
Webbasierte Musikvisualisierung in VR - Möglichkeiten und Grenzen

Bachelorarbeit zur Erlangung des akademischen Grades
Bachelor of Science
im Studiengang Medieninformatik
an der Fakultät für Informatik und Ingenieurwissenschaften
der Technischen Hochschule Köln

vorgelegt von: Aaron Asgharzadeh Khorramdarrehee
Matrikel-Nr.: 11107747
Adresse: Erftstraße 2
41540 Dormagen
aaron.asgharzadeh_khorramdarrehee@smail.th-koeln.de

eingereicht bei: Prof. Christian Noss
Zweitgutachter*in: Prof. Dr. Irma Lindt

30. Juni 2025

Kurzfassung/ *Abstract*

Musikvisualisierung ist ein Forschungsfeld mit langer Geschichte, das künstlerische und pädagogische Potenziale eröffnet. Durch den Einbezug moderner Technologien ergeben sich neue interaktive Nutzungsszenarien, deren technische Realisierbarkeit noch nicht umfassend erforscht ist.

In dieser Arbeit wird ein webbasiertes System zur Echtzeit-Musikvisualisierung in Virtual Reality entwickelt, mit dem Ziel, folgende Frage zu beantworten: *Welche technischen Herausforderungen und Potenziale ergeben sich bei der Echtzeit-Musikvisualisierung in einer webbasierten VR-Umgebung?*

Hierzu wurde ein Visualisierungskonzept entwickelt, das auf bestehenden Ansätzen aufbaut und dabei musiktheoretische Prinzipien einbezieht, um harmonische Zusammenhänge visuell darzustellen. Auf Grundlage des Konzepts entstand ein funktionaler Prototyp, der MIDI-Eingaben eines Keyboards in Echtzeit visualisiert und eine immersive Darstellung über ein VR-Headset ermöglicht. Technisch basiert die Umsetzung auf webbasierten Technologien wie Web MIDI API, React Three Fiber und React Three XR.

Die Ergebnisse demonstrieren, dass eine solche Anwendung technisch realisierbar ist. Offene Webstandards und eine breite Auswahl von Werkzeugen unterstützen die erfolgreiche Integration verschiedener Geräte und Technologien. Einschränkungen traten nur in wenigen Fällen auf, etwa durch funktionale Limitationen des Webbrowsers oder durch die teilweise unvollständige Umsetzung des Visualisierungskonzepts aufgrund des begrenzten Zeitrahmens.

Die Arbeit zeigt, dass medienzentrierte Webanwendungen auch in spezialisierten Anwendungsfeldern wie der Musikvisualisierung Potenzial bieten. Für zukünftige Forschungen ergeben sich Anknüpfungspunkte in pädagogischen Anwendungen, inklusiven Zugängen für Gehörlose sowie kollaborativen Nutzungsszenarien. Zudem bieten sich Forschungsarbeiten an, die das Spieler- bzw. Zuhörererlebnis untersuchen.

Dieses Projekt schafft somit eine Grundlage für weiterführende Entwicklungen im Bereich der interaktiven Musikvisualisierung sowie medienzentrierter Anwendungen im Allgemeinen.

Inhaltsverzeichnis

Tabellenverzeichnis	III
Abbildungsverzeichnis	IV
Glossar	V
Abkürzungsverzeichnis	VII
1. Einleitung	1
2. Entwurf eines Visualisierungskonzepts	3
3. Entwicklung des Prototyps	9
3.1. Begriffserklärung zu XR	9
3.2. Vorstellung der Technologien und Geräte	10
3.3. Entwicklungsprozess	11
3.4. Möglichkeiten und Grenzen	14
4. Fazit und Ausblick	17
Literatur	19
Anhang	21
A. Technische Ressourcen	22
A.1. ChatGPT-Prompts	22
A.2. Quellcode-Repository	23

Tabellenverzeichnis

1. Stufen und zugehörige Akkorde für C-Dur	5
--	---

Abbildungsverzeichnis

1.	Lineare Farbverteilung einer Oktave über die Tonhöhe	3
2.	Quintenzirkel	4
3.	Überführung des Quintenzirkels in den <i>key spanning circle of thirds</i> durch linksseitige Rotation des inneren Rings. Rechts ist der Farb- bereich für C-Dur dargestellt	6
4.	Farbverteilung einer Oktave nach dem entwickelten Konzept	6
5.	Exemplarische Farbzuzuordnung auf mehrere Oktaven durch ansteigende Helligkeitswerte	7
6.	Erste Umsetzung des Visualisierungskonzepts in einer 3D-Umgebung .	12
7.	Aktivitätsdiagramm zur Verarbeitung der empfangenen MIDI-Daten .	14

Glossar

Diatonik Betrachtung von Tonleitern als Folge von sieben Tönen.

Dissonanz Tonkombinationen, die als nicht-wohlklingend empfunden werden.

Dreiklang Akkord aus drei Tönen. Gebildet durch zwei aufeinanderfolgende Terzen über einem Grundton. Die Reihenfolge und Art der Terzen (groß oder klein) bestimmt den Typ des Dreiklangs (Dur, Moll, vermindert oder übermäßig).

Dur Tongeschlecht in der Musik, Pendant zu Moll. Der Dur-Dreiklang besteht aus einer großen Terz gefolgt von einer kleinen Terz.

Große Terz Musikalisches Intervall mit einem Abstand von vier Halbtönen zwischen Grundton und dem zweiten Ton.

Grundton Der Ausgangston, auf dem ein Akkord, eine Tonleiter oder ein Intervall aufgebaut ist.

Intervall Bezeichnung für den Abstand zwischen zwei Tönen. Die Größe des Intervalls wird in Halbtonschritten gezählt. Beispiele sind: Kleine Terz (3 Halbtöne), Quinte (7 Halbtöne), Oktave (12 Halbtöne).

Key spanning circle of thirds Variante des Quintenzirkels, bei der der innere Ring nach links rotiert wird, sodass Dreiklänge direkt abzulesen sind.

Kleine Terz Musikalisches Intervall mit einem Abstand von drei Halbtönen zwischen Grundton und dem zweiten Ton.

Konsonanz Tonkombinationen, die als wohlklingend empfunden werden.

Moll Tongeschlecht in der Musik, Pendant zu Dur. Der Moll-Dreiklang besteht aus einer kleinen Terz gefolgt von einer großen Terz.

Quinte Musikalisches Intervall mit einem Abstand von sieben Halbtönen zwischen Grundton und dem zweiten Ton.

Quintenzirkel Ein musiktheoretisches Modell, das Dur- und Moll-Tonarten in zwei Ringen kreisförmig anordnet. Benachbarte Tonarten stehen im Abstand einer Quinte zueinander.

Stufen Die einzelnen Töne einer Tonleiter in ihrer Reihenfolge.

Verminderter Dreiklang Dreiklang der aus zwei aufeinanderfolgenden kleinen Terzen besteht.

Abkürzungsverzeichnis

ADSR *Attack, Decay, Sustain, Release*

API Application Programming Interface

AR Augmented Reality

MIDI Musical Instrument Digital Interface

MSB Most Significant Bit

VR Virtual Reality

XR Extended Reality

1. Einleitung

Die Visualisierung von Musik ist ein wiederkehrender Untersuchungsgegenstand in den Wissenschaften und der Kulturgeschichte.

Wie Klemenc et al. (2011) anmerken, stellte Isaac Newton bereits 1704 in seinem Werk *Opticks* eine Verbindung zwischen Lichtstrahlen¹ und den Farbeindrücken her, die sie hervorrufen, sowie zwischen Schallwellen und den dadurch ausgelösten Hörempfindungen (Newton, 1730, S. 320). Newton beschrieb zudem den Farbkreis, indem er einen Kreis in sieben Segmente aufteilte – proportional zu den sieben Tönen einer Tonleiter – und den einzelnen Segmenten eine der sieben Farben des Regenbogens zuwies (Newton, 1730, S. 134).

Der Mathematiker Louis Bertrand Castel rezensierte 1723 die französische Übersetzung von Newtons *Opticks* und entwickelte daraufhin eine eigene Farb-Ton-Theorie (Jewanski, 2006, S. 147). „1724 wurde er, nach eigener Aussage, von Rameau² ermutigt, ein Farbenklavier (*clavecin oculaire*) zu entwickeln“ (Jewanski, 2006, S. 148). Einer der Gründe, die Castel veranlassten diese Idee weiterzuverfolgen, war der Gedanke Gehörlosen zu ermöglichen Musik durch Farbeindrücke wahrzunehmen (Jewanski, 2006, S. 148, 151, 157).

Das Farbenklavier ist ein Instrument, das Farben projiziert. Obwohl frühe Entwürfe bereits versuchten das gleichzeitige Spiel von Farben und Tönen zu ermöglichen (Jewanski, 2006, S. 151), führten technische Limitationen der Zeit dazu, dass das Farbenklavier keine Töne erzeugte, sondern nur farbige Projektionen zeigte (Jewanski, 2006, S. 156). Dabei entwickelte sich die Erzeugung der Farbeindrücke über Jahrhunderte von Kerzenschein hinter farbigem Glas weiter zu Glühlampen und elektrischem Licht bis hin zu modernen Lichtanlagen (Jewanski, 2006, S. 151, 157, 183).

1911 veröffentlichte der Komponist Alexander Scriabin seine Komposition *Prométhée. Le Poème du feu*, op. 60, in der er eine eigene Notenspur für ein elektrisches Farbenklavier schrieb, das er zusammen mit dem Techniker Alexander Mozer entwickelt hatte (Gawboy & Townsend, 2012, S. 2).

¹Newton besaß noch nicht das Wissen über moderne physikalische Modelle. Er beschrieb Strahlen und ihre Vibrationen, nicht aber Licht- oder Schallwellen.

²Jean-Philippe Rameau war ein französischer Komponist und Musiktheoretiker. Er gilt als Begründer der Harmonielehre (Christensen, 1993).

Diese historische Entwicklung bildet die Grundlage für moderne Systeme zur Visualisierung von Musik. Es gilt weiterhin zu ergründen, welchen Nutzen Musikvisualisierungen bieten können: Ob sie etwa pädagogischen Mehrwert liefern und somit das Lernen von Instrumenten oder Musiktheorie erleichtern. Ob es – wie schon Castel versuchte – möglich ist, Menschen ohne Gehör Musik visuell zu vermitteln oder ob die Visualisierung von Musik zu einem immersiveren Erlebnis für Instrumentalisten und Zuhörer führt.

Neben diesen Fragen besteht zudem die Herausforderung der technischen Umsetzung. Es werden zunehmend computergestützte Ansätze verfolgt, wobei das Angebot vielfältiger Technologien bereitsteht. In dieser Arbeit wird ein Prototyp entwickelt, der webbasierte Technologien nutzt und für die Verwendung in *Virtual Reality* (VR) vorgesehen ist.

Ziel der Arbeit ist es, die technischen Möglichkeiten und Grenzen zu identifizieren, die sich im Verlauf der Umsetzung ergeben. Der Entwicklungsprozess dient dabei insbesondere dazu, die Komplexität der praktischen Umsetzung zu untersuchen, etwa bei der Integration unterschiedlicher Technologien in ein funktionierendes Gesamtsystem, bei schwer realisierbaren Teilen des Konzepts sowie bei weiteren Problemen, die während der Entwicklung erkennbar werden. Die zentrale Forschungsfrage lautet: *Welche technischen Herausforderungen und Potenziale ergeben sich bei der Echtzeit-Musikvisualisierung in einer webbasierten VR-Umgebung?*

Es wird zunächst ein Konzept erarbeitet, das Musik unter Berücksichtigung musiktheoretischer Prinzipien und Parametern wie Tonhöhe, Tondauer oder Lautstärke visualisiert. Die Visualisierung soll dabei durch Farben, geometrische Formen und die Positionierung im dreidimensionalen Raum erfolgen.

Nach Ausarbeitung des Konzepts wird ein Prototyp entwickelt, der die erarbeiteten Ansätze zur Visualisierung von Musik umsetzt. Grundlage der Darstellung sind Echtzeitdaten eines live gespielten MIDI-Keyboards³. Da das System auf Interaktivität ausgelegt ist, wird bewusst auf vorgefertigte MIDI-Kompositionen verzichtet.

Damit unterscheidet sich der Prototyp von bestehenden Musikvisualisierungen: Er ermöglicht die Echtzeitverarbeitung von MIDI-Daten auf Basis musiktheoretischer Prinzipien in webbasierten VR-Systemen – ein Ansatz, der bisher nicht dokumentiert ist.

³Die Abkürzung MIDI (*Musical Instrument Digital Interface*) bezeichnet eine Hard- und Software Spezifikation für den Austausch von Informationen elektronischer Musikinstrumente (The MIDI Manufacturers Association, 1996).

2. Entwurf eines Visualisierungskonzepts

In diesem Kapitel wird ein Konzept zur Visualisierung von Musik im dreidimensionalen Raum erarbeitet, basierend auf MIDI-Daten und ausgelegt für die Nutzung in VR.

Klemenc et al. (2011) entwarfen bereits ein Konzept zur Farbvisualisierung von Musik und nutzten hierfür den von Gatzsche et al. (2007) vorgestellten *key spanning circle of thirds*. Sie führen an, dass viele Ansätze Töne und Farben nach aufsteigender Tonfrequenz verknüpfen. Mit solch einer linearen Verteilung – wie in Abbildung 1 illustriert – wird die Tonhöhe über den graduellen Farbverlauf anschaulich dargestellt (Klemenc et al., 2011). Allerdings werden auf diese Weise keine harmonischen Beziehungen abgebildet.

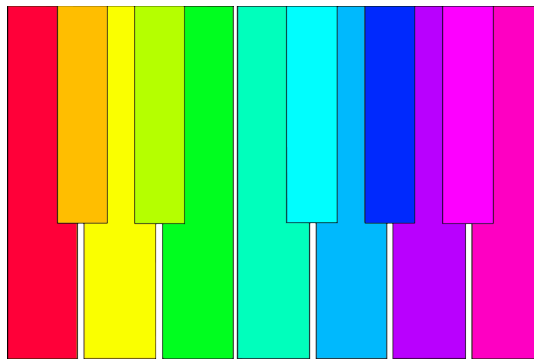


Abbildung 1. Lineare Farbverteilung einer Oktave über die Tonhöhe

Der *key spanning circle of thirds*, eine Neuordnung des Quintenzirkels (Gatzsche et al., 2007), bietet hingegen die Möglichkeit, harmonische Beziehungen besser darzustellen.

Der Quintenzirkel ist ein Modell der Musiktheorie, das Dur- und Moll-Tonarten in einem Kreis anordnet, wobei jede benachbarte Tonart im Abstand einer Quinte zueinander steht. Je näher Tonarten im Quintenzirkel zusammenliegen, desto höher ist ihre musikalische Nähe. So teilen sich zwei Tonarten mehr gemeinsame Töne, je näher sie sich im Zirkel stehen. Der Zirkel besteht aus einem äußeren Ring für die Dur-Tonarten und einem inneren Ring für die Moll-Tonarten. Für jede Dur-Tonart

gilt, dass es eine parallele Moll-Tonart gibt, welche dieselben Töne beinhaltet. Diese Paralleltonarten stehen im Quintenzirkel direkt gegenüber. Zudem ist der innere Ring eine um eine kleine Terz nach unten verschobene Version des äußeren Rings. Folglich ist jede Tonart zweimal vertreten, einmal in der Dur- und einmal in der Moll-Position¹.

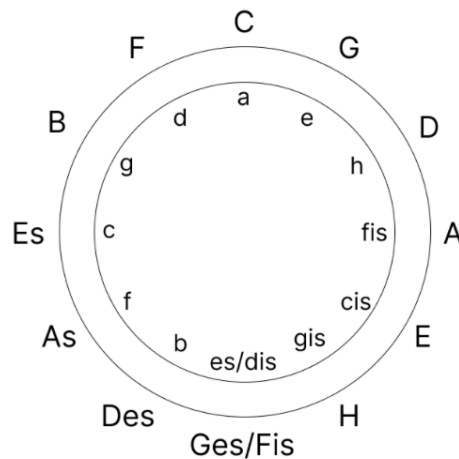


Abbildung 2. Quintenzirkel

Zur Konstruktion des *key spanning circle of thirds* wird nun der innere Ring nach links rotiert, sodass die Paralleltonarten nicht mehr gegenüberstehen, sondern versetzt nebeneinander. Dadurch ergibt sich eine neue Struktur: Betrachtet man die Tonarten des Zirkels als einzelne Töne, so ergibt jeder Ton mit seinen zwei rechtsliegenden Nachbarn einen Dreiklang (Der C-Dur Dreiklang besteht bspw. aus den Tönen C - E - G). Dieser Umstand gilt für alle Töne im Kreis, sowohl in der Dur- als auch in der Moll-Position. Somit werden harmonische Dreiklänge im *key spanning circle of thirds* in unmittelbarer Nähe zueinander angeordnet.

In der Musiktheorie existieren darüber hinaus modale Verwandtschaften zwischen Tonarten. Diese lassen sich herleiten, indem von den einzelnen Stufen einer Tonleiter ausgehend neue Tonleitern oder Dreiklänge gebildet werden, die dieselben Töne wie die ursprüngliche Tonart beinhalten. Als Stufen werden die einzelnen Töne einer Tonleiter in ihrer Reihenfolge bezeichnet.

¹Zur besseren Übersicht werden Tonarten im Quintenzirkel meist so gewählt, dass möglichst wenige Vorzeichen verwendet werden. Aus diesem Grund steht beispielsweise Des-Dur im äußeren Ring, aber cis-Moll statt des-Moll im inneren, obwohl beide gleich klingen.

Für C-Dur sind dies: C – D – E – F – G – A – H. Die zugehörigen Akkorde sind in Tabelle 1 abgebildet:

Stufe	Tonart
I	C-Dur
II	d-Moll
III	e-Moll
IV	F-Dur
V	G-Dur
VI	a-Moll
VII	h-vermindert

Tabelle 1. Stufen und zugehörige Akkorde für C-Dur

Während sich Klemenc et al. (2011) auf konsonante und dissonante Tonkombinationen beziehen, konzentriert sich diese Arbeit auf diatonische Tonleitern. Die Begriffe *Konsonanz* und *Dissonanz* wurden im Verlauf der Musikgeschichte in verschiedenen wissenschaftlichen Disziplinen unterschiedlich definiert, oftmals ohne klare Abgrenzung, was zu einer anhaltenden Unklarheit über die Bedeutung der Begriffe führt (Tenney, 1988).

Der Begriff *diatonisch* bedeutet, dass Tonleitern als Folge von sieben Tönen innerhalb einer Oktave betrachtet werden (Pazel, 2022). Dabei ist zu beachten, dass eine Oktave insgesamt zwölf Halbtönen umfasst. Innerhalb dieser zwölf Halbtöne ergeben sich diatonische Tonleitern aus einer spezifischen Abfolge von Ganz- und Halbtonschritten.

Im *key spanning circle of thirds* liegen die Stufen einer diatonischen Tonleiter ebenfalls beieinander. Zwar wird die genaue Reihenfolge der Stufen nicht beibehalten, dennoch sind sie lückenlos angeordnet, ohne dass Töne anderer Tonarten dazwischenliegen.

Das Modell ist somit in der Lage sowohl Dreiklänge als auch Tonleitern² einer Tonart in relativer Nähe zueinander anzuordnen. Wird das Modell zusätzlich mit einem Farbkreis überlagert, so liegen Dreiklänge und Tonleitern einer Tonart innerhalb eines ähnlichen Farbbereichs.

Töne, die nicht Teil einer Tonart sind, haben keinen funktionalen Bezug zu eben dieser und besitzen daher keine explizite Position im *key spanning circle of thirds*. Um die harmonische Unzugehörigkeit hervorzuheben, wird jeweils die Position gewählt,

²Verminderte Positionen werden im *key spanning circle of thirds* nicht abgebildet. In dieser Arbeit wird daher als Annäherung die jeweilige Moll-Position verwendet. Da sich Moll- und verminderte Akkorde nur in einem Halbton unterscheiden, bleibt die harmonische Nähe weitestgehend erhalten.

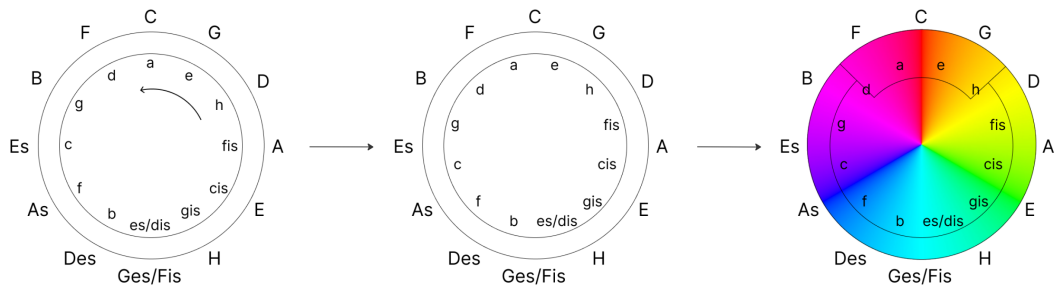


Abbildung 3. Überführung des Quintenzirkels in den *key spanning circle of thirds* durch linksseitige Rotation des inneren Rings. Rechts ist der Farbbereich für C-Dur dargestellt

die der betrachteten Tonart im Kreis am weitesten entfernt ist. Dies gewährleistet einen höchstmöglichen farblichen Kontrast. Beispielsweise wird in C-Dur, für die Farbzuzuordnung von Cis/Des – unterschiedlichen Bezeichnungen desselben Tons – die Des-Dur-Position verwendet.

Unter Verwendung dieser Zuordnung ergibt sich für C-Dur folgende Farbverteilung:

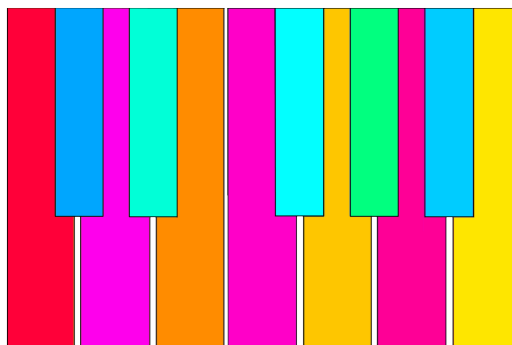


Abbildung 4. Farbverteilung einer Oktave nach dem entwickelten Konzept

Dieser systematische Ansatz zur Farbverteilung ermöglicht es zudem, den Farbbereich innerhalb des Farbkreises frei zu wählen, ohne die relativen Farbverhältnisse der Töne zueinander zu verändern. Dadurch lässt sich eine gewünschte Farbgebung gezielt umsetzen.

Die bisher vorgestellte Farbzuzuordnung bezieht sich auf eine einzelne Oktave. Unterschiede in der Tonhöhe zwischen mehreren Oktaven werden dadurch farblich nicht erfasst. Menschen assoziieren hohe Töne mit hellen Farben und tiefe Töne mit dunklen Farben beziehungsweise mit stärkerer oder schwächerer Leuchtkraft (Hubbard, 1996). Durch Anpassungen von Farbparametern wie Helligkeit und Sättigung kann

die Tonhöhe somit visuell dargestellt werden. Innerhalb einer Oktave bleiben diese Parameter unverändert, sodass die Farbzuordnung des Konzepts konsistent und nachvollziehbar bleibt. Zur Verdeutlichung der Tonhöhe über mehrere Oktaven hinweg werden die Farben zunehmend heller und blasser gewählt.

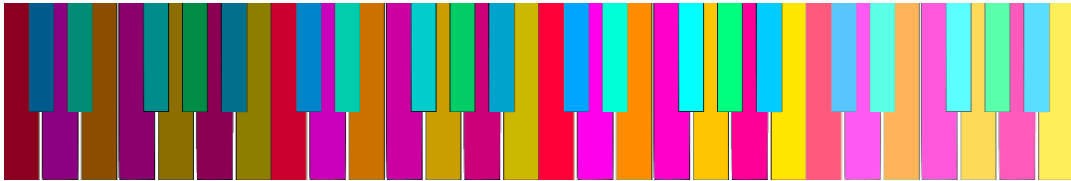


Abbildung 5. Exemplarische Farbzuordnung auf mehrere Oktaven durch ansteigende Helligkeitswerte

Der Konferenzbeitrag *A Visualization of Music* (1997) von Smith und Williams befasst sich mit der Visualisierung von Musik im dreidimensionalen Raum anhand von MIDI-Daten. Farbkugeln werden als Repräsentation für Töne genutzt und deren Eigenschaften entsprechend der bereitgestellten MIDI-Daten angepasst. Berücksichtigt werden dabei die MIDI-Parameter für Tonhöhe, Lautstärke und Klangfarbe. Letztere dient zur räumlichen Trennung unterschiedlicher Instrumente entlang der X-Achse.

Die Tonhöhe stellen die Autoren dar, indem sie den jeweiligen MIDI-Parameterwert der Y-Achse als Koordinate zuweisen. Wie bereits erläutert werden in dieser Arbeit Helligkeitswerte über mehrere Oktaven hinweg verändert, jedoch nicht innerhalb einer einzelnen. Um dies räumlich abzubilden, wird jeder Oktave eine eigene Reihe entlang der Y-Achse zugewiesen. Noten einer einzelnen Oktave werden aufsteigend von links nach rechts angeordnet. So entsprechen sie der bekannten Anordnung, wie sie etwa Tasteninstrumente verwenden. Im virtuellen Raum kann die vertikale Anordnung der Oktaven möglicherweise zu einem besseren Überblick und einem immersiveren Erlebnis führen. Zu diesem Zweck ist auch eine halbkreisförmige Anordnung der Farbkugeln denkbar.

Zur Darstellung der Lautstärke ändern die Autoren den Durchmesser der Kugeln in Abhängigkeit vom jeweiligen MIDI-Parameterwert, sodass die Kugeln mit steigender Lautstärke größer erscheinen. Dieser Ansatz wird für das Visualisierungskonzept übernommen.

Um die Tondauer darzustellen, wird nach Abklingen des Tons eine Markierung an der Position der farbigen Kugel platziert (Smith & Williams, 1997). Diese Markierung ist ebenfalls kugelförmig, wird aber relativ zur Tondauer entlang der Z-Achse verzerrt. Markierungen für längere Töne erscheinen somit zunehmend ellipsoid (Smith & Williams, 1997). Ein ähnlicher Ansatz kann für diese Arbeit genutzt werden.

Damit sind die wesentlichen Bestandteile des Visualisierungskonzepts definiert, auf deren Basis der Prototyp entwickelt wird: Die Farbzuordnung für C-Dur anhand des

key spanning circle of thirds, die visuelle Repräsentation der Töne in Form von farbigen Kugeln, die räumliche Positionierung zur Darstellung der Tonhöhe sowie die Variation von Größe und Form der Kugeln, um Lautstärke und Tondauer abzubilden.

Aspekte, die aus Zeitgründen nicht weiter behandelt werden konnten, umfassen unter anderem Rhythmus, *ADSR*-Phasen³ sowie die Bewegung innerhalb des virtuellen Raums.

³Die Abkürzung ADSR steht für *Attack*, *Decay*, *Sustain* und *Release* und beschreibt die Phasen der Signalkurve eines Synthesizers (Gaume Echeverría et al., 2010), typischerweise bezogen auf den Lautstärkeverlauf eines Tons.

3. Entwicklung des Prototyps

Auf Basis des zuvor entwickelten Visualisierungskonzepts erfolgt die Umsetzung des Prototyps. Dieser soll unter Verwendung webbasierter Technologien MIDI-Daten in Echtzeit verarbeiten und gemäß dem Konzept darstellen. Die Visualisierungen sollen zudem in VR betrachtet werden können. Dieses Kapitel beschreibt die technische Realisierung unter Berücksichtigung der eingesetzten Technologien und Geräte sowie der erkennbaren Möglichkeiten und Herausforderungen, die während der Entwicklung auftraten.

Um eine angemessene Wissensgrundlage zu gewährleisten, erfolgt zunächst eine Erklärung des Begriffs *Extended Reality* (XR).

3.1. Begriffserklärung zu XR

Der Begriff XR wird häufig als Überbegriff für *Virtual Reality* (VR) und *Augmented Reality* (AR) verwendet. In Fachkreisen fehlen jedoch klare Definitionen, was zu unklaren Abgrenzungen zwischen den Bezeichnungen führt (Rauschnabel et al., 2022). Für diese Arbeit werden Definitionen genutzt, die auf dem *xReality framework* basieren, das Rauschnabel et al. (2022, S. 13) in ihrem Artikel *What is XR? Towards a Framework for Augmented and Virtual Reality* vorgestellt haben:

- **VR:** „*Virtual Reality* is an artificial, virtual, and viewer-centered experience in which the user is enclosed in an all-encompassing 3D space that is - at least visually - sealed off from the physical environment.“
- **AR:** „*Augmented Reality* is a hybrid experience consisting of context-specific virtual content that is merged into a user’s real-time perception of the physical environment through computing devices. AR can further be refined based on the level of local presence, ranging from assisted reality (low) to mixed reality (high).“

3.2. Vorstellung der Technologien und Geräte

Der Titel dieser Arbeit verweist bewusst auf webbasierte Technologien sowie deren Möglichkeiten und Grenzen. Webbasierte Technologien sind unter beiden Gesichtspunkten von Bedeutung: Einerseits sind sie plattformunabhängig und erfordern auf Nutzerseite keine Installation oder Updates, was die Benutzerfreundlichkeit erhöht. Zudem basieren sie auf offenen Standards, wie vom W3C (W3C, 2025a) oder von ECMA International (ECMA International, 2024), die von gängigen Browsern unterstützt werden. Dies führt zu einer großen Entwicklerbasis. So zählen webbasierte Technologien laut der *Stack Overflow Developer Survey* (2024) weltweit zu den am häufigsten genutzten Technologien. Infolgedessen stehen zahlreiche Bibliotheken, Dokumentationen und Community-Ressourcen zur Verfügung.

Gleichzeitig kann die große Vielfalt verfügbarer Technologien auch dazu führen, dass unterschiedliche Komponenten nicht immer optimal zusammenarbeiten. Weiterhin können webbasierte Anwendungen erhöhte Latenzzeiten aufweisen, die sich insbesondere bei Echtzeitanwendungen negativ auswirken.

Als Grundlage für die Entwicklung wurde das Framework Next.js verwendet, das sowohl Frontend- als auch Backend-Entwicklung ermöglicht. Im Rahmen dieses Projekts wurden ausschließlich die Frontend-Komponenten genutzt. Dennoch wurde Next.js gewählt, um zukünftige Erweiterungen – etwa serverseitige Funktionalitäten – zu erleichtern.

Die Nutzung von Next.js ausschließlich für das Frontend entspricht effektiv der Entwicklung mit der *deklarativen* Javascript-Bibliothek React, die im Jahr 2024 die meistgenutzte Technologie für Frontend-Entwicklung war („Stack Overflow - Developer Survey“, 2024).

Deklarative Programmierung steht imperativer Programmierung gegenüber und zeichnet sich durch eine höhere Abstraktion aus: Technische Details werden durch intuitivere Anweisungen ersetzt. Sie vereinfacht die Zugänglichkeit von Technologien, mit denen man nicht vertraut ist, da die Notwendigkeit wegfällt, sich tief in technische Details einzuarbeiten. Deklarative Ansätze sind daher entwicklerfreundlich und besonders geeignet für dieses Projekt, das sowohl zeitlich begrenzt ist als auch die Verwendung diverser Technologien voraussetzt.

Für die Umsetzung des Prototyps waren zunächst drei APIs vorgesehen: Web MIDI API, die Web Audio API und die WebXR Device API. Die Web MIDI API ermöglicht es, MIDI-Daten zu empfangen und zu verarbeiten. Die Web Audio API sollte dazu dienen, die Visualisierungen darzustellen. Für die Umsetzung der VR-Funktionalitäten war die Nutzung der WebXR Device API vorgesehen.

Von den ursprünglich vorgesehenen drei APIs wurde letztendlich nur die Web MIDI API direkt verwendet. Die Visualisierungen, die durch die Web Audio API realisiert werden sollten, werden nun mithilfe der React-Bibliothek React Three Fiber umgesetzt. Diese ist eine React-spezifische und deklarative Umsetzung der 3D-Grafikbibliothek Three.js. Da die Darstellung im dreidimensionalen Raum einen zentralen Bestandteil dieses Prototyps ausmacht, ist die Entwicklung in einer 3D-Umgebung notwendig.

Aufgrund der besseren Optimierung im React-Umfeld wurde für die Umsetzung der VR-Funktionalitäten auch die WebXR Device API durch das deklarative React Three XR ersetzt.

Bei der Wahl des Internetbrowsers muss beachtet werden, ob dieser mit den verwendeten Technologien kompatibel ist. Für dieses Projekt wurde Google Chrome gewählt, da alle eingesetzten Technologien mit diesem Browser zuverlässig arbeiten.

Für die Wahrnehmung der VR-Umgebung wurde das VR-Headset Oculus Quest genutzt, das von der TH Köln zur Verfügung gestellt wurde. Zur Eingabe von MIDI-Daten diente das Keyboard Roland E-09.

Der Prototyp wird auf einem Desktop-PC mit dem Betriebssystem Windows ausgeführt.

3.3. Entwicklungsprozess

Die Umsetzung begann mit einer einfachen 3D-Szene (`Canvas`), in der farbige Kugeln platziert wurden. Die Farben basieren auf der im Visualisierungskonzept definierten Farbgebung und werden im Code durch entsprechende Hex-Farbcodes repräsentiert. Da das zugrunde liegende Konzept lediglich einen Farbbereich definiert, jedoch keine konkreten Farbwerte vorgibt, wurde zunächst eine Reihe von Näherungswerten manuell ausgewählt. Um die restlichen Farbwerte konsistent zu ergänzen, wurden mit Hilfe von ChatGPT weitere Farbcodes generiert, die durch angepasste Helligkeits- und Sättigungswerte das Farbkonzept aus Kapitel 2 passend abbilden. Dadurch konnte eine vollständige Farbgebung erstellt werden. Die Platzierung der Kugeln erfolgte zunächst manuell anhand fester Koordinaten. Dadurch konnte das Konzept visuell überprüft werden, auch wenn diese erste Version noch vollständig statisch war.

Im nächsten Schritt wurde die Platzierung der Kugeln automatisiert. Dies erleichtert die Wartbarkeit des Codes, da bei Änderungen der Platzierung nicht jede Kugel manuell neu positioniert werden muss, sondern nur der Algorithmus angepasst wird.

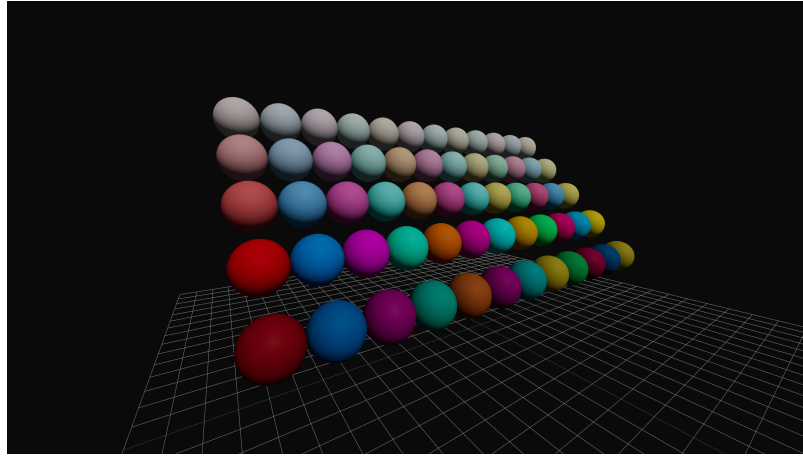


Abbildung 6. Erste Umsetzung des Visualisierungskonzepts in einer 3D-Umgebung

Zusätzlich erhielt jede Kugel einen eindeutigen Key, der dem MIDI-Wert der zugehörigen Note entspricht. Diese Verbindung zwischen Note und Farbkugel ermöglicht später das dynamische Erscheinen der Kugeln auf Tastendruck.

Technisch wurde dieser Vorgang über ein zweidimensionales Array realisiert, das die zuvor definierten Farbwerte enthält. Die inneren Arrays entsprechen dabei jeweils einer Reihe von 12 Farben, analog zur Anordnung der Töne innerhalb einer Oktave. Jedem Farbcode wurde ein eindeutiger numerischer Key zugewiesen, der einer bestimmten Tonhöhe entspricht. Diese Zuordnungen basieren auf den MIDI-Werten des verwendeten Keyboards und liegen im Bereich von 36 bis 95. Eine genauere Beschreibung des MIDI-Formats, folgt im Abschnitt zur MIDI-Logik. Die Farbzuzuweisungen wurden in einem eigens dafür definierten Objekttyp (`ColorKeyPair`) gespeichert.

Für jedes dieser Objekte wurden nun algorithmisch Koordinaten berechnet, anhand derer jedes `ColorKeyPair` als eine farbige Kugel dargestellt werden kann, die eine feste Position im dreidimensionalen Raum hat.

Anschließend wurde die MIDI-Logik implementiert. Die Web MIDI API beruft sich in ihrer Datenstruktur auf die MIDI-1.0-Spezifikation (W3C, 2025b). Die empfangenen MIDI-Daten bestehen für die in diesem Projekt relevanten Funktionen aus einem Array mit drei Zahlenwerten, die den *Status*, die *Note* und die Anschlagstärke (*Velocity*) repräsentieren. Diese Struktur entspricht dem Aufbau der MIDI-1.0-Spezifikation, bei der einem Statusbyte ein oder zwei Datenbytes folgen (The MIDI Manufacturers Association, 1996, S. 3). Ein Byte ist 8 Bit groß. Nach der MIDI-Spezifikation gibt das erste Bit (Most Significant Bit oder auch MSB) an, ob es sich um ein Statusbyte (MSB = 1) oder ein Datenbyte (MSB = 0) handelt (The MIDI Manufacturers Association, 1996). Der restliche Teil des MIDI-Bytes beträgt demnach 7 Bit und

umfasst somit einen Wertebereich von 0 bis 127. Die Tonhöhe wird in einem Datenbyte angegeben. Die bereits erwähnten Werte von 36 bis 95 entsprechen dabei genau 5 Oktaven (60 Werte für die $5 \cdot 12$ Halbtöne einer Oktave). Eine umfassende Übersicht von MIDI-Statusbytes und deren Funktionen wird von der MIDI Association bereitgestellt (MIDI Association, 2025).

Für die vorliegende Anwendung waren insbesondere die Statusbytes 147 (Note-On) und 179 (Sustain-Pedal) von Bedeutung. Obwohl das MIDI-Protokoll eigene Statusbytes für Note-On- und Note-Off-Events bereitstellt, unterscheidet das im Projekt verwendete Keyboard (Roland E-09) diese lediglich über den Velocity-Wert: Ein Statusbyte von 147 in Kombination mit einem Velocity-Wert größer als 0 signalisiert ein Note-On-Event, während ein Wert von genau 0 als Note-Off interpretiert wird. Beide Ansätze sind in der MIDI-1.0-Spezifikation dokumentiert (The MIDI Manufacturers Association, 1996, S. 10).

Um festzustellen, welche Farbkugeln erscheinen sollen, müssen aktive Noten bestimmt werden. Hierzu werden die Mengen `updatedKeys` und `sustainedNotes` betrachtet. Es werden drei Fälle unterschieden: Note-On, Note-Off und Pedal Release. Die Menge `updatedKeys` repräsentiert den aktuellen Zustand der Noten, die visuell dargestellt werden sollen.

Ein Tastendruck löst ein Note-On-Event aus. Gespielte Noten werden den `updatedKeys` hinzugefügt und gleichzeitig aus den `sustainedNotes` entfernt.

Beim Loslassen einer Taste (Note-Off) wird überprüft, ob das Sustain-Pedal gehalten wird:

- **Pedal gedrückt:** Die Note wird zu `sustainedNotes` hinzugefügt.
- **Pedal nicht gedrückt:** Die Note wird aus `updatedKeys` entfernt.

Beim Loslassen des Sustain-Pedals werden alle in `sustainedNotes` enthaltenen Noten aus `updatedKeys` entfernt. Anschließend wird `sustainedNotes` geleert.

Das nachfolgende Aktivitätsdiagramm veranschaulicht die angewandte Logik:

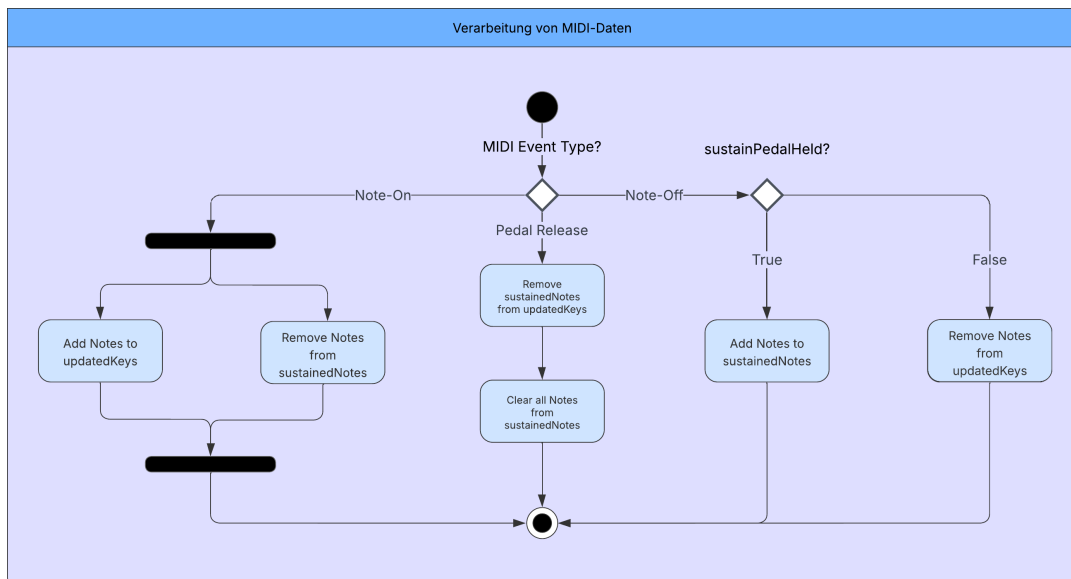


Abbildung 7. Aktivitätsdiagramm zur Verarbeitung der empfangenen MIDI-Daten

So wird gewährleistet, dass sowohl aktiv gespielte Noten als auch solche, die durch das Halten des Pedals verlängert werden, visuell durch farbige Kugeln dargestellt werden.

Die Integration der VR-Funktionalitäten erforderte keine tiefere Logik, da React Three XR hierfür einfache Lösungen bereitstellt. Zunächst muss ein sogenannter **Store** erstellt werden, der als Einstiegspunkt für die XR-Funktionalitäten dient. Dieser **Store** beinhaltet die Methode `enterVR()`, die es direkt ermöglicht die vorhandene 3D-Umgebung in VR zu betrachten. Hierbei genügt es, eine einzelne XR React-Komponente im **Canvas** zu platzieren. Die Kameraposition wurde so gewählt, dass die Farbkugeln in der 3D-Umgebung gut sichtbar sind. Ein systematischer Ansatz zur Positionierung wurde jedoch nicht verfolgt, stattdessen erfolgte die Platzierung nach subjektivem Eindruck.

3.4. Möglichkeiten und Grenzen

Im Kern dieser Arbeit soll eine Aussage darüber getroffen werden, wie die Vielfalt von Technologien einerseits Potenziale bietet, andererseits aber auch Entwicklungsprobleme verursacht. Für das vorliegende Projekt lässt sich festhalten, dass die Möglichkeiten überwiegen. Durch das breite Angebot verfügbarer Technologien konnten passende Lösungen gefunden und ein funktionierendes Gesamtsystem entwickelt werden, das den Anforderungen dieser Arbeit gerecht wird:

Der Prototyp ist webbasiert und bietet daher, wie bereits erläutert, Vorteile wie Plattformunabhängigkeit und Benutzerfreundlichkeit. Er verarbeitet MIDI-Daten eines Keyboards in Echtzeit, ohne dabei erkennbare Latenzprobleme aufzuweisen und setzt zentrale Aspekte des Visualisierungskonzepts um. Zudem wird eine immersive Nutzung durch die Verwendung von VR-Headsets ermöglicht.

Diesen Möglichkeiten stehen nur wenige Grenzen gegenüber.

Eine kleine Herausforderung stellte ein Kompatibilitätsproblem zwischen React Three Fiber und React Three XR dar. Letzteres erforderte eine ältere Version der React-Bibliothek Three.js, als React Three Fiber. Dies führte dazu, dass zur Laufzeit zwei unterschiedliche Versionen von Three.js gleichzeitig liefen, was zum Absturz der Anwendung führte. Dieses Problem ließ sich jedoch durch folgenden Eintrag in der Datei `package.json` einfach beheben:

```
"overrides": {  
  "three": "^0.176.0"  
}
```

Diese Anweisung erzwingt die Verwendung einer bestimmten Three.js-Version und verhindert somit Versionskonflikte.

Eine weitere Grenze zeigte sich bei der Nutzung des VR-Headsets: Da das Headset in VR die reale Umgebung komplett ausblendet, ist das Keyboard beim Spielen nicht sichtbar. Aktuell muss das Instrument daher blind gespielt werden.

Die Oculus Quest ist jedoch durch integrierte Kameras dazu in der Lage die reale Umgebung in der virtuellen Darstellung mit abzubilden. Aus diesem Grund wurde versucht, eine AR-Lösung umzusetzen. Durch das Einbinden der Umgebung in das virtuelle Geschehen sollte das Keyboard beim Tragen des VR-Headsets sichtbar bleiben. React Three XR stellt hierfür die Methode `enterAR()` zur Verfügung, die analog zu `enterVR()` AR-Funktionalitäten ermöglicht. Allerdings ist diese Funktion ausschließlich auf mobilen Endgeräten mit Android- oder iOS-Betriebssystem verfügbar (Google, o. D.), wodurch der Einsatz auf Desktop-Systemen derzeit ausgeschlossen ist.

Weitere technische Komplikationen traten während der Entwicklung nicht auf.

Folgende Einschränkungen ergaben sich durch den begrenzten zeitlichen Rahmen dieser Arbeit: Die Farbzuzuordnung im Prototyp basiert auf fest definierten Farbwerten und wurde nicht algorithmisch hergeleitet. Daher ist es aktuell nicht möglich, den Farbbereich dynamisch anzupassen, so wie es im Visualisierungskonzept (Kapitel 2) beschrieben ist. Zudem konnten die geplanten Darstellungen von Lautstärke und

Tondauer – durch Variation von Größe und Form der Farbkugeln – nicht realisiert werden.

Trotz dieser Einschränkungen zeigt die Umsetzung, dass die technischen Möglichkeiten im vorliegenden Projekt überwiegen. Die verwendeten Werkzeuge und Ansätze erwiesen sich insgesamt als geeignet, ein funktionierendes und responsives System zu entwickeln.

4. Fazit und Ausblick

Im Rahmen dieser Arbeit wurde untersucht, welche technischen Möglichkeiten und Grenzen sich bei der Echtzeit-Musikvisualisierung in einer webbasierten VR-Umgebung ergeben. Ziel war es, durch die Umsetzung eines Prototyps unterschiedliche Technologien in ein funktionierendes Gesamtsystem zu integrieren und dabei Potenziale und Einschränkungen zu identifizieren. Anhand dieser Untersuchung sollte die Forschungsfrage beantwortet werden: „Welche technischen Herausforderungen und Potenziale ergeben sich bei der Echtzeit-Musikvisualisierung in einer webbasierten VR-Umgebung?“

Die Ergebnisse zeigen zunächst, dass die Realisierung eines solchen Systems prinzipiell möglich ist. Insbesondere die webbasierte Natur der eingesetzten Technologien erwies sich dabei als Vorteil: Offene Standards und eine breite Entwicklerbasis führen zu einer Vielfalt verfügbarer Werkzeuge, von denen viele gut zusammenarbeiten oder sogar explizit aufeinander abgestimmt sind - wie etwa React mit React Three Fiber und React Three XR. Die Vielzahl verfügbarer Technologien eröffnet Entwicklern zudem Optionen bei der Auswahl geeigneter Werkzeuge, um sowohl individuellen Präferenzen als auch technischen Anforderungen gerecht zu werden. So kann für die Entwicklung der 3D-Umgebung das imperative Three.js genutzt werden, das mehr Kontrolle über technische Abläufe erlaubt, oder das deklarative React Three Fiber, das durch eine höhere Abstraktion sowohl die Einstiegshürde senkt als auch den Entwicklungsaufwand reduziert.

Das breite Angebot kompatibler Webtechnologien erleichtert die Entwicklung komplexer Anwendungen, die verschiedene technische Bereiche verknüpfen. Konkret konnte in dieser Arbeit gezeigt werden, dass sich externe und voneinander unabhängige Geräte wie ein MIDI-Keyboard und ein VR-Headset erfolgreich in eine gemeinsame, interaktive Umgebung integrieren lassen. Für den spezifischen Anwendungsfall der Echtzeit-Musikvisualisierung in einer webbasierten VR-Umgebung anhand von MIDI-Daten lässt sich somit eindeutig feststellen, dass eine Umsetzung gut möglich ist.

Es existieren bereits VR-basierte Lösungen zur Musikvisualisierung, die jedoch typischerweise Audiodateien oder Musikstreams von Plattformen wie YouTube oder Spotify nutzen. Eine direkte, echtzeitbasierte Verarbeitung von live gespielten MIDI-Daten in Kombination mit webbasierten VR-Technologien ist bislang nicht bekannt.

Darüber hinaus basiert die in diesem Projekt entwickelte Visualisierung auf musiktheoretischen Prinzipien. Ein solcher theoretischer Bezug konnte in bestehenden VR-basierten Musikvisualisierungen nicht festgestellt werden.

Trotz der erfolgreichen Umsetzung, weist der Prototyp einige Limitationen auf. Funktionale Einschränkungen des Browsers verhinderten die Implementierung einer AR-basierten Lösung, wodurch das Keyboard beim Tragen des VR-Headsets nicht sichtbar ist. Dies schränkt die praktische Nutzbarkeit der Anwendung ein.

Aus zeitlichen Gründen konnten nicht alle Aspekte des Visualisierungskonzepts realisiert werden: Die Visualisierung von Tondauer und Lautstärke fehlt bislang, was die Tiefe der Musikvisualisierung reduziert. Ebenso ist der Farbbereich, anders als im Konzept vorgesehen, nicht frei wählbar, wodurch die Variation der Farbgebung und damit die Vielfalt der visuellen Eindrücke eingeschränkt ist. Schließlich ist der Prototyp nicht in der Lage, eigene Klänge zu erzeugen. Aufgrund des begrenzten Zeitrahmens lag die Priorität darauf, einen funktionalen und responsiven Prototyp zu entwickeln, der die Kernaspekte Musikvisualisierung, Echtzeit-MIDI-Verarbeitung und Virtual Reality grundlegend umsetzt, auch wenn nicht alle geplanten Details des Konzepts vollständig implementiert werden konnten.

Die Ergebnisse dieser Arbeit deuten darauf hin, dass sich mit der aktuellen Technologielandschaft auch weitere medienzentrierte Anwendungen erfolgreich realisieren lassen. Selbst sehr spezifische Nischenlösungen scheinen machbar, da nur wenige technische Herausforderungen auftraten.

Für zukünftige Arbeiten ergeben sich daraus vielfältige Möglichkeiten, neue Anwendungsfälle zu entwickeln und innovative Lösungen zu gestalten. So könnten Nischenlösungen gezielt konzipiert werden, um spezifische Fragestellungen zu adressieren. Konkrete Anwendungsfälle für Musikvisualisierung in VR könnten beispielsweise pädagogische Ansätze umfassen, etwa indem sie Castels historischen Gedanken weiter aufgreifen, gehörlosen Menschen Musik zu vermitteln. Es können Projekte verfolgt werden, die das Spieler- bzw. Zuhörererlebnis zentral behandeln. Die Platzierung der Farbkugeln erfolgte im Prototyp bislang ohne systematischen Ansatz. Künftige Studien könnten sich damit befassen, ob bestimmte räumliche Anordnungen spezifische Eindrücke hervorrufen oder verstärken. Auch eine Erweiterung zur kollaborativen Nutzung ist denkbar, beispielsweise in Form digitaler Konzerte.

Insgesamt zeigt diese Arbeit, dass die Echtzeit-Musikvisualisierung in webbasierten VR-Umgebungen technisch realisierbar ist und Potenziale für weitere Untersuchungen und Anwendungen aufweist.

Literatur

- Christensen, T. (1993). *Rameau and Musical Thought in the Enlightenment*. Cambridge University Press. https://books.google.de/books?hl=en&lr=&id=170CpILluzUC&oi=fnd&pg=PR11&dq=Rameau&ots=A5TwH2Oums&sig=u6wNvAI46E8EKcHtXb2RQws3Rss&redir_esc=y#v=onepage&q=castel&f=false
- ECMA International. (2024). Standards. Zugriff 21. Juni 2025 unter <https://ecma-international.org/publications-and-standards/standards/>
- Gatzsche, G., Mehnert, M., Gatzsche, D., & Brandenburg, K. (2007). A Symmetry Based Approach for Musical Tonality Analysis. *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*. Zugriff 16. Mai 2025 unter https://ismir2007.ismir.net/proceedings/ISMIR2007_p207_gatzsche.pdf
- Gaume Echeverría, U., Guerrero Castro, F. E., & Báez López, J. M. D. (2010). Comparison between a Hardware and a software synthesizer. *2010 20th International Conference on Electronics Communications and Computers (CONIELECOMP)*, 311–314. <https://doi.org/10.1109/CONIELECOMP.2010.5440747>
- Gawboy, A. M., & Townsend, J. (2012). Scriabin and the Possible. *MTO*, 18(2), 2. Zugriff 12. Mai 2025 unter https://mtosmt.org/issues/mto.12.18.2/mto.12.18.2.gawboy_townsend.pdf
- Google. (o. D.). ARCore – Unterstützte Geräte. Zugriff 19. Juni 2025 unter <https://developers.google.com/ar/devices?hl=de>
- Hubbard, T. (1996). Synesthesia-like Mappings of Lightness, Pitch, and Melodic Interval. *The American journal of psychology*, 109, 219–38. <https://doi.org/10.2307/1423274>
- Jewanski, J. (2006). Die Farblichtmusik Alexander Lászlós. In J. Jewanski & N. Sidler (Hrsg.), *Farbe – Licht – Musik* (S. 147–148). Peter Lang. https://books.google.de/books?hl=en&lr=&id=4bmmyBhYlNoC&oi=fnd&pg=PA110&dq=Farblichtmusik&ots=LJPWaEPgK5&sig=29Wq3lDL3JjAu0w89vf0YZlHRU&redir_esc=y#v=onepage&q&f=false
- Klemenc, B., Ciuha, P., & Solina, F. (2011). Educational Possibilities of the Project Colour Visualization of Music. *Organizacija*, 44(3), 67–75. <https://doi.org/10.2478/v10051-011-0006-9>
- MIDI Association. (2025). EXPANDED MIDI 1.0 MESSAGES LIST. Zugriff 20. Juni 2025 unter <https://midi.org/expanded-midi-1-0-messages-list>

- Newton, I. (1730). *Opticks: Or, a Treatise of the Reflections, Refractions, Inflections and Colours of Light* (Fourth edition, corrected). William Innys. https://books.google.de/books/about/Opticks.html?id=GnAFAAAAQAAJ&printsec=frontcover&source=kp_read_button&hl=en&redir_esc=y#v=onepage&q&f=false
- Pazel, D. P. (2022). The Chromatic Scale and the Diatonic Foundation. In *Music Representation and Transformation in Software: Structure and Algorithms in Python* (S. 17). Springer International Publishing. https://doi.org/10.1007/978-3-030-97472-5_2
- Rauschnabel, P. A., Felix, R., Hinsch, C., Shahab, H., & Alt, F. (2022). What is XR? Towards a Framework for Augmented and Virtual Reality. *Computers in Human Behavior*, 133, 107289. <https://doi.org/https://doi.org/10.1016/j.chb.2022.107289>
- Smith, S., & Williams, G. (1997). A visualization of music. *Proceedings. Visualization '97 (Cat. No. 97CB36155)*, 499–503. <https://doi.org/10.1109/VISUAL.1997.663931>
- Stack Overflow - Developer Survey. (2024). Zugriff 16. Juni 2025 unter <https://survey.stackoverflow.co/2024/technology#most-popular-technologies-webframe>
- Tenney, J. (1988). *A History of Consonance and Dissonance*. Excelsior. https://books.google.de/books?id=_4MIAQAAMAAJ
- The MIDI Manufacturers Association. (1996). MIDI 1.0 Detailed Specification. Zugriff 20. Juni 2025 unter <https://midi.org/midi-1-0-core-specifications>
- W3C. (2025a). About W3C web standards. Zugriff 21. Juni 2025 unter <https://www.w3.org/standards/about/>
- W3C. (2025b). Web MIDI API - W3C Working Draft. Zugriff 20. Juni 2025 unter <https://www.w3.org/TR/webmidi/#terminology>

Anhang

A. Technische Ressourcen

A.1. ChatGPT-Prompts

Nachfolgend sind ChatGPT-Prompts aufgeführt, die während der Entwicklung des Prototyps verwendet wurden. Ein Pfeil kennzeichnet, dass unmittelbar auf eine Antwort der nächste Prompt folgte.

```
1 <mesh position={[-11, 5, 0]}>
2   <sphereGeometry />
3   <meshPhongMaterial color="#ff0000" />
4 </mesh>
5 <mesh position={[-9, 5, 0]}>
6   <sphereGeometry />
7   <meshPhongMaterial color="#00A6FE" />
8 </mesh>
9 <mesh position={[-7, 5, 0]}>
10  <sphereGeometry />
11  <meshPhongMaterial color="#FE00EC" />
12 </mesh>
13 <mesh position={[-5, 5, 0]}>
14  <sphereGeometry />
15  <meshPhongMaterial color="#01FFD7" />
16 </mesh>
17 <mesh position={[-3, 5, 0]}>
18  <sphereGeometry />
19  <meshPhongMaterial color="#FF8C00" />
20 </mesh>
21 <mesh position={[-1, 5, 0]}>
22  <sphereGeometry />
23  <meshPhongMaterial color="#FF00C8" />
24 </mesh>
25 <mesh position={[1, 5, 0]}>
26  <sphereGeometry />
27  <meshPhongMaterial color="#02FFFF" />
28 </mesh>
29 <mesh position={[3, 5, 0]}>
30  <sphereGeometry />
31  <meshPhongMaterial color="#FEC600" />
32 </mesh>
33 <mesh position={[5, 5, 0]}>
```

```
34     <sphereGeometry />
35     <meshPhongMaterial color="#01FF7F" />
36   </mesh>
37   <mesh position={[7, 5, 0]}>
38     <sphereGeometry />
39     <meshPhongMaterial color="#FE0096" />
40   </mesh>
41   <mesh position={[9, 5, 0]}>
42     <sphereGeometry />
43     <meshPhongMaterial color="#00CCFF" />
44   </mesh>
45   <mesh position={[11, 5, 0]}>
46     <sphereGeometry />
47     <meshPhongMaterial color="#FEE600" />
48   </mesh>
```

1. This JSX but 2 more in the y axis and 2 less in the z axis and each color a little bit lighter

2. Give me 3 sets of those, instead of the current two, the third again added further up and back but now the middle one has the colors of my original JSX set and make the lowest one darker accordingly

3. The first mesh is correctly set up. Do the rest accordingly → Do a more even spread of the brightness between all 5 rows

4. The change from Row 4 to Row 3 is very drastic. The rows above seem closer in brightness than the bottom two.

5. Type out all color hexcodes from top to bottom in a list → Do it in Code

A.2. Quellcode-Repository

Der vollständige Quellcode des Projekts ist unter folgendem Link einsehbar:

<https://github.com/Aaronasgh/bachelor-app>

Erklärung

Ich versichere an Eides Statt, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Dormagen, 30.06.2025

Ort, Datum

A handwritten signature in blue ink, consisting of stylized, overlapping loops and strokes, positioned above a horizontal line.

Unterschrift