Isha Mahadalkar
12th October 2022

JOY/FRUSTRATION ANALYSIS

Part 1: Reading Response

Paper:

The paper makes a compelling argument for fast-tracking the innovation of creativity support tools. It's not a far-fetched thought that for an artist to produce exemplary work they need to be dutifully supported, not hindered, by the tools that they are using. One of the first questions that I ask myself as a software engineer when designing a new application is "What is the function of the tool?". Simply, what problem am I trying to solve with this piece of software and additionally, "who are my users and what are their needs that I am trying to meet?".

I agree to a certain extent with Scheiderman that the tools should be designed with "low thresholds, high ceilings, and wide walls". It is important for the tool to be easily accessible to novices to not scare them away before they even try to learn more about the technology but at the same time the tool should be an adequate resource for power users. But, at the same time it can be one of those things that is easily said but difficult to implement in practice. There will be some projects, like interactive art projects, which are just not feasible to implement using just one type of tool. Even the prototype that we presented in class, we started off using Excel for cleaning and understanding the data, then Tableau for playing around with the data, and creating the initial static chart design but we soon found out that we would need to shift to Processing to do the actual interactive aspect of the visualization. I feel that there are some tools which are just more suitable for a particular task that you are trying to do and putting the onus on just *one* tool to satisfy all the requirements of a user is not viable. Having said that I do believe that there should be some tools which have a wide variety of features that can help novice and power users to solve most of their problems which we see with tools like Final Cut Pro, Photoshop, Jupyter Notebooks etc.

Now talking about what tools should be considered as a CST we would need to consider the definition of a CST. According to Schneiderman, "Creativity support tools extend users' capability to make discoveries or inventions from early stages of gathering information, hypothesis generation, and initial production, through the later stage of refinement, validation, and dissemination." It makes perfect sense to consider most of the tools in the list as CSTs like MATLAB, Processing or Visual Studio. It becomes a bit ambiguous when we start thinking about tools like YouTube or Google Maps. If these are considered as CSTs then can we think of applications like Instagram, Pinterest, or even just plain Google Search as CSTs? If they supplement your creative process, then I guess we can. Do we get inspiration from a post we see on social media or a graphic we come across on Pinterest which might drive us to make creative decisions in the project then I think we can add these tools to the list of CSTs as well. Personally,

if I am working on a research paper I use Google, Wikipedia, and Google Scholar. If I am working on a software project, then I gravitate towards XCode, Visual Studio, GitHub, and plain old Google. But I take inspiration from everywhere, Instagram, Pinterest, YouTube, Redditt and Twitter. Moving away from the list, we have GitHub which is an industry standard in version control, access control, bug tracking, and just plain sharing and collaborating on a software project. We have Jupyter Notebooks for interactive computing across multiple programming languages. Users can compile all aspects of a data project, the planning, documentation, code and the results, in one place making it easier to show the entire process of a project to your intended audience. Or even ArcGIS, a web-based mapping software to to build interactive web apps. We have a lot of newer cutting-edge technologies like Editor X which is a fully collaborative website creation platform for designers and agencies, Editor X helps creative professionals to streamline the design process using real-time collaboration. Or SparkAR which lets creators share augmented reality experiences and has easy to use templates and asset libraries which makes it easy for novices.

Shneiderman summarizes three definitions from different "schools" of thought regarding creativity: structuralists, inspirationalists, and situationists. While all three make compelling arguments I gravitate towards the inspirationalists' theory making the most sense to me. I think creativity comes from a place of chaos rather than structure. And even though collaboration is a wonderful resource into learning from others it was not an absolute necessity to being creative. Sometimes, you get the most creative ideas when you are doing something completely random and unrelated to the task at hand. Getting out of your own head, taking time away from the task, meditating or even taking a walk in the park can sometimes reboot your brain. It's like when I am stuck trying to find a bug in my code for hours on end, sometimes just taking time away and listening to some music or just having a random conversation about a random topic can help me get a fresh perspective. Sometimes, just seeing an aesthetic photo or a well-directed piece of film can make me more creative.

Moving on to the themes explored in the Forbes paper, I think the four themes inherently complement each other and also build off one another. Currently, the research project that I am working on involves migrating an already built application/tool to a different platform to increase usability and essentially "bettering" it. I think it can fit in perfectly in the role of *augmentation* where I am trying to make my tool more accessible and improving it by adding more interactive elements to it. I don't think it falls into the generation theme just because of the nature of the project. I can look at it like provocating a new element into the application, figuring out areas which are not that explored and increasing the scope of the project. And lastly, I think I can extend it to mediation where I am trying to find common ground between pure software engineering and data vis research how knowledge from one discipline facilitate and translate into the other to aid me in answering my research question.

Part 2: Joy/Frustration Analysis

For the interactive visualization I chose "Omana's 7 Days in April" by Rebecca Ruige and Sean Hongsheng Zhai. This project translates a bird's migration journey from South Arabian Peninsula to North Asia. They use landscape pictures of the terrain that the bird flew over and also create haikus by performing natural language processing using the flying and environmental data.


Link: [http://rebeccaxu.com/]
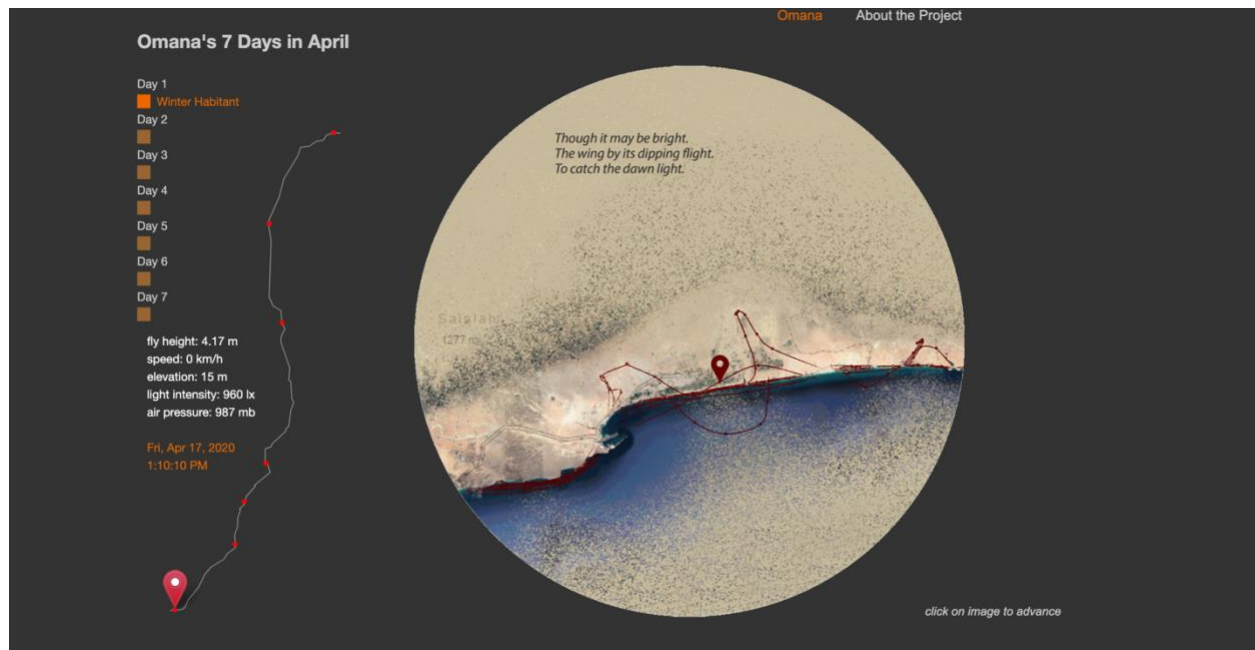
I really like the concept of this visualization. I also really enjoy the aesthetic which the artist presents. I don't know how to explain it but it gives the vibe of Arabian-Asian landscape with the colors and the font used. We learn the path taken by the bird which is shown in the circle on each day, we also have information and images of the landscape as well as a haiku which is formed using the environment data. We can click on the screen to move onto the next day which changes the data presented to us. I think what's inspiring about this is that it kind of puts the bird's journey into perspective. We can actually see the landscape with the image, understand the conditions with the information given in the legend and *feel* the journey with the haiku.

The one aspect I find a little frustrating is that I don't have granular information about the cities visited by the bird. It would also be great if the map itself was interactive (zoom in or out) with more information about the roads taken or the cities visited which we could move around. If I could redesign it I would have a time scale slider at the bottom where the visualization would have the map of the landscape which would change with time and create the path in real time. The users would also be able to jump around the time scale to a certain point as well as move around the map. But, the color scale, aesthetics and haiku would remain the same because even though they don't really add any real statistical or informational value but I think they play a great role in grabbing attention of the audience and making the visualization more entertaining.

For this project they used Python and Textblob to manipulate the data and perform the natural language processing while Folium, QGIS and three.js was used for interactive graphics and visual encodings.

For the static visualization I chose Wonyoung So and Fábio Duarte's "Cartographers of North Korea". This piece uses Volunteered Geographic Information (VGI) to map the previously unmapped areas using various technological advancements like Web 2.0 and satellite imagery. They used these tools to map North Korea and highlight its political aspects using cartography. They argue that crowdsourced efforts can make good the dearth of knowledge resulting from the physical, cultural and political barriers associated with uncharted territories.
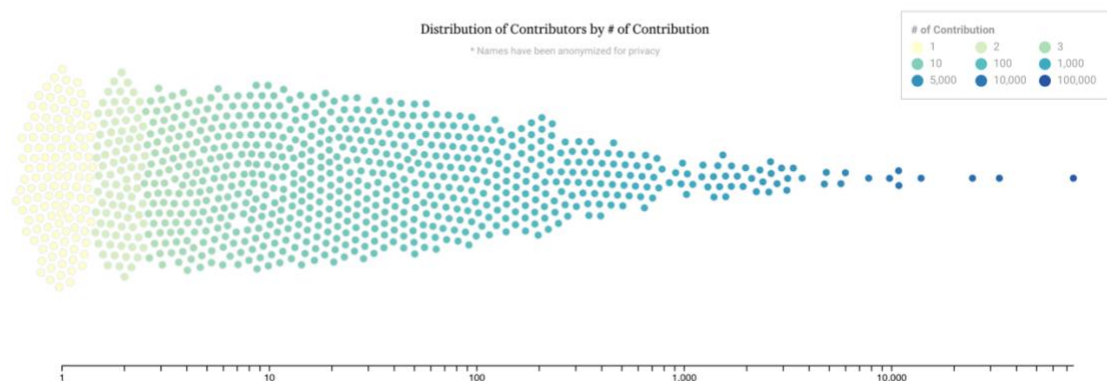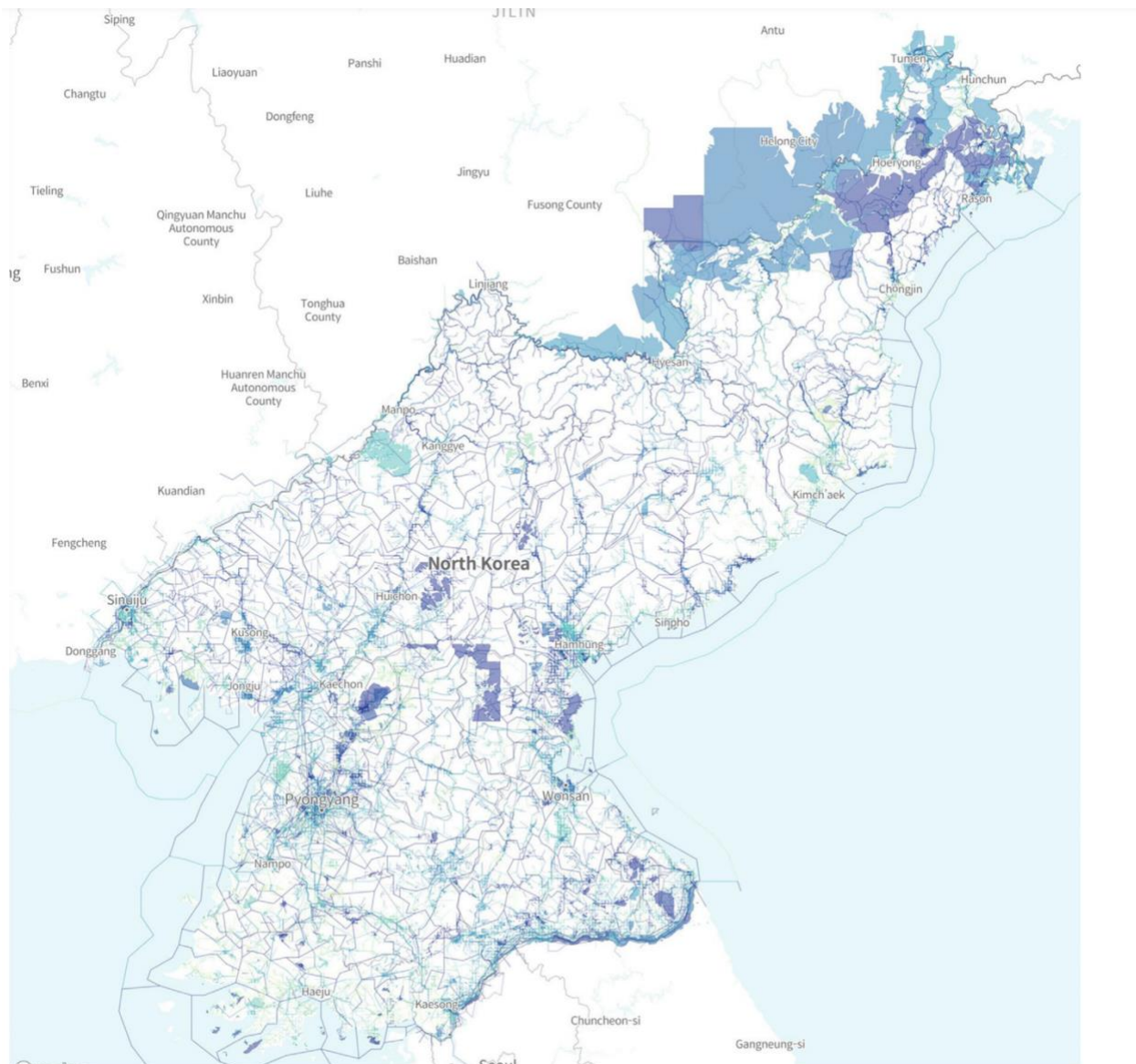
Fig. 2. OSM data for North Korea in October 2018.

I think the most novel aspect about this visualization is the actual map of the streets of North Korea just because we are not supposed to know this information. We can learn about the cities, the neighborhoods, the roads and the layout of the country and they also give us information about the distribution and the number of contributors. It's a little frustrating that I can't interact with the data, I can't filter the contributions based on the num or the region or even zoom in or out of the map of the streets. I wish there were some pictures that we could see if we clicked on the major cities but that seems really out of reach just based on political reasons. I would also like to add more information about the cities or areas itself, like local prominent places, statues, restaurants, military, and test sites, etc.

I also really like the shades of blue that they have used for the color scheme on the map, it kind of gives it a *cold and dreary* touch which we can associate with the country. They used OSM (Open Source Mapping) for the street views and for the distribution charts they might've just used Tableau or Processing.