département
Informatique

Université Claude Bernard Lyon 1

POM

Rapport de Projet
Outil de physicalisation de données

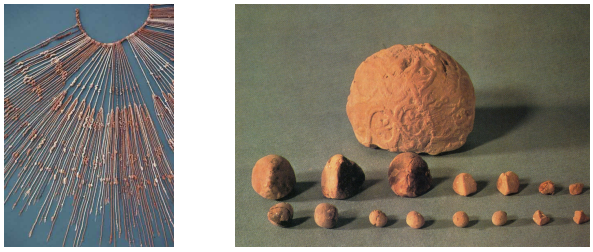
Abdelaziz SBAAI

Encadrant: Aurélien Tabard

2020/2021

1 Introduction

The concept of Data Representation existed for thousands of years, as a matter of fact, the earliest data representations were physical, dating to 5500 BC[1], stones, pebbles and clay tokens were used to represent information rather than using words. This shows for how long data representations helped us humans externalize information, support abstract thought and enrich cognition even before written language was invented.



(a) Quipus (b) Mesopotamian Clay Tokens

Figure 1: Examples of the earliest forms of data visualisations

Most data representations are digital nowadays, and that is of course due the very fast and continuous improvement of computer computation and display abilities, providing very rich representations. Also due to the natural ability of humans to perceive and understand visual objects. Consequently, the rising research area of Data Physicalization aiming to bring back physical representations to the surface and to free data representations from the limits of 2D displays.

This report discusses the result of a project that aims at the development of an application permitting the construction of a physical representation of personal Spotify listening history data.

[1] `http://dataphys.org/list/`

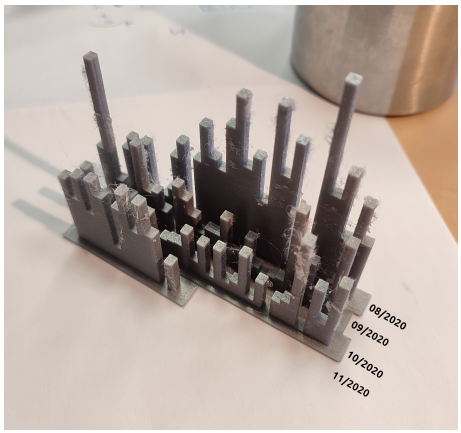


Figure 2: Physicalisation of my listening time on Spotify for each day over 4 months

2 Related Work

2.1 Music Visualisation

The process of studying and analysing music can be easily arduous, especially when dealing with large audio collections. Music research provides a solution to this problem through tools and algorithms that facilitate working with large audio sets.

> "Similar to other subdomains of the humanities, in recent years, digital methods became increasingly important in musicology to store, structure and analyse vast amounts of digitally available musicological data"[3]

Visualization plays a key role in music study, as it allows easier understanding of the data by uncovering connections between elements and aspects of complex music data. This lead to leading edge applications designed for different purposes, on from helping music professionals and researchers with interactive tools, to facilitating the communication of music concepts in a more intuitive manner to the broader public.

2.2 Quantified Self Visualization

Since a large spectrum of our lives is increasingly being covered by data, through the use of daily personal smart helpers such as phones, watches and even water bottles, ranging from our music streaming history, over daily diet and exercise logs, to all the different kinds of digital guidance we count on to keep track of our lifestyle. Thus the emerging visualization field, Personal Visualization for Self Reflection.

This field aims analysing the quantified self in form of personal data and reflecting on one's self in a personal context, D. Huang defines this latter as follows

> Internally, context could be "abstract artifacts", such as goals, skill sets, preferences, experience, etc. Externally,context could be either physical constraints (e.g., physical environments or devices) or social influence.[2]

By investigating constructive personal physicalization over few weeks, Thudt et al. [4] explored how building physical representations of an aspect one's daily, can make a change either helping maintain a good habit, improve mood or documenting bodily changes...

A good example of how integrating manually created physical representations of personal data, helped spark moments of deeper reflection and conversation.

2.3 Physicalization

> "Data physicalization has been further defined as a research area which examines how computer-supported, physical representations of data (i.e., physicalizations), can support cognition, communication, learning, problem solving, and decision making" [1]

Data physicalization shares a lot of ground with many disciplines and research areas such as information visualization, scientific visualization, tangible user interfaces, personal fabrication interfaces as well as graphic design architecture and art [1].

Existing before the invention of paper [2], physical representations have been used for analytical purposes, education, self-reflection and self-expression as well as for enjoyment and meaning [1].

Although digitally displayed visualisations are showing strong domination over physicalizations, the increase of how easy and cheap the creation of physicalizations in addition automation will eventually make physicalizations as interactive and tangible as today's visualization software

3 Physicalizing Listening data

The following diagram is an abstraction of what this project is about.

[2] `http://dataphys.org/list/mesopotamian-clay-tokens/`

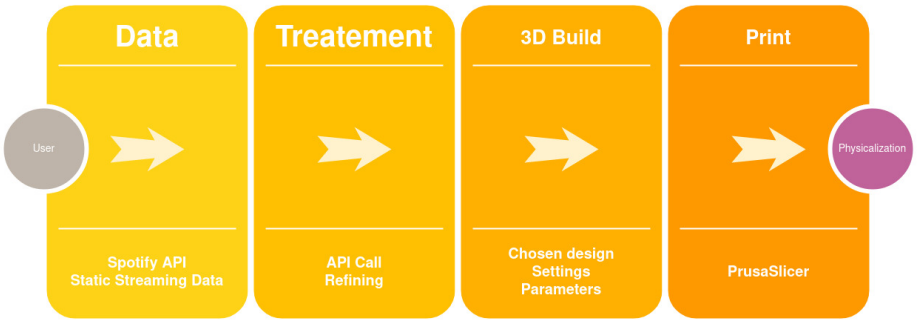


Figure 3: The physicalization pipeline.

3.1 Data Upload

First the user can input the data through one of the notebooks[3] depending on which data they want to use:

- **Static Streaming Data:** Spotify users can download their cloud stored data from their account dashboard[4]. The user will afterwards receive a .zip file that contains their streaming history for the last year, among other files with more information (playlists, email, followers ...). The user then can upload thier streaming history .json file to the notebook to start the physicalization process.

  The user can select a month among the ones that the file is about.

  The main problem with this approach is the waiting time that can be up to 30 days. Don't worry, here [5] you can find a test file.

- **Spotify API Data:** This approach requires less labor from the user. Users would only have to log in using their Spotify account. The notebook will then send queries to the Spotify REST API to get the needed data.

  The drawback of this approach is that the Spotify API doesn't provide a detailed nor long-term history endpoint, it only provides a "recently played tracks" endpoint that can only respond with the last 50 tracks, which is very little compared to the data we can get download ourselves.

3.2 Data Treatment

Data treatment consists of formatting the input data in a way that we can easily work with later.

[3] 1st notebook: `https://observablehq.com/@pillowinacoma/3d-viz-spotify-data` 2nd notebook: `https://observablehq.com/@pillowinacoma/3d-visualizing-spotify-api-data`

[4] `https://www.spotify.com/be-fr/account/privacy/`

[5] `https://raw.githubusercontent.com/pillowinacoma/printify/main/StreamingHistory2.json`

For the listening history notebook, the entered data is an array of objects show in each listening action, track and artist name,the time it ended at and the number of milliseconds played. The notebook proceeds to reduce this array to an array of days of the chosen month, each day is associated to the total milliseconds played during the day. A list of the months contained in the file is simultaneously created so the user can choose one to export.

For the second notebook, the notebook sends an request to the API to get the top 50 tracks, then a following API call to get the tracks features [6]

The "refined" data is now ready to be used to build the 3D object.

3.3 3D build

This part consists of using the prepared data to construct a 3D printable representation.

- **Streaming History** This notebook offers 2 designs:
  - **Bar Chart:** A group of cuboids [7] with each one's height representing the total listening time during a day of the given month. All the cuboids are standing on a rectangular base.
  - **Bracelet:** The more advanced design, consists of a ring (or part of a ring) composed of multiple segments, each segment's depth represents the time spent listening during a day of the month.
- **Top Tracks:** Shaped like a spiral stairway, this with each step composed of 6 segments, each segment's height represents a feature. Each step represents a track, tracks are ranked from most streamed to less streamed.

After the 3D object is built, the user can change the object's parameters (width, height, depth, size ...) then download it as an STL[8] file by clicking the button "Export". This file describes the surface geometry of the 3D object.

3.4 Print

When Downloading the STL file, The user can use any slicer software to create the physicalisation. In my case, I use a Prusa printer that comes with PrusaSlicer that creates a .gcode[9] file for the 3D printer. This file contains instruction for the printer controller that tells the motors how to move to make the print (speed, path, positions...) PrusaSlicer offers extra parameters (size, detail ...). The physicalization is created using the instructions from the .gcode file. The printing time depends on the size of the print and the detail.

[6] https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

[7] https://en.wikipedia.org/wiki/Cuboid

[8] https://en.wikipedia.org/wiki/STL_(file_format)#Use_in_3D_printing

[9] https://en.wikipedia.org/wiki/G-code

4 Implementation

Here are the tools and the technologies chosen for implementing the project

4.1 Observable

Observable [10] is a notebook platform founded by Mike Bostock and Melody Meckfessel in 2016 [11], made to explore data and code visually live in the browser. Coupled with JavaScript libraries (D3.js, Three.js ...), Observable's different features make it a great tool to experiment with data, create interactive visualisations. it can also be used make digital art and as a teaching tool especially thanks to it's beginner friendly interface and it's increasingly growing community.

Observable can be used as a social network like publication interface, as it allows users to publish their code and explore code published by other users. Notebooks can also be easily forked and edited which makes Observable even more useful as a learning tool.

All the code made for this project is contained in an Observable notebook.

Despite its strengths, Observable has few weaknesses that can make working with it harder, especially on bigger projects. Debugging complex code can be burdensome due to its limited debugger. Efforts to help with this issue have been made by implementing a dependency tree, but it has not yet been implemented on the newer version of Observable that contains some good features, the better text editor that offers more tools (better autocompletion, writing tools...).

4.2 Web based 3D frameworks

4.2.1 Three.js

Created by Ricardo Cabello (aka Mr.doob) in April 2010, Three.js is a JavaScript 3D library that can be used to create 3D lightweight Web animations, due to the advent of technologies such as WebGL.

Three.js is a very popular 3D rendering engine thanks to it's many assets. Is is considered easy to learn by many of it's new users since it abstracts low level 3D code into higher level, easier to understand concepts. Thanks to the large community, plenty of different examples are available online, in addition to third party plugins that can facilitate it's use further more.

4.2.2 A-frame

Web framework for building virtual reality experiences, it is built as an abstraction on top of Three.js. With it's html based entity component system make it is easier to deal with than raw three.js, especially while using a library like D3.js.

[10] `https://ObservableHQ.com`
[11] `https://observablehq.com/about`

Using a-frame simplifies creating 3D objects but in the same time gives full access to the Three.js API. this is particularly convenient for exporting objects to STL format as Three.js provides a built-in exporter.

The main downside to using a-frame is the camera controls, the default way to browse a scene in a-frame is to "walk" using ZSDQ keys. A better way could be to use "Orbit Controls" to turn the object around itself so we can have a easier understanding of how it would look like 3D printed, it's also more similar to how we would use the real 3D printed object.

4.3 Prusa 3D Printer

Founded Research was founded in 2012 by Josef Prusa [12], started as a startup and now one of the most influential companies in the 3D printing industry [13].

For this project I used the Original Prusa i3 MK3 model, It became first available in 2017, and is considered one on the best 3D printers of 2018 [14], It came with new features such as crash detection, power failure backup, better redesigned stiffer frame and a better documentation. Using this printer was pretty straight forward, the PrusaSlicer software take in a/multiple 3D file/s (STL, OBJ, AMF...) then we can modify the printing settings to satisfy our needs. We can then create the .gcode file that will be affiliated to the 3d printer. The software provides us with more details such as the estimated time in different printing modes, the amount of used filament and event the cost.The 3D printer then reads the .gcode file from the SD card[15], and starts printing.

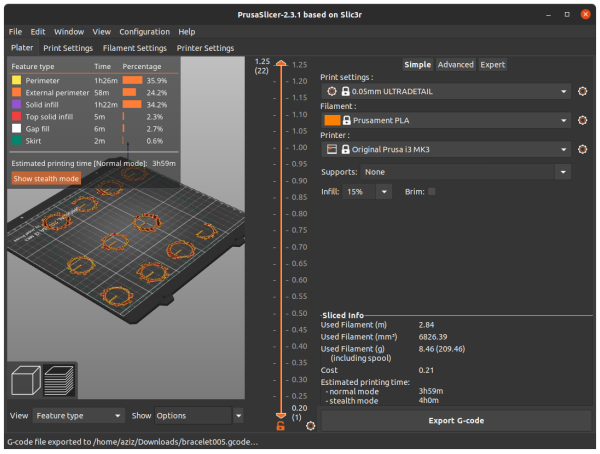


Figure 4: PrusaSlicer interface

[12] `https://www.prusa3d.com/about-us/`

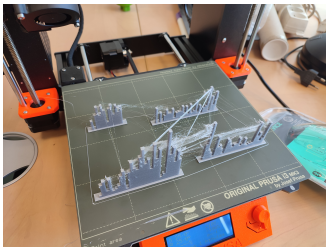
[13] Original Prusa i3 MK3S+ best 3D printer in 2021 according to ALL3DP

[14] `https://all3dp.com/1/original-prusa-i3-mk3-review-3d-printer/`

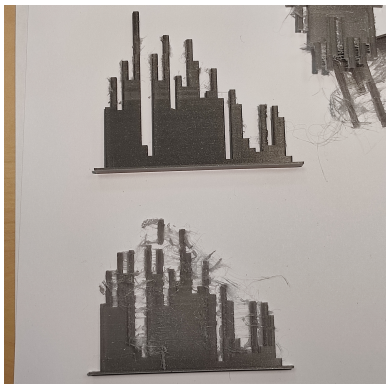
[15] There's a USB input that we can use, I am going to elaborate on my experience with it in the next paragraph

My experience with this printer was good overall, except for two main points

- **Printrun:** I did not own an SD card nor an adapter at first, so I tried to do an initial print using Printrun, the software for printing with the USB port. As a Ubuntu user, i could not find a package on the official Github repository [16], so running the source on my computer was necessary. After a couple of hours of trial and error, I could not run the source code, so i gave up and decided to buy a SD card adapter.
- **Webbing** The 3D printing quality can sometimes be unsatisfying, especially when printing thin detailed pieces. The filament left overs would form strings as the head moves around , this would cause the objects to entangle in a web in the case of multiple prints, which is hard to get rid of when dealing with thin fragile objects as they break very easily.



(a) webbing



(b) webbing effects prints differently

Figure 5: Webbing causes breakage when prints are thin

5 Design Choices

5.1 Stacking

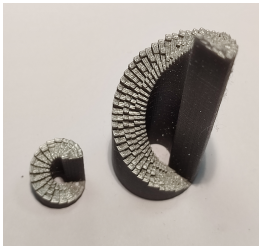
Stacking is a physicalization strategy that consists of creating segments of the physicalisation and putting them together as a stack. The pieces of the physicalisation can be arranged in different orders creating a more dynamic physicalization experience. More pieces can be created as time goes by in the case of continuous data, with is for our project.

The Touching Air [17] project is a good example of using stacking in a physicalization, each piece of the necklace represents Sheffield's air quality during a segment of time.

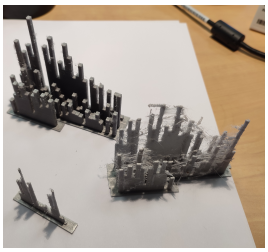
[16] https://github.com/kliment/Printrun/tree/master#ubuntudebian
[17] http://dataphys.org/list/touching-air-necklace-shows-air-pollution/

5.2 Physicalization Design

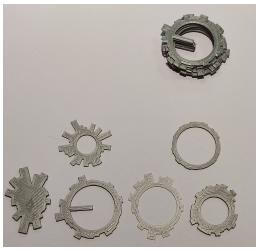
This section is a discussion of the design choices and every design's advantages and flaws.



(a) Spiral Stairway



(b) Bar Chart: first print (bottom) to last (top)



(c) Bracelets different settings

- **Spiral Stairway:** Although it's aesthetic advantage, this design has many downsides. First it is static, the result is one final object. The second problem with this design is that for each step, there is no point of reference, so it is hard to understand what it represents. This design makes for a descent printing quality (the most solid of the 3). the print time is average.
- **Bar Chart:** Unlike the previous design, Bar Chart is a more dynamic design, the user can assemble a number of charts to compare easily. Creating more charts over time would make for a growing physicalisation. However, printing bar charts was the most difficult and took the longest[18]. Parameters that would make for stiffer object, is required due to the fragility of individual bars. Webbing can make this even worse as motioned previously.
- **Bracelet:** This design offers stackability (dynamic), took the less time and effort to print. The main defect is that months have different day count, for example comparing between a 28 and 31 day month is not very convenient.

6 Conclusion

This project explores the different steps taken to physicalize personal Spotify data for self-reflection. To conclude this report I would like to talk about points I would like to explore further.

- Creating a hosted server to avoid keeping the API keys public
- Elaborating on design and stackability by creating handles for the models
- Exploring with printing text (album/track titles, atrist names...)

[18] almost 20 hours for the best prints I had

- Physicalization of track data [19] as I was trying to do this at first given the deep AI analysis Spotify offers for tracks, but creating complex visualization with thousands of components wasn't easy on my computer.
- Printing artist picture and album covers [20]
- Pushing 3D printing parameters further (Physical texture, multicolor prints ...)

7 Acknowledgments

I would like to thank Mr Aurélien Tabard for giving me the opportunity to explore this interesting project and offering all the help and guidance.

References

[1] Pierre Dragicevic, Yvonne Jansen, and Andrew Vande Moere. Data physicalization, 2020.

[2] Dandan Huang, Melanie Tory, Bon Adriel Aseniero, Lyn Bartram, Scott Bateman, Sheelagh Carpendale, Anthony Tang, and Robert Woodbury. Personal visualization and personal visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 21(3):420–433, 2014.

[3] R Khulusi, J Kusnick, C Meinecke, C Gillmann, J Focht, and S Jänicke. A survey on visualizations for musical data. In *Computer Graphics Forum*, volume 39, pages 82–110. Wiley Online Library, 2020.

[4] Alice Thudt, Uta Hinrichs, Samuel Huron, and Sheelagh Carpendale. Self-reflection and personal physicalization construction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[19] See Spotify track analysis endpoint `https://developer.spotify.com/documentation/web-api/reference/#endpoint-get-audio-analysis`

[20] You can see this video for a colored 3d printed idea`https://youtu.be/4FvtQOHzus4`