MoodMusic

## Opzet

Het programma bestaat uit een website waarop een gebruiker een stemming (mood) kan selecteren. Er worden dan nummers met die bepaalde mood gedraaid.

De applicatie geeft gebruikers ook de mogelijkheid om een stemming te koppelen aan nummers in zijn of haar muziekbibliotheek. De muziekbestanden kunnen naar de interface worden gesleept (drag and drop) en worden dan geanalyseerd door EchoNest, die een lijst met kenmerken teruggeeft (BPM, toonhoogte, etc). Deze kenmerken worden door een neural network gehaald die aan elk nummer een bepaalde mood toekent. De mood en een referentie naar het nummer worden opgeslagen in de database van het programma.

## Requirements

* Gebruikers kunnen selecteren naar welke mood ze willen luisteren.
* Gebruikers kunnen audiobestanden toevoegen aan de database.
* Audiobestanden kunnen worden afgespeeld.

## Wensen

* Social media integratie (wanneer heeft iemand naar welke mood muziek geluisterd)

## Ontwerp en Implementatie

Het ontwerp is web-based. De website zelf zal gebruik maken van HTML5 en javascript. Het backend zal geschreven worden met PHP. De javascript library die gebruikt wordt voor het neurale netwerk is hier te vinden: <http://harthur.github.com/brain/>

### Mood Selector

Op deze pagina kan een gebruiker een mood kiezen om naar te luisteren. Na het kiezen van een mood zal muziek met deze mood worden afgespeeld. Het backend van deze pagina haalt met behulp van php de moods uit de database. Via javascript wordt er dan steeds een liedje opgehaald uit de database (inclusief spotify/youtube link) als de gebruiker een mood selecteert. Een liedje wordt dan ge-embed en afgespeeld.

### Library Analyzer

Een gebruiker kan ook zijn eigen muziekbibliotheek laten analyseren. De gebruiker sleept dan de bestanden die geanalyseerd moeten worden in de webpagina. Vervolgens worden de bestanden geupload naar EchoNest en geanalyseerd. De geanalyseerde resultaten worden door ons neuraal network gehaald en de mood die daaruit komt wordt, samen met wat andere informatie, opgeslagen in de database. In het vervolg kan de zojuist geanalyseerde muziek dus ook beluisterd worden in de Mood Selector.

### Training Tool

Het neurale network moet getraind worden met voorbeelden: een trainingset. Deze trainingset gaan we zelf maken door voor een aantal nummers handmatig de mood ervan aan te geven. Hiervoor is een pagina gemaakt waar steeds een liedje van youtube wordt ge-embed, waarvoor dan een mood aangegeven kan worden door middel van het klikken op buttons met moods erop. Uit deze resultaten wordt een training- en een test set gehaald, met de trainingset wordt het network getrained, en met de testset worden de resultaten getest. Deze tool wordt in principe alleen door ons gebruikt en niet door de eindgebruiker.

### Network Trainer

De Network Trainer traint het network met de waarden die uit de Training Tool zijn gekomen. De training waarden worden uit de database gehaald. Na het trainen wordt er een JSON-representatie gemaakt van het netwerk en opgeslagen in een tekst bestandje. Dat bestandje wordt dan bij de Mood Selector weer ingeladen. Deze tool wordt in principe ook alleen door ons gebruikt en niet door de eindgebruiker.

## Planning

In de eerste sprint is er een analyse gemaakt van welke databronnen er beschikbaar zijn en hoe deze te gebruiken zijn om een nieuwe applicatie te maken. Er is een start gemaakt met de implementatie en het framework voor de applicatie staat.

### Sprint 2

In sprint twee is er een neuraal network uitgekozen om mee te werken. Het is een gebruiksvriendelijke javascript implementatie die met eenvoudige functies getrained kan worden. Er is een database opgezet met daarin de spotify id’s en EchoNest id’s geanalyseerde nummers en de bijbehorende mood. De front-end sites zijn zo goed als af: een gebruiker kan nu zijn muziekbibliotheek in de applicatie slepen en de nummers worden dan geanalyseerd door EchoNest.

### Sprint 3

De afzonderlijke onderdelen zijn af en functioneren. Het merendeel van de componenten is samengevoegd.

### Sprint 4

In de laatste sprint worden de puntjes op de i gezet. De laatste eindjes moeten aan elkaar geknoopt worden, en alle functionaliteit moet worden gebundeld in een afleverbaar geheel.

Er zijn nog een aantal dingen waar we tegenaan lopen: er staan zoveel liedjes in EchoNest dat het slechts sporadisch voorkomt dat de gebruiker een liedje krijgt dat daadwerkelijk bekend is. Er zou eigenlijk een integratie met een soort last.fm service moeten komen zodat er prioriteit wordt gegeven aan recente/populaire nummers.

In deze sprint gaan we ook veel tweaken aan het neurale netwerk om ervoor te zorgen dat de resultaten accuraat zijn. Daarvoor willen we een test set voor gebruiken en dmv een crossvalidation een error vaststellen. Als die error acceptabel is wordt het mogelijk om meer algemene uitspraken te doen als bijvoorbeeld: is de stijl van een band/muziekant (die gegeven wordt door EchoNest) hetzelfde als de stijl van de nummers die ze maken, en klopt deze waarde dus?