

Audio-Video-Programmierung

Medientechnik, HAW Hamburg

Wintersemester 2020/2021

Andreas Pläß, Jakob Sudau

Projektkonzept

„Lego Step Sequencer“

Teilnehmer:

Tim Passgang 2369620

Andrea Rauh 2369762

Kianosh Kinz 2351754



1. Einleitung

Unser Projekt entsteht im Rahmen des Kurses Audio-Video-Programmierung im Studiengang Medientechnik an der HAW Hamburg. Ziel des Kurses ist das Erlernen und Vertiefen der Programmiersprachen Python, JavaScript, HTML und CSS sowie die Verbindung von Bildverarbeitung und Audioprogrammierung. Das Projekt ist die Prüfungsleistung des Kurses und wird am Ende des Semesters in Form des Codes über gitHub abgegeben. Außerdem gibt es eine Präsentation des Projektes vor dem Kurs über eine Online-Plattform.

Hauptbestandteil des Programmcodes ist die Benutzung von openCV für die Bildverarbeitung und der Web Audio API für die Audioprogrammierung.

2. Projektziel

Unsere Software hat das Ziel einen aus Papier und Lego gebauten Step Sequencer per Webcam ins Virtuelle zu übertragen. Die in der Realität stattfindenden Änderungen sollen auf einer Website visualisiert und ausgeführt werden.

Was ist ein Step Sequencer? Ein Step Sequencer ist ein digitales Musikinstrument, welches eine Partitur speichert und an einen Tonerzeuger übermittelt. An dem Sequencer selber sind Tonhöhe, Tonlänge, Geschwindigkeit des Musikstückes und die Reihenfolge der Töne einstellbar. Diese Daten werden dann beispielsweise an einen Computer weitergegeben, der die Partitur in die gewünschten Töne übersetzt.

In unserem Fall wollen wir vier Spuren erzeugen, die jeweils einen ganzen Takt mit acht Achteln beinhalten. Pro Spur sind also acht Achtelnoten einstellbar. Die erzeugten Töne sind in unserem Fall Schlagzeugsounds, um somit einen Schlagzeugbeat zu erzeugen.

Das Layout besteht aus Pappe, unterteilt in 4 x 8 Felder. In jedes dieser Felder kann man einen farbigen Legostein legen. Jede Farbe steht für einen bestimmten Sound. Wenn beispielsweise an die erste Stelle ein roter Legostein gelegt wird und dieser für die Basedrum steht, dann wird auf den ersten Schlag in einem Takt ein Basedrum-Sound abgespielt.

Außerdem sollen neben dem 4 x 8 Feld noch zwei Fader bedienbar sein.

Diese werden einfach als Linie auf die Pappe gemalt. Man hat dann die Möglichkeit einen Legostein auf diese Linie zu setzen und einerseits die gesamte Lautstärke des Audios zu regeln und andererseits die Geschwindigkeit, in der der "gebaute" Takt abgespielt wird.

Die Position der Legosteine wird nun mit einer Kamera abgefilmt. Mittels Software wird die Position sowie die Farbe der Legosteine erkannt. Diese Daten werden genutzt, um die gewünschten Drumsounds an der gewünschten Stelle im Takt über den Computer abzuspielen. Außerdem wird die physische Benutzeroberfläche, bestehend aus Pappe und Lego Steinen, in die virtuelle übertragen, um dem/der NutzerIn Feedback zu geben. Dadurch kann der/die NutzerIn erkennen, ob die gelegten Legosteine richtig erkannt wurden und sieht den gebauten Beat noch einmal visualisiert auf dem Computer.

3. Anforderungsanalyse

Um unseren NutzerInnen eine intuitive und spaßige Erfahrung und ein angenehmes Spielerlebnis zu ermöglichen, sollte die Übertragung der NutzerInnen-Interaktionen möglichst instantan geschehen. Eine visuelle Repräsentation der physischen Gegebenheiten innerhalb weniger als einer Sekunde wäre wünschenswert.

Unsere NutzerInnen sollen die Auswahl zwischen vier Sounds und somit in der physischen Welt zwischen vier verschiedenfarbigen Lego Steinen (grün, orange, blau und rot) haben. Diese sollen dann auf der Oberfläche in kleine vorgefertigte Vertiefungen in der Pappe gelegt werden können um einen bestimmten Sound, je nach Farbe, an einer bestimmten Stelle, je nach Position, in der Takt-Schleife zu erzeugen. Während der Takt-Schleife leuchtet die jeweilige aktive Spalte in der virtuellen Repräsentation auf, um Feedback bezüglich der Sounds zu geben. So ist es sehr leicht ein Gefühl für die Geschwindigkeit des Taktes allein durch das Visuelle zu bekommen.

Wir möchten den BenutzerInnen acht mögliche Positionen für die Legosteine pro Spur geben. Somit gibt es acht Abspielmöglichkeiten pro Spur.

Mit Hilfe der vier Spuren können die NutzerInnen bis zu vier Sounds gleichzeitig abspielen und somit ein Interessantes Klangerlebnis erzeugen.

Wenn der Takt nun zu schnell oder zu laut erscheint, sollen zwei Fader rechts neben dem Step Sequenzer eine Veränderung dieser Parameter ermöglichen. Genauso wie das Plazieren der Legosteine soll auch hier eine möglichst schnelle Übertragung von der physischen in die virtuelle Benutzeroberfläche geschehen. Hier wären Latenzen von unter 100 ms wünschenswert. Ein Fader fühlt sich nur bei einer konstanten Datenübertragung und Verarbeitung der Fader Werte gut an. Ein kontinuierliche Feedback ist unserer Meinung nach für ein angenehmes Bedienen unumgänglich.

Auf der virtuellen Benutzeroberfläche sollen unsere NutzerInnen außerdem eine Start/Stop Taste vorfinden um die Taktschleife nach Belieben mit der Hilfe eines Mausklicks zu unterbrechen und wieder abzuspielen. Zudem soll es hier über ein Drop-down Menü möglich sein, für die verschiedenen Farben verschiedene Sounds einzustellen.

4. Technische Rahmenbedingungen

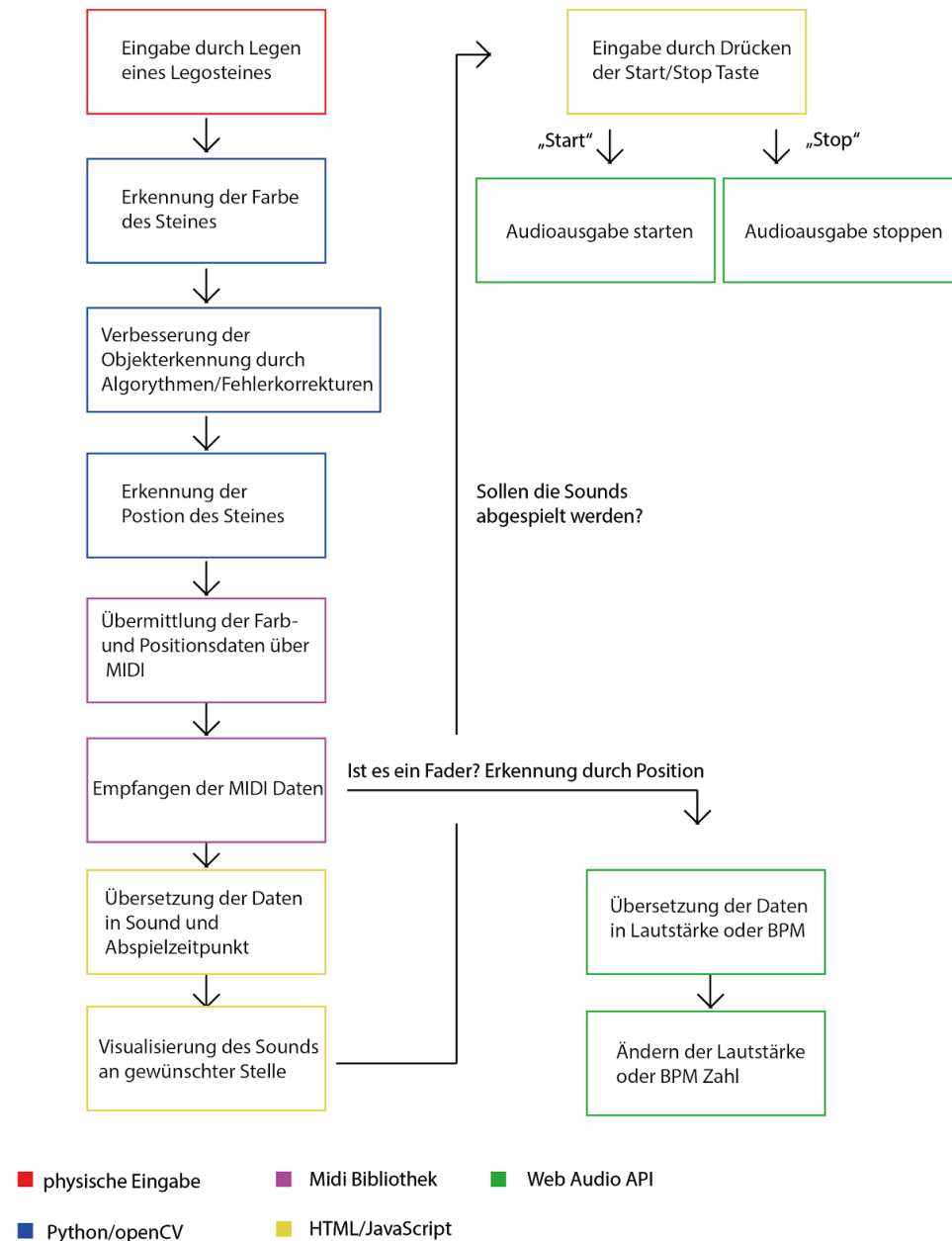
Auf der abstraktesten Ebene besteht unser Projekt aus einem Windows Computer und einer Webcam, die unsere physische Benutzeroberfläche auf den Computer in Form einer Website überträgt. Die von den NutzerInnen hergestellten Beats sollen dann über Lautsprecher ausgegeben werden.

Die entstehende virtuelle Benutzeroberfläche und Website wird mit Hilfe von HTML, CSS und Javascript umgesetzt. HTML und CSS sind in der modernen Webentwicklung Standards um eine schöne und übersichtliche Website zu erzeugen. Javascript ermöglicht hingegen eine Interaktion der BenutzerInnen mit der erstellten Website.

Die Erkennung unserer physischen Benutzeroberfläche möchten wir mit Hilfe von openCV und der Programmiersprache Python umsetzen. Die Webcam filmt die Oberfläche, openCV verarbeitet daraufhin das Video Signal und sendet über das MIDI Protokoll relevante Daten (Position und Farbe der Legosteine) an unseren Javascript-Teil. Innerhalb von Pythons verwenden wir verschiedene Bibliotheken. Zum einen die schon genannte openCV Bibliothek, sowie Numpy, rtmidi und mido. Numpy ermöglicht in Kombination mit openCV eine einfache Verarbeitung eines Videostreams. Rtmidi und Mido hingegen sind Bibliotheken mit deren Hilfe das Senden der Daten über das MIDI Protokoll erleichtert werden. Alle Bibliotheken sind uns im Rahmen der Vorlesung vorgestellt und von uns als optimal empfunden worden.

Während der Testphase verwenden wir das VNPK - Virtual Midi Piano Keyboard. Hiermit können wir einzelne Teilabschnitte des Projektes separat voneinander testen. So kann zum Beispiel ohne vom Videoteil Daten zu empfangen der Audioteil programmiert und getestet werden. Dies ermöglicht einen schnelleren und produktiveren Arbeitsfluss.

5. Technisches Konzept



Am Anfang unserer Software steht die Erkennung der Legosteine.

Um diese aus dem Video heraus zu erkennen benutzen wir openCV. Wie in der Vorlesung erlernt wollen wir das eingelesene Bild der Webcam erst einmal in den HSV Raum (Hue, Saturation, Value) konvertieren und drei einzelne Bilder daraus erzeugen.

Nun werden wir mit der inRange Funktion einen durch Testen ermittelten Farbbereich aus dem Bild filtern, um somit die Farben der Legosteine erkennen zu können.

Um den erkannten Bereich von Fehlern zu befreien und den Legostein noch eindeutiger zu erkennen wollen wir einige Filteralgorithmen nutzen, die wir in der Vorlesung kennengelernt haben.

Nachdem wir die Farbe der Steine erkannt haben, müssen wir die Position bestimmen. Um einen relativen Bezug der Steine zu der Papierfläche zu haben, wollen wir zwei bis vier Eckpunkte aufzeichnen. Diese Punkte helfen uns dabei die Position der Steine relativ zu dem gesamten Feld zu finden, auch wenn die Kamera oder das Feld etwas verrutscht oder der Abstand der Kamera zum Feld sich verändert.

Die Erkennung der Punkte soll entweder über Farbigkeit oder Form stattfinden. Wie genau wir das umsetzen müssen wir durch Testen herausfinden.

Nachdem wir also die Eckpunkte erkannt haben, können wir diese als Eckpunkte für ein Koordinatensystem nutzen.

Nun berechnen wir den Schwerpunkt der vorher erkannten Legosteine. Die Position des Schwerpunktes wird dann als x- und y-Koordinate gespeichert und mit vorher definierten Bereichen verglichen.

Liegt der Schwerpunkt in einem der vordefinierten Bereiche, dann haben wir eine gültige Eingabe und können nun die Daten über MIDI weiterschicken und verarbeiten.

Die x- und y-Koordinaten werden aber vorher übersetzt in Zeile und Spalte, um die Weiterverarbeitung zu vereinfachen.

Die MIDI-Daten, die nun die Farbe und die Position der erkannten Legosteine enthalten, werden daraufhin an den JavaScript-Teil gesendet.

In dem JavaScript müssen diese Daten nun übersetzt werden in den jeweiligen Sound und den Zeitpunkt, wann er abgespielt werden soll.

Durch eine vorher definierte Sound-Farbe und eine Position-Zeitpunkt Tabelle sollte dies sehr einfach passieren.

Wenn der Sonderfall eintritt, dass der erkannte Legostein einen Fader darstellen soll, was durch die Position erkannt wird, dann ändert das JavaScript den jeweiligen Parameter, entweder die Lautstärke oder die Geschwindigkeit.

Die gelegten Steine und die Veränderung der Fader werden über die Website visualisiert. Das JavaScript ändert dann die Farbe der jeweiligen Div-Felder oder die Position des Faders, sowie den eingestellten Wert des Faders.

Für das Abspielen der Sounds nutzen wir die Web Audio API.

Bevor die Sounds nun abgespielt werden wird erst geprüft, ob der Nutzer auf der Website die Start/Stop Taste gedrückt hat und den Beat abspielen möchte.

Ist dies der Fall, werden die Audionodes zusammengesteckt und die Sounds zu den definierten Zeitpunkten abgespielt. Außerdem wird der aktuelle Zeitpunkt im Takt auf der Website visualisiert.

6. Bedienkonzept

Unser Programm soll sowohl über ein physisches als auch über ein virtuelles Feld bedienbar sein. Hierbei ist das Haupt-Bedienfeld die physische Pappe, auf der sich das 4 x 8 Feld befindet und die Fader gezeichnet sind (siehe Abb. 1). Zusätzlich stehen mehrere, verschiedenfarbige Legosteine zur "Bedienung" des Feldes zur Verfügung. Um die Audioausgabe zu starten, muss eine Start/Stop Taste auf der virtuellen Oberfläche per Mausklick bedient werden (Siehe Abb. 2). Dem/Der NutzerIn ist hierbei überlassen ob er/sie die Audioausgabe startet bevor oder während er/sie den Sequencer einstellt.

Auf dem virtuellen Bedienfeld sieht man in der Legende, welche (Legostein-) Farbe für welchen Sound steht. Gegebenenfalls lassen sich hier auch in einem Drop-down Menü selbstständig Farben zu Sounds zuordnen. Legt man nun einen farbigen Legostein seiner/ihrer Wahl auf eines der Felder in der physischen Benutzeroberfläche, bekommt man ein direktes Feedback ob die Übertragung geklappt hat und auf dem Bildschirm erscheint das "bediente" Feld in der gewünschten Farbe. Alternativ lassen sich auch im virtuellen Bedienfeld den Feldern Farben per Mausklick zuordnen, das Hauptaugenmerk unseres Projektes liegt jedoch auf der Bedienung des Sequencers über die Legosteine.

In der virtuellen Legende lässt sich auch die Funktion der Slider ablesen. Auch hier steht gegebenenfalls wieder eine Auswahl zur Verfügung, um den Slidern eine andere Funktion zuzuordnen. Jeder Slider benötigt im physischen Feld einen Legostein zur Bedienung. Zum Einstellen der Fader nimmt man den Legostein (die Farbe spielt in diesem Fall keine Rolle), und schiebt ihn auf den gezeichneten Slidern (rechts in Abb. 1) hoch bzw. runter oder legt den Stein einfach an die gewünschte Stelle. Auch hier erhält der/die NutzerIn wieder ein Feedback über das virtuelle Bedienfeld und kann die veränderte Position des Sliders ablesen. Über den Slider 1 lässt sich in diesem Fall die Lautstärke des Audiosignals ändern. Über den zweiten Slider lässt sich die Geschwindigkeit des abgespielten Beats einstellen. Für Letzteren bekommt man ein direktes visuelles Feedback (auch wenn der Play Button noch nicht gedrückt wurde) und die 8 Spalten "leuchten" nacheinander in der angegebenen Geschwindigkeit auf und zeigen dem/der NutzerIn hiermit, welche Spalte gerade "abgespielt" wird.

Sofern die Start/Stop Taste bedient wurde, werden in der eingestellten Geschwindigkeit und Lautstärke (die Ausgangsgeschwindigkeit soll 90 BPM betragen) nacheinander die auf den acht Spalten liegenden Töne abgespielt. Sounds bzw. Legosteine, die in derselben Spalte liegen werden gleichzeitig abgespielt. Das Ergebnis ist ein Beat, der sich während des Abspielens durch Umlegen und Verschieben der Legosteine verändern lässt. Durch ein weiteres Betätigen der Start/Stop Taste stoppt die Audioausgabe.

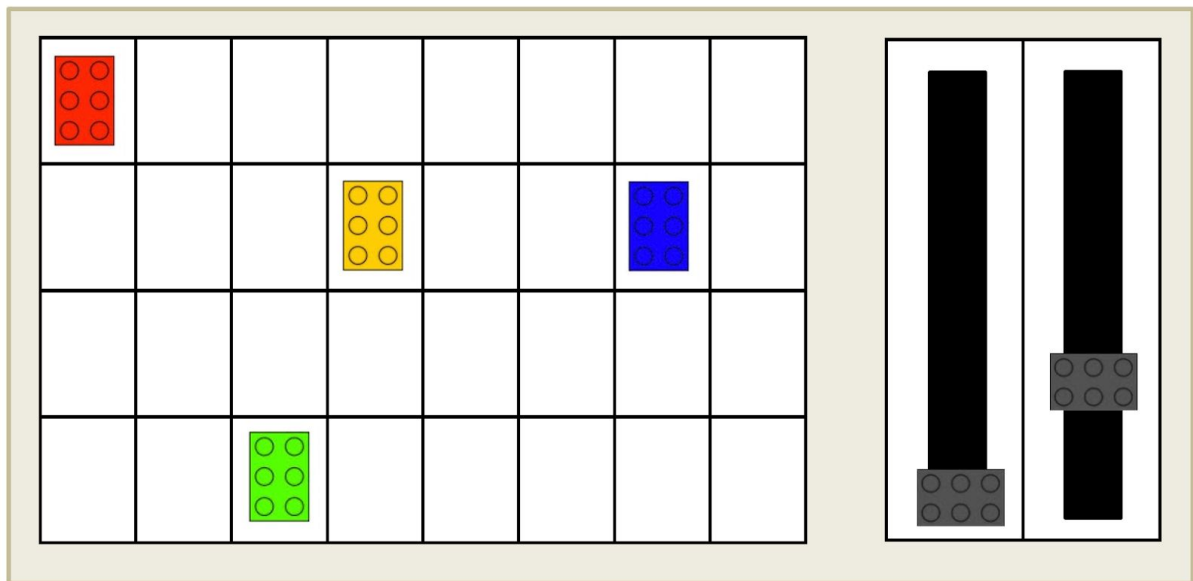


Abb. 1: Skizze physisches Bedienfeld

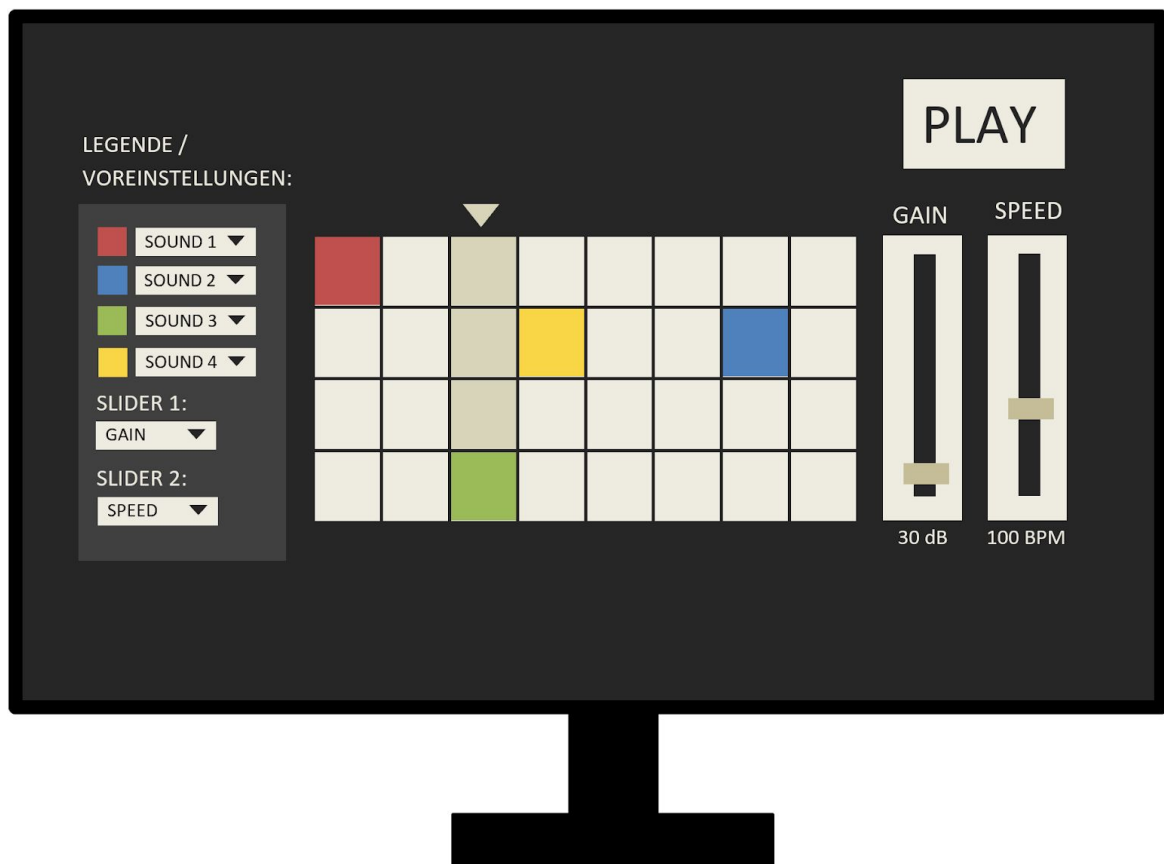


Abb. 2: Skizze virtuelles Bedienfeld

7. Zeitplan

Aufgaben	Fertigstellungszeitpunkt	Arbeitsaufwand in h
<i>Dezember</i>		
Ideenfindung und Konzepterstellung	01.12	3
Videoerkennung von verschiedenfarbigen Steinen	12.12	5
Erstellung der Website (visuell, HTML, CSS)	12.12	8
Bau eines ersten Prototypen für Testzwecke	12.12	3
Positionserkennung von den Steinen	17.12	5
Transfer von MIDI Daten von unserem Videoteil zum Audioteil	17.12	8
Erstellung der Funktionalität der Website ohne Video Anteil (Javascript)	17.12	15
Zwischenstand evaluieren	20.12	3
<i>Januar</i>		
Zusammenführung Video- und Audio Teil	03.01	15
Tests und Korrekturen	04.01	offen, je nachdem wieviel sich ergibt
Fertigstellung eines vorzeigbaren Prototyps	05.01	10
Anpassungen, weitere Tests und Korrekturen	10.01	offen, je nachdem wieviel sich ergibt
Präsentation des Projektes	15.01	1
<i>Abgabe des Projekts</i>	26.01	0,5

8. Teamplanung

Das Projekt soll von den StudentInnen Tim Passgang, Kianosh Kinz und Andrea Rauh realisiert werden. Wir planen mit der Plattform *Trello* zu arbeiten. Trello ist ein Online-Dienst zum Verwalten von Aufgaben für Projekte mit mehreren TeilnehmerInnen. Dort lassen sich Boards erstellen, die mit Listen, Terminen oder Kommentaren versehen werden können. Gemeinsam möchten wir eine Aufgabenliste mit kleineren "To-Dos" erarbeiten wie zum Beispiel "Farberkennung", "Midi-Kommunikation einbauen" oder "Abstandserkennung mit openCV". An diesen Aufgaben kann dann jeder selbstständig arbeiten und je nach Bearbeitungsstand in ein Board einfügen mit Labels wie "in Bearbeitung" oder "gelöst". An dem Code bzw. den Codes arbeiten wir gemeinsam mit der Verwendung von *Github*, ein Projekt-Verwaltungsdienst für Software-Entwicklung.

Unser Ziel ist, dass alle TeammitgliederInnen mit gleicher Verantwortung und Arbeitsaufwand an dem Projekt teilnehmen. Ein Problem, welches wir in den letzten Semestern bei Projekten mit dem Schwerpunkt Programmieren oft hatten, war, dass im Verlauf des Projektes eine Person den Großteil des Programmierens übernommen hat. Das führte dazu, dass die anderen Teammitglieder Schwierigkeiten hatten den Code zu verstehen, da dieser schon sehr fortgeschritten war, ohne dass Zwischenstände kommuniziert wurden. Das Resultat war, dass am Ende eine Person den Hauptteil des Codes geschrieben und verstanden hat und die anderen GruppenmitgliederInnen keine Möglichkeit mehr hatten den Code zu verstehen oder beim Programmieren zu helfen. Diese Problematik wollen wir von Anfang an umgehen, indem wir durch regelmäßige Treffen und Absprachen uns gegenseitig updaten und über Trello kontrollieren, dass Aufgaben gleichmäßig und gerecht verteilt werden. Auf eine Projektleitung verzichten wir.