

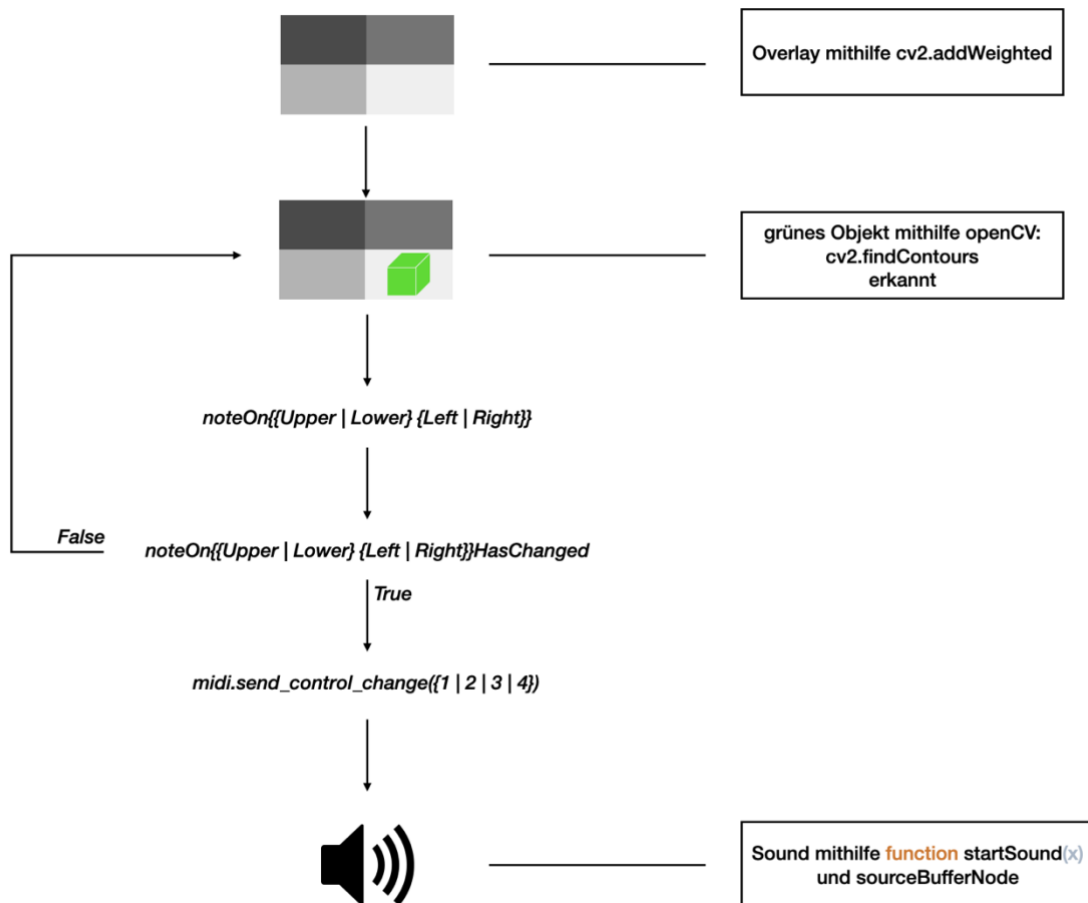
# Green Drums

## **Ein Projekt von**

Nikolas Angrabeit (2217428) ,  
Fabian Braun (2414582) ,  
Maya Kröger (2400039)

URL des github-Repositories: <https://github.com/mayakr26/GreenDrums.git>

## Kurze Beschreibung der technischen Umsetzung



Grafik 1: Diagramm zur technischen Umsetzung

Wie man anhand der Grafik erkennen kann, wird zunächst ein Overlay mit `cv2.addWeighted` über dem LiveFeed der WebCam gelegt. Wird nun ein grünes Objekt in die Kamera gehalten, wird dieses mithilfe `cv2.findContours` erkannt. Wenn sich nun ein grünes Objekt in einer der Zonen befindet, wird die Variable `noteOn{{Upper | Lower}} {Left | Right}}`, je nachdem in welcher Zone es sich befindet, auf True gesetzt. Zugleich wird auch die Variable `noteOn{{Upper | Lower}} {Left | Right}}HasChanged` auf True gesetzt.

Daraufhin wird `midi.send_control_change({1 | 2 | 3 | 4})` ausgeführt. Die `send_control_change()` Funktion wird in der Datei `midi.py` definiert und sendet mithilfe `midiOutput.send(mido.Message('note on', note=control, velocity=127))` die jeweilige Zahl (1 für Hihat, 2 für Crash, 3 für Bassdrum 4 für Snare) an Midi. In der `midi.py` wird dann die

Funktion `startSound(event.data[1])`, welche in der Datei `soundPlayback.js` definiert wird, gestartet. Diese erstellt eine `sourceBufferNode`, mit welcher der jeweilige Sound dann abgespielt wird.

Wenn die Variable `noteOn{{Upper | Lower} {Left | Right}}HasChanged` allerdings `False` ist, wird wieder geprüft, ob sich ein grünes Objekt in den Zonen befindet.

## Ursprüngliche Zeitplanung

SCHRITT	WAS	AUFWAND IN PS
1	Ideenentwicklung	2
2	Interface entwickeln	4
3	Analyse des Webcam-Inputs	6
4	Schnittstelle von Audioprogrammierung und Videoprogrammierung	7
5	Abspielen des Schlagzeugtons in Abhängigkeit von dem Webcam-Input	3
6	Testen	3
7	Dokumentation	6
		<b>INSGESAMT 31 PS</b>

Tabelle 1: Zeitplanung

Die ursprüngliche Zeitplanung war genau geplant und am Ende auch sehr akkurat. Allerdings ist von einer ausführlichen Dokumentation ausgegangen, weshalb mit sechs Stunden gerechnet wurde. Da dies nicht der Fall ist, ist der Aufwand für die Dokumentation lediglich 2 PS. Dafür war der Aufwand für das Testen und die Schnittstelle von Audio und Video höher, weshalb es sich ausgeglichen hat. Daraus resultierend lag der Gesamtaufwand in PS ungefähr bei 31 PS.