

## Projektdokumentation

### Projekt „Lego-Harry-Potter Musikspiel“

Das folgende Projekt spielt ein kleines Musikstück mithilfe von Harry-Potter Figuren ab.

### Projektmitglieder

Nicolas Kolbeck Matrikel-Nr.: 2365259

Melina Zanon Matrikel-Nr.: 2364383

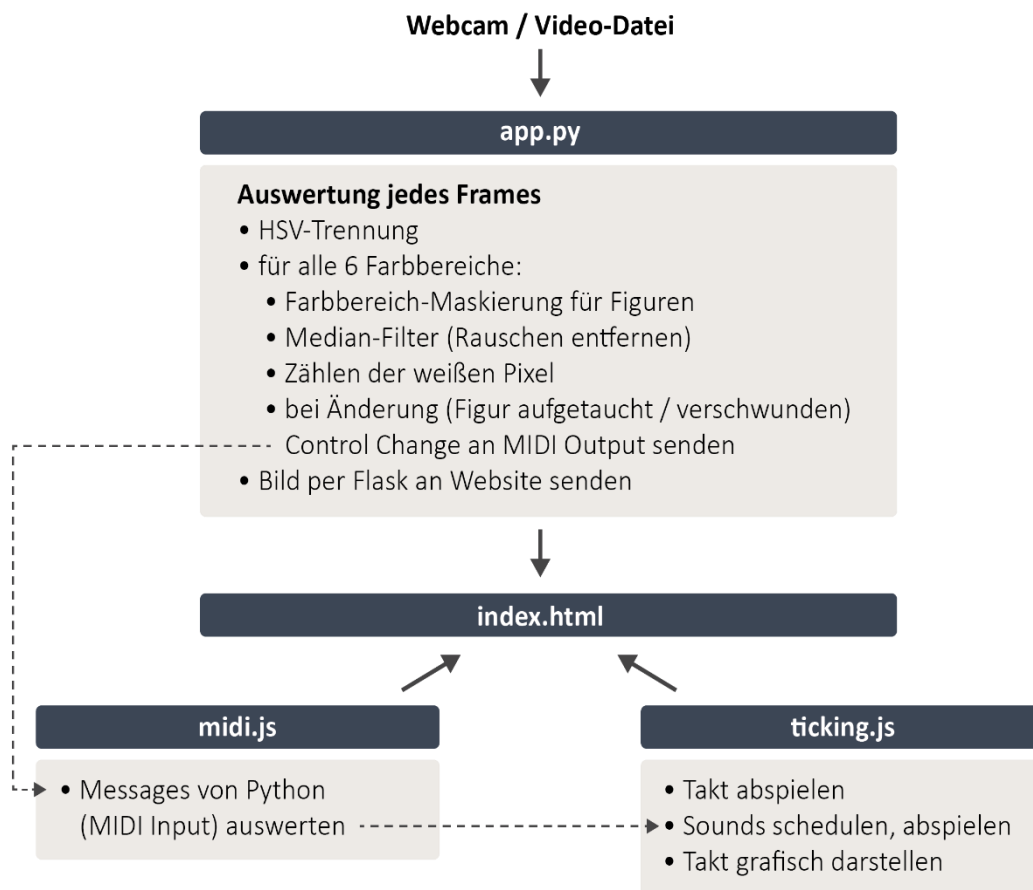
Lan Nhi Le Matrikel-Nr.: 2364449

Git-Repository-URL:

<https://github.com/melinazanon/Lego-Harry-Potter-Musik.git>

### Technische Umsetzung

Die App ist wie folgt aufgebaut:



## Video

Mithilfe von Flask wird über die Python Datei app.py ein Server aufgebaut und das verarbeitete Bild als Video-Feed ausgegeben. Die index.html Datei aus dem Template Ordner wird gerendert und empfängt den Video-Feed und gibt diesen aus.

Jede Figur hat seinen eigenen Farbbereich, welche im Bild maskiert und mit gegebenen Filtern ausgegeben wird. Bei 20 oder mehr Pixel in der Maske wird dementsprechend eine Figur erkannt. Wenn eine Änderung zum vorherigen Frame vorliegt, wird über MIDI-Events diese Daten weitergegeben und verarbeitet.

## Audio

Die Tonspuren werden in AudioBufferSourceNodes gespeichert und passend zum Beat abgespielt. Die Noten starten im Takt immer auf achte Noten. Dafür wird ein System mit zwei verschiedenen Timern benutzt, das auf dem Artikel „A Tale of Two Clocks - Scheduling Web Audio with Precision“ von Chris Wilson basiert.<sup>1</sup>

Es wird alle 25 Millisekunden eine Funktion aufgerufen, die die Tonspuren für die nächsten 2 Sekunden vorausplant. Um Verzögerungen zu vermeiden gibt es bei jedem Funktionsaufruf ein Intervall von 100 Millisekunden, indem die Funktion weitere Male aufgerufen wird, damit keine Tonspuren verpasst werden. Die Zeit ist entsprechend lang, da die längste Tonspur über 4 Takte läuft.

## Auswertung der Aufwandsschätzung

**KW: 49**

06.12.2020:

- Test-Inputs generieren, 1 Stunde
  - Tonspur aufnehmen, 1 Stunde
  - Materialien, Videos, Musikbeispiele, 1 Stunde
- = 3 Stunden

---

<sup>1</sup> Quelle <https://www.html5rocks.com/en/tutorials/audio/scheduling/>

**KW: 50**

13.12.2020:

- Beats-Generierung und evtl. Tonspuren im richtigen Takt abspielen, 5 Stunden

- Marker-Erkennung generell, 3 Stunden

= 8 Stunden

**KW: 51**

20.12.2020: Einzelne Figurenerkennung (ersten 3),

= 3 Stunden

**KW: 53**

03.01.2020: Zusammenführen des Prototyps, 2 Stunden

05.01.2020: Vorstellung bzw. Vorbereitung des Prototyps, 2 Stunden

= 4 Stunden

**KW:1**

10.01.2020: Alle 6 Figuren mit Erkennung, 2 Stunden

17.01.2020: Abschlusspräsentation vorbereiten, 1 Stunde

= 3 Stunden

Geschätzte Arbeitsstunden: 21 Stunden

**Tatsächlicher Aufwand**

**KW 49:** Tonspur aufnehmen 3 Stunden, Rest unverändert = 5 Stunden

**KW 50:** Beats-Generierung 8 Stunden, Rest unverändert = 11 Stunden

**KW 51:** Unverändert = 3 Stunden

**KW 53:** Zusammenführen und Vorbereitung des Prototyps = 7 Stunden

**KW 1:** Unverändert = 3 Stunden

**KW 2:** Verknüpfung der Python und js-Dateien im Browser mit Flask = zusätzlich 2 Stunden

Tatsächliche Arbeitsstunden: 31 Stunden