# (in)visible metrics

**Group 8: Mikael Korsholm Poulsen, Nicoline Due Smedegaard, Cathrine Vilain Dybro, Johannes Wigh-Poulsen, Mathias Zykowski Tvilling.**

In this project our objective is to address what data Facebook choose to present in their interface. By metrics, we refer to the data which is being captured, but not quantified. The interface of Facebook is filled with different kinds of metrics that shows how many likes, shares and comments a post has. These metrics are data, which has been quantified and consequently put on display in the interface. In our group we find it very interesting why Facebook choose to show some metrics and why there are some they choose to hide from the users. Therefore we have chosen to take inspiration from Facebook and make our own version for our final project.

## Technical description

Approaching the program it looks like a standard home page on Facebook. The profile is named "You" and its profile picture is the standard anonymous profile picture. In the same way the friend list is filled with profiles with the name "Friend" and a random number. This anonymisation and the name "You" was done to make the experience more immersive. Ads and groups are also displayed and these have not been made anonymous, in an attempt to keep the layout more similar to Facebook.
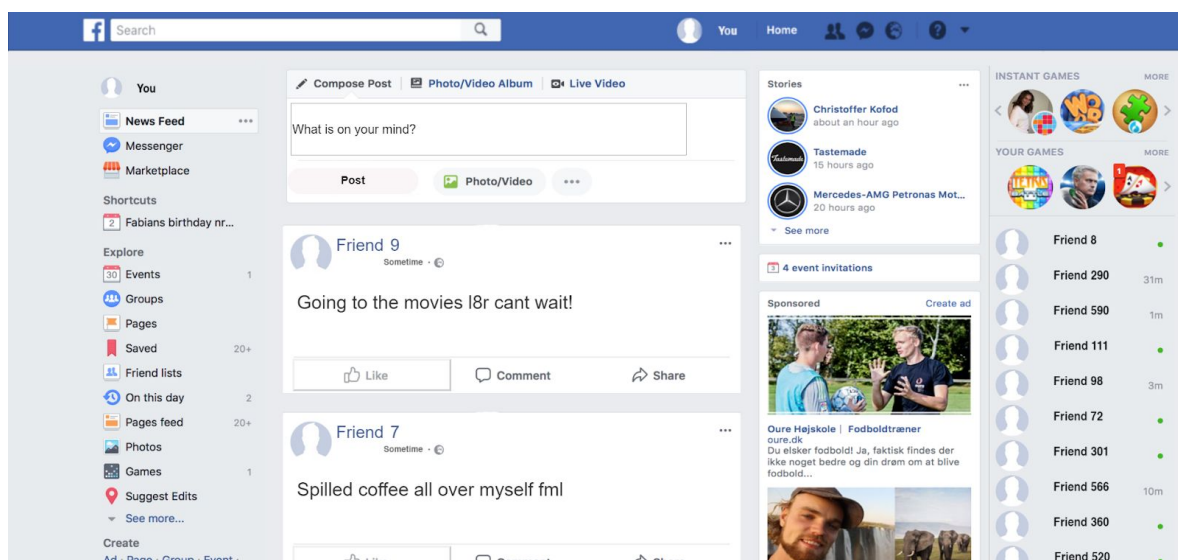
In the middle of the layout a series of randomly generated posts are displayed with text chosen from an array. At the top of these posts an input box is displayed and beneath it a button which says post, which is not a part of the regular Facebook interface. This change from the regular interface was made to highlight the feature and thereby guide the user. In the same manner "Like"-buttons are posted on the randomly generated posts, which have shaded borders to highlight the ability to interact with it. These two types of buttons and the ability to write in the input bar are the only interactable objects in the layout. Assuming that the user is familiar with the layout of Facebook we hope the users next step will be to either try using one of the "Like"-buttons or using the input bar to create a post. Clicking one of the "Like"-buttons will initiate two functions; One that changes the color of the button to indicate a registered click and another that creates an instance of a class that resembles a notification. This notification tells the user that "You liked a post after looking at it for X seconds." where X is the time the program has been open. Clicking the post button initiates two functions; one that creates a new post, which is displayed on top of one of the earlier randomly generated posts and another which reverts the color of the top "Like-button". This user-generated post is created using the same class as the randomly generated posts, but through a different function which shows some of the similarity between the two.

## Notifications

Creating a post initiates a for loop which contains a chain of timers that each set off a function that displays randomly generated notifications in response to your post. The possible notifications are: "Friend X is looking at your post", "Friend X liked your post after Y seconds", "Friend X has scrolled past your post", "Friend X is writing a reply" and "Friend X is visiting your profile" where X is a randomly generated number and Y is a set of chosen numbers. The amount of notifications increase over time but on a seemingly random rate for a set amount of time. After this time it the displaying of new notifications stop until a new post is created. Beneath is a flowchart to simplify our programs functionality.

# Flowchart

## Essential syntax

In regards to our main topic data capture, one of the most obvious syntax to highlight is millis(). In the program it is used to show how long the program has run before you like a post. This syntax is a fine illustration of how data is logged from the first millisecond of the programs runtime to its end and it is very interesting to compare it to other similar functions. When we wanted to display this recording of time we started out by using the syntax frameRate, which counts the amount of times the code has been executed. This turned out to be a challenge since our use of for loops in the draw function limits the program to one frame. These two syntaxes highlight an interesting difference in perception of machine time. In the program the syntax setTimeout() is also used, which initiates a function after a certain amount of time. Since this works in our program it must rely on the same kind of "time" as millis().

Another element of our program which exemplifies data capture is the input bar. Both the input bar and the buttons on the page are elements from DOM-library which makes including HTML-elements in p5.js possible. This exemplifies data capture in the way that the users input becomes part of the program even though the content of the input has no consequence on the generated notifications.

# Theoretical Background

Our program addresses several of the themes from the curriculum, but perhaps most prominently, it addresses data capture and datafication. In his paper *What Do Metrics Want? How Quantification Prescribes Social Interaction on Facebook* (2014), Benjamin Grosser explains how Facebook draws use of quantified data, creating counters of friends, likes, comments, shares and much more. All of these different numbers are described as metrics. Benjamin Grosser describes in the text: '' *[...] how these metrics prescribe sociality within the site's online social network. Its prescription starts with the transformation of the human need for personal worth, within the confines of capitalism, into a insatiable "desire for more"* '' (Ibid). Along with the paper, Benjamin Grosser released an artwork called *The Facebook Demetricator,* an experimental ad-on for browsers, which removes all metrics from the

Facebook interface. The demetricator aims to question what data Facebook allows its users to see and interact with, and what remains hidden. His thesis is that the shown metrics prescribe our use of the social platform, through quantifications of interactions such as the counter of a like button.

Benjamin Grosser introduces the term capitalist realism, how capitalist ideology has come to create a constant, insatiable desire for more. This idea that more is better supports the use of quantified data in the form of metrics in interfaces like the one of Facebook, since the strive for more likes, more friends etc. generates activity. Rather than only exploring Facebook through capitalist realism and focusing on metrics, our program aims to explore a different term introduced in his paper; "hidden metrics". Hidden metrics is a term used to describe captured data that is not being presented to the user. The user only has access to a limited amount of data, like the metrics on show in the interface, it does not, however, entail that the shown data is the only data they collect. By hidden metrics, we refer to data which is captured (e.g. time on page, mouse position).

In his conclusion he mentions how Facebook isn't a neutral platform:*''Demetricator intervenes into this new private social space in order to help us understand that Facebook is not a neutral facilitator of interaction''(Ibid).* We have used this understanding as a stepping stone for our project, to further explore how this facilitation of interaction and metrics is perceived.

***The like economy: Social buttons and the data-intensive web*** by Caroline Gerlitz and Anne Helmond is another text we have chosen to relate our program to. The paper focuses on: " *Facebook's ambition to extend into the entire web by focusing on social buttons and developing a medium-specific platform critique"(* Gerlitz, C. & Helmond, A. 2013*, "The like economy: Social buttons and the data-intensive web*", p.1348). Furthermore it "*Contextualises the rise of buttons and counters as metrics for user engagement and links them to different web economies"(*Ibid p.1348) . The paper explains how a like is more than just a number on a counter. A like is a button which: *"provides a one-click shortcut to express a variety of affective responses such as excitement, agreement, compassion, understanding but also ironic*

*and parodist liking"(*Ibid p.1358*).* By having a button that contains all of these meanings the Like economy is created. The like economy is an infrastructure that allows the exchange of data, traffic, affects, connections and money which are mediated, through social plugins where the most noticeable is the Like button (Ibid p.1353). Due to the datafication of social interaction within Facebook on social media it has become highly relevant for websites to capture huge amounts of data from their users. Carolin Gerlitz and Anne Helmond explore the technical infrastructure of data capture focusing on social buttons. As they describe; "*With the rise of the social web, companies realised that everyday online activities provide a rich source of information about user preferences, habits and effects that had previously only been available through consumer research techniques*"(Ibid p. 1360). This industrialisation of the web is one of the reasons data is being captured systematically. Our program illustrates how different data can change the experience and culture within a social platform. It also illustrates how much data is actually being captured by Facebook. Furthermore our program is critical towards Facebook's selection of shown data. In the article they argue that Facebook only provides the user with data that influences them to act in a certain way in order to benefit Facebook. They elaborate on this issue further; "*the Like economy is facilitating a web of positive sentiment in which users are constantly prompted to like, enjoy, recommend and buy as opposed to discuss or critique – making all forms of engagement more comparable but also more sellable to webmasters, brands and advertisers*" (ibid p. 1362). Through our program we have shown how a few different changes in shown data can change the experience dramatically, and how much the shown data actually influences an interface. We find that this enhances the understanding of Facebook's choices, and how their interface is orchestrated to be profitable.

## Capitalist Realism

In the text, Benjamin Grosser uses a capitalistic understanding of the mind. He claims that the reason why people are actively interacting on Facebook, is because of the desirable need for more. This sense of capitalist realism is what the final conclusion of his text is based on:

> *''Demetricator reveals how Facebook draws on our deeply ingrained "desire for more,"*
> *compelling us to reimagine friendship as a quantitative space, and pushing us to watch the*

*metric as our guide'' (What Do Metrics Want? How Quantification Prescribes Social*
*Interaction on Facebook*, 2014)

In this quote he points out this ''desire for more'', as the reason behind for the action. But what is more? And what is enough? Our program critically addresses this. By making notifications so extreme, perhaps people try to avoid being exposed for notifications and thereby avoid being a part of the ''more is better'' culture. More notifications is necessarily not a positive thing, it is just Facebook selecting specific metrics and showing them. We have to consider this when trying to get an understanding of Facebook and the interaction within.

Furthermore our program addresses what kind of metrics is being visualised. We show the metrics of how long a person have been looking at a specific post and how many scrolls past a specific post. By doing this we are challenging a users perception of a platform. We only focus on few elements of what is actually being logged by Facebook, there are many, many more metrics which are hidden.

Our program explores a user conception of what Facebook actually consists of. How much of the interaction is actually quantified without the user being in direct contact with it. It raises the question, why are they collecting this data and what is to be used for, if not is to improve Facebook for the user?

Our program shows different notifications, some are positive: "*friend 9 liked your post*" and some are what might be considered as negative: "*Friend 290 scrolled past your post*". When the user is exposed to both negative and positive notifications it can create a mixture of different feelings when interacting with our program.

For some users the use of our program could actively become more fun and interesting, because the user will suddenly be able to see who ignores, Like's or decides to scroll past ones post. The flow of notifications could also be an indicator of once social life, and the interface could seem empty without the constant notifications. Furthermore the need for constantly being able to see that you are visible online could become more important than if the notifications are positive or negative.

For other users the use of our program could be a very negative experience and the user would possibly not use the program, as they would use the original Facebook. The negative notifications would become too much and the feeling of being ignored could create a feeling of being irrelevant, not good or fun enough. This could result in a total stop of using Facebook because the thought of others being able to see how long you have looked at a post and if you ignore a post is too much, and the overall enjoyable experience of using Facebook is taken away.

# References

Glosser, B. 2014, "What do metrics want? How Quantification Prescribes Social Interaction on Facebook"

Gerlitz, C. & Helmond, A. 2013, "The like economy: Social buttons and the data-intensive web".