

Dokumentation

Mass Click Orchestra

Tobias Gärtner - 551145

Jonas Leitner - 551527

Dozent: Herr Tobias Hiep

Projektarbeit Medieninformatik HS Flensburg

PROJEKTIDEE

Ziel dieser Projektarbeit war, eine Webanwendung zu entwerfen, in welcher Nutzer in Echtzeit kooperativ musizieren können.

Es gibt verschiedene Räume, zwischen denen man sich frei entscheiden kann, und zu welchem man andere User einladen kann, in denen man ihnen die Raum-ID nennt oder die URL schickt.

FRONTEND

Zunächst soll es die Möglichkeit geben, nach Betreten der Seite einen Raum zu finden. Auf der Startseite kann man daher die Raumnummer eingeben oder einem zufälligen Raum beitreten. Ist man nun in einem Raum, sieht man ein Rasterlayout, welches mit Blöcken gefüllt ist. Klickt man einen Block an, färbt sich dieser und gilt solange als aktiv, bis er wieder abgewählt wird. Beim Abspielen werden nur alle aktiven Blöcke abgespielt. Das Abspielen wird durch einen „Play-Button“ verwaltet, welcher die Musik auch pausieren kann.

In der rechten oberen Ecke befindet sich eine Check-Box, mit welcher man eine Endlosschleife an- oder ausschalten kann. Der Slider-Thumb, der die Abspielposition anzeigt, kann zudem auch angefasst und verschoben werden. Zudem kann (auch während das Stück abgespielt wird) das Instrument client-seitig geändert werden.

Funktionsweise Sounds:

Bei Pageload wird ein HowlerJS-Objekt erstellt, welches für die Wiedergabe der Töne genutzt wird. HowlerJS lädt ein Soundfile jeweils als mp3 oder webm nach, welches die 9 verschiedenen Tonhöhen aller Instrumente nacheinander abgespielt beinhaltet. Nun werden die Töne durch ein wenig Javascript-Magie in Sprites unterteilt, welche parallel zueinander abgespielt werden können.

Drückt der User nun auf den Play-Button, wird eine `window.setTimeout`-Funktion angestoßen, welche sich samt Timeout rekursiv selber aufruft, solange der Play-Button aktiv ist. Innerhalb dieser Funktion werden überprüft, welche Töne in der nächsten Grid-Spalte aktiv sind, und für diese jeweils ein Ton im aktuell ausgewählten Instrument abgespielt. Jeder Ton wird mit einem Fade ein- und ausgeleitet. Die ID's der abgespielten Töne werden gespeichert, damit alle Töne per ID aufgerufen und pausiert werden können, sobald der User das Abspielen pausiert.

BACKEND

Das Backend wurde mit einem NodeJS-Server auf Basis des Express-Framework umgesetzt. Es gibt drei Routen (zu finden in `controller/index.js`). Die `"/`-Route ist für den Login zuständig, eine GET-Request auf `/` gibt die Loginseite zurück, welche ein Formular enthält, mit dem man eine POST-Request auf die Route `/roomservice` stellen kann. Die Request enthält entweder die ID des gewünschten Raums, an welchen dann man redirected wird, oder den String `„random“`, welcher den Controller dazu veranlasst, den User in den Raum mit der letzten Änderung weiterzuleiten. Die Letzte Route `/orchestra` wird mit dem GET-Parameter `room` erreicht, und gibt die Seite mit dem Musik-Grid zurück.

Funktionsweise Multiplayer:

Die Echtzeit-Multiplayer-Logik wird über SocketIO umgesetzt, welches man sich als UDP-Stream für den Browser vorstellen kann (d.h. jede Änderung wird in Echtzeit von Client an Server und andersherum übertragen).

Beim Laden der Seite meldet sich der Client beim Server an und teilt ihm seine Raum-ID mit, der Server antwortet darauf mit einem JSON, welches den aktuellen Status des Raum wiedergibt. Der Client passt nun sein Grid dem vom Server an und tritt dem SocketIO-Raum mit der entsprechenden ID bei. Jede Änderung irgendeines Clients im Raum wird nun an den Server übertragen und von diesem an alle anderen Clients im Raum gebroadcastet, welche dann die Änderungen in ihrem Frontend-Grid übernehmen können.

QUELLEN

Bild1:

<https://www.pexels.com/photo/macbook-pro-beside-black-headphones-on-gray-table-159376/>

Bild2:

<https://www.pexels.com/photo/headphones-technology-earphones-cable-30222/>

Frameworks:

<https://howlerjs.com/>

<https://socket.io/>

<http://handlebarsjs.com/>

<http://expressjs.com/de/>

<http://lesscss.org/>

<https://nodejs.org/en/>