TIME BALANCER

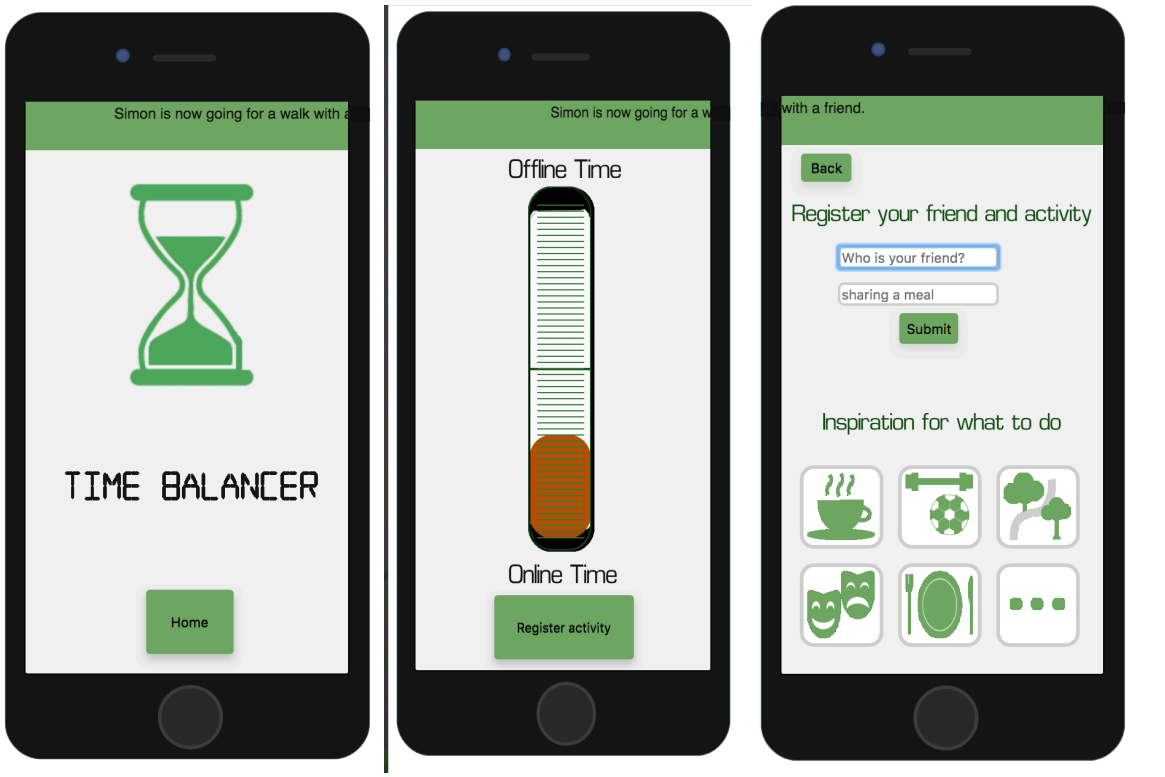
TIME AND TEMPORALITY

Authors:

Camilla Mondrup Ester Aagaard Nina Borgbjerg Sissel Raahede

Characters: 18.029

Screenshots of the program:



Link to video (Simulation of consequences):

https://rawgit.com/AUAP/AP2018_Submission/master/Final_Project/Group5/Final_Project/Final_code/empty-example/index.html

Link to video (Simulation of consequences):

https://www.youtube.com/watch?v=bc3BQV5CSU8

Introduction

Time is an interesting phenomenon. The perception of time is relative and imprinted by a lot of factors like circumstances, relation or culture. It is a human-created phenomenon and yet, there is no specific way of defining what exactly time is and how it is understood. Due to the fact that it is such a complex topic, we keep molding everything regarding time, so it becomes easier for us to understand.

An example of this is the clock as we know it with the numbers from 1 to 12, placed in a circle, with three clock hands determining the hour, the minute and the second. This concept of time tells us what time it is during our 24-hour long day, and we use it to structure our lives synchronously and to create a common ground of how to perceive time. Most people know how to understand the design of a clock and then to tell what time it is, but few know how to understand the process that is going on behind the interface design. Behind the clock hands moving around pointing at certain numbers, there is a more complex process of different cogwheels moving mechanically around in patterns to control the position of the clock hands. We are aware of this process, but it is not something we take into account in our everyday use of the clock. We are conceptualizing a complex process, to better understand and relate to the concept.

The example above is just one way to perceive time and especially with the development of more and more advanced digital and software-based technologies, the scope of how to perceive and experience time has expanded. Our online experience of time doesn't necessarily count in hours, minutes and seconds, but is maybe considered in relation to what is fast and what is slow. Our awareness of time and duration is cognitive and somehow connected to a certain amount of habitus. When we use technologies, we set expectations in our minds for how fast we should respond or expect a response and this process is shaped differently by the technologies we use: "*We have an acute awareness of duration, and that awareness is always linked to prevailing technologies that shape how we understand and experience time.*" (Farman 2017). In this use of digital technologies, the eternal data stream has a great impact on our perception of time and we can use the same example as with the clock of how we conceptualize these complex processes, to better comprehend time as an online concept: "*There is a temporal dimension to the data stream and in today's networked communication data streams indicate events that are regarded as instantaneous in capitalized economies. The now that we are experiencing through perceptible streams is entangled with computational logic.*" (Soon 2017,89). The reality and the now we see and experience in different online platforms compares to the design of the clocks interface and the computational logic

compares to the hidden process behind, that we don't really understand or take into account in our perception of the time.

In the following synopsis we want to address how our perception and experience of time and temporality is relative in regard to the backend processes in software and the frontend interface we are presented with. The approach in our program is critical towards the relation between how much or how little time you spend online and offline. We seek to increase an awareness of having a balance in the amount of offline as well as online time.

Description of program

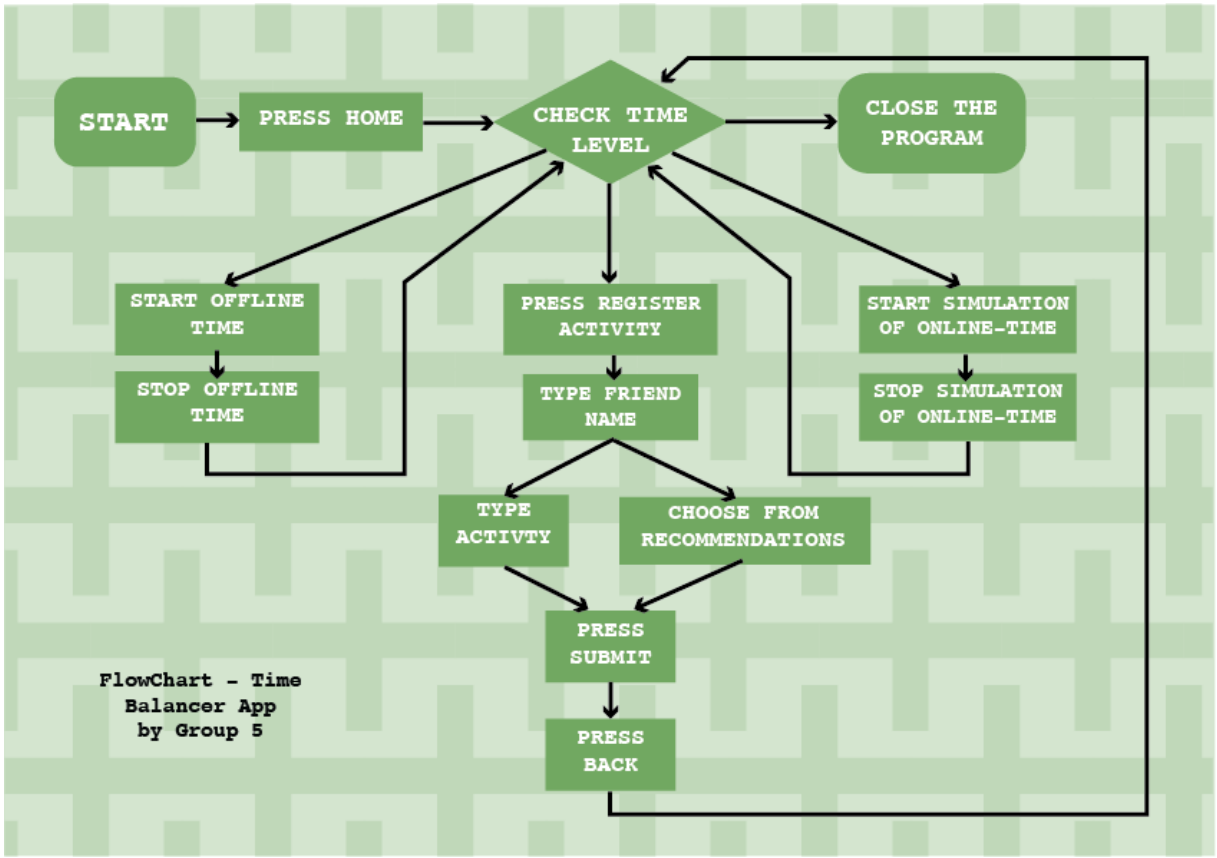
Our program is a simulation of an app that can help people to get an overview of how much time they spent online on social media as well as offline with other people. When the program starts, you see the start page with an hourglass that flips over again and again. The hourglass symbolizes time, balance and many other things related to time in general. It is a very powerful symbol and most people understand the connotation of time, so we chose to place it on the start page to signal to people from the beginning that this is something about time. There is also a button which, when pressed, directs the user to the home-page.

At the home-page the user will see a counter. The counter represents how the individual user spends their time. When a user is spending time on social media the bar in the counter will go down and if the user spends time with their friends offline, the bar will fill up again. The user earns time by spending time offline with their friends, afterwards the user can use the earned time online, and the goal is to not run out of time. Our intention was that the program would be connected to your social media accounts and by that, being able to continually track the time a user spends online. In our program the counter only works by simulations of online and offline time. You are able to press 'start-' and 'stop online time' as well as 'start-' and 'stop offline time'. Intentionally the offline time would be tracked by connecting your phone to another device, e.g. through an API when the user registers an activity with a friend.

At the top of the home-page a dark green text field is placed, with text running across the screen from right to left. This text we imagine will be constructed with information gathered from the user's friends and their current activities. Just beneath, a text states what you are doing and with whom. This text will be generated when the user registers an activity and a friend. This is possible at the next page, that you can access by pressing the 'register activity' button located at the bottom of the home-page.

At the register-activity-page there are two input-fields and a ‘submit’ button located under the text that says: “Register your friend and activity”. In the first input-field the user can type in the friend that they are going to do something with, and in the next you can type in what you are going to do. You also have the opportunity to press one of the buttons that are located under the text “inspiration for what to do”. This will create an input in the activity-input-field equivalent to the symbol on the button. We imagined that this could be helpful if you need inspiration for what to do.

When the user has typed in the information about what they are doing and with whom, the information will be shown at the top text-field and if we were to make it into a real app, then the information would also be send to the user’s friends.



(Flowchart of our program)

Analysis of time syntax

As it is stated in the description the counter is placed on the home page and is an essential part of the program. This counter functions as a visual expression of how people use their social time, either online or offline. Our program addresses the concept of time by using the syntax frameRate() from the p5.js library. In our program the framerate is set to 60 frames per second, this specifies the number of frames that is to be displayed every second. It is possible to regulate the framerate, but in our program the frame is updated 60 times each second. This makes it possible for us to translate the time a user spends being social online, and the time a user spends being social offline into a visual output. FrameRate is translated into seconds, and the seconds is translated into pixels. This we use to visualize the two different activities. This conversion determines how fast the counter will subtract time when a user is online based on a social quotient. The quotient is calculated from how much time a user spends online on the social media on a general level, in relation to the time they spend meeting with friends and family offline, which is added to the user's profile at launch. The quotient that establishes the users current time varies from user to user, and therefore might create different visual perception of how time passes when being online from user to user.

```
{
          var dailyOfflineTime=1; //calculated in hour
          var dailyOnlineTime=5; /*These values is added by the user at
launch*/

     Quotient=dailyOfflineTime/dailyOnlineTime; // for our specific user
quotient=0,2

}
```

In this code snippet the quotient is calculated from a possible user's data. This specific user spends 1 social hour offline and 5 social hours online each day. The user's social quotient equals 0,2.

```
{
          downSeconds=down/60;
          /* the variable down counts the frames that are executed in the
program during the time the
          user is online, and is then converted into seconds. */
          onlineTime=downSeconds*Quotient;

          Checker=2;
     /*The checker determines when the functions is executed, this is
when the checker is 2.*/

     if(socialtime>0.1 && socialtime<=280){
          socialtime=socialtime-onlineTime;
     }
}
```

The frames are converted into seconds. The seconds that the user is online, is multiplied with the social quotient, and is then subtracted from the variable socialtime and the variable is then used for the animation of the counter:

```
function timeCounter(){

          if(socialtime>=0 && socialtime<400){
                    if(y<=440 && y>=0){
                              counterPosition=y-socialtime;
                    /* counterPosition is the the rectangles position on
the y-axis. y is the starting position of the rectangle, which is set to
440 */
                    }
          push();
          noStroke();
          fill(r,g,0,250);
          rect(125,counterPosition,55,socialtime,20);
          pop();
}
```

This function visualizes the process for the user by drawing the counter and a rectangle which shows the amount of time, either used or earned. The program intends to make a combined understanding of the social time the user spends offline and online. This is done through the previous algorithm which is a computational process that isn't perceptible for the user: "*The now that we are experiencing through perceptible streams is entangled with computational logic.*" (Soon 2017,89) You can compare the counter with the clock, where you are presented with an interface that allows you to perceive time in a simple way, without understanding the more complex process behind.

We were not able to track the time spent on social media on the user's units, therefore we made a simulation of this input. In reality the time that the user spends online would be feeded as input to the program continually. As Soon describes it in executing micro-temporality: "*More precisely, the viewer is not watching the content as data arrives,*
instead, the viewer is watching the processed data that has arrived and
stored in the buffer." (2017,96)

When the user observes the counter counting either up or down, the user is presented with a visual interface that lets the user believe that info is presented to him or her in real-time, instead the data has already arrived, and is stored in the program. The output that the user is watching has already been transformed from one kind of input into a visualization of time. Therefore, the user does not experience this in real-time, but can possibly understand the counting as a direct stream from the use of online media. At the same time the counter in our program can be understood as an interface which communicates through a visual output, a computational process, that both translates several inputs into understandable outputs for the user.

Data-streams

Since temporality is one of the main themes in our work and what a lot of the code stems from. When speaking of temporality, we want to look at how time is understood in a digital context. The internet plays a big part in this, since it is what consumes a big part of our online time. This has changed over the years, which is something that is also mentioned in Real-Time Streams: "*The change represents a move from a notion of information retrieval, where a user would attend to a particular machine to extract data as and when it was required, to an ecology of data streams that forms an intensive information-rich computational environment.*" (Berry 2011, 143). The structure of our time spent online has changed as well. Before when we used the web for information retrieval we had a specific purpose when opening a browser. Nowadays the web has changed so all information is delivered in

a constant stream. Information is spread across the internet and we have new expectations of it. You don't have to be at home at your local connection to access the internet, but with technologies such as smartphones we have become used to being able to go online anywhere at any time. In that way we now see the web as a new *now* that is constantly updated, and no longer just a place to store information. This is also addressed in the Berry's text. "[...] *we are currently undergoing a transition from a 'slow web to a fast-moving stream* [...] *we are shifting our attention from the past to the present, and our 'now' is getting shorter'"*(ibid.,144)

Our counter can be put in relation to these fast-moving streams. It is updated constantly, showing us how much time we have left. Because everything is happening momentarily, we have new expectations of the now on the internet. Another example of this is the stream of news. We get breaking news on any news outlet on the internet, where you can get a live-feed of what is happening at any place in the world. In our program that is also portrayed in the bar at the top, where there is a live-feed - or a stream - of what your friends are currently doing. This aims to function as a motivational factor for the user to spend offline social time.

Throbbers

Computational culture has affected how we experience anything temporal. Information is in a constant stream and we expect it to flow smoothly like water in a river. This is where our concept of slowing down the stream comes into play. In Soon's text mentions the effect of a throbber.

"*A throbber represents the speed of network traffic that is also tied to our affective states and perception of time. Emotionally, it can be frustrating to encounter buffering, as it involves interruption. Things do not flow smoothly and users become impatient in waiting for an unknown period of time or for something yet to come*" (Soon 2017, 91)

When a user has exceeded their available online time in our program, whatever may consume their time online becomes interrupted by a throbber. It becomes frustrating as it is something that interrupts the flow of information. The idea of a throbber was to "[...] *shift our expectations, modifying our willingness to wait. But the image of a buffering symbol has come to trigger mainly anxiety*" (Farman 2017). The effect of a throbber has become contradictory to its intentional purpose. This is what we use to our advantage in the Time Balancer. Loading time appears wasted to us, as something

unproductive. We don't get any new information from the waiting time. What we wish is that it will encourage people to spend more time offline. It is a critical work, challenging our perception of online time versus offline time. But instead of using the online time as something punishable, it becomes the reward, something that you've earned from being offline: *"We take the moment of waiting and give it meaning; it becomes a message of its own."* (Farman 2011).

The project as a critical work

As our program seeks to increase an awareness of keeping a balance in the amount of time spent online as well as offline, we step into an area of wanting to affect and maybe even change people's lives. We approach the fact that data streams and online activities take up more and more of our time. Without being able to comprehend this data stream, the user will reach an information overload: *"Information management becomes an overriding concern in order to keep some form of relationship with the flow of data that doesn't halt the flow, but rather allows the user to step into and out of a number of different streams in an intuitive and natural way."* (Berry 2011, 143). It is important to stress that we have no intention of having the user not to spend time online at all, but as Berry is emphasizing the user should be able to step into and out of the streams. We see our app contributing with a concept that to some degree forces the user to also step out of the stream by being offline and social with friends.

According to what earlier has been mentioned about how the fast-moving stream is shifting our attention from the past to the present and that our 'now' is getting shorter (Berry 2011, 144), we don't necessarily see our app as a direct solution to this. Just because you step out of the stream, it doesn't mean that it stops our perception of a short now. Berry describes the web as: *"A real time, flowing, dynamic stream of information — that we as users and participants can dip in and out of and whether we participate in them or simply observe we are [...] a part of this flow."* (Berry 2011, 143). But if we consider the possible consequences of our app and concept as being a well-established part of society, it might lead to a new mindset of how to live a balanced social life with both online and offline activities. If we could change the focus of not circulating it around the stream of the newest updates, but rather point our focus towards when and what we opt in and opt out. If our program could support stepping in and out of the streams in an intuitive and natural way, it might lead to a balance between the shorter 'now' and what might could be considered as the longer 'now'.

Summary

To sum up what we have discovered throughout our working process, we would like to emphasize how our perception of time is relative in regard to the backend processes in the software and the frontend interface, but also how this can have an individual perspective to it. The most essential part of the program, the counter, is controlled by multiple variables that are all relative depending on the individual user of the program. This results in a relative perception and experience of the visual outcome in the form of our app interface.

Furthermore, we stress that the data streams of our computational environment have changed the structure of how we spend our time online. We experience an overload of information which lead us to come up with a proposal of how we're able to step in and out of the streams.

Bibliography

- Berry, David. 2011. “Real-time Streams”, in *The Philosophy of Software*, pp. 142-171. Palgrave Mcmillan.
- Farman, Jason. 2017. "Fidget Spinners — Real Life". *Real Life*. http://reallifemag.com/fidget-spinners/.
- Soon, Winnie. 2017. "Executing Micro-temporality."in *Executing Practices*. Eds. Helen P, Eric S and Magda T. C., 89-102. Autonomedia.