

# Streams

By Kristine, Emma O, Emma P & Maria B

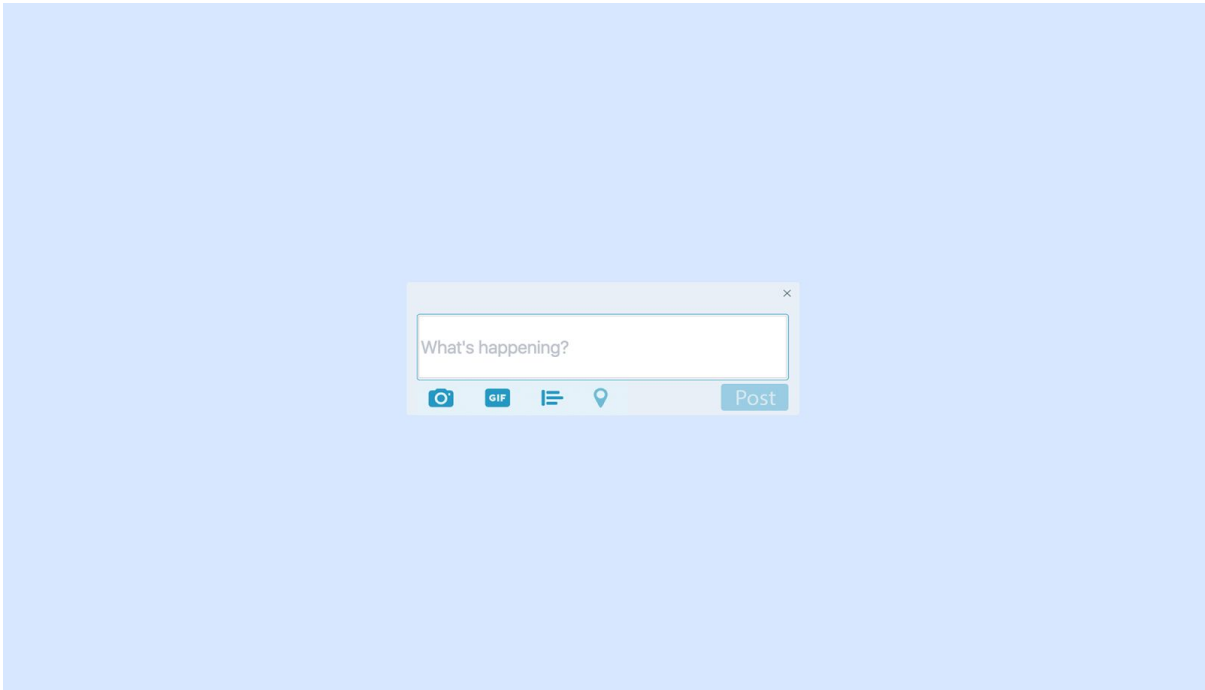
Our project is based on the chapter *Real-Time Streams* in David Berry's publication *The Philosophy of Software - Code and Mediation in the Digital Age* from 2011. The chapter elaborates the term real-time streams and later discusses the impact these streams have on society, and it is this particular discussion of Berry's that we have decided to implement in our project.

During Berry's discussion concerning the Real-Time Streams he mentions something he calls *the riparian citizen*, a term based on the philosophy of Søren Kirkegaard (2011: 144-145). Berry uses 'riparian' to refer to the act of watching a flow of stream go by. By adding Kirkegaard's view on the impact the rise of the mass media has on the citizens of society, a view claiming that "(...) *Not a single one of those who belong to the public has an essential engagement with anything.*" (2011: 144), Berry manages to compose the perception of the riparian citizen. In other words, the riparian citizen is a resident in a new kind of public created by these new streams, the real-time streams, and the citizens are continually watching the flow of data, or delegating the supervision of the data to a technical device to do so on their behalf (2011: 144).

Later in the chapter Berry refers to Twitter and in order to illustrate the complexity of Twitter he mentions a quote by Borthwick. The quote encourages you to think about Twitter as a rope of information, "*and on the outset you assume you can hold on to the rope (...) then at some point, as the number of people you follow and follow you rises - your hands begin to burn. You realize you can't hold the rope, you need to just let go and observe the rope.*" (2011: 163, Borthwick). This relates to our project that illustrates both Berry's perception of the riparian citizen and Borthwick's metaphor of data as a rope, in order to illustrate temporalities.





Given that our work expresses Berry's notion and Borthwick's metaphor, our work thereby expresses how we participate with, add to and observe the stream of data. Therefore the project depicts how the user of data activates the parasite of technology that Berry mentions, which we are now forced to live with and unable to get rid of (2011: 170-171). To completely escape the parasite and the accelerating stream of data, one would have to give up all technology and physically move away from the modern world and become a hermit.

Additionally our work seeks to portray Borthwick's concern with how: "*The cacophony of the crowd starts to erase the past and affirm the present (...) everything is out there but with little sense of priority of ability to find it becomes like a mythical library - it's there but we can't access it.*" (2011: 147, Borthwick). Because of the deluge of data Borthwick is worried that the history will be erased for the benefit of the now. Our work illustrates the society's production of the now and the user's addition to and forced observation of it.

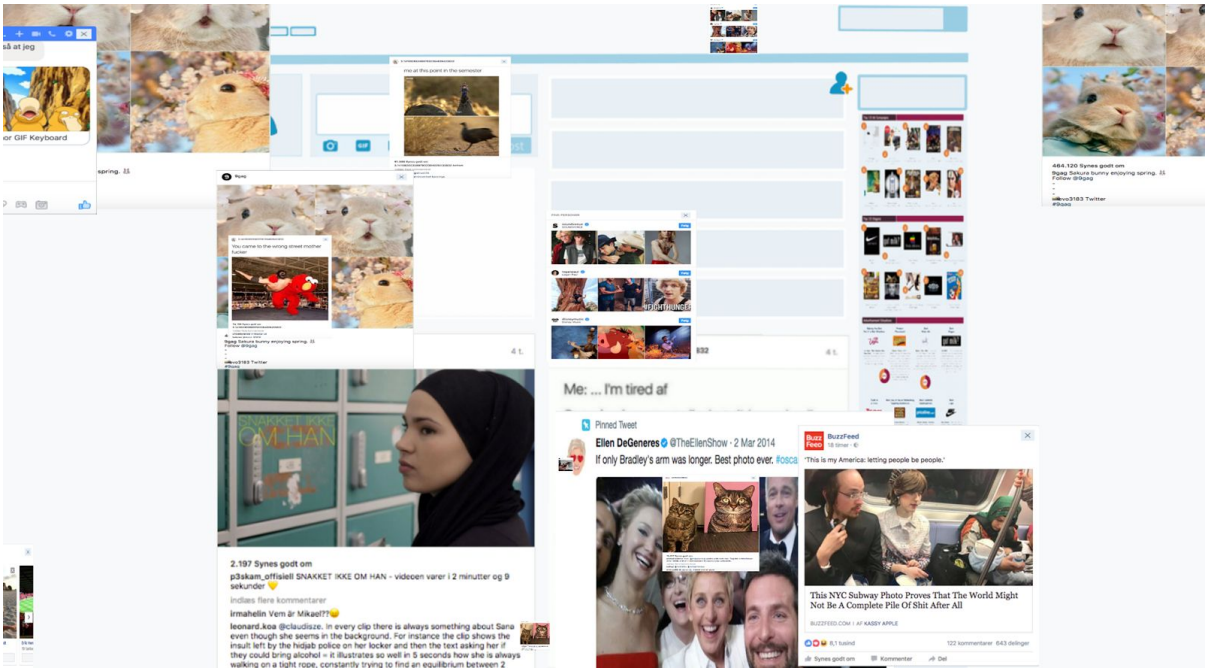


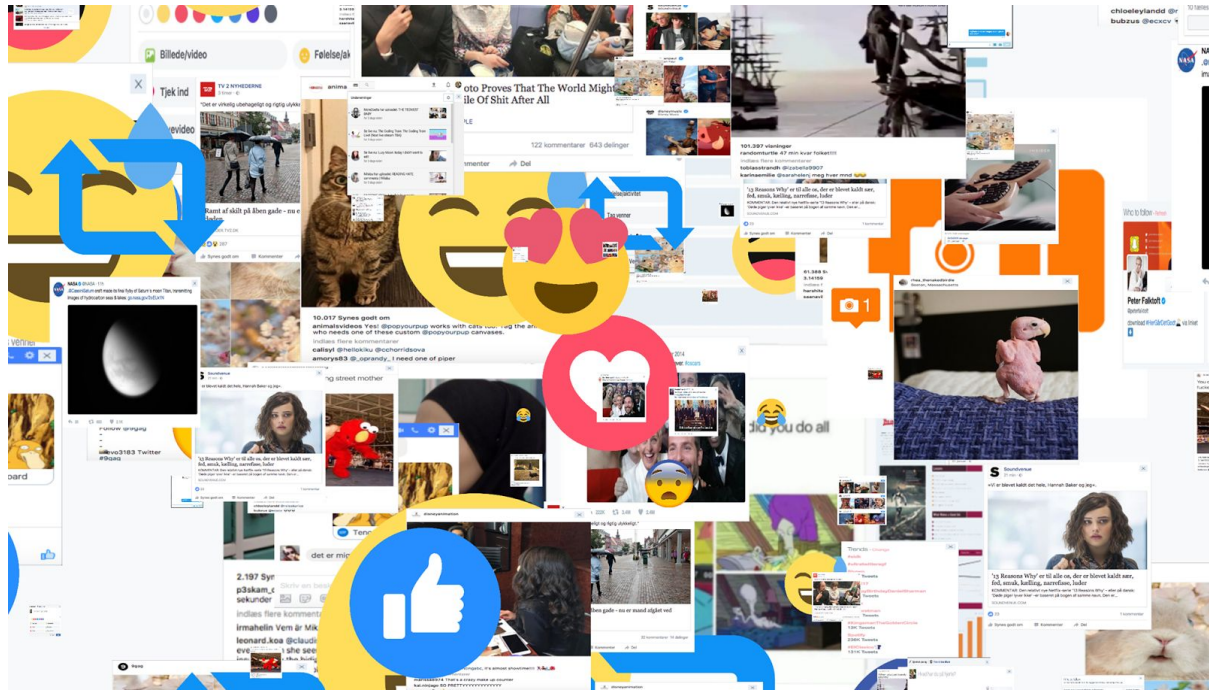
×

What's happening?



Post





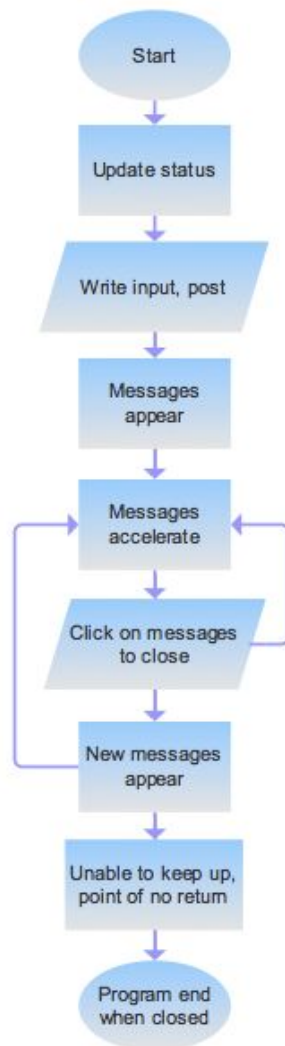
At the beginning of the programme the user is faced with a status update area with the opportunity to write something within that area. If the user decides to write something and then press *Post* the programme will start to display posts from all sorts of news feeds. The user is able to get rid of the posts by clicking on them with the mouse, which will make them disappear. At first the acceleration of the displaying of the posts is slow, but the speed gradually increases until finally the user is unable to keep up, and eventually the user have to sit back and observe the stream of posts pass by.

We decided to depict a social media site at the beginning of the programme, specifically a neutral site with no logos or recognizable features in order for the site to be all or any one of the recognised social network sites. The user has the opportunity to write an update and post it to the programme's news feed. The site consists of an image as background with a textbox created with the DOM library placed on top of the background. The *Post* button is a defined area in the mousepressed function. When the user clicks the *Post* button, they are transported to another part of the media site, resembling a news feed, with their own post emerging in their feed. The switch between the two screens is controlled by a boolean function where one screen is "true" and the other screen is "false".

Immediately after the user's post is shown, it is followed by other posts, generated by the programme. These posts include updates, images, emojis, likes etcetera, all accompanied by the notification sound-effect which is generated by a counter. The posts are called in an array of images. Their position is defined by a function which spawns the images in random locations on the screen. These generated posts move towards the four corners of the screen by dividing the object with its position according to width and height, thus moving their position. Simultaneously the posts appear to be getting closer by increasing in size relative to their position on the map.

As the posts appear the user is able to delete these posts by clicking on them and thereby remove them from blocking out the news feed. This is achieved by a function that checks if the mouse is within the image borders and then a mousepressed function to remove the image when it is clicked on. The image is then deleted from the array. However the posts keep spawning, multiplying and accelerating in speed. This is accomplished by a function that generates a new image randomly from the array, until a max of 500 images is reached. By 500 images the programme recycles the same images, in order to avoid overloading the programme. The frame rate starts at 10 and increases until it reaches 1000, the image speed will increase by 0,1 per frame. The user will at some point no longer be able to keep up with the programme and will be unable to close all posts, thereby the posts will flood the entire screen. Eventually the user will be forced to give up and simply watch the stream of post come towards them.

## Flowchart



Initially the programme may resemble a sheer demonstration of David Berry's theoretical notion of data as streams. While the mousepressed function makes it possible to delete the accelerating posts serve to prove Berry's point that we as riparian citizens are unable to keep up with the data streams, it also serves as our critical view on his notion.

Berry argues that the riparian citizen can overcome the filtration load, the deluge of data, through: "(...) *new computational abilities for them* [the riparian citizens] *to make sense of their lives (...) and to interact with both other people and the technologies that make up the datascape of the real-time web.*" (2011: 144) and additionally that these new computational abilities will be provided by new technical devices. Therefore according to Berry's opinion the riparian citizen is able to surpass the filtration load of data if he or she obtains these so called new computational abilities, but what does these new abilities contain? Even if every single citizen in the riparian community Berry describes becomes able to understand code and programming, would they be able to keep up with the data deluge? One could argue that if all citizens in the contemporary world of data streams were able to look beyond both the production and filtration of data, and master their own data creation, the amount of data would only increase and develop - in continuation of this reflection, one could consider whether or not it would be humanly possible to keep up with the stream of data, if that stream were to increase in both amount, speed, acceleration and complexity, even with these newly acquired capabilities in the field.

This reflection on Berry's notion and argument is what the mousepressed function hopefully consists of apart from simply demonstrating his statements.

## References

Berry, D. *Real-Time Streams* (2011) in *The Philosophy of Software: Code and Mediation in the Digital Age*, Palgrave, 2011, pp. 142-171.