



OSMD Native

Endbericht | Call 18 | Projekt ID 6670

Lizenz: CC-BY-SA

Inhalt

1 Einleitung	3
2 Projektbeschreibung	4
3 Verlauf der Arbeitspakete	5
Arbeitspaket 1 - Detailplanung und Formales am Projektstart	5
Arbeitspaket 2 - Erstellung der React-Native Bibliothek zum Anzeigen der MusicXML Musiknoten mit OSMD	5
Arbeitspaket 3 - Erstellung der Kotlin Bibliothek für Android zum Anzeigen der MusicXML Musiknoten mit OSMD	6
Arbeitspaket 4 - Erstellung der Swift Bibliothek für IOS zum Anzeigen der MusicXML Musiknoten mit OSMD	7
Arbeitspaket 5 - Dokumentation und Formales am Projektende	8
4 Umsetzung Förderauflagen	9
5 Liste der Projektergebnisse	10
6 Verwertung der Projektergebnisse in der Praxis	11
7 Öffentlichkeitsarbeit/ Vernetzung	12
8 Eigene Projektwebsite	12
9 Geplante Aktivitäten nach netidee-Projektende	13
10 Anregungen für Weiterentwicklungen durch Dritte	13

1 Einleitung

MusicXML ist ein offenes und weit verbreitetes Format zur digitalen Repräsentation von Musiknoten. Es wurde entwickelt, um den Austausch von Musikdokumenten zwischen verschiedenen Notationssoftware und Anwendungen zu erleichtern. Im Gegensatz zu herkömmlichen Notenblättern in Papierform ermöglicht MusicXML eine flexible und detaillierte Darstellung von Musikstücken, die auf unterschiedlichsten digitalen Plattformen verarbeitet und angezeigt werden können.

Eine zentrale Rolle bei der Darstellung von MusicXML-Daten in digitalen Anwendungen spielt OpenSheetMusicDisplay (OSMD), ein Open-Source-Projekt, das wir speziell für die Anzeige von Musiknoten in Webanwendungen entwickelt haben. OSMD erlaubt es Entwicklern, MusicXML-Dateien direkt im Webbrowser visuell ansprechend und präzise darzustellen, ohne tiefgehende Kenntnisse im Bereich der Musiknotation besitzen zu müssen. Die Bibliothek bietet eine benutzerfreundliche Lösung für die Einbindung von Musiknotation in webbasierte Projekte und wird unter anderem von vielen Musikschulen, Softwareentwicklern und Musikern weltweit genutzt.

In diesem Endbericht des Förderprojekts “OSMD Native” wird der Einsatz von MusicXML und OSMD für die Bereitstellung von Softwarebibliotheken für native mobile Betriebssysteme beschrieben, wobei die Herausforderungen und Erfolge im Laufe des Projekts detailliert dargestellt werden.

2 Projektbeschreibung

Zielgruppe und Nutznießer:

Das Projekt OSMD Native richtet sich an Entwickler, die native Android- und iOS-Apps mit MusicXML-Notenanzeige entwickeln möchten. Dies umfasst sowohl Einzelentwickler wie Studenten und selbstständige Programmierer als auch Unternehmen und Forschungseinrichtungen, die kommerzielle oder nicht-kommerzielle Anwendungen entwickeln. Die von diesen Entwicklern erstellten Apps werden schließlich von Endnutzern wie Musikschülern, Hobbymusikern oder Musiklehrern verwendet, die mit Hilfe von Noten lernen, musizieren oder Musikstücke visualisieren wollen. Diese Apps können über Plattformen wie den App Store oder Google Play bezogen werden.

Ziel des Projekts und Problemlösung:

Bisher war die Anzeige von MusicXML-Musiknoten mit OpenSheetMusicDisplay (OSMD) nur im Webbrowser möglich. Die Integration von OSMD in native iOS- oder Android-Apps stellte eine große Herausforderung dar, da sie mit erheblichen Entwicklungsaufwänden verbunden war, die oft einzelne Entwickler oder kleine Teams überforderten. Dieser hohe Aufwand führte dazu, dass viele innovative Projekte im Bereich der Musiknotation nicht realisiert wurden und wertvolle Ressourcen verschwendet wurden.

Das Projekt OSMD Native löst dieses Problem, indem es Entwicklern ermöglicht, MusicXML-Noten mit geringem Aufwand in native Apps zu integrieren. Dazu werden OSMD-Bibliotheken für die Anzeige von MusicXML-Noten in den Programmiersprachen Swift (für iOS) und Kotlin (für Android) bereitgestellt. Zusätzlich wird das beliebte Open-Source-Cross-Plattform-Framework React Native unterstützt. Hierbei wird die bestehende OSMD-Library in einen WebView eingebettet und hinsichtlich Performance und Interaktionsmöglichkeiten optimiert, um eine reibungslose Benutzererfahrung sicherzustellen.

Erwartete Ergebnisse:

Durch die Bereitstellung dieser optimierten Software-Bibliotheken wird es Entwicklern deutlich einfacher gemacht, hochwertige Musiknotenanzeige-Features in ihre nativen oder Cross-Plattform-Apps zu integrieren. Dies wird den Weg für neue, innovative Anwendungen im Bereich der Musiktechnologie ebnen und die Barriere für die Umsetzung von Projekten erheblich senken. Endnutzer können von einer größeren Auswahl an leistungsfähigen Musik-Apps profitieren, die das Erlernen und Musizieren mit digitalen Notenblättern erleichtern.

3 Verlauf der Arbeitspakete

Arbeitspaket 1 - Detailplanung und Formales am Projektstart

Die Vertragsprüfung wurde durchgeführt und das Dokument unterschrieben. Auf Basis einer Excel-Vorlage erfolgte die Projektplanung. Zudem wurde der Inhalt der Projektwebsite überprüft und der erste Blogbeitrag erstellt. Das Projektteam wurde anschließend vervollständigt.

Das AP wurde planmäßig durchgeführt.

Arbeitspaket 2 - Erstellung der React-Native Bibliothek zum Anzeigen der MusicXML Musiknoten mit OSMD

Im zweiten Arbeitspaket wurde die grundlegende React-Native Bibliothek entwickelt, um MusicXML-Noten mit Hilfe von OpenSheetMusicDisplay (OSMD) in einer React-Native App anzuzeigen. Die Umsetzung erfolgte über die Integration eines WebViews, um die Funktionen von OSMD in die native mobile App einzubetten. Die Hauptaufgaben des Arbeitspakets umfassten folgende Punkte:

1. **Umsetzung der OSMD Basisfunktionen mit WebView:** Zunächst wurde die Anzeige von MusicXML-Daten in der WebView eingerichtet, um die OSMD-Funktionalitäten wie das Rendern von Notenblättern auf Mobilgeräten verfügbar zu machen.
2. **Umsetzung der OSMD Options:** Zur Anpassung der Darstellung und des Verhaltens von OSMD wurden die Konfigurationsoptionen OSMD Options implementiert, um den Entwicklern Flexibilität bei der Handhabung von Parametern wie Layout und Darstellung zu bieten.
3. **Implementierung des Audioplayers in WebView:** Ein Audioplayer wurde in die WebView integriert, um das Abspielen von Musik in Synchronisation mit den angezeigten Noten zu ermöglichen.
4. **Blockieren der Standard WebView Funktionalitäten:** Um ein störungsfreies Nutzungserlebnis zu gewährleisten, wurden nicht benötigte Standard-Funktionen der WebView deaktiviert, damit die Benutzerinteraktionen auf die Steuerung der Notenanzeige und des Audioplayers fokussiert bleiben.
5. **Unterstützung für das Hinzufügen von MXL-Dateien:** Die Bibliothek wurde so erweitert, dass Entwickler die Möglichkeit haben, MXL-Dateien (komprimierte MusicXML-Dateien) hinzuzufügen und anzuzeigen.

6. **Zoom-Funktionalität hinzufügen:** Zur Verbesserung der Benutzerfreundlichkeit wurde eine Zoom-Funktion implementiert, die es den Nutzern ermöglicht, die Größe der angezeigten Notenblätter flexibel anzupassen.
7. **Testen:** Im Anschluss an die Implementierung wurden umfassende Tests durchgeführt, um sicherzustellen, dass alle Funktionen reibungslos arbeiten und die Bibliothek stabil in verschiedenen Anwendungsszenarien funktioniert.
8. **Erstellung der Dokumentation:** Abschließend wurde eine vollständige Dokumentation der Bibliothek erstellt, um Entwicklern eine einfache Integration und Nutzung der Funktionen in ihren eigenen Projekten zu ermöglichen.

Das Arbeitspaket 1 legte somit den Grundstein für eine funktionsfähige React-Native Bibliothek, die die Anzeige von MusicXML Musiknoten und die Audiowiedergabe in nativen mobilen Apps ermöglicht.

AP 2 wurde planmäßig zu 100% durchgeführt.

Arbeitspaket 3 - Erstellung der Kotlin Bibliothek für Android zum Anzeigen der MusicXML Musiknoten mit OSMD

Im dritten Arbeitspaket wurde eine Kotlin-basierte Bibliothek entwickelt, um MusicXML-Noten in nativen Android-Apps mit OpenSheetMusicDisplay (OSMD) darzustellen. Die Vorgehensweise orientierte sich an den Schritten aus AP 2, um eine einheitliche und benutzerfreundliche Lösung auch für Android-Plattformen zu bieten. Die Hauptaufgaben dieses Arbeitspakets umfassten:

1. **Umsetzung der OSMD Basisfunktionen:** Die grundlegende OSMD-Integration wurde in Kotlin umgesetzt, indem die Musiknotation direkt in die Android-Umgebung eingebunden und die Basisfunktionen von OSMD zur Anzeige von MusicXML-Dateien ermöglicht wurden.
2. **Implementierung der OSMDOptions:** Für eine flexible Anpassung an verschiedene Anwendungsfälle wurden Konfigurationsmöglichkeiten durch OSMD Options bereitgestellt, damit Entwickler Layout- und Anzeigeeoptionen auf einfache Weise steuern können.
3. **Audioplayer-Integration:** Ein Audioplayer wurde in die Bibliothek integriert, um synchronisierte Audiowiedergabe mit den angezeigten Notenblättern zu ermöglichen. Dies erweitert die Funktionalität der App und verbessert das Musikerlebnis für den Endnutzer.
4. **Deaktivierung unnötiger Standardfunktionen:** Um die Benutzererfahrung zu optimieren, wurden unnötige Standard-Funktionen der Android WebView gezielt deaktiviert, sodass der Fokus auf der Notenanzeige und der Audiosteuerung liegt.

5. **Unterstützung von MXL-Dateien:** Die Bibliothek wurde so erweitert, dass komprimierte MusicXML-Dateien (MXL) unterstützt werden, um ein breiteres Spektrum an Musikdateien verarbeiten und anzeigen zu können.
6. **Zoom-Funktionalität hinzufügen:** Eine Zoom-Funktion wurde integriert, um dem Nutzer die Möglichkeit zu geben, die Musiknoten an seine Bedürfnisse anzupassen, was besonders auf mobilen Geräten von großem Nutzen ist.
7. **Testen der Bibliothek:** Um die Stabilität und Leistungsfähigkeit der Bibliothek sicherzustellen, wurden umfangreiche Tests durchgeführt. Diese umfassten die Überprüfung der Performance und Kompatibilität auf verschiedenen Android-Geräten und unter verschiedenen Anwendungsszenarien.
8. **Erstellung der Dokumentation:** Eine ausführliche technische Dokumentation der Bibliothek wurde erstellt, um Entwicklern eine einfache Integration der MusicXML-Notenanzeige in Android-Apps zu ermöglichen.

Das Arbeitspaket 3 lieferte somit eine stabile und funktionsreiche Kotlin-Bibliothek, die es Entwicklern erleichtert, MusicXML-Noten und Audiowiedergabe in ihre Android-Apps zu integrieren. Dies bildet eine solide Grundlage für die Entwicklung nativer Musiknoten-Apps auf Android-Geräten.

AP 3 wurde planmäßig zu 100% durchgeführt.

Arbeitspaket 4 - Erstellung der Swift Bibliothek für iOS zum Anzeigen der MusicXML Musiknoten mit OSMD

Im vierten Arbeitspaket wurde eine Swift-basierte Bibliothek entwickelt, um die Anzeige von MusicXML-Noten in nativen iOS-Apps mit Hilfe von OpenSheetMusicDisplay (OSMD) zu ermöglichen. Die Vorgehensweise orientierte sich an den vorherigen Arbeitspaketen für React-Native und Kotlin, um eine konsistente Funktionalität auf der iOS-Plattform zu gewährleisten. Die wesentlichen Aufgaben dieses Arbeitspakets umfassten:

1. **Umsetzung der OSMD Basisfunktionen:** Die OSMD-Grundfunktionen wurden in Swift implementiert, sodass MusicXML-Dateien auf iOS-Geräten nativ angezeigt werden können. Dies ermöglichte die Darstellung von Musiknoten direkt in der App.
2. **Implementierung der OSMDOptions:** Zur flexiblen Anpassung an unterschiedliche Anforderungen wurde die Bibliothek um OSMD -Options erweitert, sodass Entwickler die Darstellung und das Layout der Noten individuell anpassen können.
3. **Integration eines Audioplayer:** Auch in die Swift-Bibliothek wurde ein Audioplayer integriert, um die Wiedergabe von Musik mit der Notenanzeige zu synchronisieren. Dies verbessert das Musikerlebnis für Endnutzer, die Noten und Musik gemeinsam nutzen möchten.

4. **Deaktivierung von Standardfunktionen in WebView:** Um die Benutzerfreundlichkeit zu maximieren, wurden Standardfunktionen der WebView deaktiviert, um störende Interaktionen zu vermeiden und die Nutzererfahrung zu optimieren.
5. **Unterstützung von MXL-Dateien:** Die Unterstützung für das Laden und Anzeigen von MXL-Dateien (komprimierte MusicXML-Dateien) wurde hinzugefügt, was eine größere Vielfalt an Notenformaten in der App ermöglicht.
6. **Hinzufügen der Zoom-Funktion:** Eine Zoom-Funktion wurde integriert, um den Nutzern die Möglichkeit zu geben, die Notenansicht individuell zu vergrößern oder zu verkleinern. Diese Funktion ist besonders auf mobilen Geräten nützlich, um die Lesbarkeit zu verbessern.
7. **Testen der Bibliothek:** Umfangreiche Tests wurden durchgeführt, um die Stabilität und Leistung der Bibliothek auf verschiedenen iOS-Geräten sicherzustellen. Hierbei wurden sowohl funktionale Tests als auch Leistungstests in realen Anwendungsszenarien durchgeführt.
8. **Erstellung der Dokumentation:** Abschließend wurde eine umfassende Dokumentation erstellt, die Entwicklern eine klare Anleitung zur Integration der Swift-Bibliothek in ihre iOS-Projekte bietet. Sie enthält Details zur Implementierung, Konfiguration und Nutzung der Funktionen.

Mit diesem Arbeitspaket wurde eine voll funktionsfähige Swift-Bibliothek bereitgestellt, die es Entwicklern ermöglicht, MusicXML-Noten und synchronisierte Audiowiedergabe in iOS-Apps zu integrieren. Dies eröffnet neue Möglichkeiten für die Entwicklung nativer Musik-Apps auf Apple-Geräten und stellt eine wichtige Ergänzung zum plattformübergreifenden Angebot dar.

AP 4 wurde planmäßig zu 100% durchgeführt.

Arbeitspaket 5 - Dokumentation und Formales am Projektende

Der Projektverlauf wurde im Projektendbericht, in der Zusammenfassung und der Endabrechnung dokumentiert. Anschließend wurden der Endbericht, die Zusammenfassung und die Endabrechnung erstellt. Diese sowie die Anwender/innen- und Entwickler/innen-Dokumentation wurden auf die Projektwebsite hochgeladen. Abschließend wurde ein finaler Blogbeitrag verfasst.

Das AP wurde planmäßig durchgeführt.

4 Umsetzung Förderauflagen

Es gab keine speziellen Förderauflagen.

5 Liste der Projektergebnisse

1	Projektzwischenbericht	CC-BY-SA	https://www.netidee.at/sites/default/files/2024-05/prj6670_Call18_Zwischenbericht_V03.pdf
2	Projektendbericht	CC-BY-SA	https://www.netidee.at/sites/default/files/2024-10/prj6670_call18_Endbericht_V02.pdf
3	Entwickler_innen Dokumentation	CC-BY-SA	https://www.netidee.at/sites/default/files/2024-09/prj6670%20Entwicklerdoku%20OSMD%20Native.pdf
4	Anwender_innen Dokumentation	CC-BY-SA	<i>OSMD Native ist eine reine Entwicklersoftware dh. AnwenderInnen Dokumentation ist vom Entwickler in Abhängigkeit vom jeweiligen individuellen Projekt zu erstellen.</i>
5	Veröffentlichungsfähiger Einseiter	CC-BY-SA	https://www.netidee.at/sites/default/files/2024-10/prj6670_netidee_call18_Zusammenfassung_V02.pdf
6	Dokumentation Externkommunikation	CC-BY-SA	https://www.netidee.at/sites/default/files/2024-09/prj6670_netidee_call18_Externkommunikation_Projekte_V01.pdf
7	Software Bibliothek für die Anzeige von MusicXML Musiknoten mit OSMD in React Native Apps.	BSD-3-clause	https://github.com/opensheetmusicdisplay/osmd-native/tree/7d3e7fc37b098f8ab82ab3af93fc03d9dc44bce8/react-native-osmd
8	Software Bibliothek für die Anzeige von MusicXML Musiknoten mit OSMD in Kotlin (Android) Apps.	BSD-3-clause	https://github.com/opensheetmusicdisplay/osmd-native/tree/7d3e7fc37b098f8ab82ab3af93fc03d9dc44bce8/osmd-kotlin
9	Software Bibliothek für die Anzeige von MusicXML Musiknoten mit OSMD in Swift (iOS) Apps.	BSD-3-clause	https://github.com/opensheetmusicdisplay/osmd-native/tree/7d3e7fc37b098f8ab82ab3af93fc03d9dc44bce8/osmd-swift

6 Verwertung der Projektergebnisse in der Praxis

OSMD Native kann für verschiedenste Anwendungen verwendet werden, bei denen das Abspielen bzw. Anhören von Musiknoten sinnvoll ist:

- Digitale Anwendungen im Musikunterricht
- Anwendungen zum Lernen von Instrumenten
- Websites von Verlage bzw. Notenplattformen

Werden die Anwendungen nun nativ für mobile Endgeräte angeboten ergeben sich folgende Vorteile:

1. **Verfügbarkeit:** Die Handydichte liegt in den meisten Ländern Europas bei 100% und darüber. Dh. jeder (erwachsene) Mensch besitzt ein Mobiltelefon und führt es mit sich.
2. **Höhere Leistung:** Da sie speziell für ein Betriebssystem (z.B. iOS oder Android) entwickelt wurden, können native Apps die Hardware und Software des Geräts effizienter nutzen, was zu einer schnelleren und reaktionsfähigeren Benutzererfahrung führt.
3. **Bessere Benutzererfahrung:** Native Apps passen sich an die Benutzeroberfläche des jeweiligen Betriebssystems an, was eine vertraute und intuitive Interaktion ermöglicht.
4. **Zugriff auf Gerätefunktionen:** Sie können direkt auf Funktionen wie Kamera, GPS, Mikrofon und Benachrichtigungen zugreifen, was eine umfassendere Nutzung der Gerätefeatures ermöglicht.
5. **Offline-Funktionalität:** Native Apps können oft auch ohne Internetverbindung verwendet werden, da sie lokal auf dem Gerät installiert sind.
6. **Höhere Sicherheit:** Sie bieten in der Regel bessere Sicherheitsstandards, da sie über die offiziellen App-Stores (Google Play, App Store) installiert werden, die strenge Prüfungen durchführen.

Diese Vorteile machen native Apps besonders attraktiv für Unternehmen, die leistungsstarke und stabile mobile Anwendungen bereitstellen möchten.

7 Öffentlichkeitsarbeit/ Vernetzung

Die Vernetzung vor allem mit Entwicklern, die OSMD für die Umsetzung ihrer Projekte verwenden oder in Betracht ziehen, erfolgt größtenteils über unsere Organisationsseite GitHub.com (<https://github.com/opensheetmusicdisplay>). Vereinzelt erfolgen Anfragen über unsere Facebook Seite oder direkt per Email. Diese werden aber dann gebeten, ihre Anliegen ebenfalls öffentlich auf Github zu formulieren, damit auch die Community davon lernt.

Seit Sommer 2021 bieten wir den Entwicklern auf der Kommunikationsplattform "Discord" einen direkten Zugang zu OSMD, dh. es findet ein unkomplizierter, direkter Austausch statt. Unter folgendem Link kann man beitreten: <https://discord.gg/Vmj8qSuWau>

OSMD Native wurde natürlich auch im Blog unserer Website (<https://opensheetmusicdisplay.org/osmd-blog/>) sowie in unseren sozialen Kanälen kommuniziert:

Instagram: https://www.instagram.com/open_sheet_music_display/

Facebook: <https://www.facebook.com/opensheetmusicdisplay/>

Twitter: <https://twitter.com/osmdengine>

Youtube: https://www.youtube.com/channel/UHCMvITdC1dX2esPXDe_7Hw

8 Eigene Projektwebsite

Die eigene Projektwebsite ist unter <https://opensheetmusicdisplay.org> zu finden.

9 Geplante Aktivitäten nach netidee-Projektende

Nach Projektende ist geplant, die Library weiter zu verbessern und um Features zu erweitern. Welche das genau sein werden, richtet sich nach den Anforderungen der Entwicklercommunity.

Für sämtliche geplanten Aktivitäten ist jedoch eine entsprechende Finanzierung nötig. Als Geschäftsmodell testen wir derzeit gerade drei Ansätze parallel:

- Individuelle Anpassung der Notenanzeige bzw. des Audio Players gegen Bezahlung. Dies ist vor allem für kommerzielle Projekte interessant
- Freemium Modell bei WordPress Plugin: Bestimmte Features sind nur gegen Bezahlabo verfügbar
- [Sponsorware](#) Modell für TypeScript Bibliothek: Bestimmte Features der Entwickler Library sind nur mit einem aktiven [GitHub Sponsorship](#) zugänglich.

Erste Erfahrungen mit diesen Ansätzen waren sehr vielversprechend. Wir sind zuversichtlich den OSMD Native in Zukunft aus eigener Kraft weiterentwickeln zu können.

Zusätzlich werden die OSMD Social-Media Kanäle bedient.

Ein eigener Bereich für die Visualisierung von mit OSMD erstellen Projekte ist ebenfalls in Planung.

10 Anregungen für Weiterentwicklungen durch Dritte

Die am meisten nachgefragte Weiterentwicklung ist die “Zuhören”-Funktion für OSMD. OSMD kann dabei die erzeugten Töne der Musiker mit dem Notenblatt abgleichen und so Rückmeldung über die Qualität hinsichtlich Tonhöhe und Rhythmus geben.