

HAW Hamburg

BSc Media Systems

WiSe 2018/2019

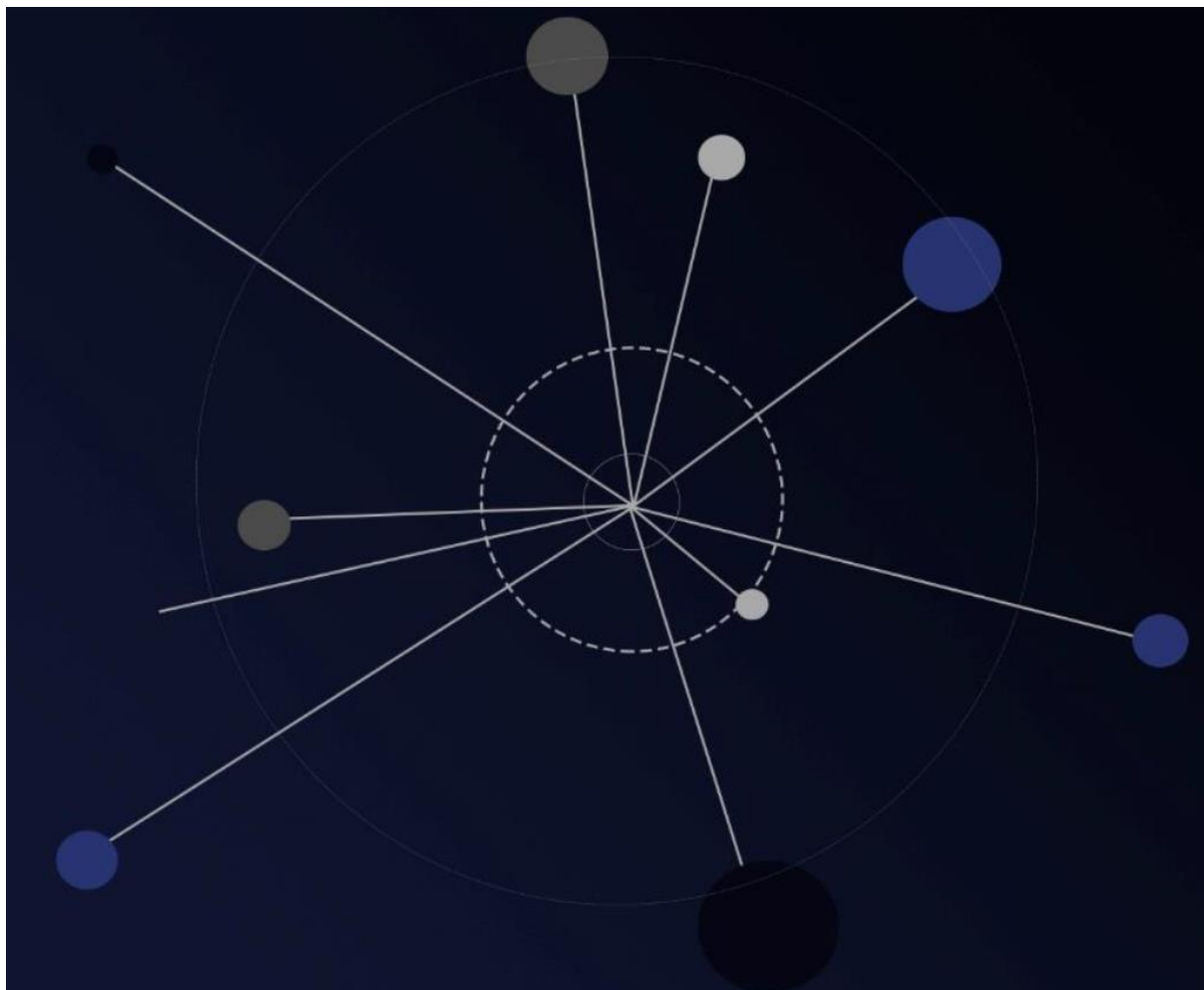
Audio-Video Programmierung

Dozent: Andreas Pläß



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

AVPRG-Projekt Orbit



Inga Fuhrmann 2220971, inga.fuhrmann@haw-hamburg.de

Tobias Pleß 2069863, plesso@gmx.de

Inhalt

1 Einleitung

2 Projektziel

3 Anforderungsanalyse

4 Technische Rahmenbedingungen

5 Technisches Konzept

6 Bedienkonzepte

7 Zeitplan

8 Teamplanung

1 Einleitung

Im Rahmen des Seminars Audio-Video Programmierung soll eine Web-Applikation entworfen werden, welche, die in den Vorlesungen behandelten Themen, der Modifikation von Audio-Dateien, enthält und anwendet. Der Bereich des „Machine Learning“ ist hierbei als optionale Erweiterung vorgesehen. In unserem Projekt *Orbit* sollen die erlernten Methoden im Zusammenhang mit einem künstlerisch-abstrakten Konzept kombiniert werden. Das Projekt wird zum Ende der Arbeitsphase mittels einer Live-Präsentation vorgestellt. Zu einem früheren Zeitpunkt werden die Funktionsweise und das Konzept anhand eines Papier-Prototypens deutlich gemacht.

2 Projektziel

Ziel des Projektes ist es einen funktionsfähigen Equalizer für Audio-Dateien zu erstellen, dessen Bedienung anhand abstrakter grafischer Elemente erfolgt.

Für eine Vorführung des Projekts wird eine User-Story nachempfunden, die Schritt für Schritt durch die Funktionen der Web-Applikation führt. Diese wird beim Upload der zu verwendenden Audio-Datei beginnen. Anschließend wird eine Einführung in das Strahlenmodell, die Symbolik und die Farben der Web Audio API Modifikationen erfolgen. Schlussendlich soll gezeigt werden, wie zusätzliche Audio-Dateien in das Audio-Projekt eingearbeitet werden können.

3 Anforderungsanalyse

Zur Sicherstellung des Erfolgs des Projektes werden nachfolgend im Rahmen einer Anforderungsanalyse alle funktionalen und nichtfunktionalen Anforderungen, die zum Erreichen des Projektziels notwendig sind, beschrieben.

Zum Hauptbestandteil der Webanwendung gehört eine grafische Benutzeroberfläche. Die dazu gehörenden grafischen Elemente sollen per „Drag and Drop“-Funktion veränderbar sein und dabei unterschiedliche Parameter eines vom User hochgeladenen oder selbsterstellten Songs verändern. Zu den grafischen Konfigurationselementen gehören Linien, Kreise, Symbole und Farben.

In der Ausgangslage umspannen zwei Ringe einen, durch einen gefüllten Kreis angezeigten, Mittelpunkt.

Der äußere Ring dient zur Darstellung einer Tonspur. Die Tonspur wird erzeugt, indem eine Audio-Datei aus einem Ordner in die Fläche der Web-Applikation gezogen wird. Die Tonspur soll Zeiten anzeigen und das bekannte Klangmuster.

Der innere Ring dient als Fixpunkt für weitere Einstellungsmöglichkeiten, indem er als Null-Linie für die Werte hinzugefügter Modifikationen dient.

Durch einen Klick in die Mitte des Kreises wird ein Menü mit wählbaren Modifikationen eingeblendet. Diese können durch „Drag and Drop“ hinzugefügt werden. Die Modifikationen werden durch ein Symbol und eine Linie dargestellt. Vom Mittelpunkt ausgehend, sollen die Linien, an deren Ende sich jeweils ein Symbol befindet, in der Länge veränderbar sein. Durch die Lage des Symbols in Relation zum inneren Ring erhalten die Einstellungen der Modifikationen einen negativen oder einen positiven Wert.

Damit ein Bereich der Tonspur durch eine Modifikation beeinflusst wird, muss die Linie per „Drag and Drop“ zur Seite gezogen werden. Hierbei entsteht eine zweite Linie, die durch eine dritte Linie in visueller Relation zur Ersten steht. Das Symbol wird dabei auf den Mittelpunkt der dritten Linie gesetzt. Hilfslinien helfen bei der Orientierung auf der Tonspur.

Sollen weitere Sounds der vorhandenen Tonspur hinzugefügt werden, muss zunächst die Einstellung „Add Sounds“ aktiv sein. Ist sie dies nicht, wird die bisherige Tonspur durch den neuen Sound ersetzt. Weitere Sounds werden ebenfalls per „Drag and Drop“ Funktion hinzugefügt. Die hinzugefügten Sounds werden als Linien oberhalb des äußeren Ringes angezeigt. Die Länge dieser Linien betrifft die Dauer des hinzugefügten Sounds in der Tonspur. Die Dicke hat Auswirkungen auf die Anzahl der Wiederholungen des Sounds innerhalb des gesetzten Zeitraumes.

Eine nebenstehende Info-Box gibt jederzeit zusätzliche Informationen und Hilfestellungen über ausgewählte grafische Elemente der Web-Applikation.

Informationen zu den gesetzten Einstellungen werden bei zusätzlich hinzugefügten Sounds und Modifikationen dadurch angezeigt, indem man den Cursor über die entsprechende grafische Oberfläche hält. Das entstandene Informationsfeld kann auch angewählt werden um Werte direkt, ohne Einbeziehung der grafischen Regulierung, zu ändern.

Über einen Slider kann eine Gesamt-Stimmung für die Tonspur gewählt werden. Diese beeinflusst die Werte eingestellter Modifikationen anhand gesetzter Parameter. Der Hintergrund der Web-Applikation ändert sich dementsprechend um eine passende Farbe zur gewählten Stimmung wiederzugeben.

Es ist auch möglich keine Audio-Datei in die Tonspur hochzuladen, sondern stattdessen nur Sounds zu verwenden um ein eigenes Musikstück zu erstellen. Dahingehende Einstellungen können durch Auswahl des äußeren Ringes vorgenommen werden.

Um den Fortschritt des Projekts zu sichern, kann dieses über einen Explorer gespeichert werden. Hierbei ist eine Sicherung als MP3-Datei vorgesehen.

Durch „Machine Learning“ wird die Tonspur der gewählten Stimmung entsprechend bearbeitet.

„Machine Learning“ kann auch dafür eingesetzt werden eine Abfolge von Sounds automatisch zu ergänzen. Auch hier wird auf die gewählte Stimmung Bezug genommen.

Über eine Aufnahmefunktion lassen sich direkt per Mikrofon aufgenommene Sounds der Web-Applikation hinzufügen.

Die Benutzeroberfläche soll minimalistisch und aufgeräumt sein, um den Fokus allein auf den Equalizer zu lenken und die Neugier des Users zu wecken. Da durch die grafische Benutzeroberfläche des Equalizers jedoch auch ein hoher Abstraktionsgrad gegeben ist und der User die Funktionen eventuell nicht intuitiv erfassen kann, ist es notwendig, Informationen zur Funktionsweise bereitzustellen. Diese sollen am rechten Rand in einer Informationsgrafik werden. So kann sichergestellt werden, dass der User ohne großen Aufwand zum Ziel kommt.

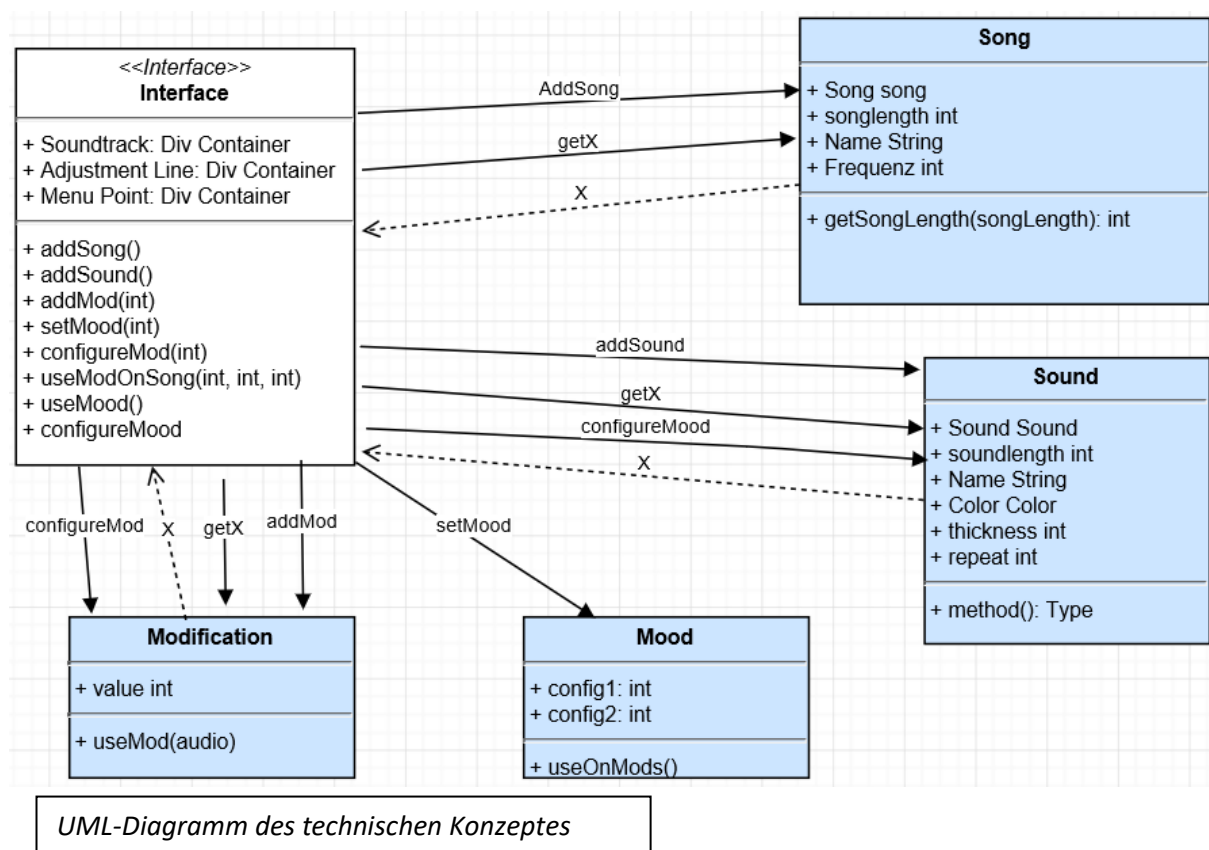
Von großer Wichtigkeit in diesem Projekt ist außerdem die Veränderung der Parameter in Echtzeit. Die Anwendung verliert ihren Nutzen und lässt Frustration beim User entstehen, wenn die Filter und Modifizierungen verzögert auf Songs angewendet werden. Daher soll der Code so effizient wie möglich programmiert sein. Des Weiteren soll sichergestellt sein, dass die Anwendung in allen gängigen Browsern funktioniert und sich auch für alle Bildschirmgrößen eignet.

4 Technische Rahmenbedingungen

Das Projekt wird vornehmlich für das Windows-Betriebsprogramm und die gängigen Browser konzipiert. Als gängige Browser sehen wir hierbei Chrome, Firefox und den Windows eigenen Browser Edge. Safari wird nicht unterstützt, da hier die technischen Möglichkeiten zur Anpassung fehlen. Um der grafischen Benutzeroberfläche eine Gestalt zu geben, wird HTML und CSS verwendet. Die so erzeugten Elemente sollen durch die Einbindung von JavaScript bzw. jQuery der Bedienung der Web-Applikation dienen. Zur Bearbeitung der Audio-Dateien wird die Web Audio API im Code eingebunden. Abrufbar soll die Web-Applikation über die Internetseite GitHub sein.

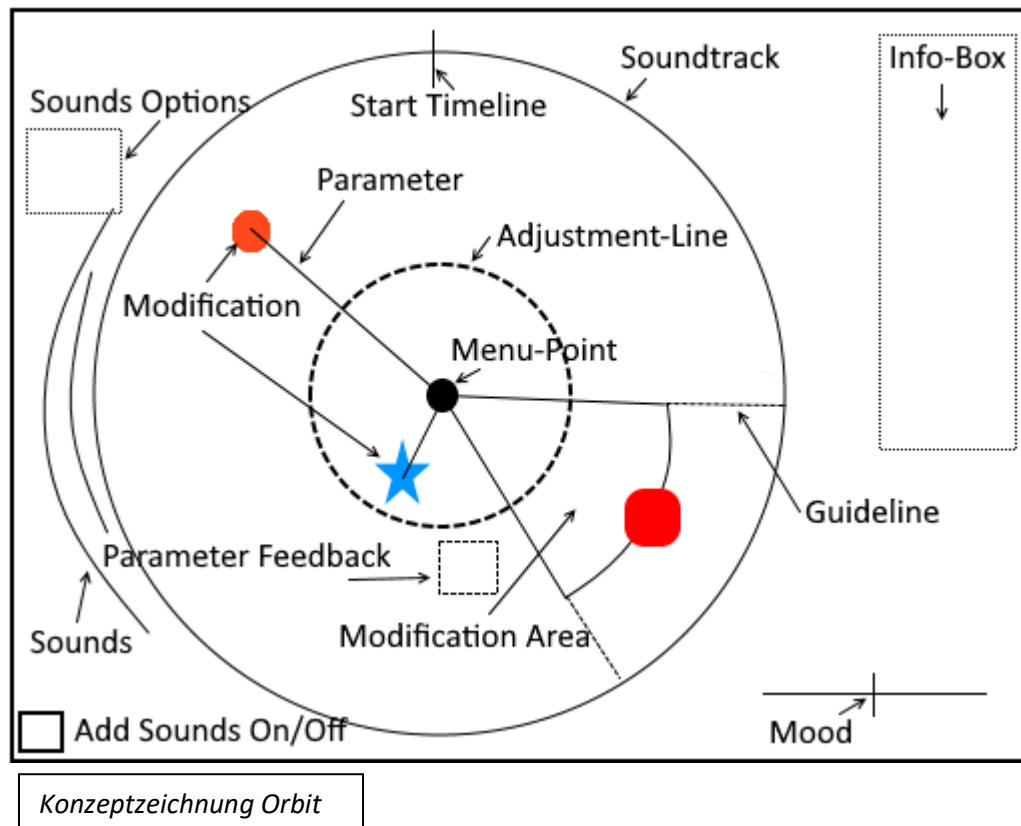
5 Technisches Konzept

Nachfolgend wird der geplante Lösungsweg anhand eines UML-Klassendiagrammes dargestellt.



6 Bedienkonzept

Die Nutzung der Web-Applikation kann durch die Eingabemöglichkeiten von Maus und Tastatur erfolgen. Die Web-Applikation ist aber so ausgelegt, dass die alleinige Nutzung einer Maus zur Bedienung ausreicht.



Audio-Dateien können über eine „Drag & Drop“-Funktion der Web-Applikation hinzugefügt werden. Die entsprechende Audio Datei wird anschließend, als Tonspur dargestellt, dem *Soundtrack* hinzugefügt. Hierbei muss jedoch das Optionsfeld *Add Sounds* in der Web-Applikation aktiv sein. Dieser Zustand ist standardmäßig aktiv, bis eine erste Audio-Datei hinzugefügt wurde. Danach ist dieses Optionsfeld inaktiv. Der Startpunkt des Audio-Tracks befindet sich bei *Start Timeline*.

Weitere Audio-Dateien werden als *Sounds* dem Projekt hinzugefügt. *Sounds* werden als Linien außerhalb des Kreises des *Soundtracks* dargestellt. Die Länge der Linie gibt den Bereich an, in dem der *Sound* abgespielt wird. Die Dicke der Linie gibt die Intensität der Wiedergabe des *Sounds* wieder. Beim Verharren des Mauszeigers über der *Sounds*-Linie wird ein Info-Feld (*Sounds Options*) mit weiteren Informationen angezeigt.

Durch das Anwählen des *Menu-Points* werden Funktionen zur weiteren Bearbeitung des *Soundtracks* eingeblendet. Einzelne *Modifications* lassen sich hier auswählen und dem Projekt hinzufügen. Die

Relation des Symbols zur *Adjustment-Line* gibt den Wert des *Parameters* an. Ist das Symbol außerhalb der *Adjustment-Line* befindet sich der Parameter im positiven Bereich, befindet sich das Symbol innerhalb, im negativen Bereich. Die *Parameter*-Linie kann mit dem Mauszeiger gegriffen werden um auf einen größeren Bereich des *Soundtracks* angewendet zu werden. Über eine *Guideline* kann der Bereich genauer festgestellt werden. Beim Verharren des Mauszeigers über der *Parameter*-Linie wird ein Info-Feld (*Parameter Feedback*) mit weiteren Informationen angezeigt.

Die *Mood*-Funktion beeinflusst das gesamte Projekt und verändert die Ausgabe des Songs anhand gesetzter Parameter.

Die *Info-Box* gibt zu jeder Zeit weiterführende Informationen zu den grafischen Bedienelementen der Web-Applikation.

7 Zeitplan

Die grafische Darstellung des Zeitplans ist im Anhang des Konzeptentwurfes enthalten. Es wird von einem geschätzten Aufwand in Höhe von ca. 200 Personenstunden gerechnet. Die Personenstunden sind aufgeteilt auf drei Bereiche, deren Abschluss zugleich die Bewältigung eines Meilensteines darstellt.

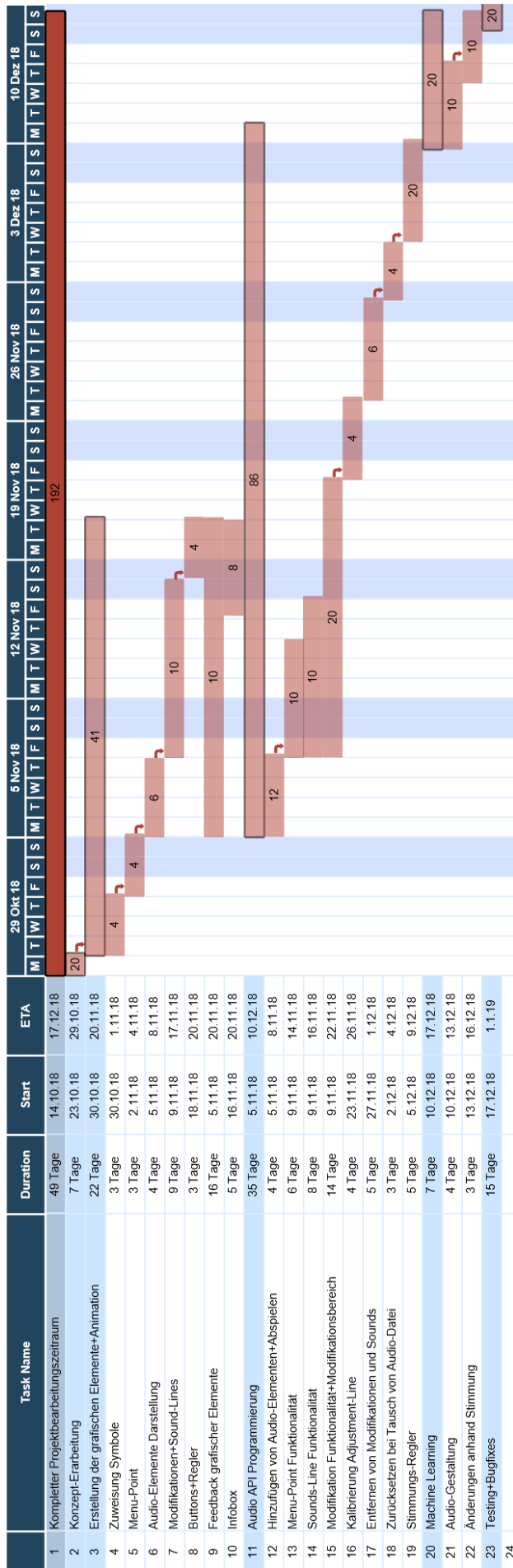
Der erste Meilenstein ist die grafische Benutzeroberfläche der Web-Applikation. Durch die Einstellungen der grafischen Elemente werden Werte zurückgegeben, die auf dem Weg zur Vollendung des zweiten Meilensteines, der Audio-API Programmierung, in den Code eingeflochten werden. Da beide Phasen voneinander abhängig sind, beginnt die Programmierung bereits im frühen Stadium der Erstellung der grafischen Benutzeroberfläche. Ab der vierten Woche sollten alle Teile, die eine direkte Kommunikation zwischen der grafischen Benutzeroberfläche und der Audio-API Programmierung erfordern, zum Abschluss kommen. Nach weiteren zwei Wochen der Programmierung erfolgt als letzte Phase und Meilenstein die Implementierung von „Machine Learning“. Die Zeit nach der Präsentation am 18.12.18 wird dazu genutzt den Code zu bearbeiten, zu verbessern und eventuelle Fehler zu entfernen.

8 Teamplanung

Die am Projekt beteiligten Personen sind namentlich Inga Fuhrmann (2220971) und Tobias Pleß (2069863). Bei einem so kleinen Projekt ist es unumgänglich, dass jede der Beteiligten Personen am vollen Umfang beteiligt ist. Als Ansprechpartner zwischen den Studenten und dem Dozenten sei Tobias Pleß genannt. Inga Fuhrmann kennt sich bereits gut mit den Gestaltungsmöglichkeiten von HTML/CSS und jQuery aus, sie wird hier deshalb den Großteil der anfallenden Aufgaben bewältigen.

Die Audio API-Programmierung und die Implementierung des „Machine Learnings“ teilen die Studenten flexibel unter sich auf.

Die Bereiche des Konzeptentwurfes wurden zur Erstellung unter den Studenten aufgeteilt und anschließend bei einem gemeinsamen Termin zusammengefügt. Der Papierprototyp wurde von Tobias Pleß angefertigt.



Anhang