

Bau deinen Beat

Technische Dokumentation für das Projekt im Wahlpflichtmodul

Audio-Video-Programmierung bei Herrn Plaß.

Von Paul Giese (2270913) und Vincent Holtorf (2276905)

Ziel

Unser Ziel war es, eine Anwendung zu erstellen, bei der man mit Legosteinen einen sich wiederholenden Beat bauen kann. Ob der User nun einfach nur versucht, ein schönes Bild zuzubauen, um sich dann anzuhören, wie es klingt, oder ob er semiprofessionell versucht, einen gutklingenden Beat zu bauen, ist ihm frei überlassen. Durch verschiedene Farben wollten wir unterschiedliche Instrumente darstellen.

Je nachdem wie weit links oder rechts der Stein in der Lego Wand platziert ist, sollte das Instrument früher oder später im Beat gespielt werden. Die Höhe des Steines sollte dann noch angeben, in welcher Tonhöhe das Instrument abgespielt wird. Wir haben uns zunächst für vier klar unterscheidbare Farben entschieden und uns so erstmal auf vier Instrumente limitiert.

Auf der Website sollte eine Timeline den gebauten Beat darstellen, um so die Übersicht zu erleichtern und dem User zu zeigen, wo im Beat er sich gerade befindet. Auf Dieser sollte er einzelne Instrumente auswählen können, um verschieden Filter auf diese zu wirken.

Außerdem sollte man den Beat speichern können, um so mehrere Beats hintereinander zu legen und so einen Song zu kreieren, den der User dann auch speichern kann.

C++

Wir haben mit der Videoverarbeitungs-Library openCV in C++ gearbeitet. Hier haben wir das aufgenommene Bild zuerst in ein 16 mal 15 Raster geteilt. Später haben wir uns dann entschieden das Bild in 8 mal 7 zu rastern, um die Datenmenge zu reduzieren.

Für jedes Rasterfeld werden Pixel ausgewertet und verglichen. Hierzu werden zunächst die Farbwerte im RGB-Farbraum in Farbwerte im HSV-Farbraum umgewandelt und dann verglichen, ob der H-Farbwert, mit denen der Legosteine übereinstimmt. Dabei wird ein Counter für die jeweilige Farbe oder eine sonstige Farbe hochgezählt. Dies bedeutet, dass entweder der Counter für rot, grün, blau oder gelb hochgezählt wird, oder, falls der Farbwert von einen dieser Farben abweicht, zu hell oder zu dunkel ist, sich der Counter für Sonstiges erhöht.

Dann wird verglichen, welche dieser Farben überwiegt und diese dann für das jeweilige Rasterfeld gespeichert. Die Farbwerte wie auch die Koordinaten geben wir in ein Array, der unter anderem die vom MIDI erwarteten Start wie auch End Bytes enthält. Durch die eingebundene „Drumstick-Library“ findet dann die Übertragung durch MIDI statt.

Zuerst haben wir versucht für jeden Frame die Daten des Arrays per MIDI an unsere Website zu senden. Es stellte sich jedoch heraus, dass die Datenmenge für diese zu groß war. Weshalb wir einen Counter implementiert haben, der nur noch einmal pro Sekunde die Daten für einen Frame übermittelt. Der so entstandene Delay von einer Sekunde wirkt sich aber nicht negativ auf die User Experience aus.

JavaScript

Auf die durch MIDI übermittelten Daten werden per JavaScript in der Funktion „getMidiMessage“ zugegriffen. Hier werden die Positionen der ermittelten Farben in der Lego Wand in zeitliche Positionen im Beat umgewandelt. Ein Counter, der durch einen Timer die Sekunden im Beat zählt, vergleicht die Position der Farbe mit der Timeline und spielt den Sound zur gewünschten Zeit ab. Für Rasterfelder in denen keine Farbe sondern sonstiges gespeichert wurde, wird auch kein Sound gespielt.

Die Farben wurden erst verschiedenen Instrumenten zugewiesen, die aus eingebundenen .wav Dateien bestehen. Statt verschiedener Instrumente haben wir später jedoch Soundfiles eingebaut, da wir feststellten, dass sich mit diesen schönere Beats zusammenstellen ließen. Der gesamte Beat wird automatisch wiederholt, damit der User nicht immer nach acht Sekunden den Beat von neuem starten muss.

Über zwei Buttons kann der User den Beat jederzeit stoppen und an der gestoppten Position wieder starten. Des Weiteren sind acht Filter eingebaut, die der User nach Belieben auf den Beat anwenden kann. Durch Slider wird die Intensität der Filter angepasst. Aufgrund der knapp bemessenen Zeit konnten wir leider keine Timeline zur Visualisierung implementieren. Daher werden die Filter auch auf den gesamten Beat und nicht nur auf einzelne Sounds angewendet.

Fazit

Zum Zeitpunkt der Präsentation waren wir mit dem Stand des Projektes sehr zufrieden, da wir trotz vieler aufgetretener Schwierigkeiten die meisten unserer Projektziele umgesetzt haben. Unsere größte Hürde war unser Problem mit der Datenübertragung über LoopBe. Wir haben mehrere Tage damit verbracht das Problem zu lösen, was uns im Zeitplan sehr zurückgeworfen hat.

Wie sich herausstellte, war die Datenmenge für eine Frame per Frame Übertragung deutlich zu groß. Deshalb haben uns letztlich entschieden, die Daten nur sekundlich zu übertragen. Auch mit den Soundfiles haben wir uns lange beschäftigt, da keiner von uns beiden zuvor Erfahrung mit Sounddesign gesammelt hatte.

Dennoch haben wir es geschafft alle Hauptfunktionen, die wir uns vorgenommen hatten, einzubauen. Als nächsten Schritt könnten wir die Timeline zur Visualisierung und Übersicht implementieren, sowie eine Speicherfunktion für die einzelnen Beats, da wir dies aufgrund der verlorenen Zeit nicht mehr vor der Präsentation geschafft haben.

Außerdem lässt sich das Projekt gut weiterentwickeln, indem man beispielsweise dem User die Möglichkeit gibt, aus einem Pool aus unterschiedlichen Soundfiles zu entscheiden, welche er für welche Farbe nutzen möchte und es würde sich noch anbieten mehr Farben einzubauen oder den Takt des Beats zu ändern.

Alles in Allem sind wir mit dem Ablauf des Projektes, unseren umgesetzten Zielen und der finalen Anwendung sehr zufrieden. Es hat viel Spaß gemacht, das Projekt umzusetzen und es ist schönzusehen, dass auch andere an unserem Projekt Spaß hatten.