



# OSMD Audio Player

Zwischenbericht | Call 14 | Projekt ID 4605

Lizenz: CC-BY-SA

# Inhalt

<b>Einleitung</b>	<b>3</b>
<b>Status der Arbeitspakete</b>	<b>4</b>
Arbeitspaket 1 - Detailplanung und Formales am Projektstart	4
Arbeitspaket 2 - Projektmanagement	4
Arbeitspaket 3 - Multithreading	4
Arbeitspaket 4 - Selektierbare Noten	4
Arbeitspaket 5 - Automatisches Scrolling	5
Arbeitspaket 6 - Wiederholungszeichen	5
Arbeitspaket 7 - Implementierung eines Playback Threads	6
Arbeitspaket 8 - Implementierung eines Metronoms mit Einzähler	6
Arbeitspaket 9 - Dokumentation und Formales am Projektende	6
<b>Zusammenfassung Planaktualisierung</b>	<b>7</b>
<b>Öffentlichkeitsarbeit/ Vernetzung</b>	<b>7</b>
<b>Eigene Projektwebsite</b>	<b>7</b>

## 1 Einleitung

Das vorliegende Projekt ist voll im Plan. Es gab bis jetzt keine unvorhergesehenen Herausforderungen. Im Gegenteil - wir freuen uns ab Juli Jimmy Utterström aus Schweden, der im vergangenen Jahr bereits einen ersten einfachen Prototypen eines OSMD Audio Players entwickelt hat, in beratender Funktion im OSMD Kernteam begrüßen zu dürfen. Er wird die Qualität des Projekts nochmals erhöhen.

Stichtag für den Zwischenbericht ist der 30.6.2020.

## 2 Status der Arbeitspakete

### 2.1 Arbeitspaket 1 - Detailplanung und Formales am Projektstart

Das AP wurde planmäßig durchgeführt.

### 2.2 Arbeitspaket 2 - Projektmanagement

Durchführung von internen Meetings, Protokollen, Projektcontrolling.

AP 2 wurde planmäßig zu 50% durchgeführt.

### 2.3 Arbeitspaket 3 - Multithreading

Implementierung von Multithreading der Anzeige und des Playbacks mittels Webworker Threads.

Mittels Webworker werden im Hintergrund des Browsers Thread-ähnliche Strukturen zur Verarbeitung von parallel durchzuführenden, rechenintensiven Aufgaben ermöglicht, dh die komplexen Berechnungen für die Tonwiedergabe beeinträchtigen nicht die Usability und halten die Webseite responsive.

Das Herausforderung hierbei ist der Austausch der Daten zwischen dem Browser-Main-Thread und den Webworker-Threads, da immer nur ein Thread auf die Daten zugreifen darf, um gleichzeitiges Verändern der Daten zu verhindern. Für die Berechnung der vertikalen Position im Notenblatt wurde hier ein Messaging zwischen Main-Thread und Worker verwendet, wo nur die notwendigen Daten übergeben werden, die für die Berechnung der Position gebraucht werden.

AP 3 wurde vollständig durchgeführt. Es wurden sogenannte Webworker geschaffen, die im Browser die rechenintensiven Aufgaben wie zB. der Tonwiedergabe parallel durchführen.

### 2.4 Arbeitspaket 4 - Selektierbare Noten

Implementierung von selektierbaren Noten. Modellierung von Bildschirmkoordinaten und Koordinaten am Notenblatt sowie BoundingBoxes der Notenobjekte.

Die Klickposition am Bildschirm wurde auf die Notenblattposition umgerechnet. Abspielen der selektierten Noten.Über einen Klick am Schirm kann die dort befindliche Notenposition bestimmt werden und somit der Cursor an diese Stelle gesetzt bzw. die zugehörigen Noten abgespielt werden.

AP 4 wurde zu 80% fertiggestellt. Noten können am Bildschirm angeklickt werden. Dadurch wird der Cursor gesetzt und die geklickte Note abgespielt. Es folgt Integration in das Playback.

Das Optimierungsproblem hierbei betrifft die Performance: Um nicht für eine angeklickte Bildschirmposition das gesamte Datenmodell der grafischen Objekte durchsuchen zu müssen, kann hier durch Optimierungen wie Filter auf Objekttypen (z.B. eine Note oder ein Notenschlüssel) und deren Eigenschaften die Performance verbessert werden. Dadurch ist es auch möglich bei bestimmten Modi nur bestimmte Objekttypen anklicken zu können.

## **2.5 Arbeitspaket 5 - Automatisches Scrolling**

Berechnung Position für das automatische Scrolling.

Automatisches Scrollen der Anzeige, abhängig von der Playback-Position, inklusive Rücksprung bei Wiederholungen.

AP 5 wurde zu 50% fertig gestellt. In diesem AP wird Code generiert, der sicherstellt, dass das Notenblatt auf dem Bildschirm automatisch mit der Playbackposition mitgescrolled wird inklusive Rücksprung bei Wiederholungen (siehe auch AP 6).

Hierbei wird über die aktuelle Cursorposition, welche über eine hochzählende Zeitvariable bestimmt wird, das aktuelle System im Notenblatt bestimmt. Durch den vorher durchgeführten Layout-Prozess des gesamten Notenblattes hat dieses nun eine definierte vertikale Position. Das Display kann nun bei jedem Cursor-Update auf die berechnete System-Position geschoben werden. Beim Sprung zum nächsten System soll ein weicher Übergang der Display-Position berechnet werden.

## **2.6 Arbeitspaket 6 - Wiederholungszeichen**

Einlesen und Interpretieren von Wiederholungszeichen (spezielle Taktlinien, da capo, dal segno, al fine). Anzeige und Playback reagiert auf Wiederholungszeichen.

AP 6 wurde zu 25% durchgeführt. In diesem AP wird Code generiert, der auf die verschiedenen Wiederholungszeichen reagiert, Notenblatt und Cursor an die korrekte Stelle "schiebt" und an dieser Position wieder die Audiowiedergabe startet.

Wiederholungszeichen werden bereits korrekt eingelesen und im internen Musik-Objektmodell verarbeitet, um letztendlich Wiederholungen korrekt zu erkennen und interpretieren zu können. Der Iterator, der die Folge der Noten erstellt, kann somit die Wiederholungszeichen korrekt verarbeiten, und erstellt demnach Notensequenzen inklusive der Wiederholungen für das Playback.

## **2.7 Arbeitspaket 7 - Implementierung eines Playback Threads**

Wartezeiten in Millisekunden zwischen aufeinanderfolgenden Noten werden korrekt berechnet und abgewartet. Dann werden Samples abgespielt und die Position im Notenblatt gesetzt. Die Töne werden gleichzeitig zu den gezeigten Noten abgespielt.

AP 7 wurde zu 20% durchgeführt. In diesem AP wird Code generiert der die Zeitspanne der Noten korrekt berechnet um dann die Audio Samples abzuspielen. Dadurch wird der Effekt generiert, dass gleichzeitig zu den Noten die Töne abgespielt werden.

Die Planung und Tests hierzu haben ergeben, dass die Berechnung der Zeitspannen bis zur nächsten zu spielenden Note auf eine unabhängig laufende hochzählende Zeitvariable bezogen werden muss, um keine Schwankungen und Verzögerungen durch die Berechnungszeiten zu bekommen.

## **2.8 Arbeitspaket 8 - Implementierung eines Metronoms mit Einzähler**

AP 8 wurde noch nicht begonnen.

## **2.9 Arbeitspaket 9 - Dokumentation und Formales am Projektende**

AP 9 wurde noch nicht begonnen.

### 3 Zusammenfassung Planaktualisierung

Das Enddatum folgender Arbeitspakete wurde nach hinten verlegt:

AP 4 - Selektierbare Noten: Neuer Endtermin ist der 30. September 2020

AP 5 - Scrolling: Neuer Endtermin ist der 9. Oktober 2020

AP 6 - Wiederholungszeichen: Neuer Endtermin ist der 28. Oktober 2020

Der Aufwand bleibt bei allen drei APs gleich. Der Grund für die Änderungen liegt in der Verschränktheit mit dem eigentlichen Audio Playback. Vieles kann bereits vorbereitet werden, einiges muss aber dann noch im Zuge der Entwicklung des Playback Threads angepasst werden.

### 4 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 kommen 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.

### 5 Eigene Projektwebsite

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