Rhythmus-Spiel

Jonathan Held und Julian Heinken

1. Einleitung

Für unser Projekt werden wir ein Rythmus-basiertes Spiel zum spielen im Browser erstellen. Wir werden es Rhythmus-Spiel nennen.

Wir erhoffen uns damit unsere Fähigkeiten in JavaScript und Musik-Analyse zu verbessern.

Des weiteren wollen wir unser Wissen aus der Lehrveranstaltung vertiefen, indem wir gelerntes (JS Audio API, HTML und CSS) anwenden.
Wir erhoffen uns von diesem Projekt nicht nur persönlich etwas zu lernen, sondern auch ein Spiel zu entwickeln, an dem wir und andere Spaß haben.
Um das zu erreichen werden wir das fertige Spiel vor dem Kurs vorführen und online zugänglich machen.

2. Projektziel

Mit unserem Projekt möchten wir drei Ziele erreichen.
Erstens möchten wir ein Spiel entwickeln, mit dem es möglich ist auf eine interessante und intuitive Weise mit Musik zu interagieren.
Zweitens möchten wir lernen Musik mithilfe der JavaScript WebAPI zu analysieren und Wege lernen diese Analysen sinnvoll und begreiflich darzustellen.
Zu guter letzt möchten wir lernen Websites mit HTML/CSS und JavaScript zu erstellen.

3. Anforderungsanalyse

Unsere User sollen in der Lage sein unser Spiel direkt mit dem Laden der Seite zu beginnen. Konkret bedeutet dass, das wir einige Lieder bereitstellen wollen, aus denen der User wählen kann. Alternativ können User ihre eigenen Lieder hochladen und spielen.
Wenn der User ein Lied ausgesucht hat wird es abgespielt und je nach gespielten Tönen werden einige Knöpfe oder Tastatur-Inputs zeitweise freigeschaltet. Wenn der User diese Knöpfe zur richtigen Zeit drückt erhält er Punkte.
Wenn das gesamte Lied abgespielt wurde werden die Punkte angezeigt und der User kann ein neues Lied auswählen/hochladen, das er Spielen möchte.
Wichtig ist hierbei, dass die Lieder auf eine Intuitiv richtige Weise analysiert und zerlegt werden, sodass der User in der Lage ist die richtigen Knöpfe bereits am Ton zu erahnen.

Zusätzlich werden Balken von oben nach unten durch den Bildschirm fallen und am unteren Ende ankommen, wenn der User einen Input tätigen soll.
Hierbei ist es nötig dem User klar verständlich zu machen, wann ein Input erwartet wird und welcher erwartet wird.
Wenn der User eine Taste drückt muss klar erkennbar sein, ob dieser Input korrekt, oder falsch war.

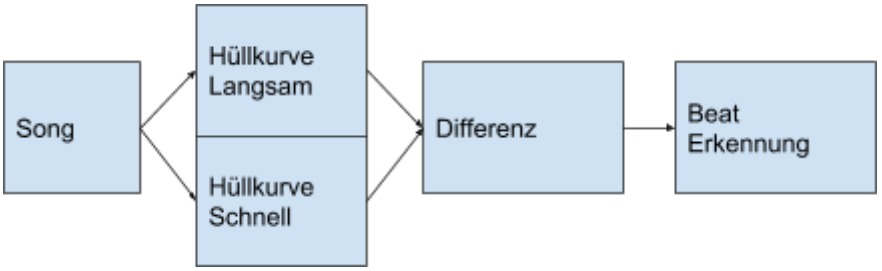
4. Technische Rahmenbedingungen

Wir wollen unser Projekt als Website mit Fokus auf Desktop entwickeln. Somit decken wir nahezu alle möglichen Geräte ab. Sofern es zeitlich möglich ist kann auch eine Optimierung für Mobilgeräte zusätzlich relativ problemlos nachgereicht werden. Zur Handhabung des Audiosignals wollen wir die Web Audio API verwenden. Diese bietet alle Funktionalität, die wir benötigen um das Audio Signal zu zerlegen. Speziell die Analyser-Node (https://developer.mozilla.org/en-US/docs/Web/API/AnalyserNode).
Durch die Möglichkeit auf einzelne Samples eines Signals zugreifen können, sind wir - sollte es nötig sein - in der Lage, bestimmte Algorithmen „manuell“ zu implementieren. In solch einem Fall könnte jedoch die eher schwache Performance von JavaScript hinderlich sein.

5. Technisches Konzept

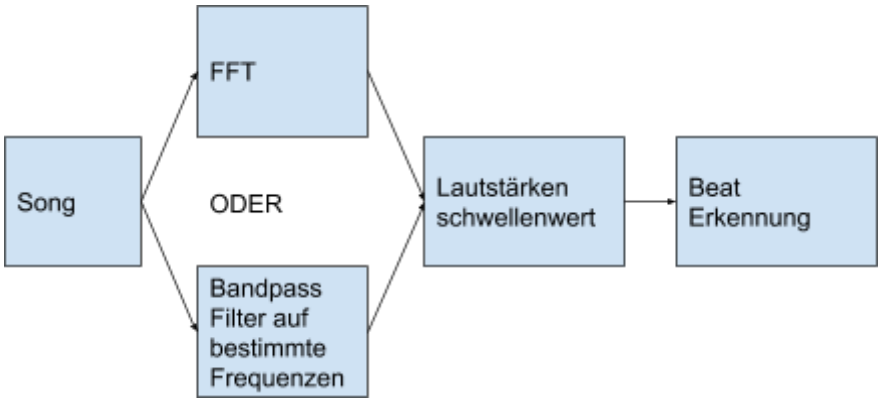
Es bieten sich folgende Methoden an:

Die Erkennung der Transienten passiert anhand der Lautstärke. Dabei versuchen sich zwei Werte der Lautstärke anzupassen, eine schnell und eine träge. Wenn die Differenz beider



Werte groß genug ist, haben wir einen Transienten. Dies ist u.a. die funktionsweise des ersten Transienten-Designers von SPL. Jene Hüllkurven können mit zwei Kompressoren realisiert werden, bei dem nur die Gain-Reduction abgefangen wird.

Erkennung über FFT:



Eine andere Methode ist es die einzelnen Bänder einer Fast-Fourier-Transformation herauszupicken. Da wir jedoch eine geringe Anzahl an Ausgaben anpeilen (~4), die FFT aber deutlich mehr braucht um überhaupt zu funktionieren, fassen wir einfach mehrere Bänder zusammen. Alternativ könnte man jedoch auch mehrere Band-Pass Filter benutzen, was vielleicht sogar effizienter ist.

Jedes dieser Bänder verfügt über einen eigenen Schwellenwert. Wird dieser überschritten, wird der Bereich als aktiv angesehen.

Ein Problem wird die unterschiedliche Lautstärke und Dynamik verschiedener Lieder sein. Hier könnte ein langsamer Kompressor Abhilfe schaffen. Alternativ könnten aber auch die Schwellenwerte an die Lautstärke angepasst werden.

6. Bedienkonzept

Wenn die Seite geladen wird startet sie im Hauptmenü.
In diesem besteht die Möglichkeit mithilfe eines Dropdown-Menüs einen Song auszuwählen.
Alternativ wird der User in der Lage sein einen eigenen Song hochzuladen.
Weiterhin wird eine Anleitung abrufbar sein.
Wenn der User auf diese drückt wird eine Anleitung angezeigt und der User kann wieder zum Hauptmenü zurückkehren.
Wenn der User einen Song ausgewählt oder hochgeladen hat kann ein Start-Button im Hauptmenü gedrückt werden.
Wenn der User sowohl einen Song hochgeladen, als auch ausgewählt hat wird der hochgeladene Song verwendet.
Wenn der User den Start-Button drückt beginnt ein kurzer Count-Down, währenddessen die ersten Blöcke vom oberen Bildschirmrand fallen können.
Nach dem Countdown beginnt das Lied und die ersten Blöcke können am Unteren Bildschirmrand angekommen sein.
Hierbei wird der Countdown gleich mit der Verzögerung des Liedes sein.
Wenn der User einen Knopf drücken soll wird als zusätzlicher Hinweis der Hintergrund aufgehellt.

Wenn der User einen Knopf zur richtigen Zeit drückt gibt es einen Effekt (der noch festgelegt werden muss).
Sollte der User einen Knopf zur falschen Zeit drücken wird es einen visuellen Effekt (der noch festgelegt werden muss) und eventuell einen Audio-Effekt geben.
Nachdem der Song komplett abgespielt wurde wird dem User die erreichte Punktzahl angezeigt und der User kann den Song nochmal starten oder zum Hauptmenü zurückkehren.

7. Zeitplan

Prototyp (27.11.2018):
Erste Visualisierung ausgewählter Songs
Proof-of-Concept Audioerkennung

Präsentation (18.12.2018):
Dynamische Analyse bereitgestellter Songs
Pre-Finale Darstellung
MVP-Hauptmenü
Finaler Player-Input

Abgabe (13.01.2018):
Finales Hauptmenü
Finale Darstellung
Upload-Funktion
Code-Clean-Up
(Mobile-Ansicht)
(Dynamische Darstellung)

8.Teamplanung

Jonathan Held: Front-End, Implementierung der Oberfläche, Inputerkennung

Julian Heinken: Back-End, Beat-Erkennung¶