

“MUSIFIKATOR” - Ein Projekt zur Vorlesung AVPRG

EINLEITUNG

Im Rahmen der Vorlesung “AVPRG” soll ein Projekt entstehen, welches sich mit der Programmierung von Audio und Video Elementen befasst. In diesem Projekt fiel die Entscheidung hauptsächlich für die Audio Programmierung, bei der es darum geht, komplexe Audioausgaben zu erzeugen und mit dynamischen Effekten zu versehen. Konkret geht es dabei um die laufende Manipulation auf ein Musikstück.

[Namensklärung: Musik + Modifikator = Musifikator]

Da sich die Musik fortwährend selbstständig verändert, hat der User die Möglichkeit durch verschiedene Parameter auf die Art der Veränderung Einfluss zu nehmen. Durch einige wenige Angaben kann der User experimentell ausprobieren und erforschen, auf welche Art und Weise sich die Musik wandelt, wie schnell und in welchem Ausmaß dieses Veränderung stattfindet und wie das ganze visuell angezeigt werden kann. Die Einzelnen Faktoren können dabei nach Belieben verstellt werden.

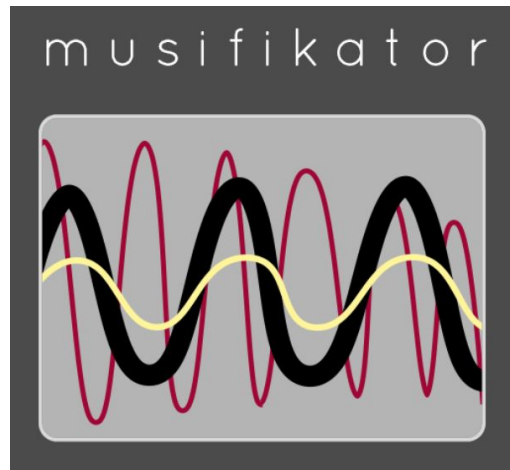
Die Neugier und der Forschergeist sollen dabei geweckt werden, und der Wunsch, den Zusammenhang der Werte zu verstehen, da auf dem ersten Blick nicht sofort erkennbar ist, welcher Regler, für welche Wirkung verantwortlich ist.



[Visuelle Ausgabe der Musik-Werte in Wellenform]

PROJEKTZIEL

Die Anwendung erzeugt ein Musikstück, welches sich stetig von alleine weiter verändert. Der User kann dabei zwischen einzelnen Audiodateien auswählen und nach eigenem Belieben die abgespielten Elemente einstellen. Die Veränderung der Musik kann der User durch die Eingabe von Parametern beeinflussen und diese direkt akustisch und visuell wahrnehmen. Die Eingabe dafür erfolgt klassisch mit Maus und Tastatur.



ANFORDERUNGSANALYSE

Der User soll eine Melodie aus einzelnen Tönen erstellen können. Diese Töne wiederholen sich in einem anpassbaren Intervall. Diese Intervalle werden optisch dargestellt. Die Töne sind entweder einzeln abspielbare Sounds, oder konstant wiedergebbare Klänge. Der User kann in die Intervalle einfache Anweisungen einpflegen.

Alle Anweisungen sollen durch einfach verständliche Bedienelemente, wie Buttons und Schieberegler, übergeben werden. Auf der rechten Seite des Bildschirms sind Steuerungsfelder für das ganze Musikstück und für jeden einzelnen Ton dargestellt. Der User kann einen einzelnen Ton auswählen um ihn zu bearbeiten.

Zum Beispiel kann die Lautstärke eines Tons während der Laufzeit angepasst werden. Sobald der User eine Einstellung vorgenommen hat, nimmt er eine sofortige akustische Veränderung wahr. Jeder Anwender kann dabei nach eigenem Belieben zwischen den Tönen auswählen und diese einzeln individuell anpassen. Das Ganze wird auf der linken Seite des Bildschirms visuell dargestellt und passt sich der Veränderung der Musik fortlaufend an.

KONZEPTION DER TECHNISCHEN UMSETZUNG

Das Programm wird eine Webanwendung, die mit HTML5, JavaScript und WebAudio umgesetzt wird.

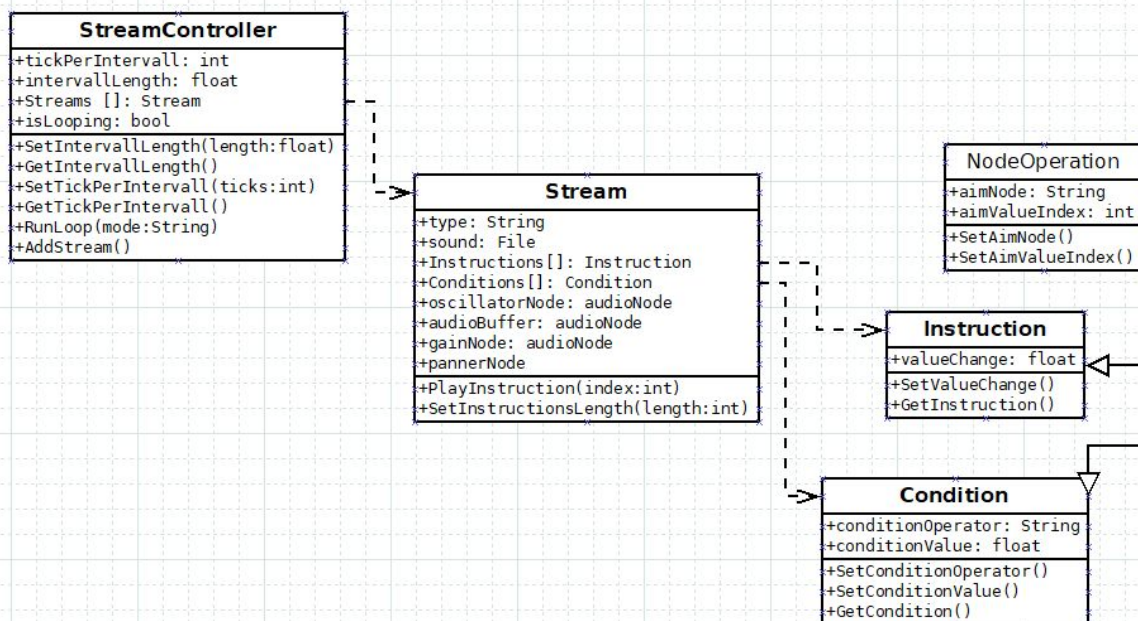
Die Usereingaben sollen über gewöhnliche Web Elemente wie Schieberegler und Eingabefeldern gesammelt werden. Die grafische Darstellung der Töne soll durch ein JS-Script erfolgen. Das Programm ist im Großen und Ganzen ein JavaScript-Programm, das ein Array an Sound Streams durch einen Taktgeber Befehle ausführen lässt.

TECHNISCHES KONZEPT

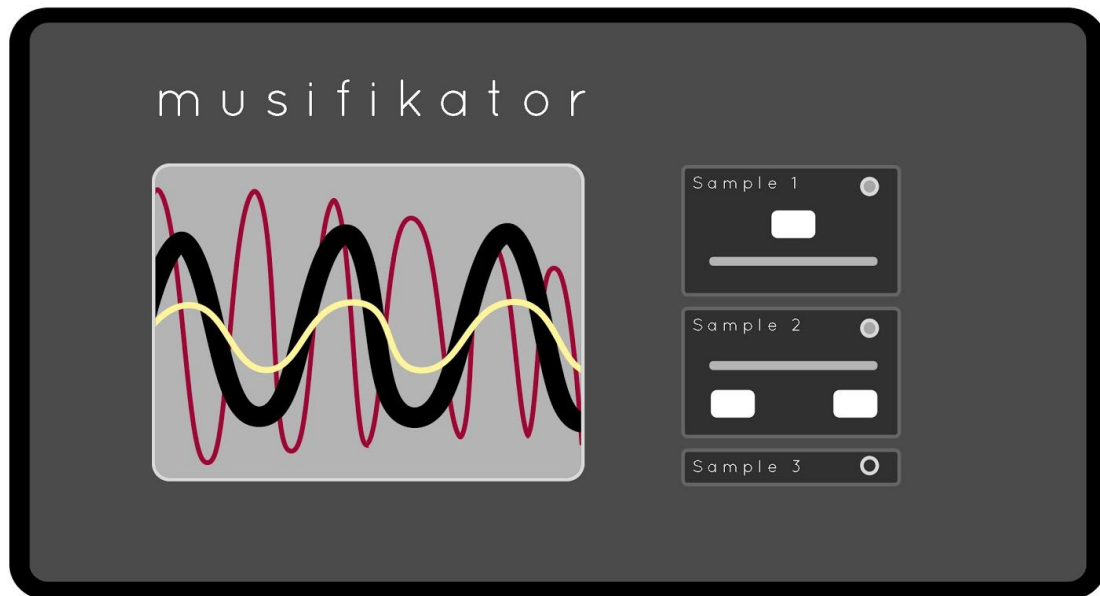


Das Kernstück des Programms ist der StreamController, der jeden einzelnen Stream organisiert und sich um die Taktgeschwindigkeit des Programms kümmert.

Ein einzelner Stream wiederum sammelt in Arrays Anweisungen und Bedingungen, die zum gewählten Zeitpunkt ausgeführt bzw. abgefragt werden.

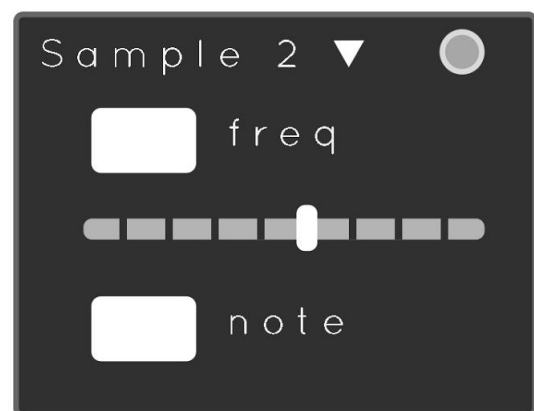
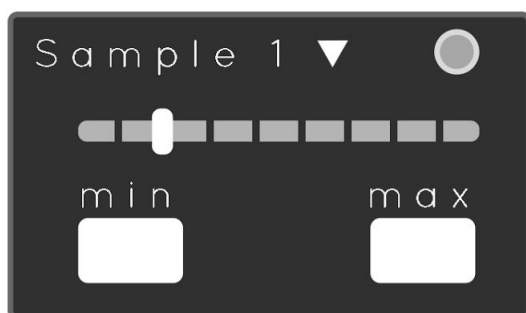


BEDIENKONZEPT



An einem fest installierten Monitor kann der User über ein Webinterface auf einer Seite des Bildschirms die visuelle Ausgabe der Musik sehen und auf der anderen Seite die einzelnen Einstellungen vornehmen. Er kann die unterschiedlichen Kanäle der Musik mit einer Checkbox aktivieren und deaktivieren. Sobald eine Checkbox für den jeweiligen Kanal aktiviert wurde, öffnen sich weitere Einstellungen.

Z.B.: Er öffnet Kanal 1 und findet folgende Segmente: Eine Zahl für Frequenz und einen Schieberegler. Durch Verändern der Werte, wird dem User klar, um welche Einstellungen es sich handelt. Das Eingabefeld für Frequenz hat direkten Einfluss auf die Höhe eines Tones und der Schieberegler stellt die Geschwindigkeit ein, mit der die Angleichung zwischen dem aktuellen Wert und dem eingegebenen Wert erzielt wird. So führt das der User mit jedem einzelnen Kanal durch.



PROJEKTKONZEPT

Besprechung und persönliche Treffen erfolgen einmal die Woche. Dabei wird auch der Wochenplan festgelegt. Online-Meetings bei Bedarf mehrmals die Woche.

- **20. November 2018:**
Ein Musikstück ist komponiert und ein Kanal lässt sich bereits aktivieren und deaktivieren. Erste Visuelle Ausgaben sind erstellt.
Erste Kanaleinstellung ist programmiert
- **27. November 2018:**
Prototyp
- **4. Dezember 2018:**
Weitere Kanaleinstellungen sind programmiert
Musikstück angepasst und erweitert
Alle Einstellungen funktionieren
- **11. Dezember 2018:**
Visuelle Ausgabe verfeinert
Einstellungen erneut angeglichen
Fehler beseitigt
- **18. Dezember:**
Präsentation des Projektes
- **13. Januar:**
Abgabe des Endgültigen Projektes

TEAMPLANUNG

- Daniel Gehn (2288086)
JavaScript - Audio Programming
- Natalie Perschkow (2221122)
UI Design - JS / CSS / HTML / Komposition Musik

Im folgenden sollen kurz die Kapitel eines Projektkonzeptes vorgestellt werden:

1. Die Einleitung ordnet das Projekt in einen allgemeinen Kontext ein. Hier wird die Frage beantwortet: "was ist das übergeordnete Ziel?". Beispiele könnten sein: externer Auftraggeber, Studienfach, Kooperation mit anderen Lehrveranstaltungen, Erlernen und Erproben einer Programmiersprache, Erforschung einer neuen Technologie. Wenn das Projekt Teil einer Prüfungsleistung ist, beschreiben Sie das, sowie den Kontext der Lehrveranstaltung. Wenn es ein Studienprojekt ist und Sie sich ein eigenes Thema gewählt haben, beschreiben Sie Ihre Motivation für diese Wahl. Zum allgemeinen Kontext gehört dann auch die Art der Präsentation: Folien-Präsentation/Vorführung, Ausstellung, Abnahme beim Auftraggeber, usw. Geben Sie dem Projekt einen Namen. Das macht das Projekt greifbar und fasst die Zielsetzung zusammen und unterscheidet es von vergleichbaren Projekten. Der Name muss nicht Englisch sein (was Studierende lieben), sehen Sie es als Herausforderung, einen guten deutschsprachigen Titel zu finden!
2. Als nächstes sollten Sie das Projektziel beschreiben. Sie definieren, was Sie konkret von Ihrer Software erwarten. Beschreiben Sie es so, wie es sich für einen Benutzer darstellt. Achten Sie darauf, dass Sie hier nicht Interna der verwendeten Technologien beschreiben. Wenn Sie zum Beispiel zum Abschluss eine Ausstellung oder eine Vorführung planen, dann beschreiben Sie detailliert den Ablauf der Vorführung.
3. Die Anforderungsanalyse ist meiner Ansicht nach der Kernabschnitt des Konzeptes. Hier definieren Sie die konkrete Aufgabe. Und am Ende messen Sie hieran, ob Ihr Projekt erfolgreich war, d.h. ob es die Anforderungen erfüllt. Die Anforderungsanalyse kann unterschiedlich detailliert erfolgen. In der professionellen Praxis würde man zum Beispiel User Stories definieren. Für ein Projekt im Rahmen einer Vorlesung wäre das wohl zu viel. Aber Sie sollten die wichtigsten Aufgaben oder Anwendungsfälle beschreiben. Bestandteil der Anforderungsanalyse sind auch Überlegungen zu nicht-funktionalen Anforderungen. Konkret kann das zum Beispiel die Latenzzeit (in Millisekunden) sein, d.h. die Zeit die es dauert, dass das System auf eine Benutzeraktion reagiert. Auch hier gilt wieder: beschreiben Sie aus Anwendersicht und gehen Sie nicht auf technische Details ein, das kommt später.
4. Als nächstes beschreiben Sie Ihre Konzeption der technischen Umsetzung. Zunächst sollten Sie die technischen Rahmenbedingungen definieren. Das heisst, mit welcher Hardware/Software-Umgebung wollen Sie arbeiten (Z.B. Windows Desktop-Computer, Android Smartphone mit NFC, Raspberry Pi im WLAN, usw.). Weiter beschreiben Sie hier eventuell verwendete Frameworks, oder Programmiersprachen, Beispiele wären: Java, openCV, Unity, NodeJs, usw. Dies kann entweder vorgegeben sein, dann könnten Sie die Limitierungen einer solchen Umgebung erwähnen. Wenn Sie eine Umgebung selbst gewählt haben, begründen Sie, warum Sie zu dieser Wahl kommen – vielleicht sogar, indem Sie mit Alternativen vergleichen (das sollte dann aber konkret sein).
5. Im technischen Konzept beschreiben Sie den von Ihnen geplanten Lösungsweg. Beschreiben Sie es so, dass ein Kommilitone/Herr Pläß eine Idee davon bekommt, was und wie Sie es umsetzen wollen. Stellen Sie es in Form eines Diagramms dar: UML-Klassendiagramm, UML-Sequenzdiagramm, Flussdiagramm, Blockschaltbild zum Signalfuss, Allgemeines Blockdiagramm zur Hardwareverteilung. Das ist sicherlich nicht einfach, aber es wird Ihnen helfen, Klarheit zu erlangen und Schwachstellen zu finden.
6. Die Beschreibung des Bedienkonzeptes ist wahrscheinlich nicht so schwierig. Es stellt ja das äußere Erscheinungsbild der Software dar und ist deshalb gut vorstellbar. ein Absolvent unseres Departments erzählte mir, dass in seiner Firma mit Papierprototypen gearbeitet wird. Versuchen Sie es auch mal! Sie sollten beschreiben, wie die Bedienung der Software geschieht. Was ist über ein Userinterface einstellbar? Welche Parameter kann ich in Echtzeit verändern? Gibt es unterschiedliche Zustände im Verhalten der Software (z.B. eine Kalibrationsphase)? Bedenken Sie auch, dass häufig Pfade, URLs oder ähnliches eingestellt werden müssen. Wird dies direkt im Quelltext definiert (eher schlecht), oder gibt es eine Konfigurationsdatei, oder sogar eine Konfigurations-Oberfläche? Natürlich bietet es sich an, Zeichnungen von Benutzeroberflächen in das Projektkonzept einzufügen.
7. Ein Projektkonzept sollte einen Zeitplan enthalten. Darin beschreiben Sie, wie hoch Sie den Aufwand in Personenstunden schätzen. Versuchen Sie, eine realistische Schätzung durchzuführen und sich nicht bloß am Workload einer Lehrveranstaltung zu orientieren. Wenn Sie dann später die tatsächlich benötigte Zeit protokollieren, erhalten Sie so die Chance, dies konkret zu üben! Der Zeitplan sollte auch Meilensteine (mindestens einen pro Monat) definieren. Wie kann evaluiert werden, ob ein Meilenstein erreicht wurde? Was soll dann geschafft sein?
8. Und schließlich bleibt die Teamplanung. Sie arbeiten mit mehreren Studierenden an einem Projekt. Nennen Sie die Namen! Nennen Sie die Aufgabenverteilung. Benennen Sie einen eventuellen Projektleiter!