Technische Dokumentation AVPRG – Colorband

1 Allgemein

Im Rahmen des Wahlpflicht Studienfachs Audio-Video Programmierung des Studienganges Media Systems erstellten wir eine Videosoftware, die Musik steuert. In der folgenden Dokumentation sollen die Funktionsweise des Video- und Musikteils, sowie die Schnittstellenkommunikation beschrieben werden.

2 Das fertige Projekt

Das Ergebnis unseres Projekts war eine funktionierende Software, mit der man durch die Interaktion mit der Kamera des Computers Musik erzeugen kann.

Dem Anwender steht eine Auswahl verschiedenfarbiger Objekte (z.B. Kugeln, oder Pappausschnitte) zur Verfügung. Sobald er eines dieser Objekte in den Aufnahmebereich der Kamera führt wird ein Instrument zu hören. Durch die Bewegung entlang der y- Achse wechselt der Anwender zwischen 16 verschiedenen Instrumenten, während die Bewegung auf der x- Achse die Lautstärke verändert. Die Bewegung nach rechts lässt das abgespielte Instrument lauter werden, ganz links ist kaum noch etwas zu hören. Die Farbe des Objektes bestimmt, welcher Filter über die abgespielte Musik gelegt wird. Im aktuellen Stand des Projektes stehen 3 Filter zur Auswahl: „Allpass“, „Lowpass“ und „Highpass“.

Seine aktuelle Position kann der Anwender einer Anzeige in der Mitte des Interfaces entnehmen. Rechts davon befinden sich 3 Buttons.
Durch Betätigen des obersten der Buttons wird das derzeit abgespielte Instrument, zusammen mit seiner aktuellen Lautstärke und dem gewählten Filter, gespeichert. Dieses Instrument wird ab jetzt in einer Dauerschleife abgespielt. So kann der Anwender beliebig viele verschiedene Instrumente gleichzeitig abspielen lassen.
Der unterste Knopf macht den letzten Schritt rückgängig, sprich er entfernt das zuletzt gespeicherte Instrument aus der Dauerschleife.
Der mittlere Knopf setzt alles zurück, so dass der Anwender von vorne mit einer neuen Komposition beginnen kann.

3 Technische Umsetzung

Bei der Entwicklung haben wir das Projekt in 2 Teile aufgeteilt: Den Videoteil, der mit C++ entwickelt wurde und den Musik-/Webteil, für den wir mit JavaScript, HTML und CSS arbeiteten. Außerdem gibt es die MIDI Schnittstelle zur Kommunikation zwischen den beiden Teilen, die sowohl im C++, als auch im JavaScript Code eingebaut ist.

3.1 Videoteil, C++

In diesem Teil der Software wird das Kamerabild aufgenommen und verarbeitet. Als Ergebnis wird ein Array erstellt in dem die x- und y- Koordinaten, sowie die aufgenommene Farbe gespeichert werden. Dieses Array kann dann per MIDI übertragen werden.

Die Aufnahme und Verarbeitung des Kamerabildes erfolgt in 2 Schritten:
Das Kamerabild wird nacheinander nach den drei von uns vorgegebenen Farben rot, blau und grün durchsucht. Die Funktion „getColoredArea“ findet im Kamerabild die größte zusammenhängende Fläche der vorgegebenen Farbe und prüft ob diese eine von uns gesetzte Mindestgröße besitzt. Ist diese Bedingung erfüllt wird im gleichen Schritt der Mittelpunkt dieser Fläche und somit deren x- und y- Koordinaten berechnet.

Im nächsten Schritt werden mit der Funktion „getDominantColor“ die gefundenen Farbbereiche vergleicht und die größte dieser Flächen wird als Ergebnis ausgegeben. Diese Fläche sollte das Objekt sein, das der Anwender vor die Kamera hält.

Als letztes soll dieses Ergebnis, sprich die Farbe, sowie die Koordinaten, nun per MIDI an unser JavaScript zur weiteren Verarbeitung übertragen werden. Dafür geben wir die berechneten Werte in einen Array der bestimmte, von MIDI erwartete Start- und Endbytes enthält. Mithilfe der eingebundenen „Drumstick“- Library findet die MIDI-Übertragung statt.

3.2 Audioteil, JavaScript, HTML, CSS

Im Zweiten Teil wird das bearbeitete Kamerabild visualisiert. Dies geschieht in Form einer Website. Der Benutzer sieht jetzt wo sich die Kugel befindet und kann damit beginnen die verschiedenen Instrumente zu spielen.

Die Instrumente sind fertige .wav Dateien, die wir dann eingebunden haben. Damit der Nutzer das Gefühl hat, dass es endlose Töne sind, haben wir diese in einen „loop“ gesetzt. Die Dateien werden in Dauerschleife abgespielt.

Der Videoteil überträgt die Farbe und die Koordinaten in Form eines Arrays. Mit der Funktion „getMidiMessage(midiMessage)“ kann man ganz einfach auf die einzelnen Teile des Arrays zugreifen.

Die Visualisierung erfolgt mithilfe von HTML und CSS. Die Website sollte einfach gestaltet sein und intuitiv. Mit JavaScript haben wir dann die verschiedenen Formen erzeugt. Es wurde zuerst ein weißes Rechteck gezeichnet als Hintergrund, um klar zu stellen in welchem Bereich die Kamera aufnimmt, und darauf werden die kleineren Rechtecke zur Kennzeichnung der Kugel gezeichnet.

Um die 16 Instrumente klar getrennt darzustellen, sodass der Nutzer bemerkt wann das nächste Instrument kommt, haben wir die y-Achse so unterteilt, dass alle 30 Pixel ein Instrument abgespielt wird. Die x-Achse hingegen funktioniert wie ein Schieberegler in einem Audio-Interface. Hält man die Kugel links wird es leise und wenn man weiter nach rechts geht wird es lauter. Die x-Koordinate wird durch den größten möglichen x-Wert geteilt und wird dann von dem Wert 1 subtrahiert. So kommt man auf Dezimalwerte zwischen 0 und 1. Der erlangte Wert ist dann unser „gain.value“.

Die drei Farben führen zu unterschiedlichen Filtern. Mit der Roten Kugel wendet man den Filter „allpass“ an. Blau hat den Filter „lowpass“ und Grün „highpass“. Mit einer switch-case haben wir dann den Filter den jeweiligen Instrumenten zugewiesen, sobald diese von der Kamera erkannt wurden.

Mit den Buttons hat der Nutzer die Möglichkeit mehrere Instrumente abzuspielen oder diese Rückgängig zu machen. Der „Instrument Speichern“-Button sichert das momentan abgespielte Instrument, inklusive der Lautstärke und dem Filter, in einem Array. Falls einem das gerade gespeicherte nicht zusagt, kann man das mit dem „Schritt zurück“-Button rückgängig machen. Der „Reset-Button“ löscht den gesamten Array Inhalt.

4 Schlusswort

Insgesamt sind wir sehr zufrieden mit unserem Projekt. Alle Funktionen, die wir haben wollten, haben wir implementiert.