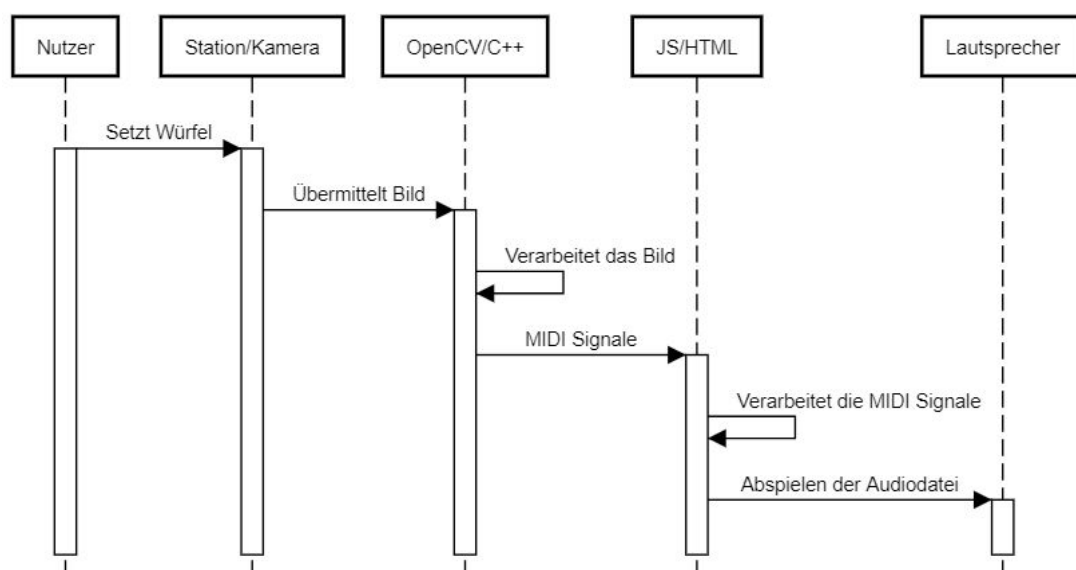


Abbildung 1: UML Sequenzdiagramm

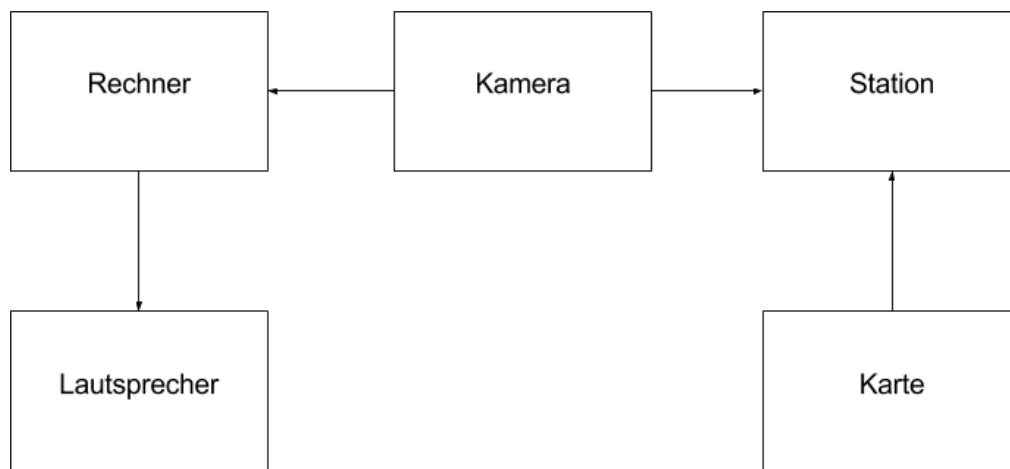


Abbildung 2: Allgemeines Blockdiagramm zur Hardwareverteilung

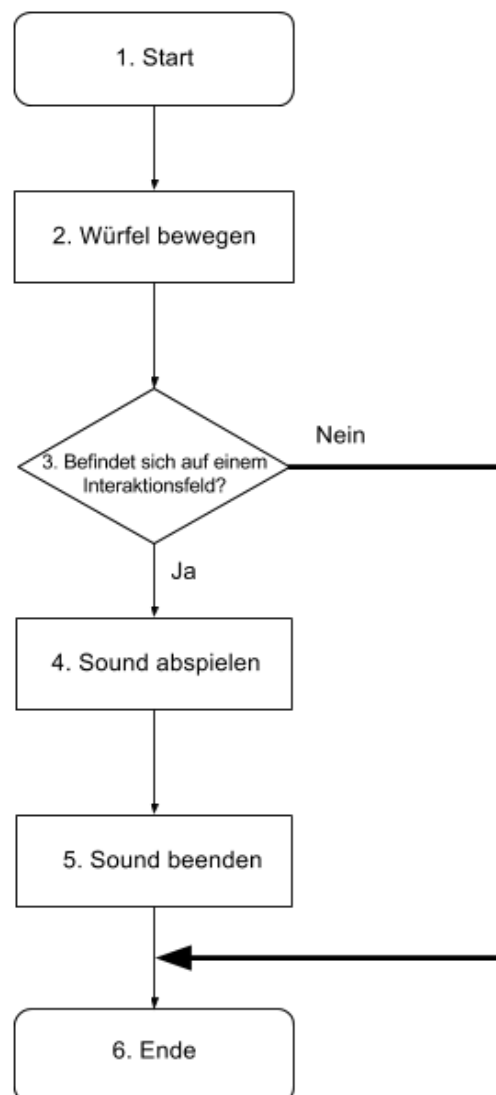


Abbildung 3: Flussdiagramm

## Bedienkonzept

Der Nutzer selbst sieht die Software nicht. Deswegen benötigt die Software auch keine eingerichtete Benutzeroberfläche. Interaktionen finden auf der Station statt. Dort befindet sich eine Karte mit unterschiedlichen Symbolen wie z.B einem Fisch für den Fischmarkt. Der Nutzer hat eine Auswahl aus vier verschiedenfarbigen Würfeln. Ein Würfel soll auf ein beliebiges Symbol gesetzt werden, dadurch wird abhängig vom Würfel-Typ eine Audiodatei abgespielt. Das Abspielen der Audiodatei soll unmittelbar nach Platzieren des Würfels beginnen. Die Audiodatei läuft weiter, auch wenn der Würfel entfernt wird. Der Nutzer kann die Audiodatei stoppen, indem er den Würfel oder einen anderen Würfel seiner Wahl auf ein anderes Symbol legt. Der Nutzer soll mit Hilfe der Würfel und der Karte interaktiv Informationen zu Hamburg bekommen können z.B welche Verkehrsmittel führen ihn zum Fischmarkt. Würfel, die außerhalb der markierten Symbole platziert werden, sollen keine Funktion auslösen.

Optional:

Der Nutzer soll zum Start die Station "kalibrieren", indem er jeweils einen Würfel an jede Ecke platziert.

## Probleme und Schwierigkeiten

Probleme die im Verlauf des Projektes auftreten können:

- Die Kamera erkennt die Würfel auf der Karte nicht richtig. Die Symbole auf der Karte sind vielleicht zu groß oder deckend und die Kamera kann die Farbe nicht korrekt aufnehmen.
- Farbige Lichter, die von der Kamera registriert werden und dadurch die falsche Audiodatei abspielt. Das Problem könnte auch durch stark reflektierende farbige Gegenstände erzeugt werden.
- Die Kamera befindet sich innerhalb der Station und unzureichende Beleuchtung kann dazu führen, dass sie kein gutes Bild aufnimmt. Demzufolge kann das Programm vielleicht keine Auswertung der Farben durchführen.
- Wenn die Qualität der Kamera zu niedrig ist, können wir im schlimmsten Fall die Position des Würfels nicht genau bestimmen. Aber auch eine zu hochauflösende Kamera stellt ein Problem da, denn sie könnte durch die Menge an gesendeten Daten die Anwendung verlangsamen.
- Verzögerung der Anwendung unabhängig von der Kamera.
- Verschiebung der Kamera innerhalb der Station durch Transport oder Stöße.
- Handflächen/Jackenärmel, die auf der Karte liegen, werden falscherweise als Würfel registriert.

## Zeitplan /Meilenstein

### November:

- Abgabe des Konzepts, gegebenenfalls Anpassungen durchführen
- Planung der Station/Würfel. Welche Teile werden benötigt?
- Besorgung von nötigen Materialien
- Javascript Audio-Funktionen testen

Im November sollten die JavaScript Funktionen, die wir nutzen wollen getestet werden und wenn möglich die Grundstruktur des Javascript Teils vorhanden sein, sodass wir sie im nachhinein erweitern können.

### Dezember:

- Station/Würfel Prototypen anfertigen
- Karte anfertigen
- Audiodateien aufnehmen/suchen
- Mittels openCV Würfel erkennen und Werte an Javascript senden

Im Dezember muss die Erkennung der Würfelfarbe mit Position fertig werden. Der Prototyp für die Station wird gegen Ende Dezember fertig gebaut. Audiodateien werden im Verlauf des Dezembers aufgenommen und angepasst.

### Januar:

- Dokumentation anfertigen
- Projekt fertigstellen
- Präsentation und Plakat anfertigen

## Aufwandseinschätzung

- Konzept/Planung: 15h
- Bau der Station mit Würfeln und Karte: 30h
- Aufnahme der Audiodateien/Recherche: 20h
- JavaScript/HTML: 30h
- openCV/C++: 20h
- Präsentation/Plakat: 10h
- Dokumentation: 12h

## Teamplanung

### **Wolf Engeland 2268379 (Projektleiter)**

- Planung und Bau der Station
- Anfertigung der verschiedenen Würfel
- Design der Karte und Symbole
- Teamkoordination

### **Darren Landig 2257637**

- Aufnahmen der benötigten Audiofiles
- Umsetzung der Soundwiedergabe mit JavaScript und HTML
- Verarbeitung von Signalen in Javascript

**Henry Tieu 2223359**

- Umsetzung der Live Videoerkennung mithilfe von openCV/C++
- Verarbeitung der Signale in Javascript
- Design der Karte und Symbole

**David Harder 2269846**

- Bau der Station
- Präsentation und Dokumentation
- Hilfe bei der Videoerkennung mit openCV/C++