



Quinten Bast
2 april 2019
Interaction Design 3

Seeing Music

Voor Interaction Design 3 hebben we ons gefocust op 'data visualisatie'. Ik zelf ging in de eerste week opzoek naar data voor de basis van mijn concept. Ik kwam op 3 verschillende concepten, een concept die gebruik maakt van de Here Traffic data, met deze data kan je bijvoorbeeld zien waar er opstoppen zijn en hoe hard mensen rijden. Als tweede heb ik gekeken naar hoe ik AR kan gebruiken om radio golven te visualiseren want er zijn een heleboel aspecten in onze wereld die we niet geloven of kunnen inbeelden, simpelweg omdat we ze niet met het blote oog kunnen zien.

Als laatste heb ik gekeken naar de API van Spotify waarmee je onder andere grondige analyses van muziek kan maken. Voor deze laatste heb ik uiteindelijk gekozen omdat ik erg visueel ingesteld ben en ik daardoor heel erg enthousiast werd over de gedachten om muziek om te gaan zetten in een esthetisch "mooi kunstwerk", een "vingerafdruk" van een nummer. Een beeld dat de emotionele laag van een nummer zou kunnen visualiseren. Het bedenken en grafische uitwerken was mijn eerste stap maar het daarna omzetten naar een soepel programma die de visualisatie zal afbeelden was een veel lastigere stap waardoor ik soms compromissen moest sluiten.



Iteratie 1

Seeing sound^{v1.0}

Music fingerprint

Seeing Sound was mijn eerste versie voor de Mid-term, toen heb ik onderzocht wat voor bruikbare data ik uit de spotify API kon halen en wat voor visualisatie ik daar aan kon koppelen. Ik heb toen onderzoek gedaan naar synesthesia, een extra "zintuig" die ervoor zorgt dat je kleur ziet bij geluid. Daarnaast heb ik ook nog onderzoek gedaan naar kleuren en welke associaties mensen bij deze kleuren hebben. De uitkomst daarvan is te zien en te lezen op de volgende pagina.

Basis information

Duration in ms

Name of track

Artist name

Popularity

Tempo (bpm)/ Energy - intensity activity

Loudness (dB)


Length of the line

In the background - custom font?

In the background - custom font?

Opacity, 100 = 100 % opacity | 0 = 0% opacity

Hartslag 0.0   1.0  

Physical strength 0.0  1.0 


In depth info

Danceability

Valence (happy - sad)

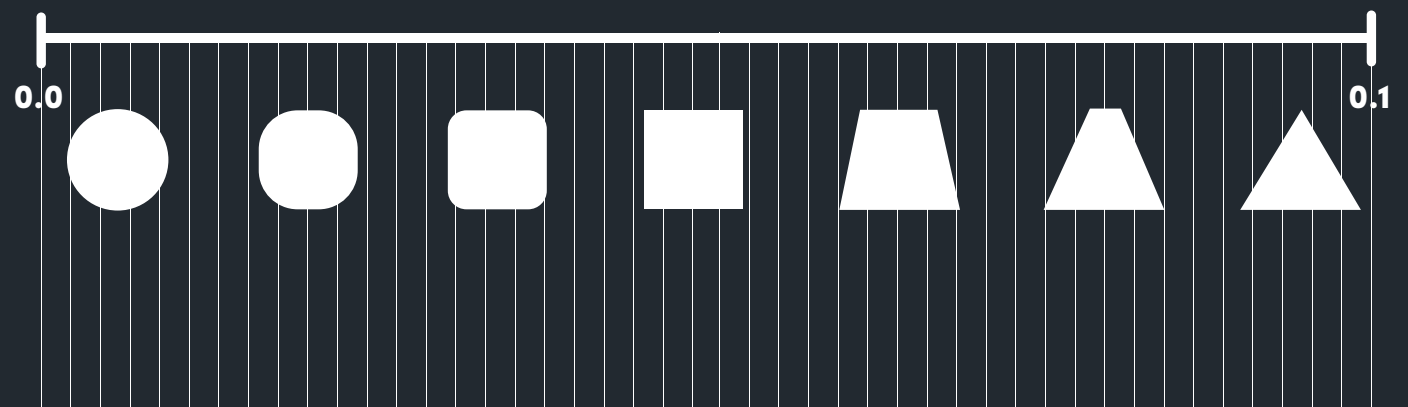
Acousticness- Instrumentalness - speechiness

Liveness (live performance)

0.01  .0  

Color panel

Energy



Blue

Tranquility, love, loyalty,
security, trust, intelligence

Coldness, fear, masculinity

Red

Love, energy, power, strength,
passion, heat

anger, danger warming

Brown

Friendly, earth, outdoors,
longevity, conservative

Dogmatic, conservative

Purple

royalty, nobility, spirituality,
luxury, ambition

mystery, moodiness

Green

money, growth, fertility,
freshness, healing

envy, jealousy, guilt

Pink

healthy, happy, feminine,
compassion, sweet, playful

weak, feminiity, immuaturity

Tan/Beige

Dependable, flexible, crisp

dull, boring, conservative

Turquoise

spiritual, healing, protection,
sophisticated

envy, femininity

Orange

courage, confidence, friendli-
ness, succes

ignorance, sluggishness

Yellow

bright, energy, sun, creativity,
intelect, happy

irresponsible, unstable

Black

protection, dramatic, classy,
formality

death, evil, mystery

Gray

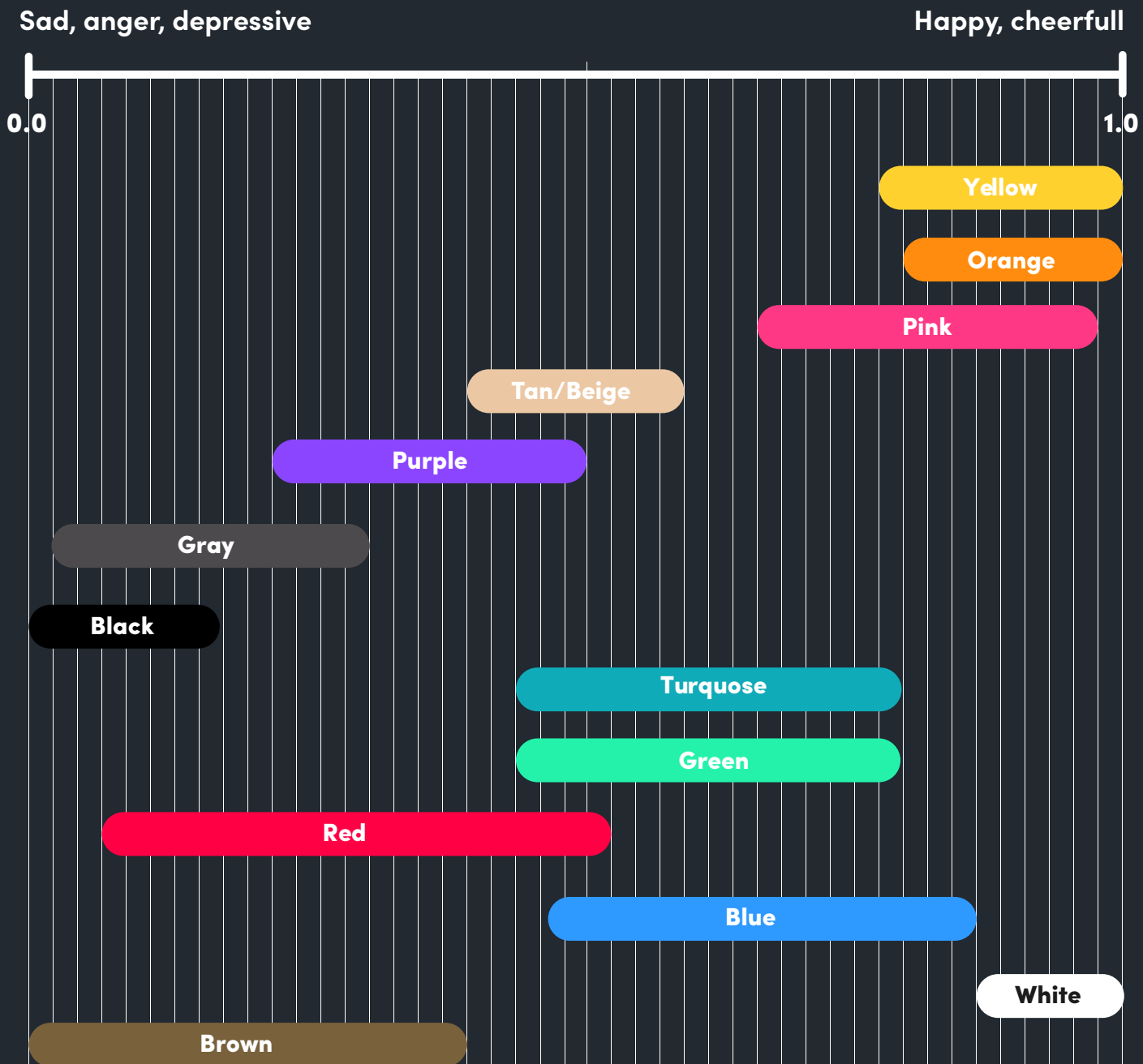
security, reliability, inteligen-
ce, solid

gloomy, sad, consevative

White

goodness, innocense, purity,
fresh, easy, clean.

Valence



Dit is hoe ik mijn visualisatie na onderzoek te hebben gedaan, in mijn hoofd had. Lijnen die gevormd werden door de dancebility en kleuren die gebaseerd waren op valance. Het moest een vingerafdruk van de muziek representeren. Maar toen ik aan de slag ging met de technische uitwerkingen, stuitte ik op een aantal uitdagingen.



Iteratie 2

Seeing Sound wordt syn

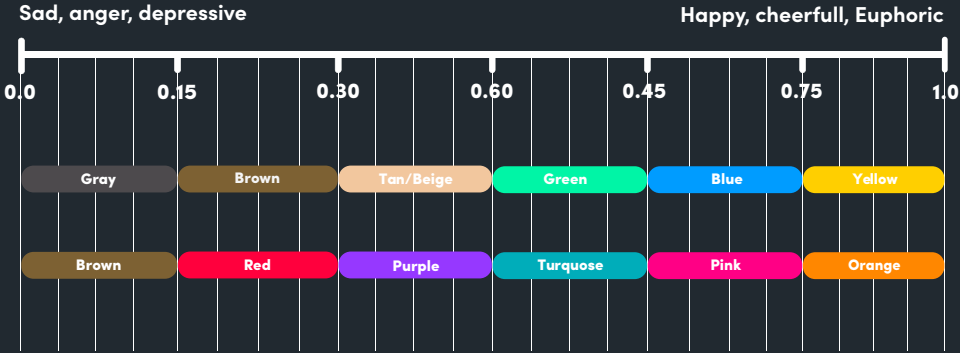
Uitendelijk moest het natuurlijk ook uitgewerkt worden, ik had zelf al een beeld hoe het er uit moest gaan zien. Maar door de technische kant werd ik uiteindelijk beperkt in mijn mogelijkheden. Ik heb ervoor gekozen om een iPad app te maken met Swift in Xcode omdat ik op dit moment deze programmeer taal aan het leren ben. Ik heb er toen voor gekozen om 4 verschillende parameters te gebruiken van de Spotify API namelijk: valance, danceblity, tempo en Energy. Valance gaat over wat de stemming van een nummer is van 0,0 zielig tot 1.0 blij. Danceablity is een maatstaaf voor hoe dansbaar een

nummer is. Tempo (bpm) bepaald de snelheid van een nummer en nummers met veel Energy zijn vaak snel, luid en lawaaiërig. Zo heeft, Death metal een hoge energie, terwijl een Bach-prelude laag scoort op de schaal.

Ik heb gekozen voor 6 strepen, dit visualiseert een muziek lijn. Uitendelijk wil ik dat het meer een beeld heeft van een vingerafdruk omdat naar mijn mening-muziek niet iets is dat je kan afbeelden als iets dat zo statisch is als een rechte lijn.



Valence



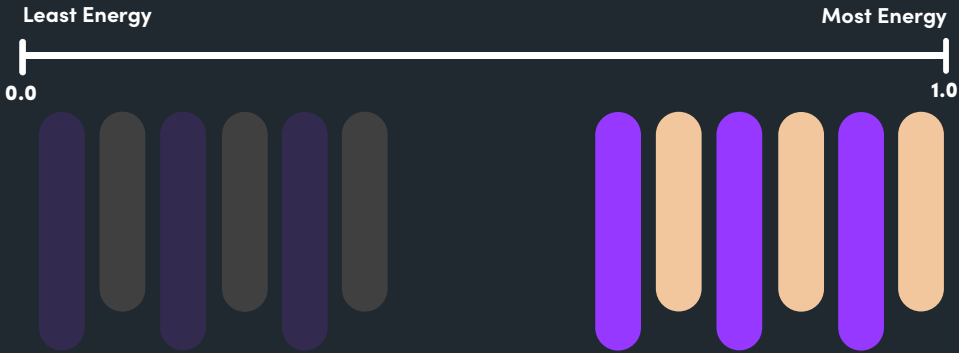
Danceability



Tempo



Energy



Launching application



Look up song ID



**Visualization
appears**

Enter Song ID



Press the button

Voorbeelden van het eindresultaat

Als het avond is - Suzan en Freek



7 Rings - Ariana Grande



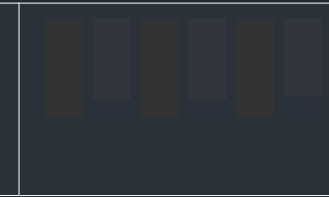
Heartbeat - Dastic



Stay with me - Sam Smith



Light a candle - Nemsis



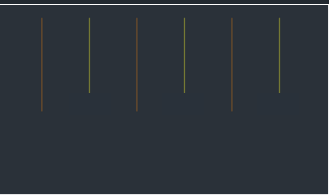
Repentless - Slayer



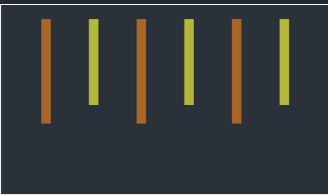
Happy - Pharrell Williams



Another one Bites the dust - Queen



Rythem of the night - De Barge



I got love - Don Diablo, Natt Dog



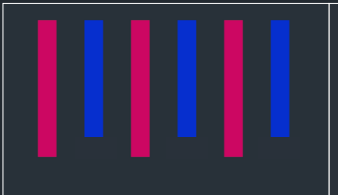
Let me down slowly - Alec Benjamin



Ik leef niet meer voor jou - Marco Borsato



Don't stop me know - Queen



High Hopes - Panic! At the disco



I'm So Excited - The pointer sisters

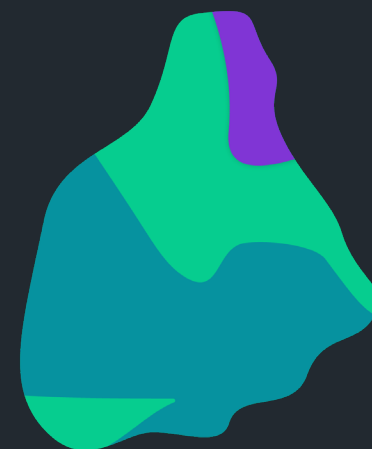


Mogelijke visualisaties

Syn, is een project dat nog lang niet klaar is en schreewt om meer aandacht. Ik heb in dit project op technisch gebied veel kunnen leren van het gebruik van JSON data in een iOS app tot visualiseren van die JSON data in Swift. Ik zie deze 6 - 8 weken die ik er aan heb gewerkt dan ook als een eerste stap en ik heb nu verschillende ideeën hoe ik dit project kan aanpakken. Van een aura van je persoonlijkheid op basis van de muziek waar je naar luistert of een cover art van je favoriete nummer op basis van de muziekanalyse van Spotify. Maar ik heb ook gedacht aan een

visualisatie van gesproken tekst, zodat je gemakkelijk in een oog opslag de sfeer van het gesprek kunt zien. Daarnaast zou je het ook kunnen gebruiken voor een live performance die de gezongen tekst direct visualiseert op een groot scherm achter de zanger, zangers of dj.

Er zijn heel veel use cases te bedenken dus wie weet komt er nog een interessante uitwerking uit deze eerste stap.



De technische documentatie

voor Edwin

Voor mijn project 'syn' heb ik gebruikt gemaakt van de Spotify API. Mijn eerste stap was om de API zo werkend te krijgen dat ik JSON data van elk nummer dat op Spotify staat kon binnenkrijgen. Dit was nog lastiger dan gedacht omdat Spotify verschillende beveiligingen in heeft gebouwd waarbij je moet verifiëren dat jij de data gebruikt. Hierdoor kwam ik er niet in mijn eentje uit daarom heb ik hulp gevraagd aan een mede interaction designer (Max), die mij heeft geholpen met het schrijven van een php bestand die de data van Spotify kan halen.

Toen ik de data van Spotify kon opvragen, ging ik aan de slag met het maken van de applicatie die de JSON data om gaat zetten naar een visualisatie. Ik heb gekozen om aan de slag te gaan met Swift omdat ik deze programmeer taal op dit moment ook aan het leren ben.

Mijn eerste prototype was een versie waar ik wilde dat ik de JSON data van mijn server kon uitlezen in mijn Swift project, zodat ik het daar kon gaan gebruiken. Ter visualisatie heb ik er toen voor gekozen om alleen 1 kleur te gebruiken zodat ik alleen nog hoeft te focussen op het maken

van een programma die de JSON data kan uitlezen. Ook moest er een invulveld komen zodat je in de app de Song ID in kan vullen. In twee dagen kreeg ik dit uiteindelijk werkend en had ik een canvas waar ik op kon gaan werken. Toen kwam de uitdaging om met de vier componenten (Valance, Danceblity, Tempo en Energy) die ik uit de Spotify data ging gebruiken. Er een esthetisch mooie visualisatie van te maken maar die ook een patroon in zich heeft zodat het niet "willekeurig" aan gaat voelen. Het was vooral lastig omdat ik nog veel moest opzoeken waardoor

ik pas na 2 - 3 dagen een interessante visualisatie had met een patroon, maar het was een erg plat en relatief saai beeld. Daarna kreeg ik een beter beeld hoe ik in Swift code een "interessantere" visualisatie kreeg, dat meer op een vingerafdruk zal gaan lijken. Maar om dat te bereiken betekende dat dat ik mijn code eigenlijk helemaal opnieuw moest gaan opbouwen omdat de basis waar de visualisatie "getekend" werd niet goed aanpasbaar was. Door de tijdsdruk heb ik er daarom voor gekozen om niet alles om te gooien en het gene wat ik had te gaan finetunen. En er een prettige user interface bij te ontwerpen.

Dit project was een goede leerweg voor mij omdat ik helemaal nog niet bekend was met JSON. Hierdoor heb ik veel kunnen bijleren op technisch gebied en zie ik nieuwe mogelijkheden voor projecten in de toekomst.

