Visualization of Quantified Self data from Spotify using avatars

Author Stefan Aleksikj
Supervisor Aris Alissandrakis
Exam date 30 August 2018
Subject Social Media and Web Technologies
Level Master
Course code 5ME11E-VT18
## Contents

1 Introduction  
  1.1 Background ............................................. 1  
  1.2 Motivation ............................................. 3  
  1.3 Research questions ................................. 6  

2 Related work  
  2.1 Visual perception of facial expressions .............. 8  
  2.2 Use of avatars in data visualization ................. 10  
  2.3 Quantified Self Spotify projects .................. 11  

3 Methodology  
  3.1 Prototyping ........................................... 13  
    3.1.1 Exploration of various avatar implementations 13  
    3.1.2 Development of the avatar mockup ............... 14  
    3.1.3 Concept design questionnaire .................. 14  
  3.2 Data exploration ..................................... 14  
  3.3 User study ............................................ 15  
    3.3.1 Data collection ................................. 15  
    3.3.2 Post-study interviews ........................ 20  
  3.4 Limitations ........................................... 20  
    3.4.1 Finding participants ............................ 20  
    3.4.2 Spotify data limitations ..................... 20  

4 Prototype development  
  4.1 Design of the prototype ............................. 22  
    4.1.1 Design principles ............................... 22  
    4.1.2 Graphic design of the avatar .................. 23  
    4.1.3 Final avatar design ............................ 24  
  4.2 Mapping data on the avatar .......................... 26  
    4.2.1 First prototype ................................ 27  
    4.2.2 Second prototype ............................... 28  
  4.3 Data calculation ..................................... 30  
  4.4 Developing the avatar ................................ 32  
  4.5 musicAvatar implementation ........................ 34
List of Figures

4.1 musicAvatar inspiration ........................................ 23
4.2 Refined mockups ............................................... 25
4.3 Final prototype design ......................................... 25
4.4 First prototype examples ....................................... 28
4.5 Second prototype examples .................................... 29
4.6 Avatar pixel grid ................................................ 33
4.7 Login steps ...................................................... 35
4.8 First implementation UI ........................................ 36
4.9 First implementation UI examples ............................. 37
4.10 myDay musicAvatar example ................................. 38
4.11 Second implementation UI ..................................... 38
4.12 Second prototype tooltip ...................................... 39
4.13 Basic MongoDB Schema ....................................... 40
4.14 System schema .................................................. 41

5.1 Question samples from the initial survey ..................... 45
5.2 Survey results chart ............................................. 46
5.3 Number of visit chart, first implementation .................. 47
5.4 Fetch data chart, first implementation ........................ 48
5.5 Average music features value per user, first implementation 49
5.6 Average values for danceability, first implementation .... 49
5.7 Users opinion, first implementation ........................... 51
5.8 Number of visit chart, second implementation ............... 52
5.9 Average music features value per user, second implementation 53

C.1 Privacy page, first implementation. ......................... 98
C.2 Privacy page, second implementation. ...................... 99

D.1 Instruction page, first implementation. ...................... 101
D.2 Instruction page, second implementation. .................. 102
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>User logs</td>
<td>16</td>
</tr>
<tr>
<td>3.2</td>
<td>User record logs</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>User activity logs</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>Users seen data</td>
<td>19</td>
</tr>
<tr>
<td>3.5</td>
<td>Users comment log</td>
<td>19</td>
</tr>
<tr>
<td>3.6</td>
<td>Users opinion log</td>
<td>19</td>
</tr>
<tr>
<td>4.1</td>
<td>Data mapping of the avatars</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>Music feature thresholds</td>
<td>31</td>
</tr>
<tr>
<td>5.1</td>
<td>Overall user activity, first iteration</td>
<td>50</td>
</tr>
</tbody>
</table>
Abstract

The increased interest for self-tracking through the use of technology has given birth to the Quantified Self movement. The movement empowers users to gain self-knowledge from their own data. The overall idea is fairly recent and as such it provides a vast space for exploration and research. This project contributes to the Quantified self movement by proposing a concept for visualization of personal data using an avatar. The overall work finds inspiration in Chernoff faces visualization and it uses parts of the presentation method within the project design.

This thesis presents a visualization approach for Quantified Self data using avatars. It tests the proposed concept through a user study with two iterations. The manuscript holds a detailed overview of the designing process, questionnaire for the data mapping, implementation of the avatars, two user studies and the analysis of the results. The avatars are evaluated using Spotify data. The implementation offers a visualization library that can be reused outside of the scope of this thesis.

The project managed to deliver an avatar that presents personal data through the use of facial expressions. The results show that the users can understand the proposed mapping of data. Some of the users were not able to gain meaningful insights from the overall use of the avatar, but the study gives directions for further improvements of the concept.

Keywords: Quantified Self, Chernoff faces, avatars, data visualization, Spotify
1 | Introduction

1.1 Background

In a ubiquitous computing world, almost every aspect of our daily online and offline activities are logged somewhere over the Internet. This information is collected through various sensors placed in the surrounding infrastructure, smartphones, online accounts etc. The collected data potentially holds meaningful understandings about us.

Possibly, the user can master new skills and grasp personal insights by reflecting on its own data. This can lead to self-improvement in various aspects of our personal development [Mols et al., 2016]. For example, tracking exercise performance by using a wearable can provide us with a deeper understanding of our current physical condition. This can be done by examining the information which is collected from the device. With this knowledge the user gains meaningful insight that can lead to improvement of its exercise habits, food diet, general health etc. Because of this benefit and overall increased access of technology, self-tracking is becoming very popular among people who exercise regularly.

This phenomenon of using technology for self-tracking in order to gain self-knowledge is called Quantified Self (QS). The term was first introduced by Wired Magazine editors Gary Wolf and Kevin Kelly in 2007 [Quantified Self Labs, 2012]. Furthermore the authors created a movement that explores the potential of QS data through user groups, conferences, articles, online discussion etc. They focus on self-improvement, for self-discovery, self-awareness, self-knowledge through self-tracking. Wolf in one of his talks gave the following statement:

We know that new tools are changing our sense of self in the world – these tiny sensors that gather data in nature, the ubiquitous computing that allows that data to be understood and used, and of course the social networks that allow people to collaborate and contribute. But we think of these tools as pointing outward, as windows and I’d just like to invite you to think of them as also turning inward and becoming mirrors. [Wolf et al., 2010]
1.1. Background

The QS movement brings empowerment for the user. It allows us to take ownership of our data and gain added value from their analysis. The benefits mentioned by the founders of the movement are realistic and achievable. Still this concept is quite new and as such it raises a lot of questions. This thesis aims to contribute to the general discussion of the QS movement. Hopefully some of my findings can lead to better understanding of how users perceive QS visualization. This potential knowledge can ultimately improve the way we design interaction for QS apps.

This thesis aims to contribute to the movement by analyzing new ways of visualizing quantified-self data with the use of data-generated avatars. It is part of an ongoing project\(^1\) that explores creative ways to visualize QS data, aggregated from social media and Internet of Things services. The initial work has produced a thesis [Nake, 2015] and two conference papers [Nake et al., 2016; Alissandrakis and Nake, 2016]. Through its research the project aims to provide meaningful insight for the user.

The project compliments the ongoing trend for self-tracking. In recent years self-tracking apps have gained massive popularity. According to a study [Rocket Fuel, 2014] conducted by Rocket Fuel, a provider of artificial intelligence (AI) marketing and advertising solutions, 31% of consumers in U.S are currently using a QS tool to track their life habits. Furthermore, 20% of consumers know someone close who uses a this kind of tool, and 25% of consumers are interested in obtaining a QS tool for health and fitness. Nearly one in five planned to give a QS tool to someone as a gift, which further shows determination for these type of interest to track health and fitness behavior digitally. Apps such as: Habit Tracker\(^2\), Instant - Quantified Self, Track Digital Wellbeing\(^3\), Sleep as Android: Sleep cycle tracker, smart alarm\(^4\), Fabulous: Motivate Me! Meditate, Relax, Sleep\(^5\) etc., have millions of downloads all over the Internet stores. The QS movement holds more than 500 examples of apps\(^6\). They cover a broad spectrum of topics such as: personal wellbeing, exercise habits and performance, productivity, socialization etc. Most of these applications tend to provide meaningful insights for the user.

In an article [Waltz, 2012] through self reporting in period of two months, Waltz provides observations of how the consumers of today can quantify

\(^{1}\)Project details: https://lnu.se/en/research/searchresearch/forskningsprojekt/projekt-visualizing-quantified-self-data-using-avatars


\(^{4}\)Sleep as Android: Sleep cycle tracker, smart alarm: https://play.google.com/store/apps/details?id=com.urbandroid.sleep

\(^{5}\)Fabulous: Motivate Me! Meditate, Relax, Sleep: https://play.google.com/store/apps/details?id=co.thefabulous.app

\(^{6}\)QS official list: http://quantifiedself.com/guide/
1.2 Motivation

They themselves. The report provides an analysis of how she recorded and analyzed biometric data on daily basis with the use of the five most popular devices on the market. The article claims that with a proactive use of the tools the outputted data offers a numeric picture for the users health, fitness and sleeping condition. Besides consciously collecting data with various devices and wearables, everyday users are generating vast data traces through their smartphones and social media accounts. The QS movement provides tools for analysis of this type of data logs, but it is worth to note that they are not as popular as the health and fitness tracking tools.

As the trend grows so is the need of providing interactive presentation of data. A current familiar problem of the QS movement is that users often lose interest in the usage of the self tracking tools [Gouveia et al., 2015; van Berkel et al., 2015]. A consumer report [Ledger and McCaffrey, 2014] that confirms this claim, presents a study in which one third of 6223 users have dropped using self-tracking apps within a span of six months. This issue can be addressed with rethinking the design approach for QS tools. This is where the thesis finds inspiration in applying Chernoff [Chernoff, 1973] type of visualization in QS applications. Potentially this approach could add value for the user experience and maintain his/her interest. More on this idea will be elaborated in the upcoming sections.

1.2 Motivation

As it was noted in the previous subsection of this thesis self-tracking fitness and health apps are on the spotlight of the QS movement. They mostly focus on measuring and displaying the user performance. This data provides guidance and motivation for future improvement. These QS apps offer a direct benefit for the user which makes them popular. People have interest in tracking these aspects of their lives long before the digital revolution. Still there is much more data about ourselves that we unconsciously create on a daily basis. Everything that we undertake over the Internet with our personal computers or smartphones leaves a data trace. This thesis considers that there is potential in exploring how this data can be visualized.

One of the challenges for the QS movement is found in the presentation and visualization of the QS data. There are potentially many hidden meanings in each QS dataset. Through proper data visualization we can contribute in discovering these insights. It is important to note that the choice of how we visualize QS data directly affects the quality of the self-reflecting process. Effective visualization will give to the user an added value [Marcengo and Rapp, 2014]. We need to address the methods of how we display this information in a meaningful manner.

7Zeo Sleep Manager Pro: Mobile, Fitbit, BodyMedia FIT Core Armband, iHealth Blood Pressure Monitoring System, and iHealth Digital Scale.
1.2. Motivation

Plain charts are not enough to display multivariate QS data [Marcengo and Rapp, 2014]. This claim is based on a study that shows that the viewer attention focuses only on partial segments from the displayed data visualization. Furthermore the article argues that the visualization should govern the points of focus especially in cases of large datasets such as QS data. This creates the need of exploring new ways of representing our data trace. As a solution Marcengo and Rapp suggest a model of storytelling. As the saying goes, data presented in a picture can hold thousand words. The pre-attentive visual processing of an image can provide the users with meaningful insight without applying complex analysis. This can make the understanding of the personal data very straightforward. Following up on this, the thesis raises the question “How to generate the visualization of a story?”. The use of avatars in presentation of QS data can contribute to the idea of storytelling through an image. Complex aspects of our everyday digital records can be simplified and presented within an avatar. This method of data representation could provide clear insight of our life habits and increase of accessibility of our own data.

In [Nake, 2015; Nake et al., 2016] as well as [Bogdanov et al., 2013], the authors use items to represent the information. The position and the size of the item should also signify some sort of meaning. This approach seems visually appealing and playful for the users. It is also worth to notice that this type of visualization is limiting because it can only be used for the project it was created for. Reusing those visualizations for a different purpose demands adjustments and rethinking some visual aspects. This project avoids using items and visualize the values from the data as part of the face and body features. This takes the project to the core concept of Chernoff faces. The Chernoff method uses facial expressions to present multivariate data. This concept can be used as an avatar face. Additionally it will allow the prototype to be reused for different type of QS datasets.

In the Merriam Webster dictionary, “avatar” is defined as an electronic image that represents and is manipulated by a computer user in a virtual space (as in a computer game or an online shopping site) and that interacts with other objects in the space. Data generated avatars can create a bond between the users and their data. They can represent embodiment of our data records. There is a potential for creating a prototype that can provide interactive data visualization with the use of an avatar.

Recent events such as the Cambridge Analytica scandal [Cadwalladr and Graham-Harrison, 2018] have raised the public awareness about Internet privacy and safety of personal data. Events like these have shaken the trust between companies that collect data and their users [Greenfield, 2018]. A possible solution to overcome this mistrust can be in increasing

---

1.2. Motivation

the transparency from the company’s side. Providing the collected data back to the user offers an added value for both sides. The company can show in details what kind of information it collects, while the user can dive into the world of QS. There is a potential to use the avatar visualization for this form of data. Very often the collected data can be seen as irrelevant or boring. The avatar is meant to deal with multivariate complex data. It can provide meaningful insights about the user activity in an engaging and interactive manner. This type of visualization can be used to raise awareness among users about their online presence. Having frequent updates for the user activity which are easy to consume will improve the understanding and perception of how the services actually work. It can show in detail aspects of their usage for which the users are not completely aware of or they do not directly correspond with their personal impression.

Furthermore, early QS apps show that users give up on Quantified Self and fitness-tracking applications in a short timespan [Ledger and McCaffrey, 2014]. This creates the need for exploring new interactive ways of data visualization in order to maintain the user attention in a longer timespan. A potential solution is found in visualizing passive collected data from the user. This data is mostly generated for commercial purpose. Although the structure of the information is complex, through the use of the avatar it can be refined for mass usage. This visualization does not require any active participation from the user in the process of data collecting. Taking an advantage of something that already exist could be one of the core concept for maintaining the user attention.

As mentioned, the QS movement has found most of its popularity among users who are interested in tracking their health and workout performance. It is important to note that this type of tracking is done intentionally with the aim to gain useful insight about our wellbeing and physical condition. Still we have to consider the fact that there is much more in the movement besides health and fitness tracking. On an average, a user generates between 0.33 - 0.79GB of data per day. This data is neglected by their creators. The avatar visualization can provide context for this vast sea of data. It is envisioned to simplify the way we present data which can ease the information consumption for the user. The avatar can serve as a gateway for the user in the QS movement.

This section offers potential in researching the idea of generating avatars from QS data. For me as an author this field holds uncharted knowledge

---

1.3. Research questions

which is challenging to explore. Additionally I firmly believe that the main 
focus of technology should be improvement of human lives and the values 
of the Quantified Self movement goes along with this cause.

1.3 Research questions

Complementing on the motivation which was brought up in the previous 
section, the following research questions were composed:

$RQ_1$ How to create an avatar based on the Chernoff faces technique for 
data visualization?

$RQ_2$ How to visualize multivariate Quantified Self data using an avatar?

$RQ_3$ How to visualize passively collected personal data in an avatar?

$RQ_4$ How to provide meaningful insights for the users through visualization 
of their personal data in an avatar?

With an attempt to answer the first three questions the thesis will de- 
velop an online application that visualizes QS data in an avatar. The avatar 
will use facial expression to present data. This exploratory project will use 
data from Spotify. Through every user activity on the platform a log is cre-
ated in the background of the service. Using users consent, the platform 
offers access of their activity. Additionally it provides advanced meta data 
for the streamed music. The logs combined with the meta data could pro-
vide intriguing avatar for the users. This extensive API represents a pas-
sively collected multivariate QS data which makes it ideal for the research 
project. Although the implementation will be conducted on data from one 
social network, the overall work will provide a know-how and a technical 
solution for reusing the proposed concept in a different project.

For the fourth question a study will be conducted to observe the user 
behavior. The study will provide answers of how the users perceive and 
understand the proposed concept. The results of the study will be used for 
creating a second prototype. Through reflecting on the user behavior into 
the design process, the second prototype should provide deeper insight 
than the first iteration. This reiterative method will provide a model of how 
to generate a meaningful data visualization using an avatar. More on the 
approach and methodology is elaborated in the Chapter 3.

It is worth to notice that the thesis finds a potential in researching ap-
proaches that can extend the usage of QS applications through a longer 
time span. As elaborated in the previous section, dropping out (due to 
loss of interest in self tracking) is one of the current problems of the QS 
movement. The author of the thesis considers that the time frame in which 
the thesis project needed to be implemented, analyzed and documented
1.3. Research questions

did not allow for exploration of this problem. This type of issue requires a longer study through several iterations. As it is not intended to provide an incomplete answer, this additional question will be postponed to future work – through the fourth research question of this thesis contributes in partially addressing this problem. Providing meaningful insight could spark interest for longer usage of the avatar. This finding could be reused in other QS apps.
2 | Related work

2.1 Visual perception of facial expressions

People grow up studying and reacting to faces all of the time. Small and barely measurable differences are easily detected and evoke emotional reactions from a long catalog buried in the memory. [Chernoff, 1973]

Chernoff face visualization tends to exploit this human capability for the purpose of presenting multivariate data. The core concept of the visualization is to show different aspects of a dataset on 18 different face elements. According to Chernoff the visualization holds mnemonic advantages when it is used for comparative analysis of large numeric data. In an analysis [Bruckner, 1978] Bruckner argues that one of the advantages for using the Chernoff visualisation method is that faces are easily recognized and described. He provides an analogy which states that a common user could face obstacles in the differentiation between sine curves. An untrained user could struggle to recognize and describe different curves. On the other side for the user it would be easy to pinpoint each element of the human face without any training. Additionally the user could easily communicate and notice changes in the facial expression.

There is an ongoing debate among the scientific community about the effectiveness of the Chernoff face visualization. Although this visualization concept has shown its practical implementations, research has also given results that the faces do not offer pre-attentive visual processing [Morris et al., 1999] as claimed by their author. In general Chernoff faces do not offer any particular advantage over other iconic type of visualization. Another study [Sivagnanasundaram et al., 2013] which aimed to explore Chernoff claims search showed that there is no advantages in the method for supporting efficient spatial search of data. The study tested the claims by measuring perception on various set of faces on 65 participants.

When it comes to recognizing facial expression, the human mind perceives the face as a whole [Robert Kosara, 2007]. In the context of Chernoff faces, this means that the user needs to be trained to break down different aspects of the face visualization in order to read and understand the
2.1. Visual perception of facial expressions

presented dataset. Additionally the facial expression recognition follows a strong hierarchy [Robert Kosara, 2007]. This set of cognitive rules determinants in which facial features we focus at, and how we identify people. For example pupils and eyebrows have much more attention then all the other features [Morris et al., 1999]. This leads to the conclusion that some different segments of the data can be neglected.

Despite the argument that Chernoff faces can present vast spectrum of multivariate data, not every case is appropriate for applying this type of visualization. This is due the fact that the facial expressions can represent a particular emotion. In situations where the dataset holds sensitive information which can temper with human emotions, applying Chernoff face can lead to confusion or even insult. In order to avoid this type of situations the designing process must be done with percussion and reiterative analysis.

As mentioned, Chernoff faces is the main inspiration for the overall project. With the anticipation that the previous arguments can also apply to the avatar, the thesis added supportive features in the application UI to overpass the possible limitations. The features included detailed instructions of how the avatar works, tooltip\(^1\) and email support. During the designing process of the avatar, additional attention was made for the segment of linking the data with the body features.

Chernoff also argues that if the faces are presented individually they provide a small value as a communication tool. Presenting the data as a personal avatar while using the human capabilities for noticing changes in facial expression could hold the key of using this type of visualization as a communication tool. Active use of the avatar can communicate different states of the user’s recent activities. Additionally displaying multiple avatars to represent a longer time span could help the user to conduct a comparative analysis of its data trace.

In their work Ekman et al. propose a Facial Action Coding System (FACS) for analysis of facial expressions [Ekman and Rosenberg, 1997]. They recognize various muscles which when contracting they display different facial expressions. Those combinations of facial muscle contractions are defined by Ekman as Action Units (AU). Different AU can display various emotions, something that the human mind is programmed to understand. Chernoff claims that the visualization of the facial expressions can be cartoonish instead of realistic. The human mind is capable of recognizing facial appearance even in illustrations [Chernoff, 1973]. Different positions of FACS can be used for selecting various features from the avatar face as placeholders for data. Still the concept of AU is merely an inspiration and loose guidance for the creation of the avatar. This project does not aim to present QS data as emotions of an avatar.

---

\(^1\)A tooltip was implemented in the second iteration.
2.2 Use of avatars in data visualization

As part of the “Visualizing quantified self data using avatars” project\(^2\), Alissandrakis and Nake proposed a design for avatars using data from Instagram, Twitter, LastFm and Human API. In Nake’s work [Nake, 2015] she conducted a user study with the goal to explore the possibilities of using an avatar in visualization of QS data. Based on the findings from the user study she created a prototype [Nake, 2015; Nake et al., 2016]. The project showed that most of the participants were able to recognize their avatar generated from their activity within a set of avatars. This finding provided a solid foundation for further research.

An interesting study [Farshid et al., 2012] suggests the use of Chernoff faces as a tool for visualization of consumer data from social media. They see the use of Chernoff faces as a powerful tool for comparison. In the study they analyze data from six Sauternes wine brands based on social conversation measurement. Although this is not QS data it still is a good example of using the faces for visualization of data. This paper provides a good insight of the thinking process for data visualization by using Chernoff faces. The result of the study shows that Chernoff faces could become invaluable tool for comparison of data for brand managers particularly in cases where there is a need of simplified data. For this case Chernoff faces showed that it could be a useful visualization of clustered data.

A visionary project [The Discipulus Project, 2013] funded by the European Commission aims to implement a medical avatar for each patient in the European Union. The project aims to generate the avatar based on health data for the patient from different health institutions. With this project, The Discipulus Project aims to solve the problem of dispersed data from the European health care system. Furthermore the aim of the avatar is to provide for the patients more detailed information about their condition, treatment options and possible means to improve their lifestyle. An opinion article [Brown, 2016] based on the project sees a mnemonic potential of the future avatar by directly pinpointing for the user their health issue. Project like this show the overall research and commercial potential for using an avatar for presenting personal data.

The QS movement holds a significant number of visualization projects. One particular from Jana Beck uses Chernoff faces for visualization of her diabetes (insulin) tracking system [Beck, 2014]. The project uses the faces in order to establish an emotional connection with the data. This work is also interesting from a technical perspective since it is using D3js library, which is the same technology that was planned to be used for implementation segment of the thesis.

2.3 Quantified Self Spotify projects

An interesting visualization by Skiptune LLC\(^3\), demonstrates how to link musical metrics to facial features of Chernoff Faces. The faces are constructed using melody data from different types of music genres. The results provide interesting oversight in the variation of different music features according to their genre. The overall project shows how music can be visualized using Chernoff faces. This approach inspired me to reflect on music data extracted from user profiles. Music holds a great importance in our everyday lives. Reflecting on our daily music listening activity can provide deeper understandings of our habits. This type of data seems suitable for the first iteration of the project.

A study that aims to visualize music metadata into avatar was conducted by Bogdanov et al., collecting music data from LastFM user profiles [Bogdanov et al., 2013]. They tracked the most popular songs for a user in order to determine his/her music style through an algorithm analysis of the songs. In the first part of the study they analysed different features from a song. Afterwards they interpreted the results in a music avatar. The avatars vary depending on the most dominant type of music which was discovered by the algorithm. This research is very similar to my thesis prototype since it includes music analysis and visualization of the results into an avatar.

2.3 Quantified Self Spotify projects

The Internet holds many different “homebrew” projects of Spotify users analyzing their own data. In the blog post, *Is my Spotify music boring? An analysis involving music, data, and machine learning*\(^4\), the author shows how the music features for a song provided by the Spotify API can provide deeper insight of the music listening habits. In a similar project, *Visualizing Hundreds of My Favorite Songs on Spotify*\(^5\), by using the song features, the author demonstrates how visualization can improve the understanding of our music preferences. As a follow up on this article a visualization tool called *SpotRecs*\(^6\) was developed. The tool utilizes Spotify’s API in order to process and analyze songs, allowing users to know more about what music they enjoy. Popular visualization tools such as *Tableau* can also directly access Spotify personal data. This feature allows the users to explore their

\(^3\)Skiptune LLC is a self-funded corporation established in the Commonwealth of Virginia in 2014. The Skiptune project seeks to understand how melodies are structured and have evolved across cultures, genres, and years, and is as much of a process as a project. [Skiptune LLC, 2014]

\(^4\)Detailed link for the blog post: https://towardsdatascience.com/is-my-spotify-music-boring-an-analysis-involving-music-data-and-machine-learning-47550ae931de, last open on 28/07/2018

\(^5\)Detailed link for the blog post: https://medium.com/cuepoint/visualizing-hundreds-of-my-favorite-songs-on-spotify-fe50c94b8af3, last open on 28/07/2018

\(^6\)Detailed link for the application: https://jyu.github.io/spotrecs/, last open on 28/07/2018
2.3. Quantified Self Spotify projects

music preferences. Projects like these are providing interesting overview of how Spotify data can be utilized for discovering personal insights. They also offer solid guidance how to collect and analyze Spotify data.

As a part of Spotify DevX 2017 alongside with other team members, the thesis author had the opportunity to develop a web application\(^7\) that generates art based on a user’s last fifty Spotify songs. *musicScapes* generates a landscape based on the Spotify music features for every extracted song. The landscape changes depending if the user has listened to happy or sad songs, energetic or calm, major or minor mode and other track features. The purpose of the project is to provide the users with a bigger insight of their Spotify activity and establish a healthier relationship with their data. We saw this app as a daily reflection tool for the music lovers. The thesis work at the time inspired the creation of musicScape. Although the app never got any official user testing, the design and the implementation processes provided a valuable knowledge for the ongoing work on this thesis.

\(^7\)Detailed link for the musicScapes application: [https://musicscapes.herokuapp.com](https://musicscapes.herokuapp.com), last open on 28/07/2018
3 | Methodology

The methodology for this thesis is divided in three segments. The first segment offers methodology for developing a visual system for the avatar. The second segment aims to explore the data collected from the user Spotify accounts. The third part of the methodology is focused on user testing of the proposed concept.

3.1 Prototyping

3.1.1 Exploration of various avatar implementations

With the goal to get inspiration for the prototype design an online research was conducted to explore various online avatars and Chernoff visualizations. The research took in consideration examples of avatars that can potentially hold user data in their elements. Those examples were mostly consisted of avatars with detailed illustrated faces and bodies. Some of the most influential examples are presented on Figure 4.1.

In the next step an analysis of the collected examples was made by the author of the thesis. The analysis was focused on two segments. The first explored various implementation of avatars which provided guidance for the prototype aesthetics. The second analysis aimed to envision possibilities of binding data in the avatar elements. They were individually observed by analyzing their design elements (space, shape, color, texture and position). For each example various comments were provided. The comments didn’t follow any specific structure. They aimed to serve as a mental note for the author opinion. For example for Figure 4.1.C some of the comments were that it is visually appealing approach which can be implemented in a short period of time. More on this is elaborated in subsection 4.1.2.

Based on the comments four basic requirements were generated. They served as guidance in the designing process as well as in the implementation of the web application. The requirements were as follows:

- The design should offer features that can represent different states of the user data.
- The changes of the avatar should be noticeable.
Development of the avatar mockup

- The avatar should be understandable.
- The avatar should be simple to implement.

The requirements are explained in more detail in subsection 4.1.1.

3.1.2 Development of the avatar mockup

For conceptualizing the avatar visual appearance and data presenting methods, several static graphic mockups were made. The mockups were illustrated using Adobe Illustrator. In addition of the graphical elements for the mockups, notes were added to explain the possibilities for the data presenting features of the proposed avatars.

After the designing of the mockups a discussion was made to choose the final concept for the avatar between the author and the mentor of the thesis. During the discussion the previously mentioned requirements were taken as standard for grading the mockups. At the end one of the mockup was chosen as the avatar appearance for the prototype.

3.1.3 Concept design questionnaire

With the aim to evaluate the data mapping for the chosen mockup a user study was conducted through a dissemination of an online questionnaire. This questionnaire was meant to test and validate some proposed visualization concepts. The users were tasked to answer the questions with the option(s) that they consider to be the most appropriate. The questionnaire challenged them to match examples of various state of the avatars with their equivalent data chart and vice versa. In the final stage of the questionnaire the users were asked to recognize a potential state of the avatar based on some music listening example. They were also told that there is no such thing as a correct answer. Throughout the questionnaire the participants had the opportunity to provide opinion for each feature individually and for all the features as a whole. The questions were repeated in different forms as a method to measure the participant consistency.

Behind the scenes for the purpose of the evaluation of the received results, there were answers which were considered to be close to the predicted mapping and answers which were considered to be extremely off from the envisioned visualization. This insight offered solid evaluation and guidance for improvement of the proposed concepts.

3.2 Data exploration

For this project it is important to understand the data retrieved from Spotify. Spotify offers rich data for the songs that can be streamed from the service. The data which was collected for this project is consisted of all the streamed
3.3. User study

songs for a given user profile during the user study. For each of the songs there was also a collection of audio features which are provided by Spotify. For that purpose an exploration of Spotify API documentation was made. The research was focused on the endpoints that provide audio features and audio analysis of a song, since these parts of the data seemed abstract and hard to interpret in a practical case. This part of data is one of the core features for the visualization. This research also provided understanding about how Spotify tracks user activity and how to utilize that in the project implementation.

For that purpose a set of song was used to see the spectrum results for various music genres. The findings from the research gave direction how to parse, calculate and map the retrieved data.

3.3 User study

For the user study a small online campaign was made to recruit participants. The recruitment campaign was consisted of disseminating adds for participants over Facebook and Twitter. Additionally word to mouth was used to promote the app among the students and teaching staff at the Linnaeus University. The implemented application was named musicAvatar. This was done with the intention to simplify the communication with the participants and to make the overall project more appealing for the public.

After their first login the users received an email with instructions how to use the avatar as well as details for privacy of the app. The email instructed the users to read carefully the instruction page in the application in order to learn how the avatar works and the privacy page which was consisted of details of what kind of data is extracted from the user profile and how it is used. The email asked the users to use the app as regular web application with no emphasis for the amount of usage. It also informed them that at the end of the testing phase, some of them would be contacted for an additional interview.

The study for the first iteration of the prototype lasted for a month while for the second iteration the user study period was two weeks. The users were able to use the musicAvatar application by logging in (providing access) with their Spotify records. Only Spotify users were invited to participate in the testing.

3.3.1 Data collection

The prototype was programmed to log the users activity. Additionally Likert scale and a comment box were added as optional inputs for the user opinion. The collected information was used for the analysis of the prototypes. The following list describes the data documents used for the logs during the two iterations:
Data collection

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Used name on Spotify</td>
<td>1, 2</td>
</tr>
<tr>
<td>email</td>
<td>String</td>
<td>Contact email provided to Spotify</td>
<td>1</td>
</tr>
<tr>
<td>spotifyID</td>
<td>String</td>
<td>Unique Spotify ID used for cross-referencing the User</td>
<td>1, 2</td>
</tr>
<tr>
<td>spotifyRefreshToken</td>
<td>String</td>
<td>User token for extracting data from Spotify</td>
<td>1, 2</td>
</tr>
<tr>
<td>recentlyPlayed</td>
<td>String</td>
<td>Reference for the user song record document</td>
<td>1, 2</td>
</tr>
<tr>
<td>avatar_type</td>
<td>String</td>
<td>Body type for the avatar chosen by the user</td>
<td>1, 2</td>
</tr>
<tr>
<td>updateTimeStamp</td>
<td>Date</td>
<td>Detailed date when the app fetched data for the user</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

Table 3.1: User logs. The table presents JSON object that contains the basic information for the user.

- **Users**

  The data document (see Table 3.1) kept the user contact info. It was used to reference the user data in the other log documents. The personal contact info was only used for communicating with the users.

  Some of the data was extracted from the GET /v1/me/ endpoint at Spotify. Additionally this data document kept record of the user last activity, Spotify credentials, and app preferences.

- **User record**

  The User record (see Table 3.2) holds all songs which were listened on Spotify by the users during the User testing phase. For each user there is a document that holds array of Song objects. The song objects contains all the essential information about the song. This data is used to analyze the users Spotify music activity and to generate their avatars.

  The information is generated by merging data from two endpoints: GET v1/me/player/recently-played and GET /v1/audio-features.

- **User activity.**

  The app kept a log record of all the attempts to fetch data from Spotify. The initiation of fetching the new data was done on four hour intervals

---

Data collection

by the server side or on opening the app by the user. The User Activity
document log (see Table 3.3) holds an Array of Objects. Every Object
represent an individual activity.

• **User seen data**

Every time the app displayed an avatar a record log was created. This
was used to explore the user experience. In the analysis this logs
offered a detailed insight of what type of avatar state the user experi-
enced. This log only kept track when data was displayed. Whenever
there was no data to be seen the activity was logged only with the
User Activity log. See Table 3.4.

For the First and Second prototype as well as the MyDay feature the
app kept a separated log document.

• **User comment**

All comments provided by the user on the MyDay feature and the sec-
ond prototype were tracked by the User comment log. This log kept
a simple record of the comments provided by the user. Analysis of
these logs provided direct insight of the user opinion. The comments
were also used in the process of selecting the participants for the
interview. See Table 3.5.

• **User opinion**

In the first prototype the users were able to provide an opinion through
a Likert scale with a comment box. They were asked if they think that
the avatar accurately represent their last two hours of Spotify music
listening. The option 1 represented Strongly Disagree and option 5
Strongly Agree. The comment section was optional. This logs pro-
vided material for analysis of the user opinion. The document struc-
ture is presented on Table 3.6.

• **User download img and User change body**

The final two type of logs recorded when the user used the change
avatar and download image feature. The documents included a time
stamp of the record and user Spotify ID. For the change of avatar a
record of the new chosen type of avatar was also kept.

The data was stored and analyzed in JSON format. All the data were
anonymised with the intention to preserve the user privacy. The recorded
logs provided a detailed overview for each user. Through the User study the
project tested the core concept for the avatar. This study offered detailed
feedback for the user experience and gave directions for further improve-
ment.
### Data collection

#### Table 3.2: User record logs. The table presents a JSON object that holds data for all the recent played songs for each user user. The recentPlayed property present an array of Song Objects.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>spotifyID</td>
<td>String</td>
<td>Unique Spotify ID used for cross-referencing the User</td>
<td>1, 2</td>
</tr>
<tr>
<td>recentlyPlayed</td>
<td>[array of Song objects]:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>played_at</td>
<td>Date</td>
<td>Time stamp when the song was played</td>
<td>1, 2</td>
</tr>
<tr>
<td>artist</td>
<td>Array</td>
<td>Array that contains a unique Artist/s ID/s of the song</td>
<td>1, 2</td>
</tr>
<tr>
<td>spotifySongID</td>
<td>String</td>
<td>Unique String for referencing the song</td>
<td>1, 2</td>
</tr>
<tr>
<td>duration_ms</td>
<td>Number</td>
<td>duration of the song in milliseconds</td>
<td>1</td>
</tr>
<tr>
<td>popularity</td>
<td>Number</td>
<td>Popularity of the song according to Spotify based on the number of overall streams; Measured from 0 to 100</td>
<td>2</td>
</tr>
<tr>
<td>valence</td>
<td>Number</td>
<td>Spotify song feature representing how positive the song appears to be; Measured from 0 to 1</td>
<td>2</td>
</tr>
<tr>
<td>instrumentalness</td>
<td>Number</td>
<td>Feature representing the use of real instruments for each song; Measured from 0 to 1</td>
<td>2</td>
</tr>
<tr>
<td>speechiness</td>
<td>Number</td>
<td>Feature representing the use of spoken words in each of the songs; Measured from 0 to 1</td>
<td>2</td>
</tr>
<tr>
<td>danceability</td>
<td>Number</td>
<td>describes how suitable a track is for dancing based on a combination of musical elements; Measured from 0 to 1</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Table 3.3: User activity logs. The table presents a JSON object that holds information for an event when the app tried to fetch data from Spotify.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>spotifyID</td>
<td>String</td>
<td>Unique Spotify ID used for cross-referencing the User</td>
<td>1, 2</td>
</tr>
<tr>
<td>time_stamp</td>
<td>Date</td>
<td>Time stamp when the activity was logged</td>
<td>1, 2</td>
</tr>
<tr>
<td>didFetchData</td>
<td>Boolean</td>
<td>Boolean that shows if data was retrieved from Spotify</td>
<td>1, 2</td>
</tr>
<tr>
<td>auto</td>
<td>Boolean</td>
<td>Boolean that shows if the data was fetched automatically by the server</td>
<td>1, 2</td>
</tr>
</tbody>
</table>
Data collection

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>spotifyID</td>
<td>String</td>
<td>Unique Spotify ID used for cross-referencing the User</td>
<td>1, 2</td>
</tr>
<tr>
<td>time_stamp</td>
<td>Date</td>
<td>Time stamp when the activity was logged</td>
<td>1, 2</td>
</tr>
<tr>
<td>obj</td>
<td>JSON Object</td>
<td>Object that contains all the data parameters for the displayed avatar</td>
<td>1, 2</td>
</tr>
<tr>
<td>body_type</td>
<td>String</td>
<td>String value of the displayed avatar body</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

Table 3.4: Users seen data. The table presents a JSON object that holds the information of all the time the app displayed an avatar.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>spotifyID</td>
<td>String</td>
<td>Unique Spotify ID used for cross-referencing the User</td>
<td>1, 2</td>
</tr>
<tr>
<td>time_stamp</td>
<td>Date</td>
<td>Time stamp when the activity was logged</td>
<td>1, 2</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>String value of the provided comment by the user</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

Table 3.5: Users comment log. The table holds a JSON object for a comment provided by the user for the displayed avatar.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>spotifyID</td>
<td>String</td>
<td>Unique Spotify ID used for cross-referencing the User</td>
<td>1</td>
</tr>
<tr>
<td>time_stamp</td>
<td>Date</td>
<td>Time stamp when the activity was logged</td>
<td>1</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>String value of the provided comment by the user</td>
<td>1</td>
</tr>
<tr>
<td>value</td>
<td>Number</td>
<td>Value between 0 - 5 representing the opinion provided by the user</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.6: Users opinion log. The table holds a JSON object for a comment and opinion provided by the user for the displayed avatar.
3.3.2 Post-study interviews

At the end for both of the user studies the methodology included interviews with some of the users. For the first iteration few participants were chosen for an oral interview. The interview questions were composed after a brief analysis of the collected records. Three users participated in the user testing. The most active participant who was also provided constructive criticism towards the prototype in the comments was chosen for the interview. The other two participants represented “average” active users on the application. In their initial feedback before the interview they provided both high and low values for the avatar. The interviews lasted between 30 and 40 minutes.

For the second iteration a list of written question was send to all the testers. Same as before, the questions were composed after a brief analysis of the collected records. All testers received questions. The questions were identical for every user.

The goal of this phase to get a detailed insight from an user perspective. The answers helped to improve the prototype design for the second iteration and they provided inspiration for future work.

3.4 Limitations

3.4.1 Finding participants

The study struggled to find appropriate testers. Although there was a significant number of users only few of them were actually interested in tracking their music behavior. The project depended on a specific profile of users. The ideal candidate was considered to be a proactive Spotify user who is interested in tracking his/her daily music habits. A person with this profile could provide a meaningful feedback for the avatar.

The early results, retrieved after the first week of the user study of the first prototype, showed that the initial users are not as active on Spotify as the project wanted them to be. This issue was tackled by promoting the app in various Facebook groups for music and data visualization. The assumption was that in these groups there is a higher chance of finding the proper candidates for testing. Still considering that this project is conducted by one person the overall outreach didn’t manage to acquire a larger number of proactive Spotify users interested in tracking their activity.

3.4.2 Spotify data limitations

Spotify offers comprehensive support for developers using their API. Still during the user study there were cases in which the project was limited due to the rules set by the provider of the data, malfunction of the company.
Spotify data limitations

system and bugs in the prototype deployment. Spotify limits the number of songs which can be retrieved from the recent_played endpoint to 50 per user. This meant that musicAvatar was able to see only the last 50 songs streamed by the user. This policy presented a limitation because the prototype was designed to present time periods. For the avatar to be accurate it needed a stream of linear data, while Spotify provided desecrate data. This issue was resolved with an implementation of an interval of http requests towards Spotify for each user individually. The server side on every four hours made a call to Spotify to see if there are any new songs for the users. Cases in which the user has listened to more than 50 songs in a time span of four hours the web application was not able to see all of the data. Although it was considered that cases like these are not very likely to happen, there was no mechanism to spot or record them.

As an additional limitation was considered the server side of the prototype. There were times were due to unhandled error or malfunction by the server host provider the intervals did not fully complete their iterations. The reasons for the unhandled errors were mostly because of inconsistency in the received data. This cases were not anticipated during the development of the application back-end. They were fixed “on the go” during the user testing period. Although these type of bugs were reported and handled in a short period, there is a potential of missing some records for the users.

The data testing mentioned in the subsection 3.2 showed that some music features can appear to be incoherent. For example the API documentation from Spotify claimed that the speechiness for rap music should be above 0.33. In many cases for particular rap songs the value was bellow 0.05. This was considered confusing. Similar cases were spotted with the instrumentalness feature. This inconsistency of the data presented challenging burden for providing proper thresholds.

Finally there were reported cases in which some of the Spotify API servers were reported to be down. This proved limited access to the API which made the application significantly slow. During this period there might be cases were the app missed data for the users. Additionally these problems affected the quality of the user experience due to increased waiting time.
4 | Prototype development

4.1 Design of the prototype

4.1.1 Design principles

The following design requirements were used as a guidance for creating the mockup of the avatar. They were generated based on an analysis of various avatars and Chernoff visualizations.

- **The design should offer features that can represent different states of the user data.**

  Since the avatar is meant to represent data the design should envision features that can display different states. This can be achieved by changing the position of some of the avatar elements according to the data input. For example similar to the concept of Chernoff face, the mouth of the avatar can hold data. Open type mouth can hold one state of data, while closed mouth can hold the opposite state. Following this logic different body and face elements can be used for data visualization.

- **The changes of the avatar should be noticeable.**

  It is important for the user to notice the changes on the avatar. On the opposite the visualization can appear as a static image. For that purpose the dynamic features should be emphasized while the rest of the body should appear neutral. This can be achieved through the use of various design elements such as color, size, shape, form, space and position. For example a strong color for the mouth which is placed on a neutral skin color can get the attention of the user.

- **The avatar should be understandable.**

  The user should be able to easily understand the proposed concept for visualization. For that reason the avatar should apply clear differentiation between the extremes in the dataset. The viewer should be able to recognize and separate the presented minimum and maximum. The in-between states should follow a logical progression. This
Graphic design of the avatar

Figure 4.1: The image presents the main inspiration for the musicAvatar, A) Facebook Space, B) Bit emoji, C) Unknown project which implemented pixel like Chernoff faces.

could allow for the user to start consuming the avatar without the need of any complex instructions. This approach excludes the possibility of using symbolic presentation of data.

- **The avatar should be simple to implement.**

Considering the fact that the time for research is limited and that the focus of the thesis is a user study, the design should be simple to implement as a web application. This limits the space for applying details during the designing process. Additionally the used technology for implementation should take into consideration all limits. This includes my scope of developer skills. The final design should be simple to implement using the chosen technology without any need for learning additional skills.

### 4.1.2 Graphic design of the avatar

Before starting the avatar design I browsed over the Internet to find inspiration. I mostly focused on different types of avatars and Chernoff faces type of visualizations. Through the survey of existing projects I mostly found inspiration in the implementations of avatars by Facebook Spaces\(^1\) (see

---

\(^1\)Facebook Spaces main page [https://www.facebook.com/spaces](https://www.facebook.com/spaces), last open 28/07/2018
Figure 4.1.A) and Bitemoji\textsuperscript{2} (see Figure 4.1.B). Both of the projects use cartoonish design. Besides finding this styles visually appealing, they provide a vast scope of facial expressions. This gave solid direction how various face elements can be displayed in different “emotional” states.

The second inspiration that defined the technical aspect of the project came from a Chernoff face implementation\textsuperscript{3} (see Figure 4.1.C), which used “pixel” art design. This project contained shapes which are simple to code and changing features which are easy to spot. The overall cartoonish characters which I found appealing can be designed using square forms. This seemed interesting and straight-forward.

For the development of the mockups I initially used pencil and paper. The more promising concepts were afterwards designed in Adobe Illustrator. For each potential visualization I created multiple states of the facial expressions and body positions. This provided a solid basis for analysis of the proposed avatars. The end goal was to extract the most suitable features from each mockup and to apply them in a final avatar.

In the final phase of designing the mockups the process mostly went with a human and a robot representation. Figure 4.2 displays the most refined versions of the proposed avatars with their face and body features. In an analysis with the mentor of the thesis the robot representation was dropped. The arguments for this decisions claimed that coding three types of body representation can be time consuming. Furthermore the features of the robot were not completely synced with the features of the human. Finally we thought that having a robot alongside with humans can be confusing and misleading for the user.

4.1.3 Final avatar design

The final appearance (see Figure 4.3) of the avatar was created by selecting the most suitable features from the refined mockups. The big head approach seemed most applicable because this element is the focal point of the data presentation. The face features were designed to have multiple different positions. Still, with the intention to have clarity in the visualization of data only the positions who follow linear progression were chosen for the final version. Since the body didn’t presented any data it was implemented with neutral appearance in order not to distract the user attention. Both of the implementation had the same visual appearance although they have some variations in the functionality of the features.

In the first implementation (see Figure 4.3.A) of the avatar, data was

\textsuperscript{2}Bitmoji main page https://www.bitmoji.com, last open 28/07/2018

\textsuperscript{3}This project was found through browsing over the Internet. I didn’t manage to find more details about the actual implementation. Still the style of the design provided valuable inspiration for the technical implementation. https://www.slideshare.net/kalleheinonen/superanalytics-cfe, last open 28/07/2018
Final avatar design

Figure 4.2: This image shows the most refined versions of the mockups. A.1) Human mockup versions, A.2) possible states of the body and face feature for the human version, B.1) Robot mockup versions, B.2) possible states of the body and face features for the robot version.

Figure 4.3: A) First prototype iteration; A.1) eyes, A.2) arms, A.3) mouth; B) Second prototype; B.1) eyebrows, B.2) ears, B.3) arms, B.4) eyes, B.5) mouth. See also Table 4.1 for more details.
4.2. Mapping data on the avatar

<table>
<thead>
<tr>
<th>Avatar feature</th>
<th>First Iteration</th>
<th>Second iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eyes</td>
<td>Variation of artists and songs (emphasizing the eyes)</td>
<td>Variation of artists Spotify popularity of songs Instrumentalness</td>
</tr>
<tr>
<td>Eyebrows</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>Ears</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mouth</td>
<td>Amount of music</td>
<td>Speechiness</td>
</tr>
<tr>
<td>Arms</td>
<td>Danceability</td>
<td>Valence</td>
</tr>
</tbody>
</table>

Table 4.1: The table contains all used avatar features for data presentation in both of the prototypes.

presented in the eyes, mouth and arms (see Table 4.1). The eyebrows in this version were used to reinforce the eyes. They tilted based on the position of the eye pupils. For the mouth (see Figure 4.3.A.3 feature the selected displays were opened, semi-opened and closed, thus presenting three different states of one information. The arms (see Figure 4.3.A.2) also had three different positions, lowered, horizontally spread and up in the air. The arms and the mouth presented one dimensional data. The eye pupils (see Figure 4.3.A.1) were used as dot in a scatter plot with a possibility of four step movement, horizontally and vertically. This allowed for the eyes to have sixteen different states.

The second prototype presented data (see Figure 4.3.B) in the eyebrows, eyes, mouth, ears and arms (see Table 4.1). The eyebrows (see Figure 4.3.B.1) had three possible positions by moving vertically on the avatar forehead. In this iteration the size of the eye pupils was used to present data (see Figure 4.3.B.4). The avatar was consisted of three possible sizes for the pupils, small, medium and big. The ears (see Figure 4.3.B.2) presented binary data with two possible displays, pointy and normal looking ears. The mouth (see Figure 4.3.B.5) and the arms (see Figure 4.3.B.3) had the same design as the first iteration. In this avatar all features presented one dimensional data.

The avatar features are quite flexible. As it can be seen the features can adjust according to the project needs. Future implementations can also explore presentation of data in other body parts. The avatar provides mostly space for experimenting with colors and shapes. For example as it can be seen from the mockups the project suggested use of the hair as a placeholder for data.

4.2 Mapping data on the avatar

On Table 4.1 is presented the mapping of data through the avatar features in both of the prototypes. The process of choosing which music feature should be linked with the first implementation of avatar elements was done through a discussion. Since music perception is a complex topic and there
is no straightforward answer of how we perceive melody, the decisions for the mapping were mostly made from a personal experience using intuition. This was done to get a starting point for the user testing. The insight provided from the first testing gave guidance for improvement of the mapping and complexity of the avatar for the second user study.

4.2.1 First prototype [Figure 4.3.A]

- **Eye pupils** [Figure 4.3.A.1]
  The position of the eye pupils represents the “diversity” of music that the user has listened to by measuring the variety of artists and songs. The variation of songs moves the pupils horizontally; more to the right means that mostly unique songs have been played. The variety of artist moves the pupils vertically; upwards indicates big diversity of artists.

- **Mouth** [Figure 4.3.A.3]
  The size of the mouth represents the amount of music the user has been listening to in the past two hours. A small mouth indicates that the amount of music was low for the given period. A more open mouth means more time has been spent listening to music.

- **Arms** [Figure 4.3.A.2]
  The position of the arms represents the danceability of the music being analyzed. A low danceability is indicated by arms next to the avatar body; raised arms indicate higher danceability.

  According to Spotify, danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.

  The avatar displayed data for the user Spotify music listening activity in the last two hours. The moment of opening the application was used as a referent point. For example, if the application was opened at 15:07 (Today), the avatar would use the songs which played at time stamps are between 13:07 and 15:07 (Today). On Figure 4.4 are presented three avatars from the first prototype as an example:

1. **Avatar A** [Figure 4.4.A] has the lowest state of the arms, which represents music with little danceability. The mouth shows medium amount of played music in the presented period. The eye pupils displays high amount of repetitive songs and artists.

2. **Avatar B** [Figure 4.4.B] has the medium state of the arms, which represents music with intermediate danceability. The mouth shows
Second prototype

Figure 4.4: The image contains some example states for the first prototype.

high amount of played music in the presented period. The eye pupils displays significant number of unique artists with small repetition and mostly repetitive songs.

3. Avatar C [Figure 4.4.C] has up pointing arms, which represents music with high danceability. The mouth shows low amount of played music in the presented period. The eye pupils displays significant number of unique artists with small repetition and mostly unique songs.

4.2.2 Second prototype [Figure 4.3.B]

- **Eyebrows** [Figure 4.3.B.1]
  The position of the eyebrows represents the popularity of the music listened in a period of one day. The higher are the eyebrows positioned the more popular the music is according to Spotify.
  The popularity is calculated by a Spotify algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are.

- **Eye pupils** [Figure 4.3.B.4]
  The size of the eye pupils represents the “diversity” of the songs by measuring the variety of their artists. Small eye pupils represent small variety of artists and bigger eye pupils represent larger variety.

- **Ears** [Figure 4.3.B.2]
  The “spikiness” in the ears represents the use of instruments in the songs. Spike ears means the songs were mostly instrumental. Flat looking ears presents that the songs contained vocals.
Second prototype

The Spotify music feature instrumentalness, predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”.

- **Mouth** [Figure 4.3.B.5]
  The size of the mouth represents the amount of speech in the music. Smaller mouth represents music that is mostly instrumental and bigger mouth music that contains significant amount of lyrics.
  
The Spotify music feature Speechiness detects the presence of spoken words in a track.

- **Arms** [Figure 4.3.B.3]
  The position of the arms represents the positiveness of the music which is being analysed. A low positiveness is indicated by arms next to the avatar body; raised arms indicate higher positiveness (cheerful, happy).
  
  Valence is a measure by Spotify for describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

In this prototype the avatar displays data for a whole day of Spotify activity. The twenty four hours are calculated starting from midnight. The application showed seven avatars, thus presenting data for a whole week. When opened, the first presented avatar in the row was from the previous day. For example if the application is opened on Saturday, the first displayed avatar would be from Friday, the second would be from Thursday,
third Wednesday, etc. In Figure 4.5 are presented three avatars from the second prototype as an example:

1. **Avatar A** [Figure 4.5.A] eyebrows are in their highest state which represents that the analyzed music was very popular among Spotify users. The size of the eye pupils shows that the listened music was mostly from few repeating artists as they are in their smallest state. The mouth tells that there was high use of spoken words in the songs. The ears show that the music was consisted of analog instruments. The up pointing arms represent high valence of the listened music.

2. **Avatar B** [Figure 4.5.B] eyebrows represent that the analyzed music was with medium popularity. The size of the eye pupils shows that the listened music had medium uniqueness of artists. The mouth tells that there was medium use of spoken words in the songs. The spiky looking ears shows that the music was consisted of analog instruments. The medium state of the arms, presents music with intermediate valence.

3. **Avatar C** [Figure 4.5.C] eyebrows are in their lowest state which represents that the analyzed music was not popular among Spotify users. The size of the eye pupils shows that the listened music was mostly from unique artists as they are in their largest state. The mouth tells that there was low use of spoken words in the songs. The ears shows that the music was consisted of digital sounds. The down pointing arms represent low valence of the listened music.

### 4.3 Data calculation

All songs played during the presented time period are pushed in an array. This array is iterated once in order to calculate how the avatar features should be displayed. More on this is explained in subsection 4.5.2, Backend. The array members are selected from the `recent_played` property in the `UserDocument` object (see Table 3.2). They all present a `SongObject`. Their properties are used to calculate a JSON object for the avatar instantiation. Table 4.2 holds all properties and their thresholds used in the JSON object for both of the prototypes. According to the value of the properties the thresholds determinants the avatar display.

The properties for danceability, popularity, valence, speechiness and instrumentalness (see Table 4.2, Property) are calculated with an identical method. The iteration of the array, mentioned in the previous paragraph, makes a summation for each property by adding together their equivalents from the `SongObject(s)`. The sum is divided with the array length thus providing an average. The value of the average is used as a property in the
4.3. Data calculation

<table>
<thead>
<tr>
<th>Property</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prototype 1</strong></td>
<td></td>
</tr>
<tr>
<td>songs_uniques</td>
<td>( s_1 &lt; 25% &lt; s_2 &lt; 50% &lt; s_3 &lt; 75% &lt; s_4 &lt; 100% )</td>
</tr>
<tr>
<td>artist_uniques</td>
<td>( s_1 &lt; 25% &lt; s_2 &lt; 50% &lt; s_3 &lt; 75% &lt; s_4 &lt; 100% )</td>
</tr>
<tr>
<td>danceability</td>
<td>low &lt; 40% &lt; medium &lt; 75% &lt; high &lt; 100%</td>
</tr>
<tr>
<td>time_played</td>
<td>low &lt; 34% &lt; medium &lt; 64% &lt; high &lt; 100%</td>
</tr>
<tr>
<td>avatar_type</td>
<td>male or female</td>
</tr>
</tbody>
</table>

| **Prototype 2**      |                               |
| artist_uniques       | low < 34\% < medium < 64\% < high < 100\% |
| popularity           | low < 34\% < medium < 64\% < high < 100\% |
| valence              | low < 34\% < medium < 64\% < high < 100\% |
| speechiness          | low < 15\% < medium < 25\% < high < 100\% |
| instrumentalness     | low < 50\% < high < 100\%       |
| avatar_type          | male or female                |

Table 4.2: On the table are displayed the thresholds for all of the presented music features. This table also contains all the properties required for instantiation of the avatar.

The properties for artist_uniques and songs_uniques are calculated with an exact formula. The number of unique of songs/artist is divided with the number of songs/artist. The result is converted to percentage by multiplying it by 100. In cases when a song had multiple artist, they were merged and treated as one. The threshold for this properties is presented in Table 4.2.

The time_played property was calculated by dividing the actual amount of played music with the presenting period of two hours. The result is converted to percentage by multiplying it by 100. The time unit is milliseconds, because the time_stamp property is in UNIX format\(^4\). The actual amount of played music is calculated by subtracting \( \text{time} \_\text{stamp}_n - \text{time} \_\text{stamp}_{n+1} \) of the recent_played array (see Table 3.2). If the difference is smaller then the song duration\_ms, the difference value is used for the summation. If not, the duration\_ms is added. With this step the application filters if a song was played in a full length or skipped. For the first member of the list the duration\_ms is always used. The array list is ordered in chronologically, starting with the oldest song and ending with the most recent played song.

The threshold were set based on exploration of the endpoint of Spotify. For that different songs were used to get hold of the variation of values. In

\(^4\)UNIX Timestamp: [https://www.unixtimestamp.com/](https://www.unixtimestamp.com/), last open 05/08/2018
most cases the thresholds were set proportionally depending on the number of states the avatar feature has. In some cases where the music feature seemed disproportional. For example the threshold for speechiness was set lower since value above 0.33 would typically characterize rap music, and above 0.66 spoken word. Lower threshold excluded podcasts and poetry.

4.4 Developing the avatar

The avatar was developed using JavaScript. Following up on the idea to contribute to the QS movement by generating a visualization that can be used across different platforms the avatar was developed as a JavaScript library. This allows for the proposed visualization to be implemented in different projects with few easy steps. In its core the visualization works by generating a grid of squares with different colors. In this case the squares can be seen as “pixels” on a SVG canvas. A row-column based arrangement of different colored “square-pixels” generates the appearance of the avatar. This approach allowed straightforward translation of the static graphic design for the visualization into a programmatic dynamic solution.

The grid is built on top of D3js\(^5\). This library was chosen because it offers powerful data binding methods and advanced SVG elements manipulation. The data is injected in the DOM using the D3 append feature. The whole process depends on two types of iterations. In the first iteration all the rows of the grid are generated. Every row holds information for the number of squares, their position in the grid and their color. This information is stored as array of JSON objects. The second iteration generates each square of the grid. This iteration uses the data which was binded in the row.

Although the core concept of how the visualization works appears simple, it is important to note that it depends on a strict and complex data structure. If the intention is to provide a coherent display of the avatar the data structure needs to be respected in all the details during implementation. As mentioned each square in the visualization is defined by a JSON object. Concerning the fact that the avatars in both of the prototypes are 29 x 49 squares (1421 squares), a subroutine was created to generate each “pixel”. This subroutine is meant to take-in human readable JSON file and convert it into multiple squares according to the given commands. For this purpose the grid is defined as some sort of a Descartes coordinate system (see Figure 4.6). Each square holds an id based on the position in the grid. For example the first square on the top left holds the id of “X0Y0”. The last square in the bottom right holds an Id of “X28Y48”. With this kind of structure it comes much more intuitive to pin-point to a square and set its attribute of color. The subroutine exploits this taxonomy by taking in

\(^5\)Link for the JS library: https://d3js.org/, last open 03/07/2018
4.4. Developing the avatar

Figure 4.6: The image shows the grid used for defining the avatar elements. It holds examples of: definition of a “square-pixel”; definition of a rectangle; and the color palette for the avatar.
4.5. musicAvatar implementation

directives for how the Avatar looks in the Descartes system and outputs “square-pixels” JSON object array.

The human-readable JSON is used as middleware between the graphic design and the code. It is manually created using the grid as a map (see Figure 4.6). The JSON describes the avatar as matching color rectangles consisted of squares from the grid. These rectangles are organized based on the body elements. The JSON holds the position id of the starting and the ending square of the rectangles alongside the color of the rectangles. This information is enough for the subroutine to translate the data into JSON object for each square of the grid. In the interest of fast rendering the middleware JSON is divided into static body elements and dynamic. The dynamic body elements (eyes, eyebrows, ears, mouth, arms) are defined based on the input of the data from the user, while the static elements are predefined based on the chosen avatar body. This division significantly lowers the number of loops in the process of parsing the data.

The process for coding the avatar was identical for both of the prototypes. The second prototype is build by reusing the code from the first prototype. Changes were only made on how the eyes and the eyebrows display data. The ears were changed from static to dynamic element that can be binded with data.

4.5 musicAvatar implementation

4.5.1 Front-end

With the intention to achieve accessibility musicAvatar was developed as a web responsive application, thus making the application reachable from a mobile and desktop browser. This flexibility allowed the users according to their convenience to use the application on every type of device that has a browser.

The musicAvatar object is binded by a DOM element on the front-page. The application uses Bootstrap for the layout. The use of this framework allowed for a straightforward implementation of a responsive design for a desktop and mobile browsers. JQuery and plain JavaScript is used for the front-end logic (redirection, Ajax calls, creating cookies etc.)

When a user opens the application for the first time s/he is prompted with a login screen (see Figure 4.7.A). The login screen contains a warning box which explains that the application will use and store cookies on the user machine. Additionally the screen is consisted of a title of the application and a Spotify login button. When the button is clicked the application redirects the visitor to Spotify in order to get his/her credentials for musicAvatar. If the authorization is successful the user is redirected back to musicAvatar where s/he is given a page to choose an avatar body representation (see Figure 4.7.B). This is the final step for completing the user
Figure 4.7: The image presents all initial screens during the first login on the musicAvatar. A) Login with Spotify screen, B) Choose avatar body representation screen, C) Loading screen.
profile. After the user choice a loading screen appears (see Figure 4.7.C). This screen is used for events when the application is fetching data for the user from Spotify. When the fetching and parsing of the data is complete the user is taken to the avatar page. For further technical elaboration of how a user is registered see subsection 4.5.2.

The front-end side of the system uses cookies for maintaining a user session. The cookies are set to keep the user logged in for four days. They are refreshed on each opening of the application. This provides an improved experience since the user is not required to login every time s/he is accessing the app. The cookies are also used for limiting the number of server request calls to one per five minutes. Additionally they limit the feedback to one per one displayed avatar state.

First prototype

On Figure 4.8 is displayed the appearance of the avatar page for the first prototype. The figure shows the display of the page on a mobile and web browser. The main item of interest is placed in the middle of the page. The design uses minimal number of elements on a white canvas, so it can put the avatar on the spotlight. Below the avatar an optional Likert scale with an comment box is placed.

This iteration of the prototype offers the user to download a .png image from the presented avatar. It also allows for the user to change the avatar body representation. This features are placed in the menu of the application. Additionally the menu provides access to the privacy page, instruction page, myDay and logout (see Figure 4.9.B). The menu was dynamic which meant that it adjusted its content based on the displayed information on
Front-end

Figure 4.9: A) this is the warning box which is prompted when there is no data for the past two hours; B), C), D) examples of the dynamic menu.

the screen. For example when the privacy page was opened, the privacy button in the menu was replaced with go back to musicAvatar button (see Figure 4.9.C and D).

When there was no records to be presented for the past period of two hours the application showed a yellow box which informed the user about the situation (see Figure 4.9.A). In these cases the page offers a redirect button for the section myDay. This page displays a whole day of music activity divided in intervals of twelve hours (see Figure 4.10). Essentially it shows the past avatars from the last twenty four hours. The avatars are presented in a sequence of five seconds. The point of time for each avatar is displayed by a list of dots bellow the avatar. The dots switch colors according to the change of sequence. When there is no records for a certain sequence a yellow box is put as a placeholder. The user is able to share an opinion for this section of the application through a comment box. More on how the avatar displayed data can be seen in the section 4.3 and sub-section 4.2.1.

Second prototype

The second prototype used the same layout for the front-end as the first implementation. As mentioned in subsection 4.2.2, this iteration of the avatar
Figure 4.10: The figure presents an example of the myDay feature. On the left is displayed a sequence that holds an avatar for certain part of the day, while on the right is displayed a sequence that holds no data.

Figure 4.11: The image presents the second implementation of the avatar page layout for mobile and desktop version.
displayed data for a twenty four hours period. On the front page it shows a whole week of music activity by presenting data in seven avatars. When there was no activity for certain days an empty box was displayed (see Figure 4.11). Below every avatar the application provided the name of the displayed day.

The front end for the second prototype was much more lightweight. In this case it only offered an opinion box. It’s stripped down from features such as download the avatar and myDay. It also includes the privacy and the instruction page. This version of the prototype added a tooltip for each avatar. The tooltip appeared on a mouse hover over the avatar. The tool described with words each position of the avatar features. An example of a tooltip is presented in Figure 4.12.

4.5.2 Back-end

The back-end of the application is build on Node.js and MongoDB. The routing is handled by Express⁶, while the database connection is managed with Mongoose⁷. This part of the application handles the http requests

---

⁶Link for ExpressJS: https://expressjs.com/, last open 03/08/2018
⁷Link for Mongoose: http://mongoosejs.com/, last open 03/08/2018
Back-end

Figure 4.13: The Figure presents the basic MongoDB documents and their relations through a class diagram.

Towards Spotify, data parsing, data storing, page routing, activity logging and provides API for the front-end.

When the user logs in for the first time on musicAvatar through his/her Spotify account, the application fetches the user basic information (see Table 3.1). These data are used to create the user profile. It's stored in a Mongo document named User (see Figure 4.13). In the next step, the last played fifty songs are retrieved from the user Spotify account via /v1/me/player/recently-played endpoint. If the request call is successful, the server logic searches for the songs features in the MusicFeatures document. Assuming that the songs feature is found within the database the data from recent played song is is merged with music features. The merged data is stored in a document called UseData. In cases when the songs are not found in the MusicFeatures the app makes additional request towards Spotify /v1/audio-features to fetch the missing information. When retrieved they are added to MusicFeatures document. Then the app continues with the merging. These three documents are the core records for each user.

Recent played songs are collected from Spotify on two instances (see Figure 4.14). The first is on an interval of four hours for each registered user on musicAvatar. The second instance is when the user opens the app. In this case the application collects data only for the user. The reason for this
Figure 4.14: The diagram presents a basic schema for the back-end implementation.
kind a setup as mentioned in the Limitations section (see subsection 3.4.2) is because the API endpoint provides only the last fifty played songs. After each fetching of data a merger is conducted with the music features, as it was explained in the previous paragraph.

As it can be seen in the User document (see Figure 4.13), the app stores a Spotify refresh token for each user. This token is acquired after the user approval on Spotify during the first login on musicAvatar. The refresh token is used to gain an access token, in cases when the application needs to extract information from the user profile. Each access token has a validity of one hour upon which it needs to be refreshed. It is worth to acknowledge that Spotify offers a comprehensive and handy JavaScript library as an npm module. The library was used in the back-end of musicAvatar. This module optimizes the http requests and the management of the access tokens.

The back-end provides API data for generating the user avatars. The data object needed for generating the visualization was calculated every time the application displayed an avatar. In order to select the songs for the presented time period the server iterates once through all members of the recentPlayed array (see Table 3.2) in the User document (see Figure 4.13). The played_at timestamp in the Song object is used as a reference in the Data calculation controller (see Figure 4.14). All songs which are played between the start and the end of the presented period are pushed in a new array. The data in this array was used to calculate the position every avatar feature. How the data are calculated is elaborated in section 4.3.

Proactive Spotify users, can have extensive UserData documents. This type of documents can be a significant burden for the calculation module of the application. This leads to extending the waiting time for getting the avatar. With the intention to provide a better user experience the server side of the application uses a socket to directly communicate with the front-end. When the user creates a request s/he is redirected to a loading screen. After the calculation is done, the socket will send a message for redirecting the user to the avatar page (see Figure 4.14). In cases when the server has errors the loading page will provide a reset option for starting the whole process from scratch. This is considered to be a lightweighted solution. In further up scaling, the system should hold pre-calculated data and proper error handlers. The Spotify request controller on the server side was set to limit the number of calls per user. The limitation was set to five minuter per one call. The limit was created to prevent work overload of the server side and any misuse of the application.

The code for the back-end followed a MVC structure. It was deployed over Heroku while the database was deployed on mLab. Both of the
Back-end

services provided solid dashboard overview of the back-end performance. They were controlled through the terminal control, *GitBash*\(^\text{10}\). The Heroku dashboard logged Node.JS errors through detailed timeline charts. These charts contributed in the debugging of the server. The MongoDB was deployed on a replica server with two clusters. This type of deployment offered higher response rate and accessibility for the remote application over Heroku.

Through different relevant part of the application various loggers were placed to observe the user participation. These loggers are explained in subsection 3.3.1. All logs were accessible for observance on an url endpoint. This endpoint required a secret password in order to provide the records. The logs were presented in a raw JSON format. They were accessed in *Mozilla Firefox*, because this browser offers organized hierarchical view of the data.

For both of the prototypes the same back-end implementation was used with slight modifications in retrieving, parsing and calculating the Spotify data.

\(^{10}\)Git for Windows: [https://gitforwindows.org/](https://gitforwindows.org/)
5 | Results and Analysis

5.1 Questionnaire for the proposed concept

The questionnaire which was meant to validate the proposed concept for visualization was answered by 163 participants. The answers were collected through an online survey which was disseminated through various Internet channels.

The questionnaire consisted of 23 questions, some of which had sub-questions. The first set of 20 questions presented to the participants either a chart or an avatar feature. They were asked to match the presented item with one solution from a set of offered answers. A sample of questions from the survey is provided on Figure 5.1. For each question there was a correct answer. For example the correct answers presented in Figure 5.1 are: (A) - Avatar 3; (B) - Chart 4; (C) - Avatar 3; (D) - Chart 1. This set of questions was meant to test if the Chernoff like mapping is understood by the potential users of musicAvatar.

The second set was consisted of 2 questions. These questions were similar to the previous set but they added complexity and context for the proposed avatars. Here the participants were asked to select a whole avatar based on a described scenario. For example: “In your opinion which avatar best represents a person that listened to a small amount of music (over the last day), from a variety of artists, and the music was high tempo in major mode?”. The question was intentionally left opened for interpretation and no instructions were given for the participants. The questionnaire expected for the participants to associate elements from the avatar with parts of the question. The task allowed the user to choose two answers and it provided a Likert scale to measure the difficulty of finding the answer. This part of the survey was meant to measure how intuitive the proposed concept is.

Finally the last part was a subset of questions that aimed to get an insight of the participants music habits. The answers were used to scale the avatar data presentation.

On Figure 5.2 is presented the percentage of correct answers from all users for the first twenty questions. The y-axis presents the percentage of correct answers and the x-axis the question number. The rectangles in the plot are used to indicate group of questions based on a body feature. As it
5.1. Questionnaire for the proposed concept

Figure 5.1: The figure presents samples of the questions which were used for the initial survey.
5.2. User study

The chart presents the percentage of correct answers for the first 20 questions. *y-axis* - % correct answers; *x-axis* - question number.

can be seen a significant number of correct answers was provided during the study. This indicated that the mapping is understandable and easy to follow.

On the other side the second segment of questions provided a very mixed distribution of answers. According to our sample of participants the questions were hard to understand. Additionally they claimed that they have limited knowledge of music which lead them to misunderstand the readings of the avatar. Most of them answered according to their perception without following the established rules from the previous part. This type of problem is recognized in subsection 2.1. This result indicated that the implementation of the prototype should provide detailed instruction for the usage of the avatar.

5.2 User study

5.2.1 First prototype

The first deployment of the musicAvatar took place from Monday, April 9th until Sunday, May 20th. The users where recruited through an online campaign over social media. Additionally among the users were also participants from the questionnaire who expressed interest in the avatar. Overall 43 participants registered for the first prototype. Upon registering each user received an email with instructions how to use the avatar. Throughout the study the testers received two more emails. The first was intended for the
inactive users to visit the application, the second promoted the new feature myDay.

Figure 5.3 presents the users activity during the study period. The y-axis presents the number of users/views while the x-axis is the days. The gray curve shows all views of the application while the black curve represent the amount of unique users. The spikes in the chart correspond to the promotional activity which were conducted during the user study period. As it can be seen besides the spikes the users who were active on musicAvatar mostly opened the application once per day. This is indicated by the similarity of values for both of the curves presented on the chart. The values of the curves points out that the application was not actively used by all 43 participants.

Further analysis shows that 18 participants used the application more then once. Figure 5.4 presents a detailed view of the overall user visits throughout the whole evaluation period. The y-axis holds the time of the day while the x-axis is the days. Every element in the chart presents an user activity. The triangles are used to label cases when the application didn’t fetch any new records for the user while the circles present the opposite. As we can see the triangles dominates the chart, which implies that in most cases the visitor didn’t saw any avatar. A more detailed elaboration of this phenomena is presented in Table 5.1. As we can see 25 of the users never saw an avatar, from which 4 opened the application more than once. The table also shows that there is a core group of users that actively engaged with musicAvatar. In this case, the immediate time frame of two hours turns into an limitation for most of the users. The initial idea for the first iteration was for the users to engage with the application while they are musically active or freshly after they have listened to music on Spotify.
This challenge set a requirement to rethink the presented time period. As a possible solution is to extend the presented time frame in longer fixed periods. This approach was applied in the second iteration of the avatar. Something which is worth to note is that the analysis of the overall Spotify activity showed that most of the inactive users were not very engaged on the music platform as well.

On Figure 5.5 is presented the average displayed state for each user that saw an avatar. The \textit{y-axis} holds the threshold for every avatar feature and the \textit{x-axis} is for the users. The data for these charts is from the Table 3.4. The mapping of the avatar can be seen on Table 4.1. From the figure it can be concluded that most of the users saw the same state for each of the features as their standard deviation is most of the cases within the scope of the threshold. In some cases the users had a significant variation for some of the features, but generally the avatar features appeared to be static. This could be considered a reason for the low engagement from the users. An overall analysis of the music danceability for all 43 participants, shows that the sample of users were mostly covered by two thresholds (see Figure 5.6). This indicates that although the changes of the avatar are noticeable the data itself appears to be static, which contributes for the monotonous display of the avatar.

All the grades provided by the Likert scale are displayed on Figure 5.7. Almost all of the participants who saw an avatar provided an opinion. As it can be seen most of the users had a positive impression for the avatar. The results provided guidance for selecting users for the interview. Users “113”, “217” and “ell” agreed to participate in a short interview. “217” stated that the avatar reflected his music habits. More particularly a switch from
First prototype

Figure 5.5: The figure holds charts for the the average value of the presented music features for all 18 participants who saw an avatar; $y$-axis - feature threshold, $x$-axis - users.

Figure 5.6: The figure presents an overview of the average danceability for all 43 registered users of the first prototype; $y$-axis - danceability threshold, $x$-axis - users.
Table 5.1: The table presents the number of opens and views of avatars for each of the 43 registered users during the first iteration of musicAvatar.
First prototype

Figure 5.7: The chart contains all feedback values given in the Likert scale during the first iteration.

less to a more danceable playlists. The participant also claimed that the avatar seemed easy to understand. “ell” noted that the presented avatars didn’t match their perception of music danceability. They recommended providing a tooltip for the avatar since the prototype required memorizing the data mapping. Both “ell” and “217” had an overall good impression of the concept although they didn’t got any meaningful insight.

Participant “113” was the most active user of the application. Similar as “ell”, the participant seemed that the presented danceability is off their perception. Because of the threshold issue mentioned above, this participant saw only two states of the arms. In their words the eyes and the mouth stated the obvious and didn’t provided any additional value. Still this participant stated that the avatar would be an useful tool to share his activity with friends. Due to the static appearance of the avatar, the visualization seemed boring and didn’t provide any deep insight for “113”. That is the main reason why this participant finds a “low value” in musicAvatar. Their response on the liker scale was based mostly on how the ascetics of the avatar reflects the current mood. For further iteration they recommended in reconsidering the wording of the instruction to be more simplified and straightforward. Although the first prototype didn’t match “113” expectation, the user was very found of the general idea.

All interviewees said that the avatar seems boring as a standalone application but they would be interested to see the concept as part of an ecosystem. Additionally it seemed that all participants were not found of the presented time frame. For them it seemed very immediate and it didn’t provide any broader perspective. myDay provided a solution for this obstacle since it offered a possibility to compare different periods of the day. This type of presentation also allows for the users to refresh their memory on past activity.

The interviews and the overall feedback was considered to be helpful
Second prototype

Figure 5.8: The chart shows the amount of visits during the second implementation of the musicAvatar. y-axis - users/views; x-axis day of the year. The gray curve shows all views of the application while the black the amount of unique users.

and constructive. After a brief analysis some initial challenges were addressed in the second prototype.

5.2.2 Second prototype

As a result of the first user study, the second prototype expanded the scope of presented features. This was intend for the avatar to appear less dull and more meaningful. The presented time period was also reconsidered and it was extended for a whole day. Additionally the application presented seven avatars to cover a period of a whole week. This allowed for the user to compare its activity across the week. Finally the avatar included a tooltip (see Figure 4.12) which served as an instruction for reading the presented data.

The second user study lasted from June 30th, 2018 until Sunday, July 20th. This study was conducted over 10 participants which were recruited over the Internet. In Figure 5.8 is presented the total and unique visits of the application. For this iteration of the prototype it is worth to note that the avatar was updated on daily basis which means that the user does not have an incentive to open the application more the once per day.

As a continuing behavior from the previous study, the participants listened to mostly playlist with various artists. This resulted for majority of the users to see the avatar eyes in their highest state. This even includes the returning “113” who said that they will switch from listening albums to playlist. As it can be seen from Figure 5.9, most of the user tend to stick to songs with similar music features. Although in this case there is significantly larger standard deviation that covers all thresholds, expect for speechiness, the users in most cases fell under one or two thresholds. Again resulting into display of similarity between each presented avatar for the user. Most
Figure 5.9: The figure holds charts for the the average value of the presented music features for all participants in the second iteration of the musicAvatar; y-axis - feature threshold, x-axis - users.
of the users saw the moth only in the lowest state. Speechiness presented a problem even in the initial phase. As mentioned in section 3.4 this data appeared to be inconsistent with the described behavior in the documentation.

This version only provided comment section for sharing the user opinion. As such it revived a very small scope of opinions. Most of them were aimed towards the mismatch of speechiness and instrumentalness.

After the user testing an interview email exchange with some of the participants was conducted. User “113” stated that this version seemed more promising but the issue with data being off their perception continued to exist. User “121” and “217” had a positive impression of the visualization. According to them, the avatar generally presented their music behavior over Spotify, but it didn’t triggered any self-reflection or provided meaningful.

5.3 Discussion and Future work

Spotify was chosen as a popular platform which is quite accessible and relevant. It seemed as an appropriate solution for finding testers. The thesis never meant to explore Spotify usage. The general idea was to put into context the proposed concept by using passively collected data from Spotify and returning back meaningful insights for the users. The research has shown that a significant sample of the users of both of the prototypes were not very interested to track their music habits in order to gain a personal insight. In general most of the users were not very active on the platform. As such the users were not able to articulate any constructive ideas for the improvement of the avatar. This lead for the research to relay on the quantitative data collected from the questionnaire and the user logs. The challenge of finding the proper users is also mentioned in [van Berkel et al., 2015]. It is claimed that the issue of finding users to be engaged during long tests is one of the main challenges of the QS movement. Overall the few proactive participants in the user study claimed that they are intrigued by the idea of displaying their data in an avatar. They were supportive of the proposed concept and constructive in their comments.

The proposed concept for visualization was mostly understandable and easy to follow. Throughout the initial questionnaire and the user studies the participants were able to identify and interpret the data mapping of the avatar. It is worth to note that the users required instructions in a form of a user manual or a tooltip as an introductory for the avatar mapping. Still they were able to differentiate between all the states of the avatar features based on their visual perception. The project managed to deliver an avatar that uses facial expressions for data visualization.

1This is somehow to be expected, as the focus of QS is meant to be personal and cannot necessarily be easily generalized in a way that is relatable to multiple users.
5.3. Discussion and Future work

The research finds mismatch between the user perception and the presented data. In many cases the participants complained that the provided data from Spotify does not correspond to their music activity. As mentioned in the Limitations (see subsection 3.4.2), instrumentalness and speechiness seemed that they do not correspond to their description in the API documentation. This could have led to wrongful display of the music activity.

musicAvatar tried to provide unified visualization for all participants. As it was established in the interviews, some participants claimed that the avatar precisely corresponds to their activity, while some disagreed with the provided states. From the standard deviation in Figures 5.5, 5.9 and 5.6 it can be seen that most of the participant listened to very similar songs based on their music features. It could be said that they are stuck in a cluster. Although in their perception the music has many different specifications, the broad thresholds was not able to display the variation of music genres. This is why most of the users only saw one or two states for each avatar feature and thought that the avatar is mostly static. The challenge could be solved with two approaches. The first is to create a dynamic threshold individually for each user. Adjustable thresholds could capture each variation in the data. For example some users will have more sensitive thresholds while others will have a broader scope of values. The second approach could tackle the issue by extending the number of possible positions for each avatar feature. This could allow for the visualization to hold more thresholds. It is worth to mention that both solutions are not mutually exclusive which means they can be combined together.

Interesting point is that the music features provided by Spotify are generated by an algorithm. It can happen to be that although some participants claimed that they have listened to vast variety of music, the machine recognized that music as similar. Through the study nobody from the users mentioned Spotify as a stakeholder. One of the intention of the avatar was to present how the user is seen from the company perspective. This is one of the aspects where the visualization failed to provide meaningful insights.

The research showed that presenting the avatars for different time frames as an animation or displaying them together could serve as a good practice. This approach allows for the users to cross compare their activity through a period of time.

As an interesting suggestion from the user study is to use the avatar as part of Spotify. Adding the avatar as part of that platform could provide more context and insight. Further implementations of the project could take into consideration the use of the visualization in a supportive role as part of the user interface in broader digital ecosystem. Additionally as suggested the avatar could be used as a communication tool among users. Imaginably the visualization can communicate the users activity among their contacts.

The current implementation established itself as a measuring tool. Al-
though this could prove its usefulness, the initial goal of the visualization was to provide meaningful insight from QS data. In hindsight the research should have spent more time on understanding the user anticipation. Future implementation of the avatar should firstly conduct a qualitative exploration of the user interest before mapping the avatar. This knowledge could deliver guidance for the user needs and expectations.

The next challenge for the visualization would be to test the avatar library with different set of data. This exploration could show if music data was a big step for the avatar or there is a fallacy in the overall approach.

5.4 Reflection on the Research Questions

5.4.1 How to create an avatar based on the Chernoff faces technique for data visualization?

The overall project managed to deliver a working library that generates avatar based on a data input. The avatar uses facial expressions to map data. The project implementation uses Chernoff’s idea to hold information in the avatars face. Throughout the study the users were able to read the presented data from the facial expression.

The designing process has shown that in order for the visualization to achieve its goal the changing of states for each dynamic feature of the avatar must be visually noticeable. The changing features should also follow logical progression for the purpose of establishing visual rules of the displayed state. This will make the avatar easy to understand and it will simplify the instruction manual.

5.4.2 How to visualize multivariate Quantified Self data using an avatar?

The avatar uses the eyebrows, eyes, mouth, ears and arms to present data. This elements were predicted to change their state based on the data input. Adding dynamic elements to the avatar could broaden the scope of presented information. For example the initial mockups predicted the use of the avatar’s hair as a placeholder for data. The implementation showed that the visualization approach can be adjusted for different type of records as long there is a clear differentiation between the dynamic and static body features.

The avatar needs additional instructions for the user to link the face features with the presented data. The study has shown that there are cases when users perceive the avatar as a whole. Still in most of the cases the users were able to follow the instructions and understand the displayed state.
Different data can be linked to each avatar feature, but it would be a good practice to explore user's perception before deployment. This could tell which information works best for every dynamic body element. This could make the avatar design more intuitive.

### 5.4.3 How to visualize passively collected personal data in an avatar?

The work has shown that there is interest among users for their own passively collected personal data. The records used for the visualization were not envisioned as QS data. This type of data can hold overwhelming amount of information. Because of that, it needs to be interpreted for the user consumption. Before developing the avatar an accurate exploration should be made to understand the meaning of the data properties. The insight could provide solid direction for the data mapping. This step is crucial in offering meaningful visualization. Extra percussion needs to be made for the wording of data description. Unclarity could contribute to misunderstanding of the avatar.

### 5.4.4 How to provide meaningful insights for the users through visualization of their personal data in an avatar?

The study did not include many examples where the avatars delivered meaningful insights for the users through visualization of their personal data. This can be explained with the user claims in the comments as well as in the interviews which stated that the avatar is presenting obvious information. They were able to recognize their activity within the avatar, but they were already aware of their actions. Additionally only one user was active in tracking their music activity on their own, even before the study. The pool of users was either inactive or they listened to music with no particular interest in analyzing their habits. This means that the users were not seeking for any insight in general. The study has shown that some users were able to recognize their activity within the avatar's displayed states. Further implementations of the musicAvatar should firstly accurately explore the users needs before mapping the data into the avatar. This could provide a direction of generating an avatar that offers meaningful insights for the users.

### 5.5 Conclusion

This research proposed a concept for visualization of QS data using avatars. Based on the concept a prototype was developed and implemented online. The prototype was tested in a user study with two iterations. The lessons learned from the first iteration were reflected in the design of the second prototype. Through the study the the project faced different challenges in
5.5. Conclusion

terms of user engagement and provided feedback. The whole thesis offers a detailed analysis of the results from the user testing. Based on the analysis it delivers a discussion and proposes future steps for research. Furthermore this research provides comprehensive overview of the designing and implementation for both of the prototypes.
Bibliography


Bibliography


Bibliography


A | Mockups
Avatar mockup's
AvatarBot v.1

- Eyes: 2 dimensions, 9 positions (structured)
- Mouth: 2 dimensions, 6 positions (structured)
- Arms: 1 dimension, 3/5 or 360/180 positions (structured)
AvatarBot v.2

eyes. 2 dimensions. 9 positions (structured)

arms. 1 dimension. 3/5 or 360/180 positions (structured)

mouth. 1 dimension. 5 positions (structured), 3 (unstructured)

eyes. 2 dimensions. 9 positions (structured)

arms. 1 dimension. 3/5 or 360/180 positions (structured)
BigFootBot v.1

- **Eyes**: 2 dimensions, 9 positions (structured)
- **Arms**: 1 dimension, 3/5 or 360/180 positions (structured)
- **Mouth**: 1 dimension, 5 positions (structured), 3 (unstructured)
- **Eyebrow**: 1 dimension, chaos

```
eyes. 2 dimensions. 9 positions (structured)
arms. 1 dimension. 3/5 or 360/180 positions (structured)
mouth. 1 dimension. 5 positions (structured), 3 (unstructured)
eyebrow. 1 dimension. chaos
```

- **Eyes**: 2 dimensions, 9 positions (structured)
- **Arms**: 1 dimension, 3/5 or 360/180 positions (structured)
- **Mouth**: 1 dimension, 5 positions (structured), 3 (unstructured)
- **Eyebrow**: 1 dimension, chaos
BigHead v.1

- Eyes: 2 dimensions, 9 positions (structured)
- Eyebrow: 1 dimension, less chaos
- Mouth: 1 dimension, 5 positions (structured), + chaos (unstructured)
- Ears: 1 dimension, 2 positions (structured)
- Hair: 1 dimension, 3 positions (structured)
- Arms: 1 dimension, 3/5 or 360/180 positions (structured)
NormalHead v.1

- Eyes: 2 dimensions, 9 positions (structured)
- Eyebrow: 1 dimension, less chaos
- Ears: 1 dimension, 3 positions (structured)
- Mouth: 1 dimension, 3 positions (structured), + chaos (unstructured)
- Arms: 1 dimension, 3/5 or 360/180 positions (structured)
- Hair: 1 dimension, 3 positions (structured)
B | Visual Perception Questionnaire
Data Visualization Survey

My master thesis project aims to explore ways to visualize different types of data.

NOTE! This questionnaire is meant to test and validate some visualization concepts. Please answer the questions with the option(s) that you consider to be the most appropriate. This will allow me to evaluate and improve the proposed concepts.

For any questions related to this survey or the overall work, please contact stefan_aleksik@live.com

* Required
1. In your opinion, which avatar below best describes this chart? *

Mark only one oval.

- Avatar 1
- Avatar 2
- Avatar 3
- Avatar 4
2. In your opinion, which avatar below best describes this chart? *
Mark only one oval.

Avatar 1

Avatar 2

Avatar 3

Avatar 4
3. In your opinion, which avatar below best describes this chart? *

Mark only one oval.

- Avatar 1
- Avatar 2
- Avatar 3
- Avatar 4
4. In your opinion, which chart below best describes this avatar? *

Mark only one oval.
5. In your opinion, which chart below best describes this avatar? *

Mark only one oval.

Chart 1

Chart 2

Chart 3

Chart 4
6. In your opinion, which chart below best describes this avatar? *

Mark only one oval.

Avatar 1  Avatar 2
Avatar 3  Avatar 4
7. In your opinion, which avatar below best describes this chart? *

Mark only one oval.

- Avatar 1
- Avatar 2
- Avatar 3
8. In your opinion, which avatar below best describes this chart? *

Mark only one oval.

- Avatar 1
- Avatar 2
- Avatar 3
9. In your opinion, which chart below best describes this avatar? *

Mark only one oval.

- Chart 1
- Chart 2
- Chart 3
10. In your opinion, which chart below best describes this avatar?

Mark only one oval.

- Chart 1
- Chart 2
- Chart 3
11. In your opinion, which avatar below best describes these charts? *

Mark only one oval.
12. In your opinion, which avatar below best describes these charts? 

Mark only one oval.

Avatar 1
Avatar 2
Avatar 3
Avatar 4
13. In your opinion, which avatar below best describes these charts? *  
Mark only one oval.

- Avatar 1  
- Avatar 2
- Avatar 3  
- Avatar 4
14. In your opinion, which charts below best describes this avatar? 

Mark only one oval.

- Charts 1
- Charts 2
- Charts 3
- Charts 4
15. In your opinion, which charts below best describes this avatar? *

Mark only one oval.

- Charts 1
- Chart 2
- Chart 3
- Chart 4
16. In your opinion, which charts below best describes this avatar? *

Mark only one oval.

Charts 1
Charts 2
Charts 3
Charts 4
17. In your opinion, which avatar below best describes this chart? *

Mark only one oval.

- Avatar 1
- Avatar 2
- Avatar 3
18. In your opinion, which avatar below best describes this chart? *

Mark only one oval.

- Avatar 1
- Avatar 2
- Avatar 3
19. In your opinion, which chart below best describes this avatar? *

Mark only one oval.

- Chart 1
- Chart 2
- Chart 3
20. In your opinion, which chart below best describes this avatar? *

Mark only one oval.

- Chart 1
- Chart 2
- Chart 3
21. In your opinion which avatar best represents a person that listened to a small amount of music (over the last day), from a variety of artists, and the music was high tempo in major mode? *

Mark only one oval per row.

A B C D E F G H

First choice

Second choice (please repeat first choice, if you only have one)

22. How difficult was to decide on your choice(s)? *

Mark only one oval.

1 2 3 4 5 6
Easy, fast
Difficult, took more time

23. Can you elaborate more? (Write N/A if you don’t want to) *

__________________________________________
__________________________________________
__________________________________________
24. In your opinion which avatar best represents a person that listened to a high amount of music (over the last day), from only few artists, and the music was low tempo in mixed amount of music modes (major and minor)?

Mark only one oval per row.

A  B  C  D  E  F  G  H

First choice
Second choice (please repeat first choice, if you only have one)

25. How hard it was to find the avatar? *
Mark only one oval.

1  2  3  4  5  6

Easy  Hard

26. Can you elaborate more? (Write N/A if you don’t want to) *

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
27. What amount of time would you consider to be high in daily music listening? *
Mark only one oval.
- 0-2h
- 2-4h
- 4-6h
- 6-8h
- 10-12h
- 12h+

28. What amount of time would you consider to be low in daily music listening? *
Mark only one oval.
- 15 min
- 30 min
- 45 min
- 1h
- 1-2h
- 4-6h
- 6h+

29. Is there any aspect of your music habits that you would be interested to monitor or track on a daily basis? (Write N/A if you don’t want to) *

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

30. Can you describe your music habits and interests? (Write N/A if you don’t want to) *

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

31. If you are Spotify user are you interested in using the avatar as a tester for the prototype? If yes please share your email
C | Privacy pages
musicAvatar 2.0 Privacy

The application extracts data from your Spotify account in order to generate the Avatar. It utilizes 3 different endpoints from Spotify:

- Get Current User’s Profile [GET /v1/me] [1]
  From this endpoint we extract your name, email and spotify id. This data is used only once to create a profile for the musicAvatar.

- Get Current User’s Recently Played Tracks [GET /v1/me/player/recently-played] [2]
  From this endpoint we follow your music history.

- Get Audio Features for Several Tracks [GET /v1/audio-features] [3]
  This endpoint compliments recent played music by adding danceability attribute for each song.

We care about your privacy. That’s why the app extracts only the needed information from your Spotify account. Additionally your music history is anonymised by not using the name of the artist and songs that you have been listening to, instead we only use the songs and artists id’s in order to differentiate the data.

Due to the way how the Spotify API is designed we need to make automatic calls on daily basis to cover your listening activity in the past period. An additional call is made when you open the app in order to provide the played songs between the last automatic update and the current moment.

musicAvatar logs your activity. The log data alongside your Spotify history will be used to evaluate the app for the purposes of the thesis.

If you want your data to be destroyed after the end of the app evaluation period (somewhere around end of June) please contact us and let us know.

If for whatever reason you would like to quit before the evaluation period ends, you can revoke the access of your data by the app on the following link (The app is called QuantifiedSelfMusic). Additionally write us to remove the data from our database.

If you have any questions or you want to report a bug please feel free to contact stefan.aleksik@live.com

Figure C.1: Privacy page, first implementation.
musicAvatar Privacy

The application extracts data from your Spotify account in order to generate the Avatar. It utilizes 3 different endpoints from Spotify:

- Get Current User’s Profile [GET /v1/me] [1]
  From this endpoint we extract your name, email and spotify id. This data is used only once to create a profile for the musicAvatar.

- Get Current User’s Recently Played Tracks [GET /v1/me/player/recently-played] [2]
  From this endpoint we follow your music history.

- Get Audio Features for Several Tracks [GET /v1/audio-features] [3]
  This endpoint compliments recent played music by adding danceability attribute for each song.

We care about your privacy. That’s why the app extracts only the needed information from your Spotify account. Additionally your music history is anonymised by not using the name of the artist and songs that you have been listening to, instead we only use the songs and artists id’s in order to differentiate the data.

Due to the way how the Spotify API is designed we need to make automatic calls on daily basis to cover your listening activity in the past period. An additional call is made when you open the app in order to provide the played songs between the last automatic update and the current moment.

musicAvatar logs your activity. The log data alongside your Spotify history will be used to evaluate the app for the purposes of the thesis.

If you want your data to be destroyed after the end of the app evaluation period (somewhere around end of June) please contact us and let us know.

If for whatever reason you would like to quit before the evaluation period ends, you can revoke the access of your data by the app on the following link(The app is called QuantifiedSelfMusic). Additionally write us to remove the data from our database.

If you have any questions or you want to report a bug please feel free to contact: stefan_aleksk@live.com

Figure C.2: Privacy page, second implementation.
D | Instruction pages
musicAvatar

musicAvatar is a web app that generates an avatar based on your recent played songs on Spotify. It is a part of a research project that aims to map out aspects of human behavior from personal data on an avatar. The project’s main aim is to contribute to the Quantified Self movement, by exploring different possibilities of visualizing data using ‘Chernoff’ face inspired approach. [GitHub repo of the project](https://github.com/yourusername/musicavatar)

How to interpret musicAvatar:

musicAvatar displays your Spotify listening activity for the last two hours. Based on the retrieved data it maps out different aspects of your recent music in the avatar’s eyes, mouth and arms.

A. The position of the eye pupils represents the “diversity” of music that you have listened to by measuring the variety of artists and songs.

The variation of songs moves the pupils horizontally; more to the right means that mostly unique songs have been played.

The variety of artist moves the pupils vertically; upwards indicates big diversity of artists.

For example the avatar shown above has listened to mostly repeating songs from the same artists in the last two hours.

B. The size of the mouth represents the amount of music you have been listening in the past two hours.

A small mouth indicates that the amount of music was low for the given period. A more open mouth means more time has been spent listening to music.

For example the avatar shown above has listened to a very low amount of music in the past two hours.

C. The position of the arms represents the danceability\(^* \) of the music which is being analyzed.

A low danceability is indicated by arms next to the avatar body; raised arms indicate higher danceability.

For example the avatar shown above has listened to music with low danceability.

\(^*\text{Wikipedia Chernoff face} \cite{1}\)

\(^{\text{2}}\text{According to Spotify, danceability describes how suitable a track is for dancing based on a combination of musical elements including tempos, rhythm stability, beat strength, and overall regularity.} \cite{2}\)

---

If you have any questions or you want to report a bug please feel free to contact stefan_ukolsk@live.com

Figure D.1: Instruction page, first implementation.
musicAvatar 2.0

musicAvatar is a web app that generates an avatar based on your recent played songs on Spotify. It is a part of a research project that aims to map out aspects of human behavior from personal data on an avatar. The project's main aim is to contribute to the Quantified Self movement by exploring different possibilities of visualizing data using Chernoff® faces inspired approach. Of maps of the project.

How to interpret musicAvatar:

musicAvatar displays your Spotify listening activity for the last seven days (week). Based on the retrieved data, it maps out different aspects of your recent music in the avatar's eyebrows, eyes, ears, mouth and arms.

A. The position of the eyebrows represents the popularity of the music you listened according to Spotify. The higher your eyebrows are positioned, the more popular the music you have been listening to in the past period.

In the example above, the eyebrows of the musicAvatar represents a day of listening to music with low Spotify popularity.

B. The size of the eye pupils represents the 'diversity' of the music that you have listened to by measuring the variety of artists. Small eye pupils represent small artist variety of artist within your music and bigger eye pupils represent larger variety.

For example, the avatar shown above indicates that somebody listened to mostly the same artists.

C. The 'spikiness' in the ears represents the use of real instruments in your music. Spiky ears represent music that includes a significant use of real instruments. Note that this can also include samples or as well. Flat looking ears represent music that most probably contains sounds that are generated using digitally.

The musicAvatar ears shown in the example above represents music with high use of real instruments.

D. The size of the mouth represents the amount of speech in your music. Smaller mouth represents music that is mostly instrumental and bigger mouth music that contains significant amount of lyrics.

For example, the avatar shown above has mostly listened to music with a lot of lyrics.

E. The position of the arms represents the positivity of the music which is being analyzed. A low positivity is indicated by arms next to the avatar body; raised arms indicate higher positivity (cheerful, happy...).

For example, the avatar shown above has mostly listened to music which is considered to be cheerful.

*Wikipedia page about Chernoff faces [1]

**In order to display a full week we need to collect data for 7 days. As a day passes, the data will be displayed on the app.

If you have any questions or you want to report a bug, please feel free to contact: stefan_aleksic@iisw.com

Figure D.2: Instruction page, second implementation.