Bau deinen Beat

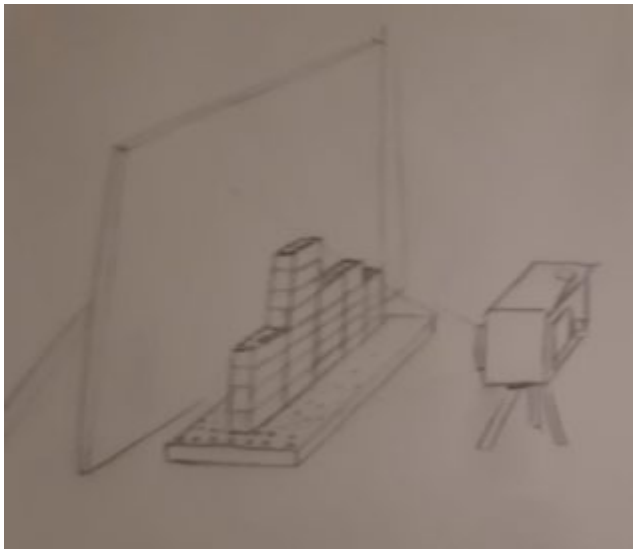
AVPRG Konzept von Paul Giese und Vincent Holtorf

1. Einleitung

Das Projekt „**Bau deinen Beat**“ oder kurz „**BdB**“ ist Teil der Prüfungsleistung für den Wahlpflichtkurs Audio-Video-Programmierung. Hierfür werden wir ein Audio-Video Projekt mit den Sprachen OpenCV und JavaScript erstellen. Da wir fanden, dass die Mischung aus dem Audio und Video Bereich ein spannenderes und interessanteres Projekt ergeben würde, als ein Projekt mit nur einem von beiden Themen Bereichen, haben wir uns für die Kombination aus beiden Themen entschieden.
Der Name des Projektes ist, wie oben erwähnt, „**Bau deinen Beat**“, da der User mit Hilfe von „Legosteinen“ mehrere Beats erstellen soll, in dem er diese nebeneinander und aufeinander stellt.
Bei der Abgabe werden wir einen Prototypen ausstellen, damit unser Projekt anschaulicher ist und die Leute bei der Ausstellung unser Programm ausprobieren können.
Zur Veranschaulichung werden wir noch ein Plakat vorbereiten, auf dem erklärt wird, was unsere Idee ist und eine kleine Anleitung zeigen, wie das Projekt funktioniert.

2. Projektziel

Das Ziel unseres Projektes ist, dass der User verschiedene farbige „Legosteine“ zur Verfügung hat, wobei jede Farbe ein anderes Instrument darstellt. So wird zum Beispiel ein grüner Stein die Gitarre und ein roter Stein die Drum darstellen. Mit diesen Steinen soll der User dann ein 2D-Gebilde ganz nach seinen Wünschen erstellen.



Dieses wird live von der Kamera aufgenommen und durch openCV, MIDI und JavaScript in verschiedene Töne umgewandelt. Diese Töne bilden zusammen einen Beat. Über den Computer wird dieser gespeichert und in einem Loop wiedergegeben. Durch das Zusammenstellen dieser verschiedenen oder gleichen Beats, erstellt der User so eine Musikspur. Über das UI können dann auch noch die Beats einzeln ausgewählt werden und mit verschiedenen Parameter bearbeitet werden wie z.B. Filter, die Dynamik oder Temperatur.

3. User Stories

- Der User kann mit jedem Stein direkt den Beat verändern, den er am Erstellen ist.
- Der User kann nach dem Bauen den Beat am Computer weiterbearbeiten.
- Ein automatischer Loop muss staatfinden.
- Umso höher der Stein im Raster ist, desto höher muss der Ton sein.
- Nimmt der User einen Stein weg, ist der Ton beim nächsten Loop verschwunden.
- Die Kamera sollte einfach und schnell neu kalibriert werden können.
- Die Übertragung durch den MIDI muss schnell geschehen.
- Es müssen mehrere Beats gespeichert werden können.
- Auch bei leicht veränderten Lichtbedingungen, muss das Programm die Farben erkennen.
- Der User kann bei der Bearbeitung die Auswirkungen der Filter direkt hören und sehen.
- Der User kann einzelne Töne lauter und leiser stellen.
- Die Filterauswahl ist leicht anwendbar (auch für Laien).
- Der User kann den Loop jederzeit starten und stoppen.
- Der User kann über die UI mehrere Beats hintereinanderlegen und -abspielen lassen.
- Die gebaute Musikspur aus einem oder mehreren ggf. bearbeiteten Beats soll lokal gespeichert werden können.

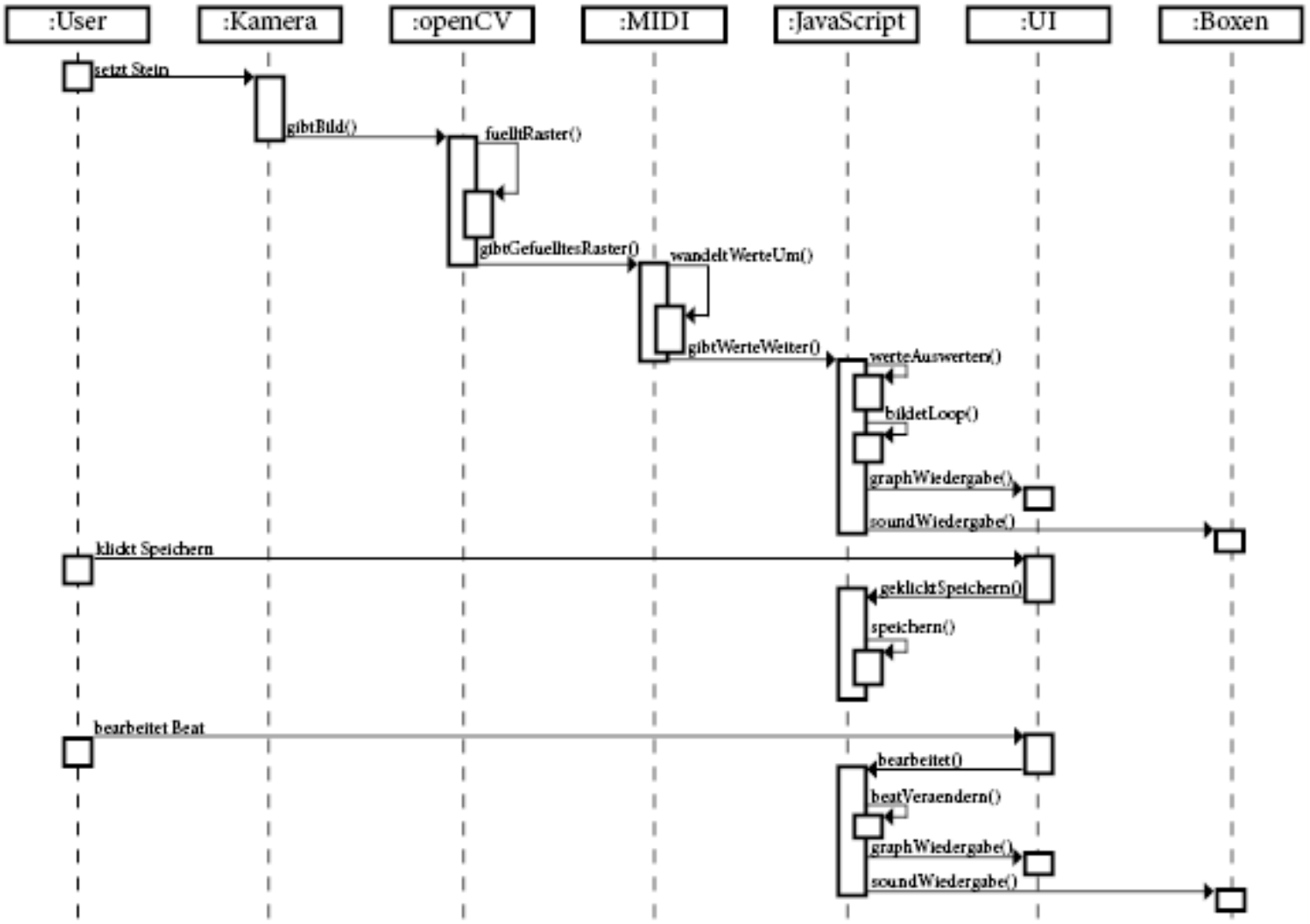
4. Technische Rahmenbedingung

Wir werden „**BdB**“ auf einem Windows Desktop-Computer mit den Programmiersprachen OpenCV(C++) und JavaScript schreiben. Verbinden werden wir diese beiden Sprachen über einen MIDI.

Diese Sprachen wurden uns durch den Kurs vorgegeben, da wir auch schon die Übungen in den Vorlesungen hiermit gemacht haben. Dazu sind diese beiden Sprachen auch gut für das Projekt geeignet, da C++ direkt die Hardware ansprechen kann und nicht erst noch über komplizierte Aufrufe agieren muss. Über den MIDI werden wir dann die Werte der Kamera umwandeln, sodass wir diese in JavaScript nutzen können. JavaScript nutzen wir, da wir mit dieser Sprache gut und einfach verschiedene Funktionen auf einer Website aufrufen können und aus unseren vermittelten Werten einen Beat bauen und veranschaulichen können.

5. Technisches Konzept

Um die Farben, die der User zuvor als „Lego Wand“ gebaut hat, anständig erfassen zu können, werden wir das, von der Kamera aufgenommene Bild, rastern. C++ entscheidet für die einzelnen Felder des Rasters in Echtzeit, ob darin nichts ist (weißer Hintergrund) oder überwiegend eine Farbe. Wie zuvor erwähnt, sollen die verschiedenen Farben der Steine für verschiedene Instrumente stehen. Alle Spalten des Rasters zusammen stellen dabei einen einzelnen Beat dar, der immer wieder wiederholt wird. Je nach dem in welcher Spalte das Rasterfeld mit der erkannten Farbe liegt, beginnt der jeweilige Ton früher oder später im Loop zu spielen. Je nach dem in welcher Zeile die Farbe erkannt wird. Wird die Frequenz des Tons erhöht oder verringert. Das bedeutet dementsprechend auch, dass nicht im gleichen Moment das gleiche Instrument in der gleichen Tonhöhe abgespielt werden kann.
Wenn ein fertiger Loop gebaut wurde und der User mit seinem aus „Lego“ erbauten Beat zufrieden ist, kann er nun an einem Computer den Beat speichern und weiterbearbeiten. Er kann einzelne Filter auf einzelne Tonspuren legen oder Töne lauter oder leiser stellen.



Sequenzdiagramm für einen Ablauf

6. Bedienkonzept

Zu Beginn muss der User die Kamera im passenden Abstand zu den „Legosteinen“ kalibrieren, um so sicher zu stellen, dass das Raster immer mit den Steinen übereinstimmt.

Dann muss ein Hintergrund gesetzt werden. Der Hintergrund muss weiß, oder eine andere Kalibrierungsfarbe haben, die nicht der Farbe der Steine entspricht. Dieser kann entweder eine Wand sein, oder ein Blatt Papier, welches hinter der Baufläche aufgebaut wird.

Daraufhin erbaut er mit den Steinen sein Konstrukt vor der Kamera auf einer geeigneten „Legoplatte“.

Der User soll dann auf dem Bildschirm seinen zuvor gebauten Beat sehen und bearbeiten können. Er soll einzelne Tonspuren oder ganze Beats auswählen können, um darauf Filter anzuwenden, die ebenfalls durch ein Dropdown Menü auswählbar und durch Schieberegler veränderbar sind. Hinzukommt noch, dass die fertigen Beats als Loop abspielbar sein sollen und dieser Loop über Knöpfe gestartet und gestoppt werden kann.
Außerdem wird es zwei Button geben, mit denen der User den erstellten Beat speichern kann, um dann einen neuen Beat anzulegen. Der User soll seinen Beat hintereinanderlegen können oder verschiedene Beats aneinanderreihen können, um daraus eine längere Musikspur zu bauen. Diese Musikspur soll er auch abspeichern können.

7. Timeline

- 06.11.18 <- Konzept fertig und Abgabe des Konzeptes
- 13.11.18 <- Eingearbeitet in C++, JavaScript und MIDI; Grundwebsite steht; Kamera erkennt bestimmte Farben
- 20.11.18 <- Bildschirm gerastert; Steine werden farblich und in der Rasterposition erkannt; Über MIDI wird die Farbe und Position weitergegeben -> Ton wird abgespielt.
- 27.11.18 <- Prototyp fertig <- UI fertig, verschiedene Farben gleichzeitig erkennen und wiedergeben
- 04.12.18 <- Höhen und Tiefen werden durchs Raster erkannt
- 11.12.18 <- Weitere Filter können durch JavaScript angewandt werden.
- 14.12.18 <- Projekt fertig, nur noch testen und ggf. Bugs fixen
- 18.12.18 <- Projekt Abgabe

8. Das Team

Paul Giese → openCV, MIDI und Projektleitung

Vincent Holtorf → JavaScript und MIDI

Wir haben uns entschieden, dass wir die beiden Sprachen unter uns aufteilen. Wir werden uns aber beide um den Bereich MIDI kümmern, da dieser als Schnittstelle uns beide betrifft.
Dazu hat Paul sich bereit erklärt, bei diesem Projekt die Leitung zu übernehmen und so sicher zu stellen, dass wir unsere Meilensteine einhalten und gegebenenfalls diese zu aktualisieren oder die Reihenfolge etwas zu ändern.