

SOUNDBOX



21/1/2018

Audio-Video Programmierung – C++, MIDI, JS

SoundBox ist eine 3D Audio Anwendung welche Nutzern ermöglicht in Echtzeit eine Soundkulisse zu manipulieren und Soundquellen zu verändern.

SoundBox

AUDIO-VIDEO PROGRAMMIERUNG – C++, MIDI, JS

Inhalt

HINTERGRUND UND INSPIRATION	1
PROBLEMATIK UND UMSETZUNG	2
Einarbeitung	2
Lösungsansatz und Architektur	2
PRÄSENTATION	3
FAZIT	3

HINTERGRUND UND INSPIRATION

Als Hintergrund für das Projekt stand die Anforderung vom Kurs AVPRG eine Anwendung zu entwickeln welche die Methoden und Techniken von Bilderkennung mittels C++, QT-Creator sowie den uns zur Verfügung stehenden Libraries über eine MIDI Anbindung an eine JavaScript Applikation weitergibt, die dann folglich mittels der WebAudio API Sounds und Geräusche erzeugt. Diese Verbindung sollte eine Interaktionsebene für einen Nutzer beinhalten.

Dementsprechend entschied ich mich für ein Projekt mit dem einen Nutzer in Echtzeit ermöglicht wird Soundquellen in einem realen 3D Umfeld zu manipulieren und somit Einfluss auf eine Soundkulisse auszuüben. Dies würde mittels Bilderkennung von farbigen Element auf mehreren Kameras ermöglicht und auf dem 3D Audio System der Fakultät Finkenau ausgegeben.

Das finale Produkt wäre dann ein Raum in der die wahrzunehmende Soundquelle gleich dem physikalischen farbigen Objekt sei.

PROBLEMATIK UND UMSETZUNG

Einarbeitung

Zu Beginn bestand die Herausforderung die angewandten und erworbenen Fähigkeiten der Vorlesung technisch auf dem Arbeits- und Projektrechner umzusetzen und zu implementieren. Dies beinhaltet das Aufsetzen und die Instandsetzung von QT-Creator sowie lokaler Server für die WebAudio Applikation.

Nachdem dies durchgeführt wurde konnten Tests getätigt und erste Elemente segmentweise entwickelt und umgesetzt werden. Dies konnte in Referenz zu vorhanden Daten und Materialien geschehen.

Lösungsansatz und Architektur

Zu Beginn war die Verarbeitung der Bildinformationen in relevante Daten zu lösen. Die Bildinformationen und Bildgenerierung wurden mittels der QT Library „opencv“ generiert. Somit konnten die einzelnen Frames der Kameras analysiert und deren Pixelwerte verarbeitet werden. Diese Werte wurden dann als „processedFrame“ weitergegeben, dessen Informationen für das folgende Vorgehen relevant waren.

Anschließend wurden aus dem „processedFrame“ mittels Schwerpunktberechnung die Koordinaten für das 3D Soundfeld gewonnen. Diese Koordinaten hingen von den jeweiligen Parametern ab, die für die Farberkennung konfiguriert wurden.

Im nächsten Schritt mussten dann diese Algorithmen für zwei Kameras parallel ausgeführt und die Daten entsprechend der Kameraposition interpretiert werden. Die daraus resultierenden Daten mussten geordnet und in Arrays gespeichert werden. Die Auflösung der Kameras spielt hier unter anderem auch eine Rolle.

Nachdem die Daten angemessen bearbeitet wurden mussten diese in byte lange arrays für die MIDI-Übertragung unterteilt und anschließend nacheinander für jede Koordinate gesendet werden. Dies wurde teilweise durch die QT library „drumstick“ realisiert. Allerdings mussten noch weitere Änderungen und Anpassungen getätigt werden um den Anforderungen der Übertragung gerecht zu werden.

Des Weiteren, um den Empfang zu ermöglichen, war es nötig die MIDI Kommunikation zu simulieren und auf die nötigen Parameter für den lokalen Webserver einzurichten.

Auch diese und alle anderen Merkmale der Übertragung mussten natürlich auf der Empfängerseite aufgefangen und beachtet werden.

Nun konnten auf der Empfängerseite diese Daten interpretiert und umgesetzt werden. Zunächst wurden hierfür die Daten zurück in relevante Zahlen übersetzt und anschließend an die „resonance audio node“ der WebAudio API weitergegeben.

Vorab wurden in JavaScript die Nodes gesetzt und der virtuelle Raum definiert. Die einkommenden Daten werden dann auf eine Soundquelle gelegt und mit jedem Update an die aktuelle Position gesetzt. Das Ergebnis ist eine Soundquelle die den Koordinaten der C++ Anwendung folgt und damit einhergehend die Position der getrackten Elemente der Kameras einnimmt.

PRÄSENTATION

Die Endgültige Präsentation wurde dann auf das 3D Audio System der Fakultät Finkenau ausgelegt und versucht zu testen, doch leider konnte aufgrund technischer Probleme die geplante Vorführung nicht stattfinden. Daher wurde die Präsentation über Kopfhörer und Kanalaudio durchgeführt, was leider einen leichten Verlust der Wahrnehmungsqualität der 3D Audio Elemente mit sich führte.

FAZIT

Das Projekt ermöglichte mir einen sehr umfangreichen ersten Einblick in die Audio-Videoprogrammierung und darüber hinaus gewährte es mir wertvolle Erfahrung der Produktentwicklung von hardwarenahen- bis hin zu rein software-/virtuellen Anwendungen.

Die Möglichkeit jeden einzelnen Schritt der Entwicklung bis ins kleinste Detail nachzuvollziehen und auszuführen schuf mir einen besseren Überblick über allgemeine und komplexere Softwaresysteme und deren Integration sowie Umsetzung.

Das entstandene Produkt hat darüber hinaus einen sehr zufriedenstellenden Effekt durch seine einfache Zugänglichkeit und Präsentierbarkeit für nicht involvierte Interessierte.

Daher würde ich das Projekt als guten Erfolg bezeichnen und freue mich auf zukünftige komplexe Softwareumsetzungen.