HAW HAMBURG

Projektdokumentation

von Shayan Shabab (2364720), Pascal Gluba (2365244), Sebastian Wehber (2365545)

Audio-Video-Pong

Media Systems - AV-Programmierung - Andreas Plaß/Jakob Sudau - Wintersemester 2020/2021

Technische Umsetzung

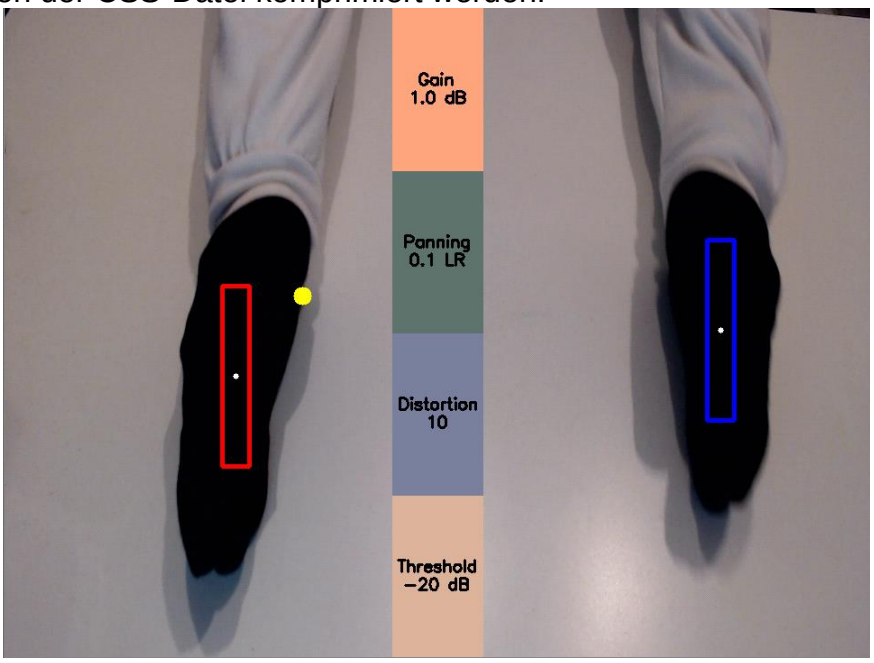
Relativ früh in der Projektarbeit wurde in Python mit openCV2 bewerkstelligen, dass die beiden Hände der Spieler über den Kamerafeed oder einem Video gut erkannt werden.
Hierfür wird das Bild in BGR aufgeilt und nach einem Farbwert gefiltert. Da das Pink der Haut zu sehr von Lichtverhältnissen abhängt, wird nach einem sehr dunklen Farbwert gefiltert. Anschließend werden BGR zu einer Maske multipliziert. Auf diese Maske wird ein 5x5 Medianfilter und eine schließende morphologische Operation angewandt. Für die Konturenerkennung wird das Bild dupliziert und je eine unterschiedliche Seite schwarz gefärbt. Auf dem einen Bild werden die Konturen- und Schwerpunktberechnungen der linken Hand, auf dem anderen der rechten Hand bestimmt.
Zeitgleich wurde über HTML/JavaScript und mithilfe von WebAudio dafür gesorgt, dass mit ersten Gesten mehrere Soundmanipulationen möglich sind. Hierbei wird in JavaScript eine Audiodatei entgegengenommen, welche in Dauerschleife über den Browser abgespielt und live durch die beschriebene Soundmanipulation verändert wird. Die Soundmanipulation geschah in der Anfangsphase noch durch das Schweben des Mauszeigers an mehreren erstellten div-Objekten, später jedoch durch entsprechende Nachrichten, die von Python an JavaScript versendet werden. Jedes Objekt sorgte hier für die Manipulation an anderer Stelle des Tons.
Insgesamt werden bei der Applikation die Werte vom Gain, Panning, Distortion und Threshold verändert. Dafür wird zu Beginn der AudioContext initialisiert und unter diesem werden dann die einzelnen Werte, die manipuliert werden, deklariert. Abschließend werden diese miteinander verknüpft und daraufhin mit dem AudioDestinationNode verbunden:

```
let context = new AudioContext();
let sound = new Audio("trampoline.mp3");
let source = context.createMediaElementSource(sound);
let gain = context.createGain();
let stereoPanner = context.createStereoPanner();
let distortion = context.createWaveShaper();
let compressor = context.createDynamicsCompressor();

source.connect(gain);
gain.connect(stereoPanner);
stereoPanner.connect(distortion);
distortion.connect(compressor);
compressor.connect(context.destination);
```

Für das Spiel Pong werden zwei Rechtecke um die zuvor errechneten Schwerpunkte der Hände gezeichnet. Der Rest ist triviale Mathematik.
Der Ball hat eine konstante X-Geschwindigkeit, die negiert wird, sobald eine der vertikalen Wände oder einer der Schläger berührt werden, sofern dieser Schläger nicht zuvor als letztes vom Ball berührt wurde.
Die Y-Geschwindigkeit tendiert mit einer Dekrementation alle paar Frames gegen Null. Die Y-Position der Schwerpunkte der Hände wird mit der des vorherigen Frames verglichen. Bei Ballkontakt wird diese Differenz auf die Y-Geschwindigkeit

des Balls addiert. Bei Kontakt des Balls mit einer Horizontalen Wand wird die Y-Geschwindigkeit negiert.
Die optischen Elemente wurden später aus der HTML komplett entfernt und in verbesserter Form in Python durch openCV2 gezeichnet. Die sorgt für eine grafisch hübschere Oberfläche und diente ebenfalls dazu den Browser, durch die HTML-Datei, nur zur Soundwiedergabe zu verwenden. Außerdem konnte die HTML durch das Entfernen der CSS-Datei komprimiert werden.



Per MIDI werden von der Python-Datei einzelne Nachrichten bei bestimmten Begebenheiten an die HTML versendet, die durch midi.js diese Nachrichten empfängt, auswertet und durch die handler.js benutzt wird, um entsprechende Soundmanipulationen durchzuführen. Als Schnittstelle für diesen Austausch wird der MIDI-Driver LoopBe1 verwendet. Die Übertragung und die Anwendung der MIDI-Nachrichten geschehen wie folgt:

Python:

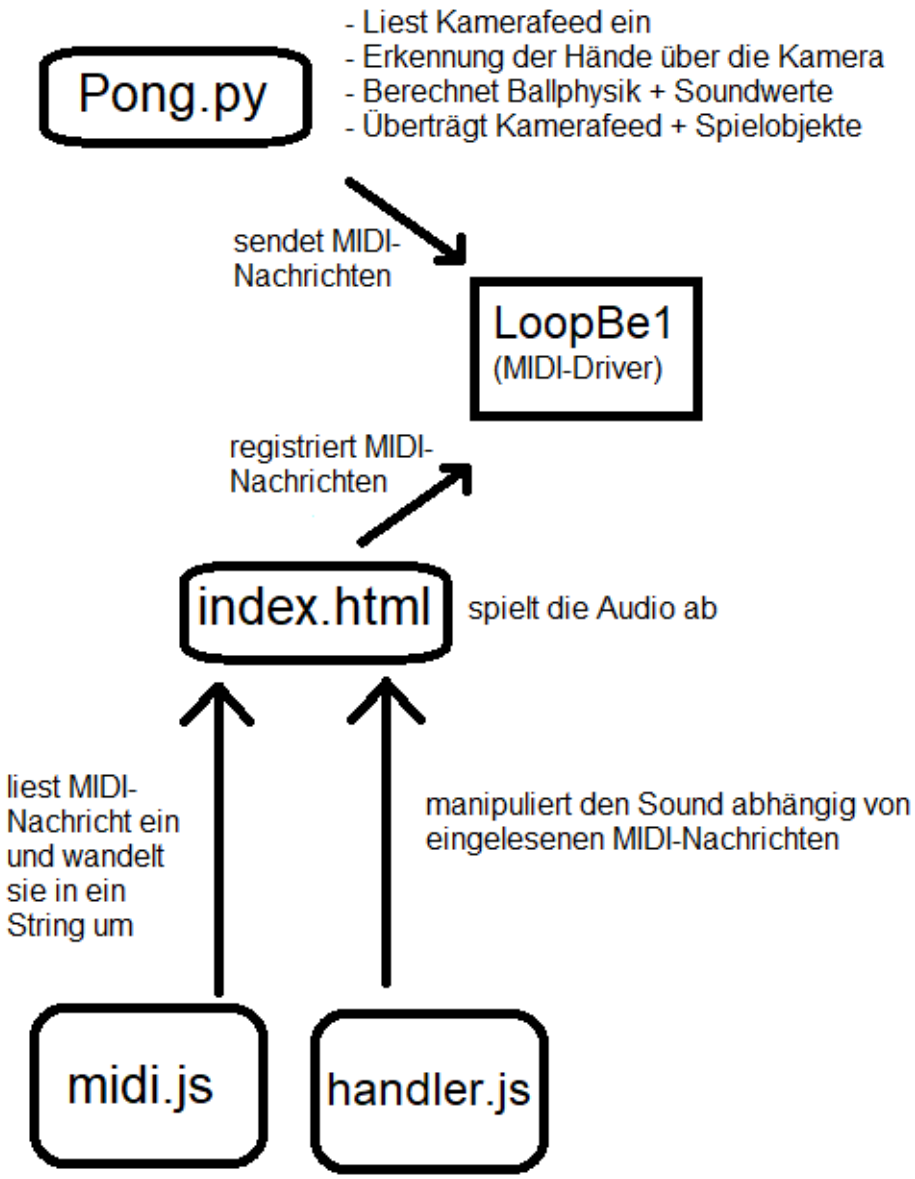
```
midiOutput = mido.open_output("LoopBe Internal MIDI 1")
def sendMIDIMessage(val1, val2):
    midiOutput.send(mido.Message('control_change', control=val1, value=val2))

sendMIDIMessage(5, 1)
```

⬇

JavaScript:

```
if (string == "b0 5 1 " && string != preString){
    gain.gain.value -= 0.1;
    console.log(Math.round(gain.gain.value * 100)/100)
    preString = string;
}
```



Auswertung der Aufwandsschätzung

Ursprünglich war es geplant die Projektarbeit innerhalb von 50 Personenstunden abzuschließen. Der Aufwand erhöhte sich jedoch durch mehrere Probleme, die es uns unmöglich machte, das Projekt nach unserer Grundvorstellung zu gestalten. U. a. sollte das Kamerabild mit den dazugehörigen Spielobjekten ursprünglich über den Browser ausgegeben werden, jedoch blieb der Versuch dies in qualitativer Art und Weise umzusetzen erfolglos. Außerdem beanspruchten größere Umstrukturierungen innerhalb des Codes gegen Ende des Projekts für weitere Verzögerungen. Somit kamen wir zum Ende der Projektausarbeit auf einen Aufwand von 70 – 75 Personenstunden.