The Metricated Social Self

Group 7 - Helena, Mathilde, Cathrine B.P. and Line

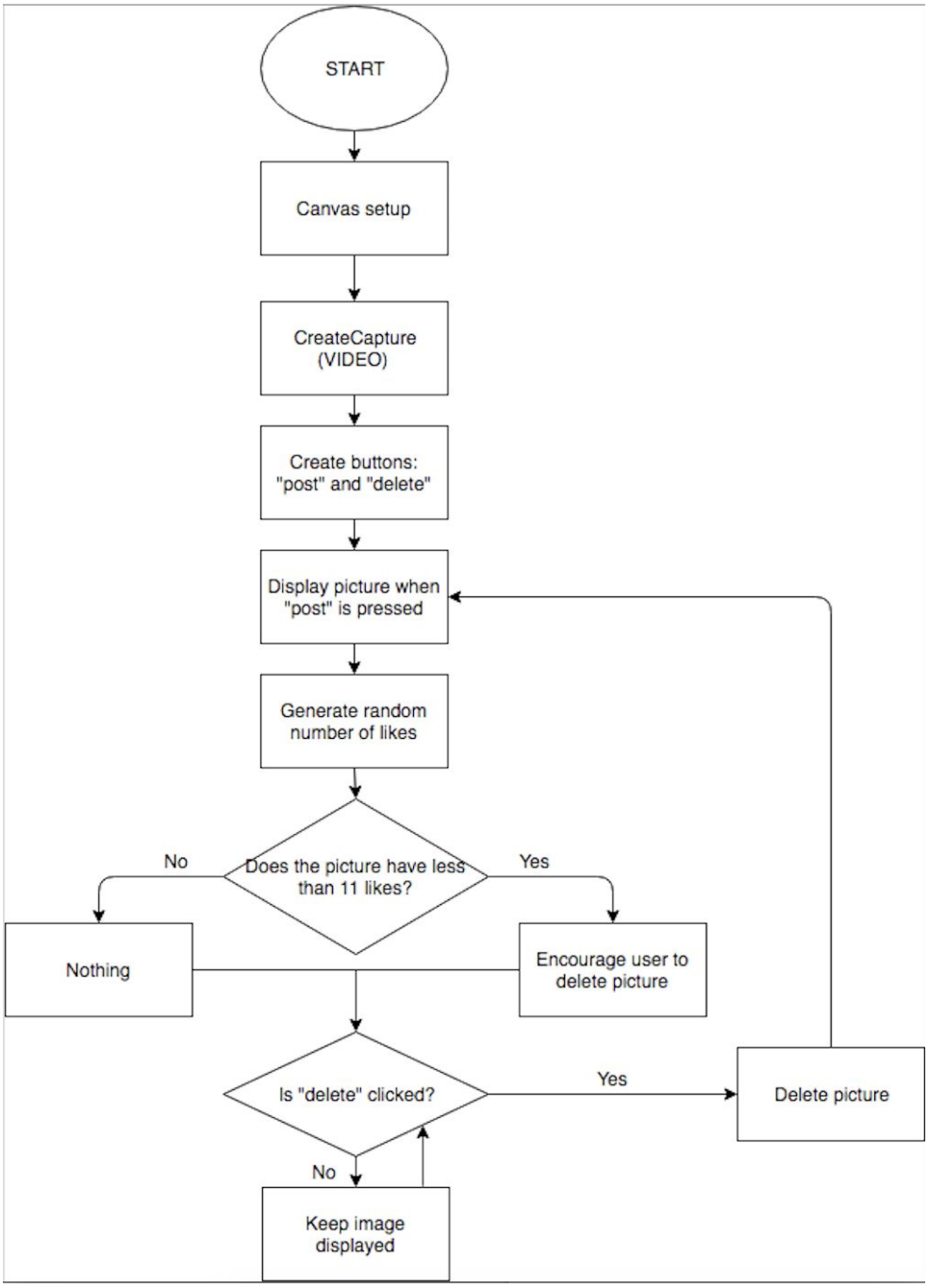
1. Presentation of Problem

The overall problem that we choose to base our final project on, revolves around social media, and how our actions on these are being affected by quantifications and buttons. Social media are a big part of our everyday lives, and therefore they change our actions and opinions to such a degree, that we sometimes doubt our own judgmental skills. *"When you affect something, you are at the same time opening yourself up to being affected in turn, and in a slightly different way than you might have been the moment before."* (Snodgrass, 2017, p. 140) We base what we post online on the acceptance of others, often in the form of "likes". If the numbers do not add up to our own expectations, we often find ourselves deleting the content again, even though we actually liked it ourselves. In this way, the opinion of others determine whether you consider yourself good enough, and the quantity of likes becomes a number that determines this.

Hereby our overall problem is: "*How do likes affect our perception of ourselves and the way we behave online?"*

The reasoning for choosing this specific problem is, first of all, based on our own personal experiences; we are a part of the social media culture, and we ourselves experience the pressure created by our need for acceptance among the people around us. The interest in the subject mainly started when reading the text "*What Do Metrics Want?"* by Benjamin Grosser. In this text he mentions how the quantity of numbers are becoming more important to us than the people who created these numbers. He also puts a focus on how the importance of metrics to the user increases the use of the platform, because the user wants a higher number of metrics, and for that reason will post something to get it.

2. Flowchart and Description of Idea

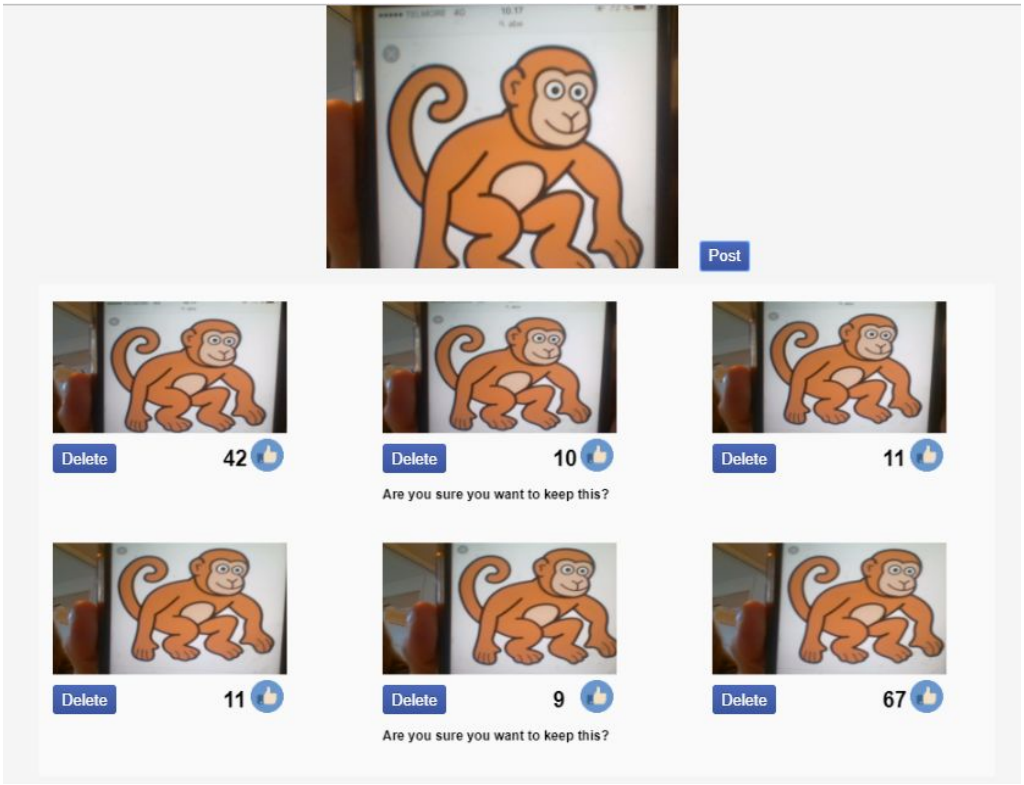


Our idea is about Data Capture using the webcam. The program contains several buttons; a “post” button and six “delete” buttons, one under each picture posted. When clicking the “post” button, a capture from the webcam is saved underneath it. The program will then randomly pick a number of likes from the array ‘likes’ for each picture. If a picture has less than 11 “likes”, a text will encouraging the user to delete this picture. Each “delete” button are connected to a picture. The pictures are saved if the “delete” button connected to it is not

clicked. We chose the two kinds of buttons “delete” and “post”, because they seem to represent the essence of social platforms. The “post” button is what constitute these platforms, that depend on user contribution to actually have content to show. The “delete” button ensures the user that if they change their mind about a post, it can be removed. The random amount of likes symbolise how many people acknowledges the content you posted. Hereby, the program is able to simulate a possible situation on a social media, and highlights how a lack of acceptance from others can ultimately lead to a removal of the content. Our program is also supposed to symbolize the importance of the amount of likes, where quantity is more important than quality; it is more important to get many likes, rather than focusing on who the specific like is from.

The number of likes, 11, is based on the early years of Instagram. If your photo did not get 11 likes, the number would not be displayed, instead all the names of the users who liked the photo would be shown. When the picture had under 11 likes, it became very obvious to everyone that the picture did not even have enough likes to get a number displayed. Therefore 11 became the minimum number that a lot of people wanted to reach for them to feel like their picture was acceptable.

Our behavior on social media is quite different from how we act in real life. On social media our behavior is highly influenced by how we think others would respond to what we do and post, and how it makes us look on the surface. In real life we cannot delete our actions, but on social media it is possible to change the data, or delete it, if the response does not live up to what was expected. This is an aspect that we also want to emphasize in our program by including the “delete” buttons.



Link: https://rawgit.com/MathildeFN/mini_exes/master/Final/empty-example/index.html

3. The Technical Aspects of our Program

Our program is combining a webcam function with the “like” and “post” possibilities as seen on Facebook and Instagram and other social media. We are using the syntax “createCapture(VIDEO);” to gain access to the users webcam. Next to the webcam, we have created a “post” button that triggers a capture of the webcam content. Underneath every single picture posted, a “delete” button will appear. To create the “post” button, we used the syntax “createButton(‘name’)”, and combined this with the “button.mousePressed(function)”. To give the buttons some properties, we have created a function that explains what happens when the button is pressed. To create the “delete” button, we too have used the syntax “createButton(‘name’)” and combined this with the “button.mousePressed(function() { deletepicture(i);});)”. The function “deletepicture(i)” works in the way that it removes the current index of the picture from the array of pictures, and then in function draw, a background is continually drawn on top, which hides the pictures removed from the array behind it, and makes it appear that they have been permanently removed.

When a picture is posted, a random number from our “like” array appears with each picture. To create these random numbers, we have created a variable consisting of an array

containing 10 different numbers, five numbers from 1-10, and five numbers from 11-100. The program assigns these numbers randomly to the different pictures by the use of the random syntax. Through a conditional statement, the program checks if the number of likes is higher or lower than 11. If the number is under 11, an encouraging text about deleting the picture appears.
Furthermore, both the “delete” button, the pictures and the numbers are defined within a class. This organisation of our code is done because these three objects has a relation and interaction with each other, but especially also because they all occur six times which means, that the code quickly feels unmanageable. By creating a class we only have to write the attributes once to the certain class, even though we call six objects.

4. Presentation of Main Theme - Data Capture

The main theme of our work is Data Capture, and this theme is being addressed both within the final runme, but also within the code. In the runme, data capture is first represented in the webcam, and in how the program is able to take a picture, and then use this as data that can be evaluated in the form of “likes”. The theme of data capture is also present in the way that the assigned “likes” are supposed to represent an opinion of others, that has been captured and simplified in the form of a “like” - what is commonly seen on other social media platforms. Lastly, an aspect of data capture that is also present in our program is the aspect of erasure, and how the data that we chose to share online is very hard to permanently delete. The pictures that have been posted to our program never truly disappear, but are being hidden behind a continuously drawn background, that makes it appear to the user as if they have the ability to permanently remove these pictures.

Within the code, Data Capture is presented in the way that the code is structured into classes. These classes represent how data is often being put into different categories, that makes this data more easily comparable and usable when companies want to sell and distribute this captured data as targeted advertising. “In this sense, 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.” (Gerlitz & Helmond, 2013, p. 1362). As mentioned in this quote, this is often seen on social media in how people get the ability to like, rather than have many different reactions. This makes the data more easily categorised.

5. Reflection on the Aesthetics of our Code

5.1. Classes

The reason for choosing to put all the pictures into the same class is because they all share the same attributes, like position and size. Even though a picture might not seem to have much complex characteristica beside maybe content, size or material, we could have chosen to add more to these pictures with e.g. a filter or opacity. "An abstraction is a generalization of an object which includes only those details necessary to define it and to differentiate it from other objects" (Lee, 2013, p. 32). The decisions about which attributes a class is defined by, is exactly what refrain them from being objective. By making decisions about what is included, you are also excluding something. This means that the object you are creating within a class might not match the real world object it is representing. Though, not many would be upset about some missing attributes in a picture object, imagine creating a human object and how many decisions about attributes one would have to make.

5.2. Randomness and array

Regarding reflections on the aesthetics of code, our program consist of the use of randomness and array. We have chosen 10 different numbers between 1-100, five numbers between 1-11 and five numbers between 12-100, to illustrate our point about the impact of getting at least 11 likes, which determines whether the encouraging text about deleting the picture will occur. We are well aware of the fact that this choice of numbers does not show how it works in real life, but the reason is to make the chance of getting less than 11 likes equal to getting more than 11 likes, so that the user have a higher chance of experiencing this encouragement to delete her/his picture. If we had chosen all the numbers between 1-100, the chance of only getting 11 likes would be very small, in that sense our point would not be as clear.

The reason for using the random syntax to decide which number of likes the pictures will get, is to illustrate how the allocation of likes on the web might have a random impact, or at least illustrating the feeling of unpredictability - not having control of whether the pictures you share will be popular or not: “*Randomness corresponds to infinite volumes of data in contemporary software culture and the random quantities included in algorithmic decisions that compute the compression of data in the form of binary digits. Additionally, such a notion*

of randomness and infinite volumes of data lead to what Parisi calls "unpredictable variabilities"" (Soon, 2016, p. 155) In this way our program addresses the theme of unpredictability by using the random syntax, by illustrating the uncertainty of how other peoples will react on social platforms, when posting personal things on the web.

6. The Project as a Critical Work

The motivation for sharing things online is no longer about sharing a part of your life with friends and family, but about how well the things we share are being received and accepted by other people on social media platforms. This is illustrated in our program by the randomly generated "like" metric, where the user is being encouraged to delete the picture, if it has received less than 11 likes. Hereby, we are trying to replicate the way that likes affects our behavior online, and we try to illustrate the thought process that a lot of people go through, when their expectations of a certain amount of likes has not been met. In this case, it is not unusual to begin to question the content of what has been posted, sometimes leaving the the user tempted to delete the post again.

The amount of likes are generated randomly, because it is supposed to represent the feeling of uncertainty that is experienced when posting something online. It is very difficult beforehand to try to predict whether a certain post and content will get few or many likes, because it depends on a lot of different factors, which can make the distribution seem random. We are well aware of the fact that likes are not randomly distributed in real life, but dependent on different parameters, like how popular the profile is, time of the day or day of the week, the content etc., but the use of randomness underlines this feeling of unpredictability that might result in frustration because you do not know what posts will eventually turn out "good enough", before they have been posted.

The encouragement to delete a picture with less than 11 likes addresses the fact that the amount of likes manipulate how we act and what we post in the future on these platforms. "*Certain kinds of posts get more likes than others, and the desire for more thus begins to guide what users write and submit. In other words, when users want more likes, they write statuses that get them and post photos that accrue them. Quantification is changing the quality of what users do*." (Grosser, 2014, p. 7). Most people are haunting this acceptance through likes which means that what they post will also be determined by how many likes the user thinks the post will get. Therefore, a user might end up not posting something they

would not have done if they knew the content would not get a lot of likes. This affects our own ability to judge the quality of what we want to post, and changes our behavior from posting what we like personally, to posting what we believe will get the biggest amount of likes. In this way, the opinion of other people become more important than our own.

By using the webcam instead of an API containing random pictures, the posted pictures become much more personal, which means that the user of our program hopefully feels more vulnerable in this evaluation from others that the likes represent. By doing this, we want to put a focus on how this evaluation that we experience online today can affect the self-worth of the user, both positively and negatively. "Personal worth becomes synonymous with quantity." (Ibid, p. 2).

7. Presentation of Theory

- Benjamin Grosser, ”What Do Metrics Want? How Quantification Prescribes Social Interaction on Facebook,” in Computational Culture no. 4 (2014):

We want to use the text *What do metrics want* to support our thesis about how like buttons generates a culture where quantity is more important than quality, where it is more important how many likes you get rather than who likes it. Also, we want to use the text to support our thesis about how all the numbers and counting’s affects the way we act on the web, and how the purpose for these metrics is to increase the use of Facebook or other social medias.

- Gerlitz, Carolin, and Helmond, Anne. “The like Economy: Social Buttons and the Data-Intensive Web. ” New Media & Society 15, no. 8 (December 1, 2013): 1348–65:

The text *The like economy: Social Button and the Data-Intensive Web* is supposed to be used as an examination of data sharing and also to support our thesis about how the like economy, data and numbers consist of performative and productive capacities, which enables generation of user effects, which enact more activities and also multiply themselves.

- Pold, Søren. "Button." Software Studies\ a lexicon. Eds. Matthew Fuller. MIT Press, 2008. 31-36:

The text *Button* is supposed to support our thesis about how pressing a button generates data capture and how buttons creates binary choices, where you only have a certain option to answer or comment on something according to what there is buttons for – in this sense it is kind of like “all or nothing”.

- Snodgrass, Eric. “one / easy / click: facebook’s like button” in Executions: Power and Expression in Networked and Computational Media. Malmö University, 2017, pp. 121-152:

The text “one / easy / click: facebook’s like button” is supposed to support and express our thesis about how the like button has an ability to capture and share certain affective response; that the like button acts as a kind of highly simplified, but still efficient and productive interface for processing affective response.

- Soon, W., 2016. *Executing Liveness: An Examination of the live dimension of code inter-actions in software (art) practice,* Aarhus: School of Communication and Culture, Aarhus University.

The section about “Randomness”, in the chapter “Generativity” in *Executing Liveness* is supposed to support our reflections about the aesthetics of our code, when using the random syntax, which substantiates our point about the uncertainty and unpredictability you might experience when posting things to be judged on social platforms.

8. Oral Discussion Points

There are several further aspects of our program that we believe could be interesting to discuss more. From the perspective described above, we would like to adress how the social media affects how you act on these platforms. Therefore, it is relevant to discuss what and why some content become popular, and what content does not leave an impression. Even though our program tries to imitate the post possibilities of platforms like Facebook and Instagram, it does not include the option to upload a picture you took on your phone the day before. It adds another perspective on the content posted on Facebook and how much thought you put into a post. Would you dare to post a picture you took directly from the webcam/app with no editing? Are factors like popularity and amount of previous likes more important, than the immediate quality of the content, when you have to decide whether or not to like a post? Are buttons like the ‘like’- and ‘share’-button neutral, or do they have an agenda that manipulate you on the platform?