Projectomschrijving CSD2d

Door: Harold Groenenboom

# Green AudioSearcher

In dit verslag beschrijf ik het concept van mijn project, het einddoel en de toekomst van dit project en ik beschrijf wat ik aan het eind van dit project lever. Ik zal langslopen hoe ik te werk ben gegaan, hoe ik heb ontworpen en hoe het resultaat er uiteindelijk uit ziet. Vervolgens zal ik kort reflecteren over mijn bezigheden.

## Inhoud

### Concept

### Ontwerp

### Planning

### Systeemontwerp

### Resultaat

### Conclusie

## Concept

Als componist of producer ervaar je vaak dat je een sample vindt en dat die net niet is wat je wilt. Toch is de toonhoogte precies perfect. Of juist de timing. Of de klankkleur. Dit is iets wat ik zelf vaak ervaar. Hiervoor probeer ik in dit project een tool te maken die de gebruiker met veel gemak naar vergelijkbare samples laat zoeken. De *Green AudioSearcher* is een audioapplicatie waarin de gebruiker audiosamples kan opzoeken die op een aangegeven audiosample lijkt. Hierbij staat user centered design centraal. Het is namelijk belangrijk dat deze tool een toegevoegde waarde kan hebben voor het creatieve proces, en dat deze tool het proces in ieder geval niet verstoord. Hierom heb ik allereerst nagedacht wat ik wil en heb ik op papier een design uitgetekend.

Voor de Green Audiosearcher heb dus ik allereerst gewerkt aan een volledig design. Zo wil ik van een volledig design terugwerken naar een prototype design waarin ik kan uittesten hoe de workflow bij de gebruiker aanslaat, in dit geval ik. Hierom focus ik in mijn prototype op de essentiële elementen om mijn concept uit te voeren en probeer ik bewust zo min mogelijk extra’s toe te voegen. Ik als componist ben de voornaamste doelgroep, omdat ik dit zelf als tool wil hebben, maar ik wil het ook zo designen dat het ook voor andere componisten en producers in ieder geval logisch werkt. Zo kan ik uiteindelijk een iteratie ontwikkelen die bij de buitenwereld aanslaat.

Een bestaande applicatie die dit doet is Sononym. Deze heb ik kort uitgeprobeerd en de rede dat ik dit nog steeds wil ontwikkelen is omdat in Sononym de gebruikte technieken wel workflow verbeterend vindt maar het design niet zo. Omdat het een kleine stap is in het compositieproces, denk ik dat je als gebruiker niet te veel opties en niet te veel visuele informatie wilt. Ook denk ik dat je zo een dergelijke tool eigenlijk alleen wilt gebruiken als een sample net niet is wat je zoekt. Voor mij is dus de zoektocht om een goede interactie te ontwikkelen voor een creatief bemiddelend product.

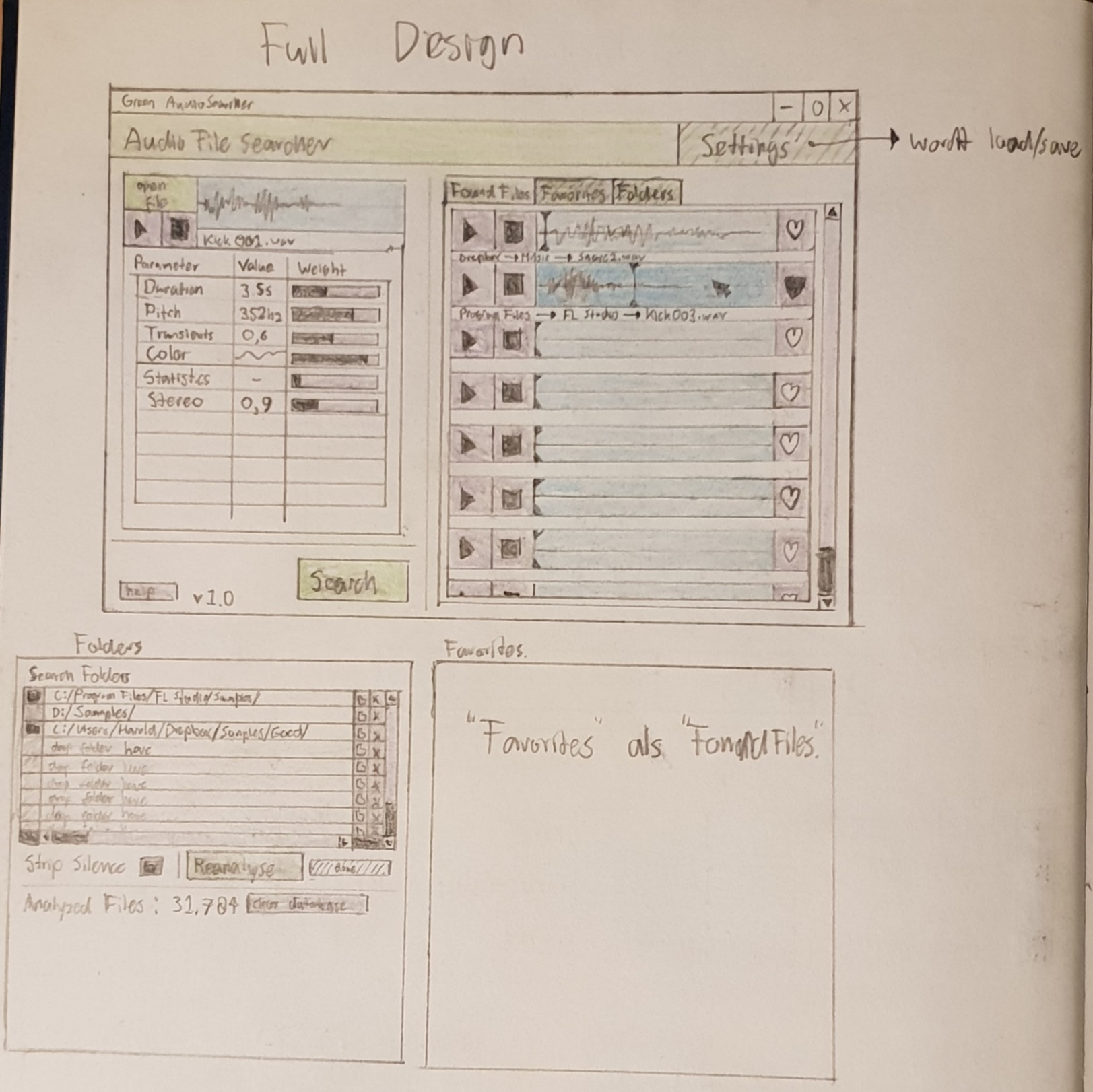
## Ontwerp

### Volledig ontwerp

Het volledige ontwerp heb ik allereerst op papier uitgewerkt. Ik ben gaan nadenken over een workflow die mij handig lijkt en over technieken waarmee samples kan zoeken die muziekaal gezien verwant zijn. Op de volgende bladzijde is het volledige design te vinden. Er is direct een belangrijke onderverdeling te onderscheiden. Zo bevindt zich links de input audiofile en rechts de gevonden vergelijkbare audiofiles. Linksonderin is de search button. De gebruiker werkt zo van linksboven naar linksonder, en vervolgens van rechtsboven naar rechtsonder. Ook heb ik de window expres klein gelaten, zodat het niet snel voor de gebruiker in een DAW in de weg zal zitten. Ik zal een paar belangrijke keuzes in het design langslopen.

Sum of Least Squares method en weights. (Euclidean Distance???)

Om de audiofiles te kunnen vergelijken analyseer ik eerst elke audiofile op verschillende parameters. Zoals piekfrequenties, lengte, dynamiek en klankkleur. Deze parameters normalizeer ik naar een range van 0-1 met een normalizatiefunctie, bijvoorbeeld de *Sigmoid function*. Het programma krijgt een lijst van mappen om te analyseren en creëert een database van geanalyseerde audiofile objecten. Om de gebruiker een specifiek zeggenschap te laten hebben op de analyseëigenschappen van de audiofile die hij of zij belangrijk acht, kan de gebruiker met weights aangeven welke waardes meer in de *similarity* vergelijking worden meegenomen. Zo wordt de uiteindelijke *similarity* tussen twee samples s en s2 met de *Least Squares Method* als volgt berekend:



Kleurgebruik

Om de interactie zo duidelijk mogelijk te maken onderzoek ik de waarde van kleurgebruik. Zo geef ik de belangrijkste knoppen de kleur lichtgroen. Deze knoppen worden door de gebruiker als het goed is het meest gebruikt. De minder belangrijke knoppen, waar de gebruiker niet per se aan hoeft te komen om het programma te gebruiken, geef ik de kleur paars. Audio geef ik de kleur blauw. Zo weet de gebruiker altijd meteen waar audiofiles zijn. Een andere belangrijke reden hiervan is dat ik alle Audio, weergegeven als blauwe waveforms, een drag-and-drop functie wil geven. Zo wordt de workflow voor de gebruiker optimaal. Elk gekleurd veld wordt ook subtiel gehighlight als de muis zich boven het veld bevindt om de gebruiker te laten weten dat er een interactie mogelijk is.

Workflow:

1. Selecteer een sample uit de DAW.
2. Sleep de sample naar de waveform van de AudioFileAnalyser en wacht op voltooien van analyse.
3. Adjust de weights naar behoefte
4. Druk op search en wacht tot de search compleet is.
5. Luister naar de audiofiles en sleep de gewenste audiofile naar de DAW of weer opnieuw in de waveform van de AudioAnalyser.

Favorites

Een gebruiker kan een gevonden audiofile als favorite markeren. Zo kan de gebruiker uit gevonden samples zijn of haar favorieten selecteren. Hiermee kan de gebruiker vrijuit spelen met het kiezen van nieuwe input audio files, zonder bang te zijn eerder gevonden audiofiles die zijn bevallen kwijt te raken.

Folders

De gebruiker kan in de folders map aangeven welke folders in de search geinclude moeten worden. Zodra de gebruiker deze folders aan heeft gegeven, moeten een keer alle folders geanalyseerd worden en vervolgens nooit meer. De gebruiker krijgt hierbij de controle om zelf het analyse proces te starten, zodat het voor de gebruiker voelt alsof hij of zij volledige controle heeft over het programma. Ik vind het zelf namelijk storend als een programma een proces start waar ik zelf controle over wil hebben. Deze soort interactie waarbij de gebruiker meer doet wil ik hier dus uitproberen.

Save / Load

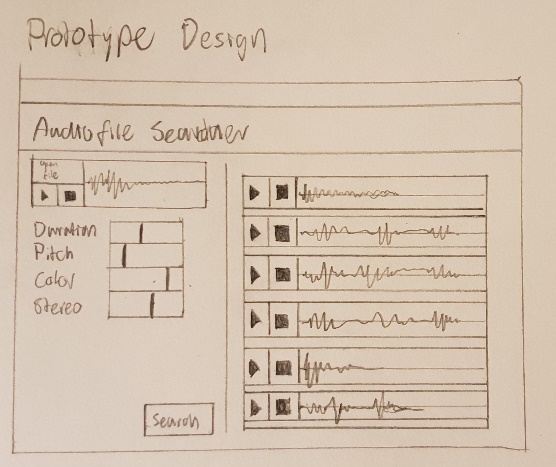
Tot slot kan de gebruiker settings laden en saven. Zo opent het programma automatisch de laatst geopende setting. De preset bevat de folderpaths, de favorites en de database. Zo kan de gebruiker makkelijk wisselen tussen verschillende ‘zoekopdrachten’ op een intuïtieve wijze.

Samengevat

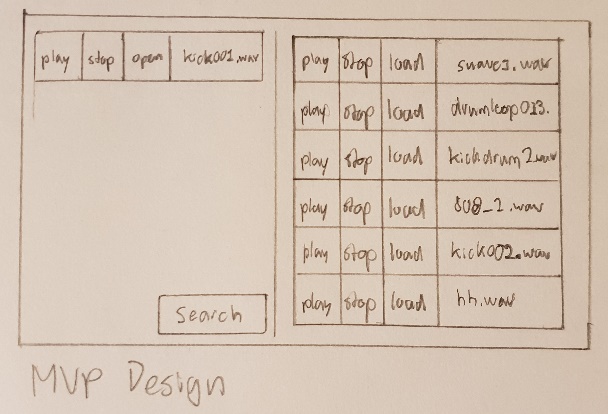
Het volledige ontwerp bevat dus al aardig wat opties. Een aantal opties hiervan zijn gebaseerd op aannames en het ontwerp is volledig op papier gedaan aan de hand van mijn smaak. De allereerste aanname, namelijk of de zoekmethode de gebruiker helpt, zal ik uittesten in dit project. Hiervoor heb ik geanalyseerd wat essentieel is uit het volledige concept en dit heb ik gebruikt om mijn prototype concept te vormen. Door het volledige concept achter de hand te hebben, hou ik in mijn systeemontwerp rekening met te verwachten vormen van uitbreiding.

### Prototype Concept

In het prototype in dit project wil ik een werkende audioapplicatie bouwen die een audiofile analyseert en een reeks vergelijkbare audiofiles teruggeeft. Hiervoor kan de gebruiker voor nu één folder met subfolders instellen. Met het prototype wil ik het kleurgebruik en de workflow onderzoeken. Zo hoop ik na het prototype op een zinnige manier het volledige concept te kunnen uitwerken.

Hiervoor heb ik een prototype ontwerp gemaakt. De layout is hierbij bewust hetzelfde. Zo werk je nog steeds van linksboven naar linksonder en vervolgens naar rechtsboven naar rechtsonder. Alle interacties hier zijn interacties die iedere DAW gebruiker al kent. In dit getekende ontwerp mist alleen de audiofile folder selectie.

## Planning

Om het project te structureren heb ik voor de zekerheid ook een MVP design gemaakt. Vanuit dit design wordt heel duidelijk wat ik precies wil gaan proberen. Zo is het echt duidelijk dat het vooral gaat om kleurgebruik, gebruikersgemak en workflow. In dit getekende ontwerp mist net als bij de prototype design ook de audiofile folder selectie.

Om goed te werk te kunnen gaan heb ik als eerst een MoSCoW opgesteld.

### MoSCoW

Must (MVP)

* Audio analyse
  + Duration
  + RMS
* Audiofile players
  + Drag and drop
  + Play, pause en stop
* Search button
* Folder selection
* Euclidean distance
* Drag n drop

Should

* Audio analyse
  + Pitch
  + Timbre
* Parameterset database
* Kleurgebruik
* Weights in GUI en euclidean distance

Coulds

* Waveform viewer
* Play-pause en stop knop ontwerp

Woulds

* Toevoegen folder tab met:
  + Meerdere folders
  + Status bar voor analyse
* Toevoegen van een viewport aan ‘Found Files’ om zo meerdere Found Files aan de user aan te bieden