Data Visualization for game popularity on Twitch/Steam platforms

Nuno Gonçalves
nuno.motagoncalves@epfl.ch

Baran Nama
baran.nama@epfl.ch

Abstract

The aim of this project was to create a website to provide *gamers* and *streamers* with a visualization of games' popularity across several platforms - namely Steam and Twitch. The goal was to give a way of comparing the amount of concurrent users (either players or viewers) associated with a given title. As a means, we used *D3* and *bootstrap* to layout the web-page, creating separate representations for each platform and allowing for user-interactions with the elements we created. Being able to synergize a customizable experience with an uncluttered and simple approach was one of our priorities - and we think the final product reflects that. In the end, there are still a lot of factors that need improving but we consider it to be a good foundation to build upon.

1 Introduction

As *gaming* being a category increasingly mainstreamed over the web - and as a personal interest of ours - we decided to create a tool for both content creators and consumers within this category (streamers and players, respectively). To do this, we first looked at the information provided by two of the biggest platforms on the market, Steam and Twitch. Despite many complex statistics some APIs and websites may provide, simply being able to compare the top games on both platforms and their concurrent users' historical data could provide interesting insights on which games to play or watch.

Having this in mind, the goal of our project was to develop an interactive web visualization that, despite being simple, would allow for a customizable user experience (UX). As a means to an end, we used D3 and bootstrap to design our visualization and align the components within the webpage, inherently making it responsive.

In this process book, the sections were organized as follows: *Data Collection* 2 briefly describes how we collected the data we used; *Dataset Description* 3 goes a bit into more detail into how the data was organized and stored for use throughout the project; *Visualization* 4 focuses both on the theoretical challenges behind our design (4.1) and on the approach we took to overcome them (4.2); *Group Evaluation* 5 simply states how work was mainly divided among ourselves; and *Conclusions* 6 reflects our final thoughs and considerations about the project.

2 Data Collection

Our data was collected from a mix of parsing/filtering on raw text files (dumps of databases) and data-scrapping - to complement the data. Our plan was to further enrich what we had with information on games' genres, provided through Steam's API, but we ended up not having time to implement this feature. Although we cannot make public the initial data we used to create our datasets, all the files inside the *data* directory represent the information handled by our website, formatted in *CSV*.

3 Dataset Description

We separated our data within two separate sets of datasets: *'data/steam'* and *'data/twitch'*. Within each one of these sets, there exists a single file for each day of the information we collected. The *metadata* for these files (or their *schema*) is consistent within the set we are referring to - either Twitch or Steam - but vary slightly from one to the other. For our purposes, we decided to use *flat tables* for our dataset types and format them using CSV.

3.1 Steam

Steam's daily datasets contain information from about 9000 different games. For each entry, they contain information about:

- **Date** - string date representation, common across all entries in a single file
- **ID** - unique identifier for each entry
- **Name** - title of the game
- **Daily Peak** - daily peak of concurrent players
- **All-time Peak** - all-time peak of concurrent players (even before we started collecting data)
- **Hourly Count** - several columns ([0-24]H) representing the amount of concurrent players at every hour

Although both the *Date* and *All-time Peak* columns end up not being used in our preliminary approach, we though we should keep them for further iterations on the project - either to ease exporting the data to different formats or to provide possibly interesting extra information, respectively.

3.2 Twitch

Twitch's daily datasets are very similar to Steam's but with a small difference: they do not contain the *All-time Peak* data. Regardless, they still have:

- **Date** - string date representation, common across all entries in a single file
- **ID** - unique identifier for each entry
- **Name** - title of the game
- **Daily Peak** - daily peak of concurrent players
- **Hourly Count** - several columns ([0-24]H) representing the amount of concurrent players at every hour

Bear in mind that the *ID*s of games on Twitch match the *ID*s of games on Steam (for the games that are common between them).

3.3 Data Filtering

The main problem while taking care of the data was handling *text encoding*. Since some of the games had weird characters in their names, we forced *UTF-8* which, while sacrificing some information, would still allow for a wide range of characters. Regardless, since we are using *ID*s even if certain games' names match after this conversion (or become ill-formatted) we can always differentiate between them and/or recover their information from external APIs, like Steam's. All this was done using external python scripts, using the *Pandas* library to manage a columnar-oriented format.

3.4 Data Enriching

To further improve on our current data we could again use Steam's API, to fetch each game's genres. More than comparing games across platforms, we would allow users to compare genres - but in the end we did not have enough time to finalize the idea.

4 Visualization

Although taking care of the data consumed a bit of our time, the biggest challenge was to design a way to visualize what we wanted, with the remaining information.

4.1 Challenges

We had to find a way for people to mainly do two things: (i) be able to see a comparison between, at least, two games regardless of platform; and (ii) be able to select which games to compare. Of course, to achieve a pleasant result we had to find a way to display as much of a games' information without cluttering the user's view - and hence reducing clarity or intuition.

4.1.1 Simplifying Data

Before attempting to show anything about a game, we should limit the amount of data we show to the user. Here, we went with the straightforward approach and filtered only each platform's top games. The actual amount of games would depend on several other factors, which we will detail later in this section, but in the end we settled on the top 18.

4.1.2 Information Layers

Even after reducing the amount of data, presenting it all at once would be an overload for the user. Here we defined what we call an *information* *layer*, as being "an abstract collection of data that requires, at most, a single interaction to be viewable" (e.g. a drop-down box's contents).

Once we decided upon the least amount of information to show, reducing the *information per layer* would increase the amount of *layers* (and vice-versa). Following the *three-click rule*, commonly used in web or mobile application development and described by some authors as a good usability guideline (1), we narrowed the *information layers* down to two (since one operation is required to navigate between levels).

4.1.3 Exploiting the system

Still in terms of usability, passively displayed information does not necessarily increase the *learning curve* of an application, it can actually be used as a good attention driver (2). While still following the rules described before in this report, we can enrich each *information level* with, for example, on-hover displayed *tool-tips* or *informative panels*.

4.1.4 Brushing

To represent a game's information, *brushing* was one of the first things that came to mind. Since we needed users to select a certain subset of games, from the displayed ones, we just had to find a way to implement an *on-click selection* to allow that to happen.

4.1.5 Navigation

Now that we established we can use several *information layers*, we have to ascertain what information they contain exactly and how we can navigate between them.

Much like *node-link diagrams*, we thought that an intuitive way of visualizing a game's genres was to create a connection to its node. This method, however, for the kind of information we want to display could easily clutter the screen and make it harder for the user to select just the items he wanted. Instead, we though a *zoom-in & zoom-out* navigation, inspired on the reading of a publication on *information visualization* (3), would be perfect. Zooming-in on a game would show its genres, bringing the user to the second *information layer*, and zooming-out would bring him back again to the first layer.

Allowing continuous zooming operations in this context (e.g. zooming-in on genres would bring the user to an even deeper layer containing that genre's most popular games) crossed our minds but this would exceed our own limit of two *information layers* to access *necessary* information and could lead to a maze-like process where the user would no longer be able to easily navigate back to where he wanted. However, because we did not have the data to fill in the second *information layer* with games' genres, we thought we could allow this continuous zooming as a timeline browser (the further a user zooms in, the further back he browses our data's history).

4.1.6 Visual Queues

Regardless of layer, all information from different sources should be visibly well-delimited. Following Gestalt's principle of *proximity*, Steam and Twitch should form their own separate clusters of games. As for *similarity*, when a certain game is selected its information should be visibly related to it. Finally, by simultaneously using *symmetry* and *enclosure* would allow us to join both clusters, their relationship and other factors into a single element - without becoming visually displeasing or confusing.

Our goal now was to define a *mark* to represent the canvas of our information. More specifically, we wanted to define areas to enforce component separation. This separation now, however, was not between the several elements of the visualization but between the visualization itself and the rest of the page (although it ended up working both ways).

4.1.7 Other Considerations

Some sections of the screen tend to be more focused than others, in terms of attention. Consciously or unconsciously, people seem to ignore certain elements on web-pages (much like in games) if they deem the information is unnecessary. Because of this, we decided to shrink our visualization to bootstrap's default content width.

4.2 Design

With clear goals and criteria to follow, we went through many iterations of the project until we settled on a design that aligned with our goal. We couldn't quite find any template that would fulfill our particular needs, so we went ahead and developed our own, unique visualization.

4.2.1 Building Blocks

A good way to keep things visually clean and organized is to use geometrical shapes. Evenly faced geometrical figures, specifically, are aesthetically appealing to create symmetrical images. Although a square would be an alternative we decided to use an hexagon as our building block, as shown in *Figure 1*.

Figure 1: Hexagonal shape

This would give our visualization a more hive-like look - which we though would be interesting.

4.2.2 Game Clusters

Taking from our base hexagon choice, we developed each platform's game clusters to follow the same concept both at a micro and macro level (*Figure 2*).

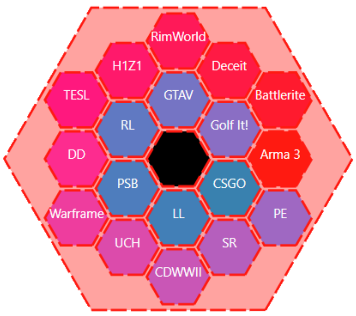

Figure 2: Hive cluster

We didn't want to enforce the notion of *absolute ranking*, so we hid that information from the first layer. Instead, we gave an idea of *relative ranking* based on color proximity, which we though would be more useful since we want to allow comparisons.

4.2.3 Combination vs. Isolation

Now that we had the games' clusters designed, we had to find a way of creating an overall shape keeping the symmetry (*Figure 3*).

With the resulting view it should be perfectly clear to the user that there are two separate groups of items, while suggesting that they should be similar in terms of information.

4.2.4 Optimizing Space

Although it seems like we are following the right path, we still had to find a way to place the items' relationships in view - ideally within the limits of a combined viewport with the previous two clusters. Following the idea of complementarity, we could add an additional shape in between the clusters (*Figure 4*).

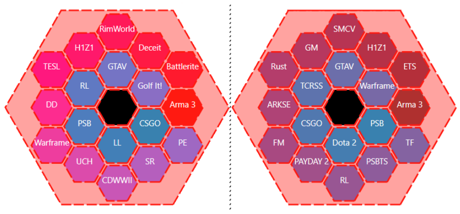

Figure 3: Combined symmetry

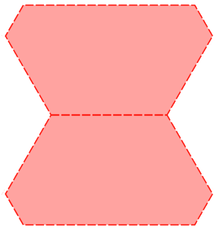

Figure 4: Connecting shapes

This would keep overall symmetry but provide empty space for additional information.

4.2.5 Stitching Together

Bringing all of the previous elements into a single one we would get our final overall visualization's canvas (*Figure 5*).

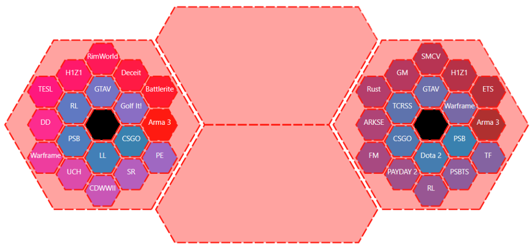

Figure 5: Overall Visualization

Now we just needed to fulfill our last step of showing a comparison between several games. We though about the *Parallel Coordinates* example given in one of the first lectures of the course but we just didn't have enough attributes to compare between games (we just wanted to see the absolute number of users).

4.2.6 KISS Principle

Keep it simple, stupid (KISS) is an acronym used in design, stating that "most systems work best if kept simple, rather than complicated". Instead of going for an all-out new design for representing the game's relationships (especially because of time concerns), we sticked to a commonly used visualization - a *line chart* on the top empty space (*Figure 6*).

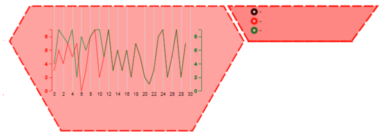

Figure 6: Relationship representation

On a color-basis, we also provided data's labels on a top-right new section (that would fit right on top of a cluster). The main problem with this is that it breaks symmetry. A possible solution would be to add labels on each side of the chart, corresponding to each clusters' games independently, but there was simply not enough time to implement this feature.

4.2.7 Final Touches

Now that the whole structure of our visualization is in place, we decided to experiment with different color-palettes and icons to make our project even more appealing. As of now, the final result of our work is portrayed in *Figure 7*.

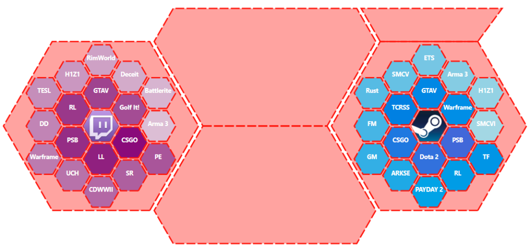

Figure 7: Final Visualization

5 Group Evaluation

5.1 Work Division

How we split our work:

Task	Contributor
Data parsing & filtering	Nuno Goncalves
Visualization Imp.	Baran Nama
Data-binding to Viz.	Nuno Goncalves
Interactions on Viz.	Baran Nama
Process Book	Nuno Goncalves
Debugging	Baran Nama

5.2 Peer Reviews

Mutual peer reviews:

Question	Reviewee	Value
1.Preparation	Baran Nama	Yes
2.Contribution	Baran Nama	Yes
3.Respect	Baran Nama	Yes
4.Flexibility	Baran Nama	Yes
1.Preparation	Nuno Goncalves	Yes
2.Contribution	Nuno Goncalves	Yes
3.Respect	Nuno Goncalves	Yes
4.Flexibility	Nuno Goncalves	Yes

Note: *Reviewee* refers to the person being reviewed (which means the other element did the review).

6 Conclusions

There are many things we know need improvement but the project is still a work-in-progress, as we plan to continue with this idea in the future. One of the first things we should focus on is better utilizing colors and animations on the visualization to more intuitively guide the user through the process. Despite this, we think our final product allows for a good preliminary study (or a *proof-of-concept*) and reflects all the factors we took into consideration during its conception.

6.1 Personal Note

There is a clear contrast between the *before* and *after* of this project. Visualization always seemed such a trivial endeavor but now we can see it's a lot more intricate than we thought - clearly, "ignorance is bliss".

References

[1] Jeffrey Zeldman. 2011. *Taking Your talent to the Web: A Guide for the Transitioning Designer*. New Riders, San Francisco, USA.

[2] Mormann, Milica Milosavljevic and Navalpakkam, Vidhya and Koch, Christof and Rangel, Antonio. 2012. *Relative Visual Saliency Differences Induce Sizable Bias in Consumer Choice*. Journal of Consumer Psychology, Vol. 22, No. 1, 2012. Available at SSRN: https://ssrn.com/abstract=1998433

[3] Ben Shneiderman. 1996. *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. University of Maryland, College Park, Maryland 20742 USA.