Audio-Video-Programmierung

Medientechnik, HAW Hamburg

Wintersemester 2020/2021

Andreas Plaß, Jakob Sudau

Projektdokumentation

„Lego Step Sequencer“

Teilnehmer:

Tim Passgang 2369620

Andrea Rauh 2369762

Kianosh Momo Kinz 2351754

1. Einleitung

Im Folgenden werden wir eine kurze Beschreibung der technischen Umsetzung unseren Projektes geben. Des Weiteren werden wir auf unsere Aufwandsschätzung eingeben, welche wir zuvor im Projektkonzept erstellt hatten. Wir werden schauen wie lange gewisse Arbeiten tatsächlich gedauert haben und können so bei kommenden Projekten bessere Aufwandsschätzungen erstellen.

2. Technische Umsetzung

openCV / Python:

Der Videoteil dieses Projektes besteht aus einer Schleife, die durchgehend läuft, solange das Bild der Webcam abgefragt wird. Innerhalb dieser Schleife werden dann unterschiedliche Funktionen aufgerufen.
Als Erstes wird das Bild in den HSV-Bereich konvertiert, weil damit bessere Farberkennung möglich ist.
Am Anfang müssen einmalig unsere beiden Eckpunkte erkannt werden, mit denen wir später die Legosteine relativ zu unserem Spielfeld verorten können.
Dadurch, dass wir dies nur einmal am Anfang machen, können die hierfür genutzten, blauen Steine auch später als Soundsteine genutzt werden und müssen nicht die ganze Zeit als Eckpunkte im Bild liegen. Dafür wird als erstes die “Maskenfunktion” aufgerufen. Diese filtert den ausgewählten Farbbereich unserer Ecksteine heraus, verbessert die Filterung mit einigen Algorithmen und gibt dann ein Schwarz-Weiß-Bild heraus, in welchem alles außer unsere Ecksteine schwarz ist. Dieses Bild wird nun in der “Schwerpunktfunktion” genutzt, um Konturen um die erkannten Steine zu ziehen und daraus den Schwerpunkt, also den Mittelpunkt des Steines als X- und Y-Koordinate auszugeben.
Diese Koordinaten werden nun als Eckpunkte gespeichert.
Um diese Funktionen zur Erkennung der Eckpunkte nur einmal aufzurufen, haben wir einen Counter eingebaut, der bei jeder Ausführung der Schleife hochzählt. Die Funktion wird nur bei Counter = 0 aufgerufen. Als Nächstes werden die oben genannten Funktionen (“Maske- und Schwerpunktfunktion”) für jede mögliche Farbe der Steine ausgeführt. Somit erhalten wir für jeden Stein die X- und Y-Koordinate. Diese werden in unterschiedlichen Arrays gespeichert, somit wissen wir auch die Farbe der jeweiligen Steine.
Dann werden diese X- und Y-Koordinaten mit der “PinKoordfunktion” in Relation zu unserem Spielfeld gesetzt. Unser Spielfeld wird dort in Felder eingeteilt, in dem sich ein Stein befinden kann.

In dieser Funktion wird auch zwischen einem Spielstein, der einen Sound abspielen soll und einem Stein, der einen Fader repräsentiert, unterschieden.
Wenn der Stein ein Fader sein soll, wird die Position in einen Wert zwischen 0-127 gewandelt, um ihn dann per Midi an die Website zu senden.
Wenn der Stein einen Sound repräsentieren soll, wird seine Position übersetzt in das Feld, in dem sich der Stein befindet.

Javascript:

Es werden Midi Daten vom Python-Part empfangen und in ein “felder”-Array in der Javascript-Datei eingetragen. Das Array hat 34 Stellen, die mit jeweils fünf verschiedenen Werten gefüllt werden können. Jede Stelle repräsentiert hierbei eines der Bedienfelder des Sequencers, der im Browser zusätzlich visualisiert wird. Die fünf Werte stehen für die fünf verschiedenen Farben der Legosteine, die zuvor im Python-Code erkannt wurden (In den Kommentaren im Javascript Code lässt sich ablesen welcher Wert zu welcher Farbe gehört).
Bei jedem Eingang einer Midi Note werden zwei Funktionen aufgerufen: Die *changeColor()*-Funktion greift auf das CSS Stylesheet zu und ändert dort aufgrund der Werte im *„felder“*-Array die Farbe des zugehörigen Bedienfeldes. Die Funktion *changeSliderParamMIDI()* ändert die Slider-Position im Browser anhand der im Array stehenden Daten. Die Midi Eingänge überschreiben immer den Website User Input.
Der Kern des Javascript-Codes ist die *playBeat()*-Funktion. Diese wird mithilfe der *setTimeout()*-Funktion jeden Takt neu aufgerufen. In ihr werden zwei for-Schleifen für die Initialisierung der Sounds aufgerufen.
Die Funktion *whichSound()* greift wieder auf den Style der Bedienfelder zu. Sie untersucht welche Farbe an den jeweiligen Feldern in der aktuellen Spalte vorliegt, ordnet jeder Farbe einen Sound zu und legt die Start Spielzeit der dann ausgewählten Sounds fest. Sie erreicht dies, indem ihr beim Starten der Funktion die genaue Spalte i und Zeile j mitgegeben wird.
Des Weiteren wird in der *playBeat()*-Funktion eine visuelle Animation des Step Sequencers im Browser durch Manipulation der Opacity festgelegt. Hier werden genau wie bei dem wiederholten Aufrufen der *playBeat()*-Funktion *setTimeouts* verwendet, die zeitlich alle Achtelnote aufgerufen werden.
Wenn kein Midi Input zur Verfügung steht, kann die Website auch autark verwendet werden. Beim Anklicken eines Feldes im Browser wird durch einen *EventListener* der Wert an der gleich bezifferten Stelle im *„felder“*-Array geändert. Zusätzlich wird wieder die *changeColor()*-Funktion durch das Anklicken aufgerufen, welche dann das Feld passend zum neuen Array-Wert einfärbt. Auch die Positionen der Slider lassen sich per Mausklick im Browser ändern.

Die *playBeat()*-Funktion wird schließlich durch einen ein Anklicken des „Start Buttons“ im Browser aufgerufen– ebenfalls durch einen *EventListener* - und kann so auch wieder gestoppt werden.

3. Abbildungen:

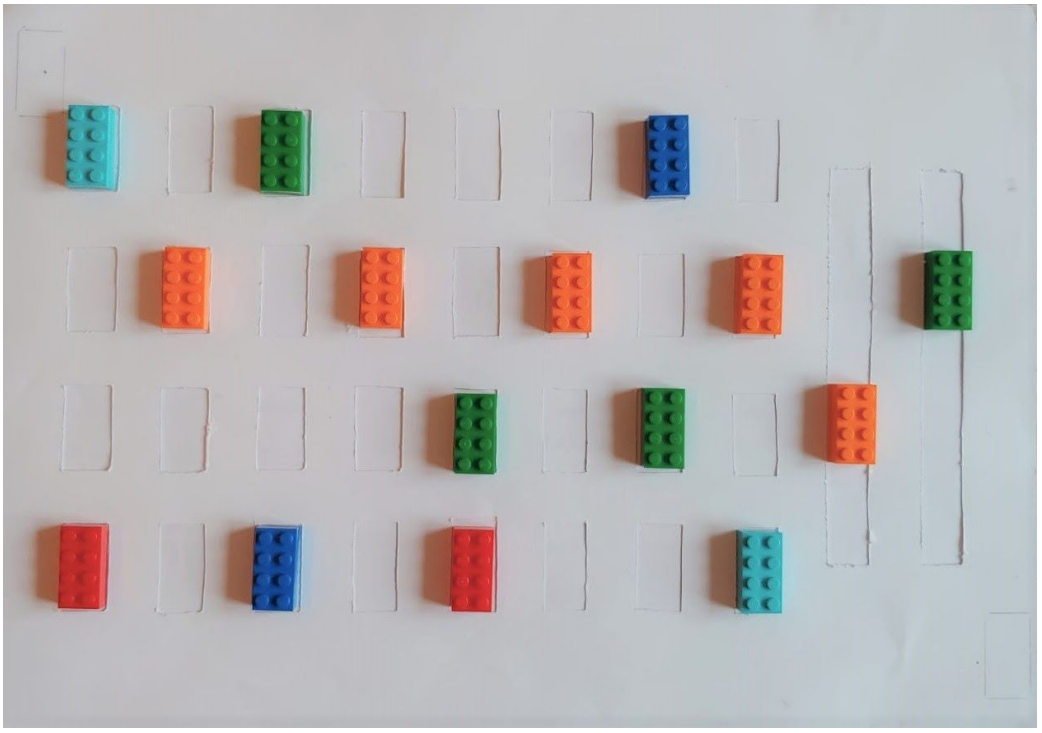

Abb. 1: Abbildung des physischen Eingabefeldes (nach Erkennung der Eckdaten)

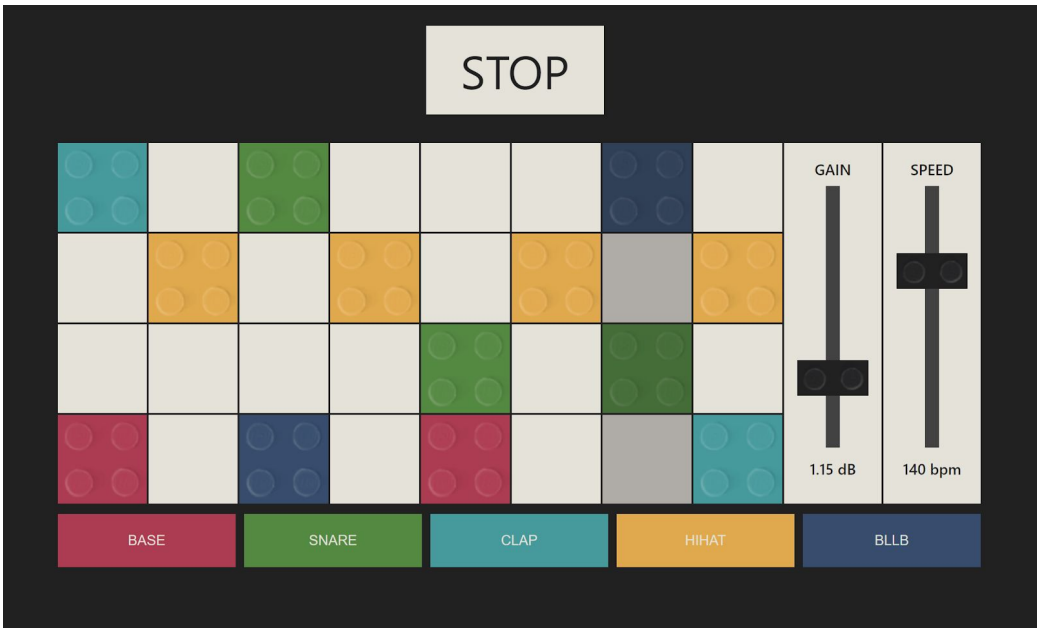

Abb. 2: Screenshot des (Ein-) und Ausgabefeldes im Browser (während des Abspielens)

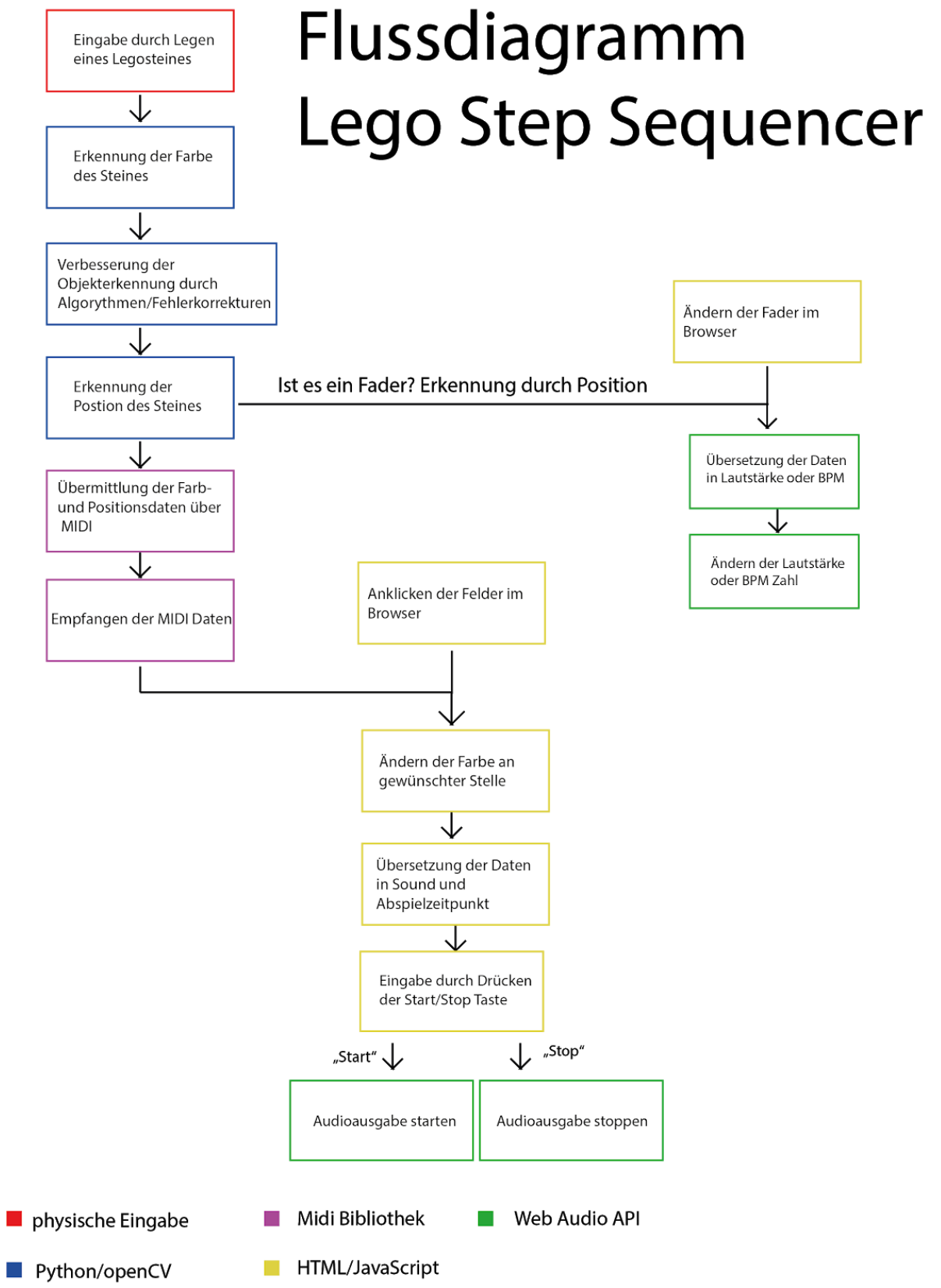

Abb. 3: Flussdiagramm

3. Aufwandschätzung

Aufgaben	**geschätzter Arbeitsaufwand in h**	**echter Arbeitsaufwand in h**
Ideenfindung und Konzepterstellung	3	5
Videoerkennung von verschiedenfarbigen Steinen	5	3,5
Erstellung der Website (visuell, HTML, CSS)	8	6
Bau eines ersten Prototypen für Testzwecke	3	1
Positionserkennung von den Steinen	5	8
Transfer von MIDI Daten von unserem Videoteil zum Audioteil	8	3
Erstellung der Funktionalität der Website ohne Video Anteil (Javascript)	15	10
Zwischenstand evaluieren	3	3
Zusammenführung Video- und Audio Teil	15	10
Tests und Korrekturen	offen, je nachdem wieviel sich ergibt	12
Fertigstellung eines vorzeigbaren Prototyps	10	2
Präsentation des Projektes	1	1
Abgabe des Projekts	0,5	0,5
Gesamt	76,5	65

Generell konnten wir das Projekt mit weniger Arbeitsaufwand als erwartet abschließen. Durch die vorhergegangenen Vorlesungen wurden uns viele Grundkonzepte mitgegeben, die wir dann “nur noch” ein wenig anpassen mussten. Natürlich ergaben sich unvorhergesehene Probleme. Allerdings konnten auch diese mit Hilfe von Google und regelmäßigen Absprachen im Team gelöst werden. Zudem funktionierte unsere Arbeitseinteilung mit Hilfe des Projektmanagement Tools Trello ziemlich gut. Die anfängliche Furcht vor einem Alleingang einer Person konnte somit nicht bestätigt werden. Alles in allem sind wir mit dem Projekt sehr zufrieden. Das Spielen und Bauen der Beats bringt sehr viel Spaß!