Data Sketches

NADIEH BREMER
SHIRLEY WU

“Shirley and Nadieh are honest, entertaining, and insightful in their retrospectives, and their openness and humility make the practice less intimidating, inviting newcomers to get started.”

– Mike Bostock, Creator of D3.js and Founder of Observable

“Written in an approachable, first-person angle, this book is a delightful behind-the-scenes look at the process for creating any sophisticated data visualization. They make it seem so easy you will want to start on your own project right away.”

– Manuel Lima, Author, RSA Fellow and Founder of VisualComplexity.com

“The work of Nadieh Bremer and Shirley Wu is some of the most beautiful and exciting data visualization work being done today.”

– Elijah Meeks, Author and Executive Director, Data Visualization Society

“The Data Sketches collaboration is a glorious tour de force: two people spur each other along a remarkable spiral of visualization creativity, and let the rest of us come along for the ride!”

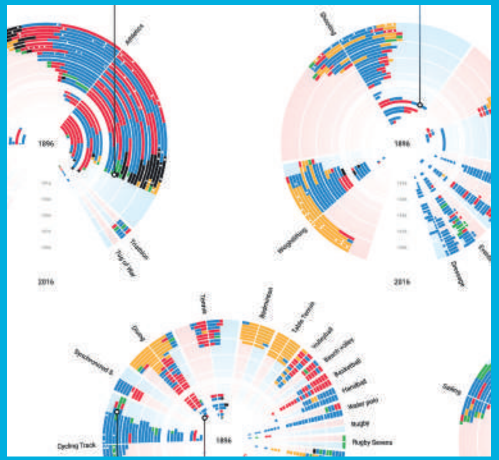
– Tamara Munzner, Author and Professor, University of British Columbia

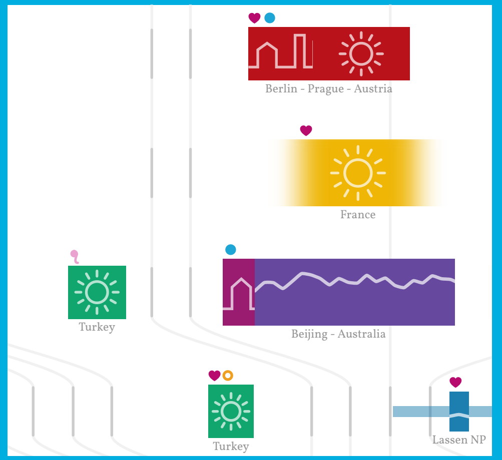
“Lay-chart readers cannot appreciate the expertise and thousands of decisions that go into a single visualization, but this wonderful behind-the-scenes peek reveals how there is never just one ‘right’ answer, but many possible answers — each of them beautiful, provocative, and shaped by the unique lens of its creator.”

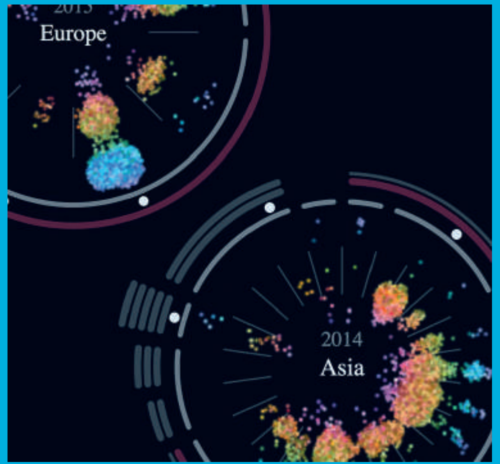
– Scott Murray, Author and Designer, O'Reilly media

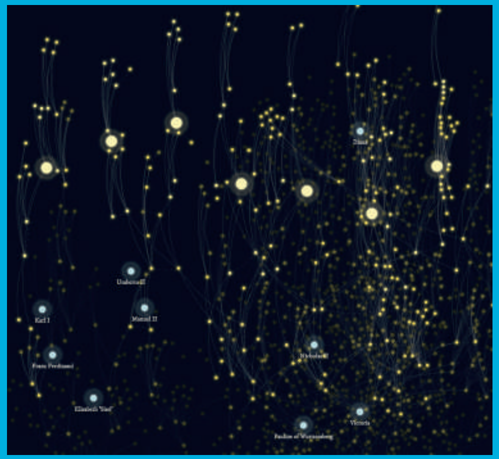
CONTENTS


Gandalf
Sam
Aragorn
Frodo
Gimli
Pippin
Merry
Boromir
Legolas

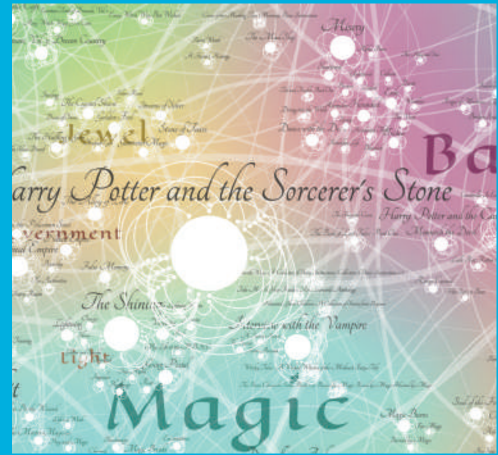


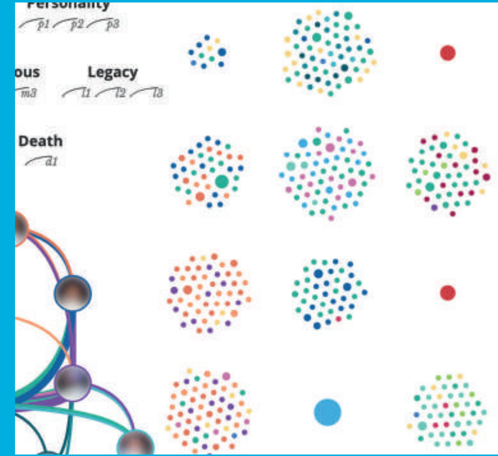

Berlin - Prague - Austria
France
Turkey
Beijing - Australia
Lassen NP

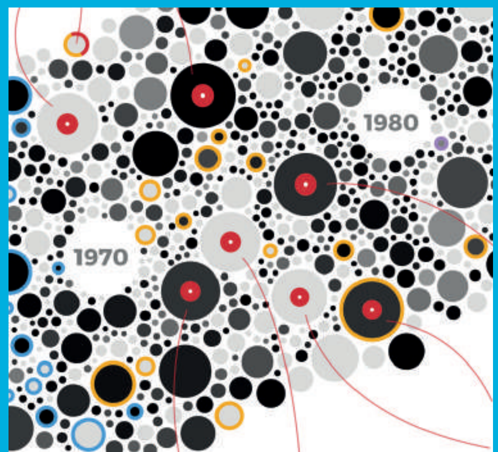

Europe
2014
Asia

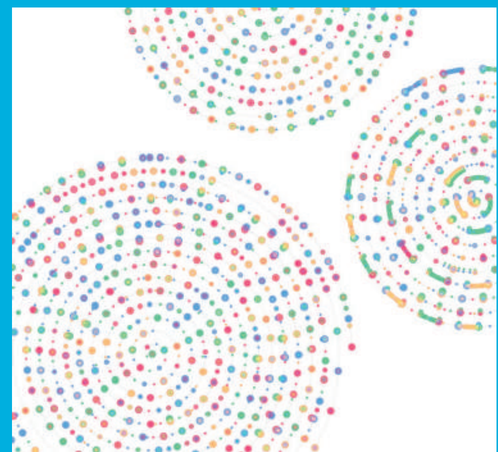


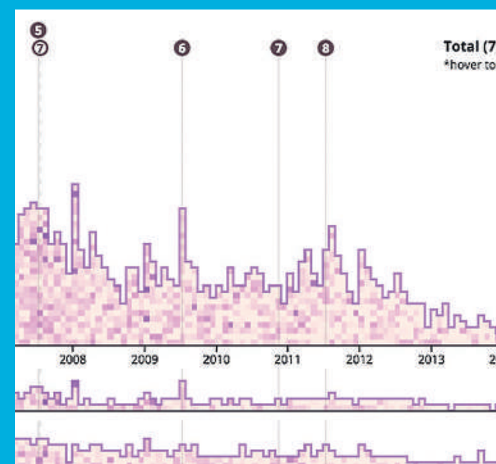

2014
2015
2016
40m
30m
20m

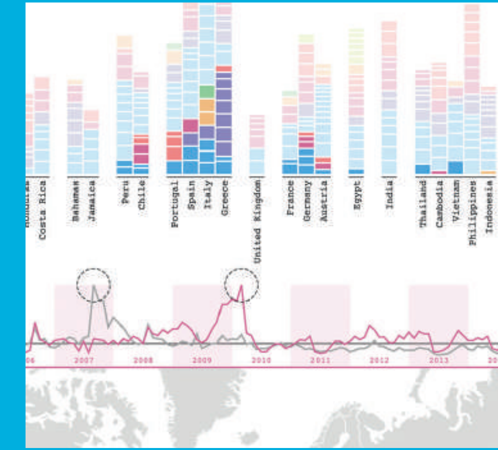


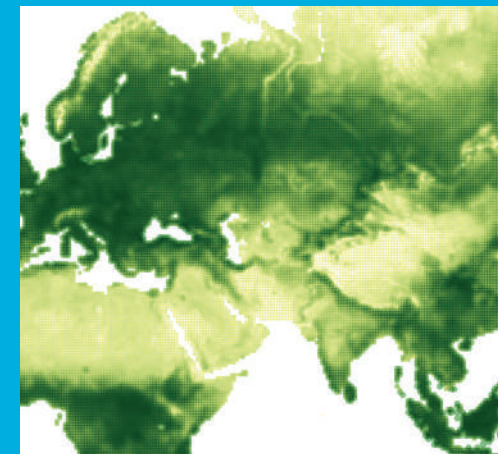


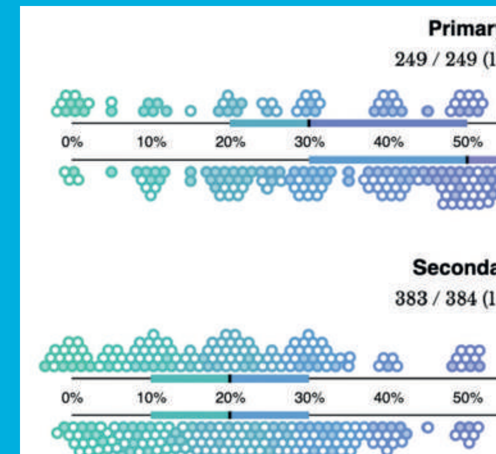


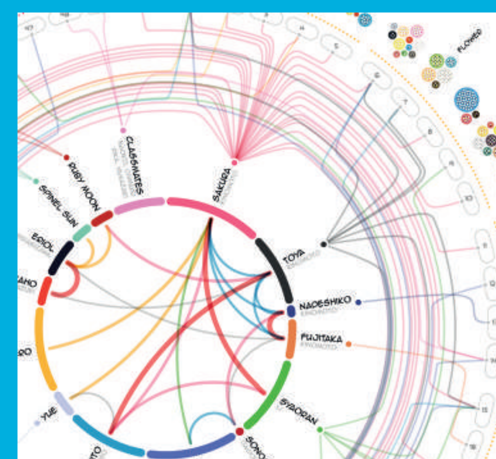












ORTHODOXY & ECCENTRICITY

Alberto Cairo

Knight Chair at the University of Miami and author of *How Charts Lie*

Orthodoxy and eccentricity are opposing but complementary forces in any field, and data visualization isn't an exception. Periods when the former prevails over the latter discourage whim, passion, and experimentation, and favor stability and continuity. When the opposite happens—when eccentrics take over—chaos and turmoil ensue, but progress becomes more rapid and invention more likely. This is a book where eccentricity abounds. That's a good thing.

The formalization and systematization of data visualization took decades and books by authors such as Jaques Bertin, John Tukey, William Cleveland, Naomi Robbins, Stephen Kosslyn, Leland Wilkinson, Tamara Munzner, and many others. It is thanks to them that we possess a common language to discuss what constitutes a well-designed chart or graph and principles that aid us when creating them. They deserve our gratitude.

What most of those authors have in common, though, is a background in statistics or the sciences, and I suspect that this has had an effect on the visual style favored for many years. Since the 1970s at least, data visualization has been governed by a vague consensus—an orthodoxy—that favors bare clarity over playfulness, simplicity over allegedly gratuitous adornments, supposed objectivity over individual expression.

As a consequence, generations of visualization designers grew up in an era of stern and pious sobriety that sadly degenerated sometimes into dismissive self-righteousness exemplified by popular slurs such as 'chart junk.'

It's time maybe not to abandon that orthodoxy outright—when the goal of a visualization is to conduct exploratory analysis, reveal insights, or inform decisions, prioritizing clarity and sticking to standard graphic forms, conventions, and practices is still good advice—but to acknowledge that other orthodoxies are possible and necessary. Visualization can be designed and experienced in various ways, by people of various backgrounds, and in various circumstances. That's why reflecting on the purpose of a visualization is paramount before we design it—or before we critique it.

Nadieh Bremer and Shirley Wu are wondrous eccentrics. Their splendid book is the product of a collaborative experimental project, *Data Sketches*, that might be one of the first exponents of an emerging visualization orthodoxy in which uniqueness is paramount and templates and conventions are seen with skepticism.

I discovered *Data Sketches* right after it was launched, back in 2016, and I was immediately enthralled, even if I couldn't understand many of its graphics. They are insanely complex and ornate, I thought, colorful, mysteriously organic in some cases, a departure from the strictures of classic graphs, charts, and maps. I felt that Nadieh and Shirley were not only pushing what was possible through technologies such as `D3.js`, but also wished to defy what was acceptable.

The book that you have in your hands reveals how Nadieh and Shirley think. This is useful. Visualization is, like written language, based on a body of symbols and a syntax that aids us in arranging those symbols to convey information. However, this system of symbols and syntax isn't rigid—again like written language—but flexible and in constant flux. That's why I've come to believe that visualization can't be taught as a set of rigid rules, but as a principled process of reasoning about how to make good decisions when it comes to what to show and how to show it.

This process ought to be informed by what we know about vision and cognitive science, rhetoric, graphic and interaction design, UX, visual arts, and many other fields. However, this knowledge shouldn't be a straitjacket. Rather, it's a foundation that opens up multiple possibilities, some more appropriate, some less, always depending on the purpose of each visualization and on its intended audience.

The education of visualization designers, whether it's formal or not, can't be based on memorizing rules, but on learning how to justify our own decisions based on ethics, aesthetics, and the incomplete but ever-expanding body of empirical evidence coming from academia. There are plenty of lengthy and detailed discussions in *Data Sketches* about how to balance out these considerations, and it's always useful to peek into the minds of great designers, if only to borrow ideas from them. Some of you will be persuaded by those discussions, and others will disagree and argue against them. That's fine. Conversation is what may help us determine whether certain novelties fail and should be discarded, or succeed and become convention. Today's eccentricity is tomorrow's orthodoxy.

Now go ahead: read, think, and discuss. And consider becoming a bit more of an eccentric.

OUR JOURNEYS INTO DATA VISUALIZATION

In the last decade, the amount of data collected has exploded, and many fields—including data visualization—have gained momentum to make sense of all this data. These data visualizations can take a wide variety of forms, such as dashboards, infographic posters, data art, and data-driven journalism. But what we enjoy the most is wielding data as a tool to explore our curiosities about the world around us. We love collecting a fun dataset, finding the insights and stories buried within it, and sharing that story in a beautiful, visual way that excites people. Our most successful projects have turned spreadsheets full of numbers into visualizations that entice people to dive in, explore, and learn all that it has to reveal.

To do this, we honed a wide variety of skills: data analysis, information design, coding, and storytelling. And because of this wide spectrum of skills (and the relative youth of the field as a whole), most data visualizers start from a variety of different backgrounds. We are no exceptions.

Nadieh's Start

In 2011, I graduated as an astronomer. I loved the topic and its gorgeous imagery, but knew that writing academic papers wasn't for me. Instead, I joined the new Analytics department of Deloitte Consulting as a data scientist. Apart from data analyses, I was creating tons of simple charts in PowerPoint, QlikView, and (mostly) R. In 2013, at a data science conference, I randomly joined an "Introduction to `D3.js`" workshop and my mind was blown by the possibilities! I didn't care that I had to learn JavaScript, CSS, and HTML; I was going to add `D3.js` to my repertoire! I still saw myself as a data scientist, though.

At another conference at the end of 2014, while waiting for Mike Freeman's talk, my eyes fell on his first slide. He called himself a Data Visualization Specialist. "Wait? What? That's a separate thing!?" It was like I was struck by lightning. I *immediately* knew that was where my true passion was, not the data analysis part. And from then on, I spent every moment of free time I had to become better in the visualization of data.

In December 2015, I joined Adyen as a full-time data visualization designer. However, after designing and creating dashboards for months, even in `D3.js`, I felt that I was still missing something; my creative side wasn't feeling fulfilled.

Shirley's Start

I loved math and art growing up. When I went to college, I studied business, but found much more enjoyment in the computer science classes I took. When I graduated in 2012, I started as a software engineer on a front-end team at Splunk, a big data company. There, I was introduced to `D3.js`, a JavaScript library for creating data visualizations on the web, and I immediately fell in love with being able to draw in the browser.

In 2013, I started frequenting the Bay Area D3 User Group, where I learned that `D3.js` was only a subset of a larger field called data visualization that people could specialize in. I loved that it was such a beautiful blend of art, math, and code—all of my favorite things.

For my second job, I joined an enterprise security start-up, specifically because they were looking for someone to specialize in creating data visualizations for their product. I grew a lot technically and loved all the data visualization aspects of my work, but not the industry I was in.

In 2016, I decided to take the leap into freelancing. I wanted to see the data-related challenges that companies faced and the problems they were trying to solve with data visualization; I wanted to find the industries I'd be the most excited to work in.

And this is where our stories converge, in early 2016...

Data Sketches, or "Shirley & Nadieh's Awesome Collaboration Marathon"

We met in April 2016 at OpenVis Conf in Boston and kept in touch when we got home (Shirley in San Francisco and Nadieh in Amsterdam). One June day, we were lamenting the fact that we'd had little free time to focus on personal data visualization projects and, as a result, hadn't completed many in the past year. But on that fateful June day, Shirley had just quit her full-time job and had plenty of free time to experiment and create. So she plucked up her courage and asked Nadieh a simple question: "Do you want to collaborate with me?" Nadieh excitedly agreed!

In a series of rapid back-and-forth messages, we defined the structure of the project: 12 months, 12 topics, 12 projects each. We would gather our own data, create our own design, and code our visualizations from scratch, while also documenting the whole process. We encouraged each other to use these opportunities to try new approaches, explore new tools, and push the boundaries of what we could create. But most of all, we promised each other that we would have lots of fun.

OpenVis Conf was an amazing annual data visualization conference in Boston that had a great blend of technical and design talks.

It's wild how dramatically this simple question and decision altered both of our careers.

We decided to call our project "Data Sketches," and went live on September 21st, 2016 with our first four visualizations on datasketch.es. We didn't think anyone beyond our friend group would care, and we certainly didn't expect the overwhelming response we got. But as it turned out, people really liked getting a behind-the-scenes look at our process, and we kept hearing how helpful and educational our write-ups were.

We first mused about a "Data Sketches" book not long after our launch, when we heard Alberto Cairo mention our project on a livestream. We were giddy with delight when we heard him say that if we ever created a book, he'd display it on his coffee table. We were enamored with the idea, but it felt like a far-away dream.

Nonetheless, we tried to talk to a few publishers. None of them seemed keen on the book we wanted to make: a beautiful coffee table book with large, indulgent images of our projects, side-by-side with our very technical process write-ups. We wanted it to be both aesthetically pleasing and educational, yet the publishers we talked to all seemed only to want one and not the other.

We were close to giving up when one summer day in 2017, Shirley reached out to Tamara Munzner for dinner during her visit to Vancouver. At the end of a long and delicious meal, Tamara asked if we had ever considered turning "Data Sketches" into a book. She wanted us to be part of her series of data visualization books, and we were more than thrilled! We had finally found an editor (and enthusiastic champion of our work) and a publisher willing to work with us to create our dream book.

Tamara gave us a renewed purpose in creating a "Data Sketches" book. She convinced us that because everything on the Internet eventually bit-rots, we needed to do it for archival purposes. We also knew that because we wanted to keep many of our online write-ups freely available, we therefore needed to make this book worth splurging on. We've tightened up our tangents, filled in gaps in explanation, and packed the book full of lessons we've learned along the way. This book is just as much about the 24 individual data visualizations we created as it is a celebration of the technical and personal growth we've gone through in this three-year journey. It is a snapshot in time to commemorate the immense impact "Data Sketches" has had on our lives, allowing us to quit our full-time jobs, launch thriving freelance careers, travel the world to talk about our work, and develop beautiful friendships.

Writing a book is hard. We knew this going in, and our original one-year project has since turned into three years. It's been a monumental three (four by the time of publishing) years, and we're so excited to hold *Data Sketches* in our hands, to flip through the pages, and see our work immortalized on paper. We want to thank you for all of your excitement and support, whether you've been with us from the very beginning, or have just picked up this book. Thank you for helping us get here.

We hope you enjoy this dream book of ours and that you have just as much fun reading it as we did working on our *Data Sketches*.

Our very first brainstorming document was titled "Shirley & Nadieh's Awesome Collaboration Marathon" before we eventually landed on "Data Sketches." Other names that we seriously considered included "Pencils&Code" and "Visual Wanderlust."

We've added the time frame of each data visualization at the beginning of each chapter to give context.

Shirley and Nadieh, on the left and right respectively, together in real life for a week in San Francisco during September 2017.

Nadieh Bremer is a graduated astronomer, turned data scientist, turned freelancing data visualization designer. She's worked for companies such as Google, UNESCO, Scientific American, and the New York Times. As 2017's "Best Individual" in the "Information Is Beautiful" Awards, she focuses on uniquely-crafted data visualizations that are both effective and visually appealing for print and online.

VisualCinnamon.com

Shirley Wu is an award-winning creative focused on data-driven art and visualizations. She has worked with clients such as Google, The Guardian, SFMOMA, and NBC Universal to develop custom, highly interactive data visualizations. She combines her love for art, math, and code into colorful, compelling narratives that push the boundaries of the web.

shirleywu.studio

HOW TO READ THIS BOOK

Each chapter represents one of the 12 topics that we've visualized with *Data Sketches*, and each chapter consists of both of our individual write-ups.

Period during which we worked on this project

Topic

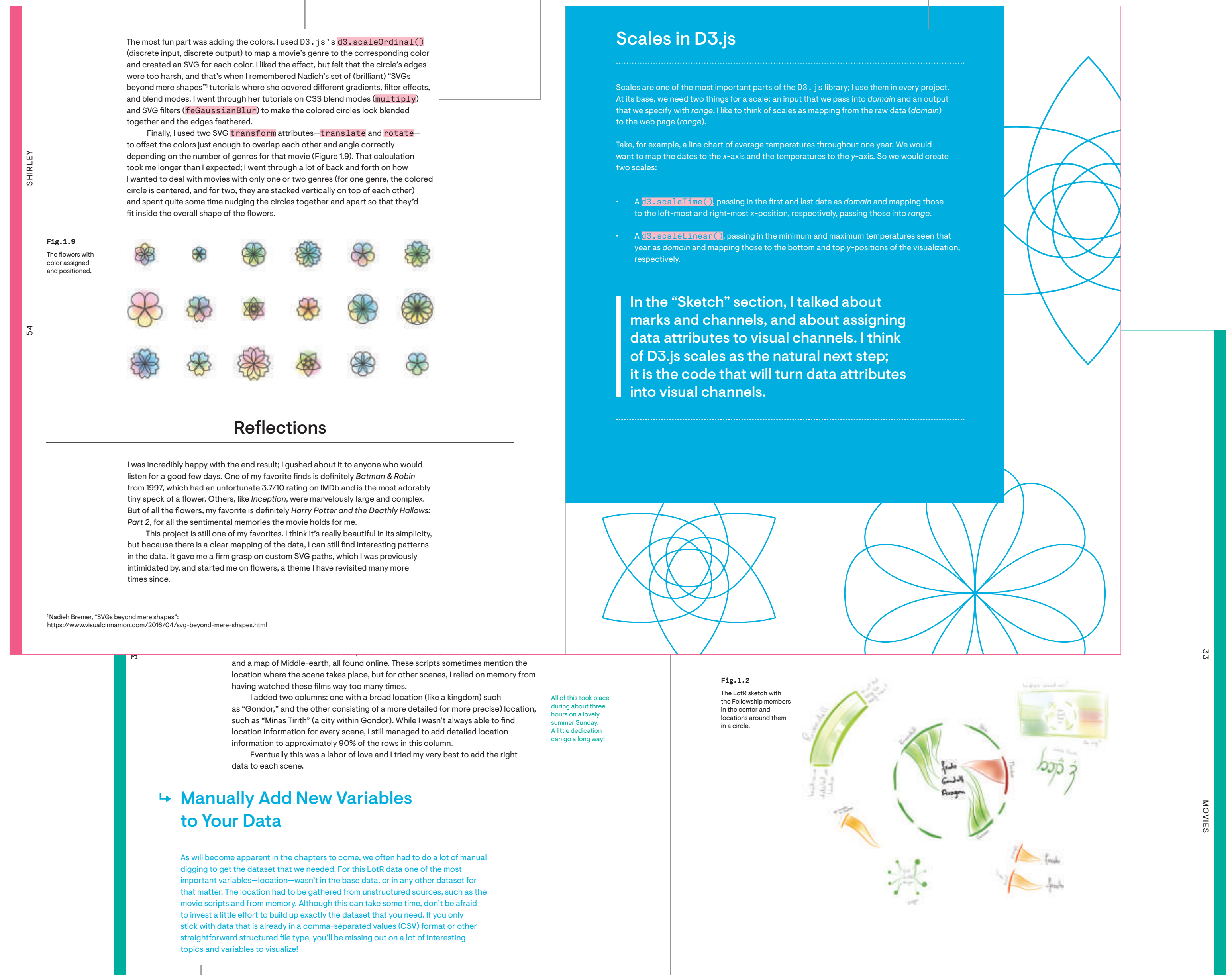
Each write-up has an introduction, **data, sketch, code,** and **reflections** (textual) section and ends with several pages showing the final visual. The data, sketch, and code titles refer to what we were doing at that point in the project, not literally what is shared in that section (i.e., we've avoided sharing actual code snippets in the code section).

The write-ups can be distinguished between Shirley and Nadieh through the base color. (Shirley's color is pink and Nadieh's color is green.)

Technology, referring to any kind of package or library, is displayed in this monotype font

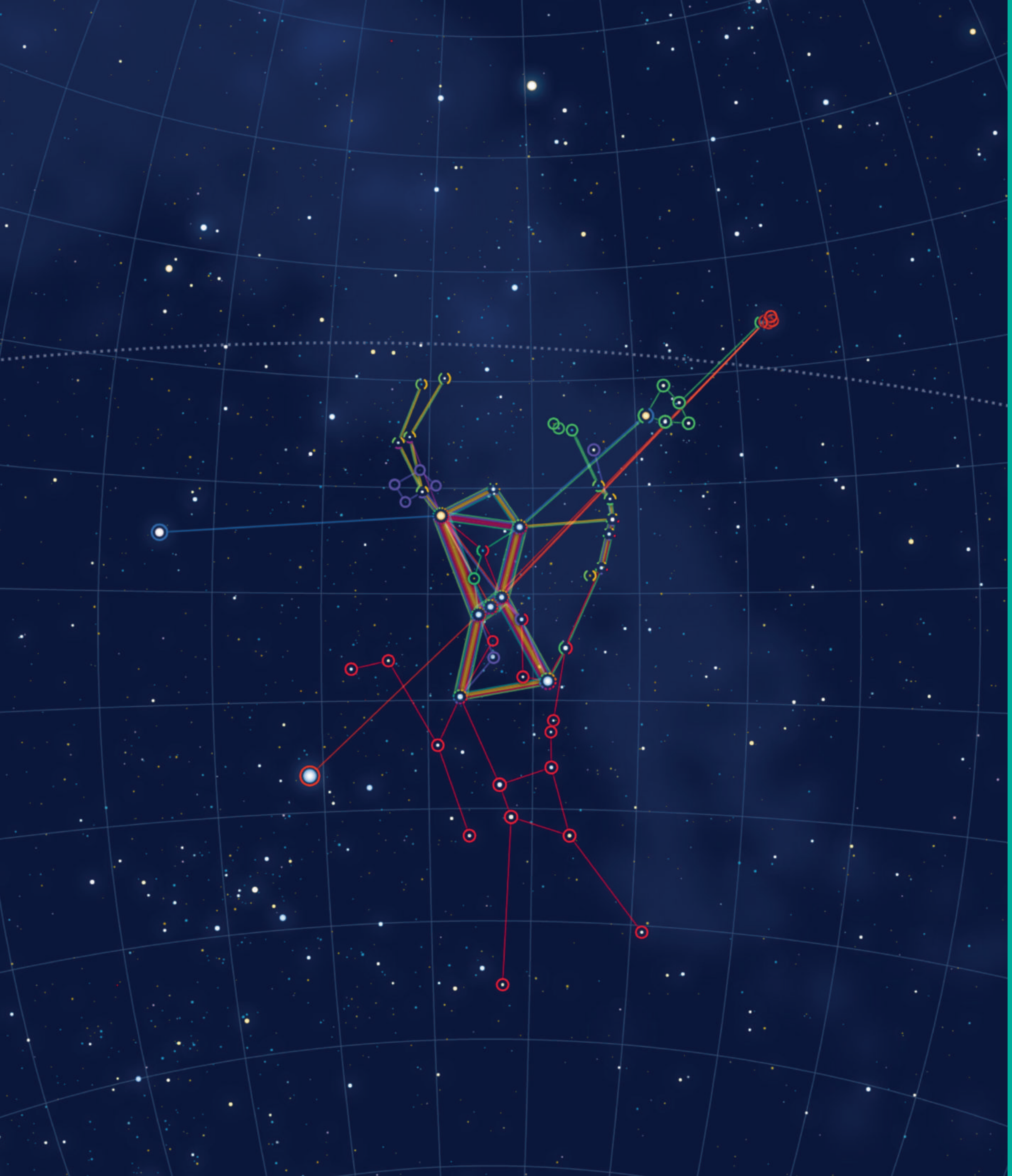
Code is displayed with a light pink or green highlight

Technical lessons have a blue background



General lessons are written in blue. Multiple chapters may refer to the same lesson, so all lessons are collected at the end of the book for quick reference.

MYTHS & LEGENDS



Figures in the Sky

NADIEH

This project took a *long* time to figure out topic-wise and to create; I even finished the next scheduled project (about "Fearless") before this one. There were several avenues that Shirley and I investigated (Cinderella, Disney), but they didn't pan out. And so, many months after my previous project, while at OpenVisConf in Paris, I decided to look for completely different ideas. The talks definitely inspired me, especially one about Google's Quickdraw dataset. I thought, maybe I'd make something about the "mythical" creatures from the Quickdraw word list and how they're drawn, like dragons and mermaids? Something about dragons in general? Or about myths from many different cultures and their timelines and similarities? Unfortunately, that would probably mean a lot of manual data gathering. But myths across cultures ... that suddenly reminded me of constellations! Many constellations have been named after characters from certain myths and legends. My favorite constellations are Orion and the Swan (officially known as Cygnus). But what did other cultures make of those same stars? What shapes and figures did *they* see in the same sky?

> **That idea sparked a feeling of enthusiasm and wonder in me in such a way that I knew it felt right.**

As an astronomer, it also felt kind of appropriate to have my final *Data Sketches* project to be connected to actual stars.

Data

Of course, that idea still hinged on data availability. I thought that the subject I had chosen would be specific enough for Google. But alas, trying to search for constellation data was heavily intermixed with astrology. (—_—*;)

I found some promising information about the "modern" 88 constellations, but nothing about constellations from multiple cultures. That is, until I came across Stellarium, an amazing open source 3D planetarium software and all its data can be accessed on GitHub. The giant cherry on the cake is a folder[1] called "skycultures" which contains information on constellations from ±25 different cultures from across the world, including Aztec, Hawaiian, Japanese, Navajo, and many more. This data was *exactly* what I needed, but it wasn't available in a simple CSV format, nor in the shape that I wanted for my visualization. Luckily, Stellarium has a very extensive user guide[2] that explains exactly how to interpret the data.

For example, Figure 11.1 shows the data to create "stick figures," or the lines between stars. Each row is one constellation, with the constellation's ID listed in the beginning, followed by the number of connections (lines) in the constellation. After that come the so-called Hipparcos (HIP) star IDs. Each pair of HIP IDs defines a line between those two stars.

I converted these files into something very similar to the typical *links* file of a network, with a `source_id` and `target_id` per row, having one row for each line to draw in the stick figure/constellation.

I pulled the full names of the constellations from a different Stellarium dataset and created another file that contains all the constellation IDs that a specific star is connected to. However, there was still one important "subject" that I was missing in terms of data: the stars.

Thankfully, that's a dataset I'm already familiar with and have used in a few other astronomer themed visualizations. The HYG database[3] contains lots of information about many, many stars. From that database I took the *right ascension* and *declination* so I could place the stars on a map (you can think of these as the *latitude* and *longitude* of the sky). But I needed more information, such as the HIP ID, to connect them to the constellation data from Stellarium. I also found the *apparent magnitude*, which represents how bright the star looks, to use as a star's size, and finally the star's *color index* to get an effective temperature, which could then be mapped to a color for the stars. It would be a shame not to color the stars the way they actually appear to us.

The HIP ID was the unique key that made it possible to link the constellation data from Stellarium to the star data from the HYG database.

Fig.11.1

Stellarium's data to create "stick figures" between the stars.

```
And 5  677 3092 3092 5447 9640 5447 5447 4436 4436 3881
Scl 3  116231 4577 4577 115102 115102 116231
Ara 7  88714 85792 85792 83081 83081 82363 82363 85727 85727 85267 85267 85258 85258 88714
Lib 5  77853 76333 76333 74785 74785 72622 72622 73714 73714 76333
Cet 20  10324 11484 8102 3419 3419 1562 3419 5364 5364 6537 6537 8645 8645 11345 11345 12390 12390 12770 12770 11783 1178
Ari 3  13209 9884 9884 8903 8903 8832
Sct 5  92175 92202 92202 92814 92814 90595 90595 91117 91117 92175
Pyx 2  42515 42828 42828 43409
Boo 9  71795 69673 69673 72105 72105 74666 74666 73555 73555 71075 71075 71053 71053 69673 69673 67927 67927 67459
Cae 2  21060 21770 21770 21861
Cha 2  40702 51839 51839 60000
Cnc 5  43103 42806 42806 40843 42806 42911 42911 40526 42911 44066
Cap 9  100064 100345 100345 104139 104139 105515 105515 106985 106985 107556 105515 105881 105881 104139 100345 102485 10
Car 14  45238 50099 50099 52419 52419 52468 52468 54463 54463 53253 53253 51232 51232 50371 50371 45556 42568 41037 41037
Cas 4  8886 6686 6686 4427 4427 3179 3179 746
Cen 16  71683 68702 68702 66657 66657 68002 68002 68282 68282 67472 67472 67464 67464 65936 65936 65109 67464 68933 67472
Cep 6  109492 112724 112724 106032 106032 105199 105199 109492 112724 116727 116727 106032
Com 2  64241 64394 64394 60742
Cvn 1  61317 63125
```

[1] Stellarium skycultures folder: https://github.com/Stellarium/stellarium/tree/master/skycultures
[2] Stellarium user guide: https://github.com/Stellarium/stellarium/releases/download/v0.19.1/stellarium_user_guide-0.19.1-1.pdf
[3] HYG database: https://github.com/astronexus/HYG-Database

To stay true to how we see the night sky, I filtered the stars to only include those that are bright enough to be seen by the naked eye, which is an apparent magnitude smaller than 6.5.

In astronomy the smaller the apparent magnitude, the brighter the star appears to us.

In addition to sky maps with constellations, I also wanted to display something more "statistical" using a bigger set of data. What sparked my interest was seeing how the data looked when I plotted a star's brightness versus the number of constellations that each star is used in. Was there a trend? If so, which stars deviated and why? I made a quick plot in R using `ggplot2` (Figure 11.2) that revealed some interesting insights, specifically, insights around which stars deviated from the general trend of "the brighter a star, the more constellations that use it."

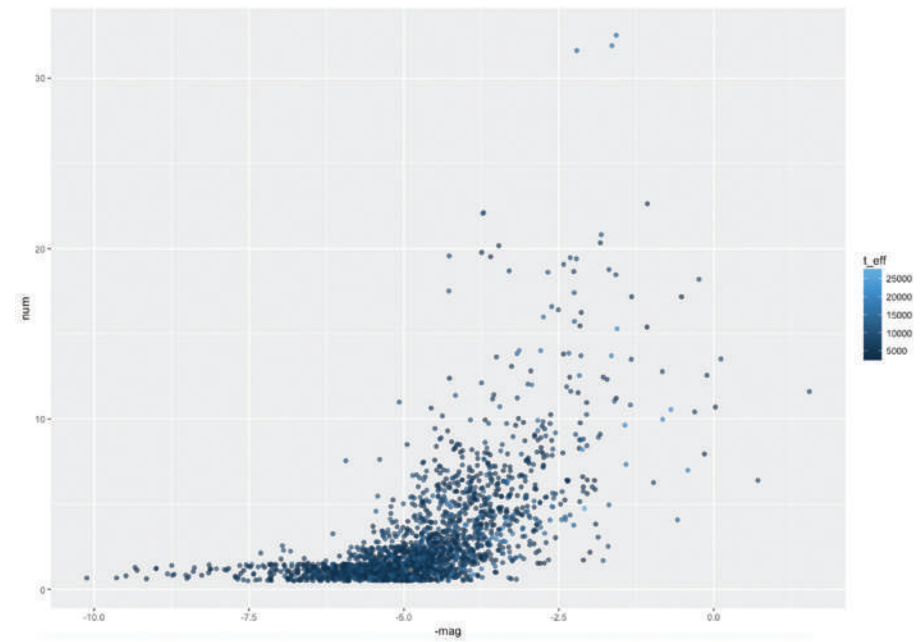


Fig.11.2

A scatter plot made in R showing apparent magnitude versus the number of constellations a star is part of, for approximately 2,200 stars.

However, while investigating this scatter plot more closely, I noticed that my star data was missing many proper star names. Almost all nine stars of the Pleiades were not named! I searched for a bigger list of named stars and found a sort-of official list of ±350 stars on Wikipedia.[4]

By proper names of stars I mean their popular/common names instead of their catalogue IDs, such as the star names Betelgeuse and Sirius.

However, these only contained the names themselves, not the HIP IDs needed to connect them to my data. Thankfully, there is a website called the Universe Guide[5] where the URLs are based on the star's name, while the page itself contains the HIP ID in the HTML's `h1` header (title) of the page. I therefore used the `rvest` package in R to download the Universe Guide page of all of the stars on the wiki list, grabbed the `h1` from the HTML, and only kept the HIP `id`. I only had to do a few manual lookups for names that didn't return results from the Universe Guide through my script. Finally, I merged this "proper star names" dataset into the original HYG dataset for a much more complete set of star names.

I copied the Wikipedia list of 350 star names into Excel using its "data from web" import option.

A final note about the data: there are no officially declared constellation figures. There are indeed 88 official constellations, but the only thing that is recorded is what area of the sky that constellation takes up (kind of like how the US states divide up the land). There is no official consensus on how the stick figure part of the constellation should be drawn. I've therefore decided to use the data from Stellarium as my "single source of truth."

[4] List of proper names of stars: https://en.wikipedia.org/wiki/List_of_proper_names_of_stars
[5] Universe Guide website: https://www.universeguide.com/star/atlas

Sketch

This project was pretty light on actual "design" sketching. That's mostly because the basic idea was quite simple: I would focus on one star and visualize all the constellations that use that star. I created a sketch that surrounded a star with a donut-like mini chart that would show the constellations it appeared in. You can see the tiny sketch from my brainstorm in Figure 11.3 (bottom of the left page).

I wanted the star map to look like a combination between current and ancient sky maps. An example of the latter is the exquisite illustration made by the author Alexander Jamieson in 1822 (see Figure 11.4).

That's three projects in a row with very little sketching—definitely unusual for me!

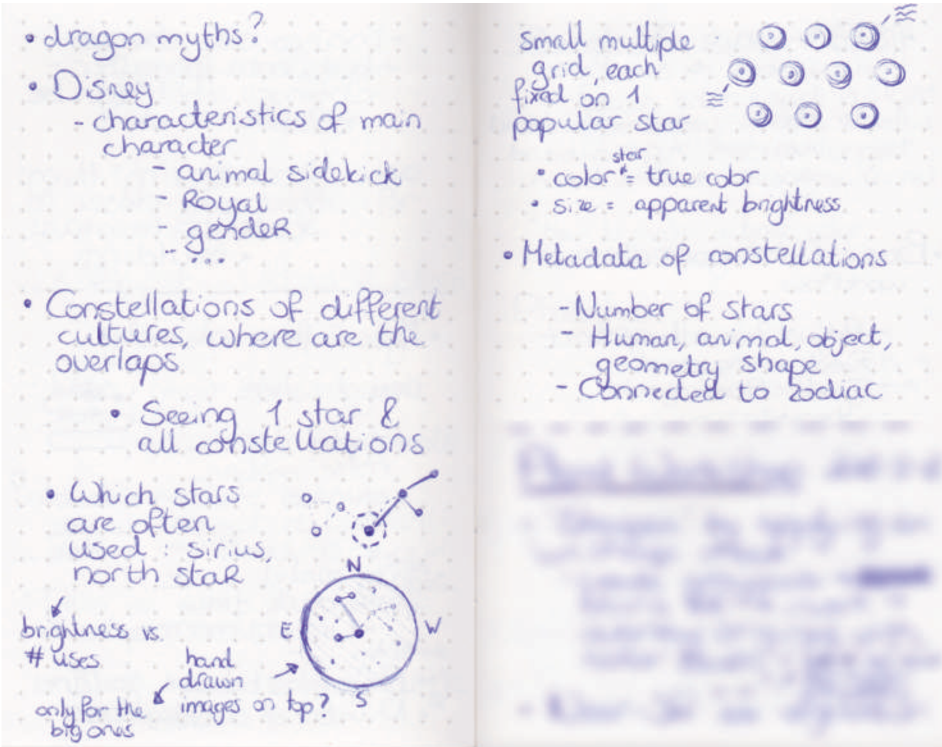


Fig.11.3

Brainstorming concept ideas and data to use for this project.

Another part of the project that took up more pages in my little notebook was math (as usual). If there were multiple constellations with a line between the same two stars, I wanted to draw those lines *side by side*. I started out thinking it would require some trigonometry, with four different cases. In Figure 11.5 you can see how I tried to conclude if there were different solutions for the four orientations. However, it eventually all came down to a little vector math; I had to find the *normal vector*, which is simple to calculate, and I needed each new line from a constellation to move up a little farther along the normal vector. The red sort-of-circle in the middle-left of Figure 11.5 b shows when I finally realized the normal vector was all I needed.

The four were cases being defined by how the target star was positioned relative to the source star, in the top-right quadrant, or any of the other three quadrants.

The normal vector lies perpendicular to the line you're focusing on, making a straight angle.

↳ Learn to Love Math

Math is truly your best friend in creating more unique data visuals. And although it's usually trigonometry, this time knowing just a little bit of linear algebra/vector math helped turn a rather difficult problem into a straightforward approach.

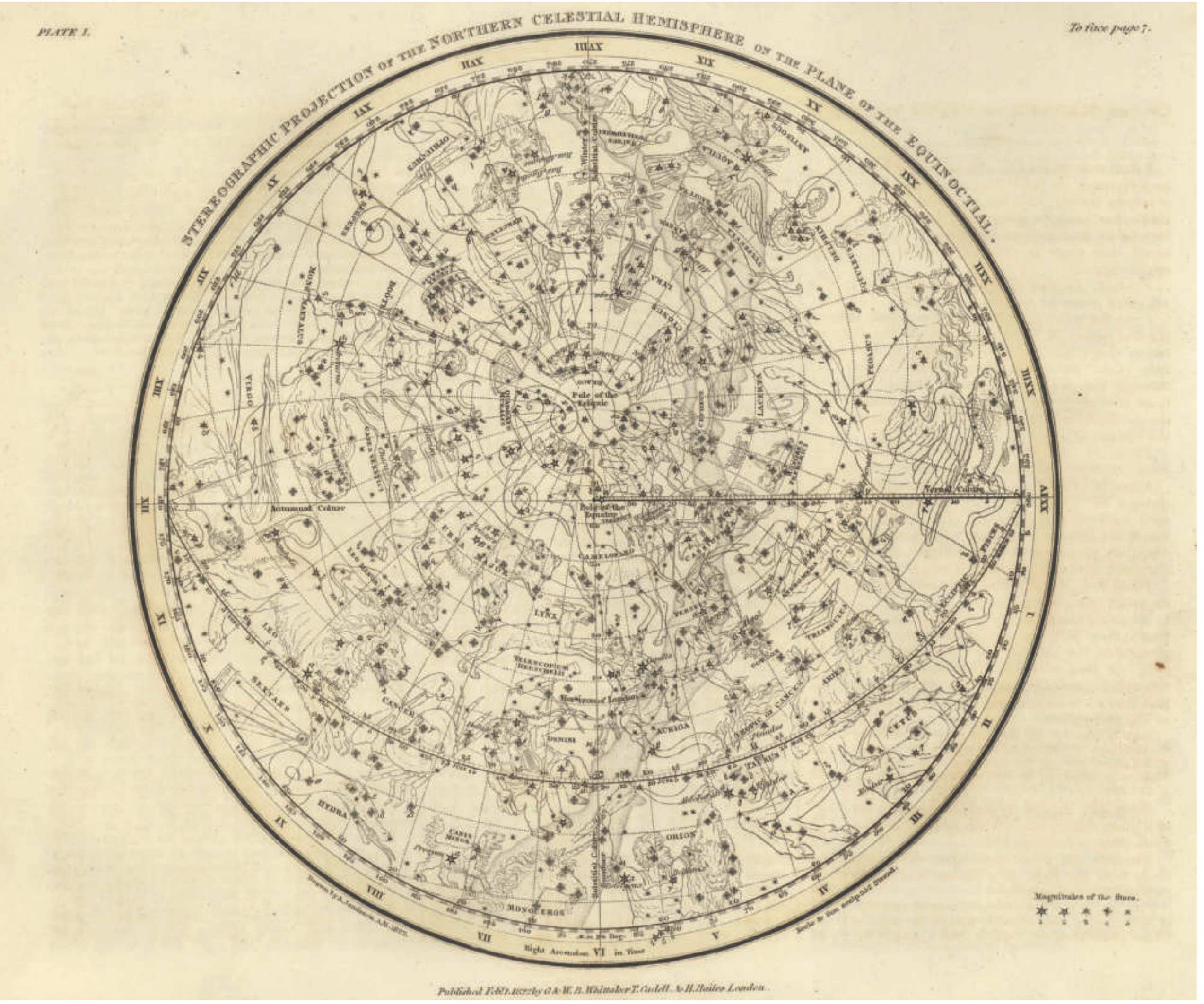


Fig.11.4

Plate 1 from A Celestial Atlas by Alexander Jamieson from 1822.

a

b

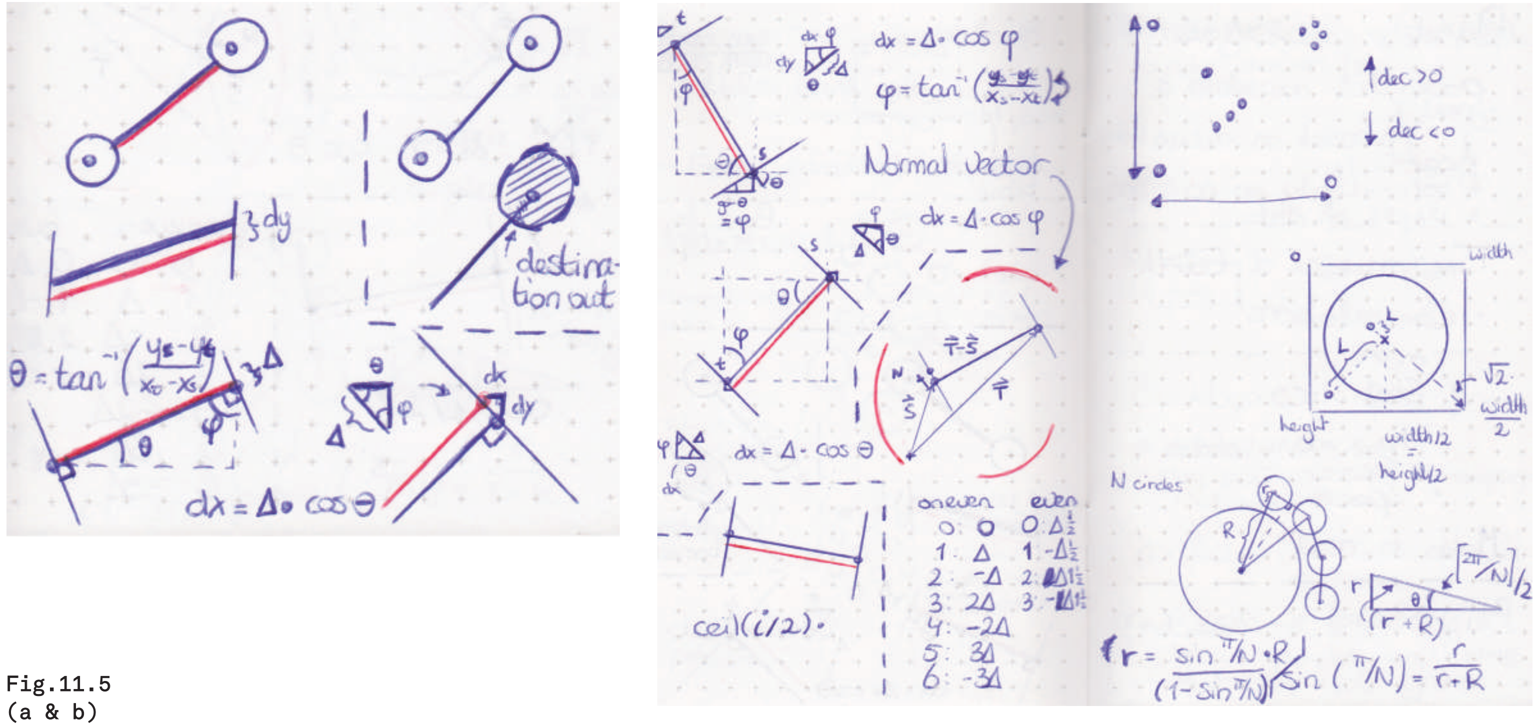


Fig.11.5 (a & b)

Sketches about figuring out how to get lines next to each other.

Finally, I made some sketches of the general page layout. The title would stand out nicely against a background of the night sky. With the sky maps themselves already providing more than enough aesthetically, I wanted to simplify the other elements of the text and layout.

a b c

Fig.11.6 (a,b,c)

Sketching out the layout of the page in later stages of the project.

Code

My first goal, before focusing on the *actual data* visualization side of things, was to create a "base map" of the sky. I've never created one, so I did a little research on what kind of map projection is typically used for sky maps. (I decided to go with a *stereographic* one.)

Given this map would include 9,000 stars for the full sky, mini donut charts, and many constellation lines, I set out to create this project with canvas due to its better performance. I loaded my star data and set up my code following several `D3.js` based examples of sky maps.[6] But all my code produced was a thin stripe of stars. ಥ_ಥ (see Figure 11.7 a).

I never truly worked with projections before outside of the default Mercator projection.

a b

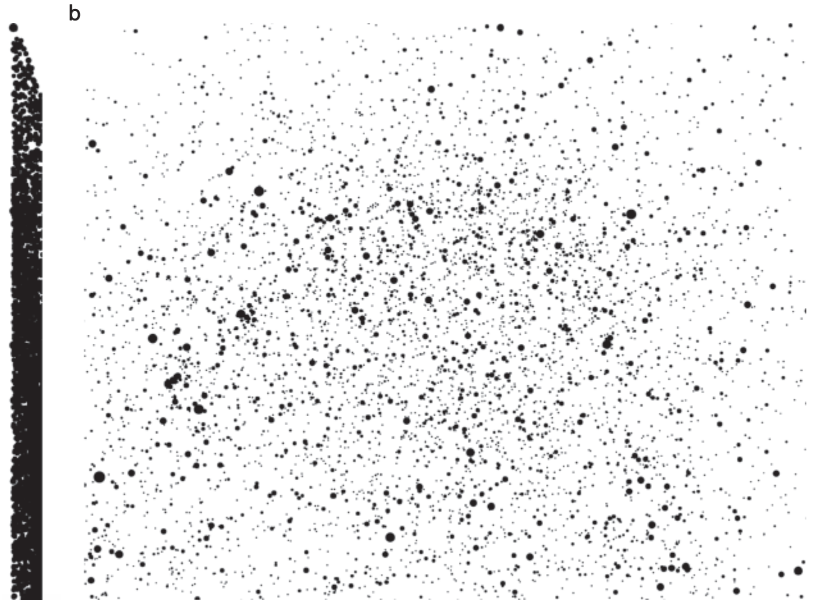


Fig.11.7 (a & b)

The very first results on screen, failing first, but getting it right the second time.

After a more careful comparison of the other sky map examples and mine, I realized I forgot an easy-to-miss transformation calculation of the RA (right ascension) and declination. A few code adjustments later and I had the map from Figure 11.7 b. However, it was still too abstract for me to see if it was correct or just a random collection of points. I felt that the one thing that would probably help me realize if the stars were correctly plotted was to add the *graticule lines*, the background grid. Thankfully the `D3.js` based sky map examples helped out again. When I rotated the projection to face North, added the stick figure lines for the modern constellations, and even recognized a few, I *finally* knew for sure that the stars were in the correct location (Figure 11.8).

For the rest of the sky map, I focused mostly on the central star of Orion, my favorite constellation for a variety of reasons: it contains many bright stars, is easy to pick out, and I could look at it from the living room window of my childhood home for many winters.

a b c

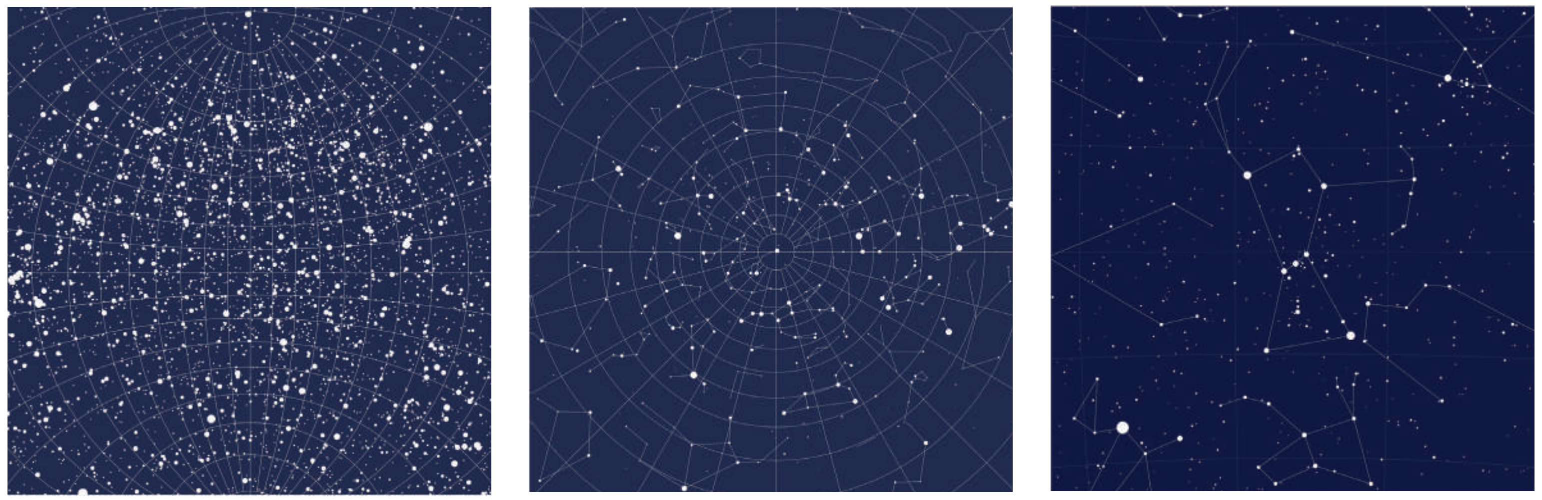


Fig.11.8 (a,b,c)

Steps in setting up the sky map, such as adding a background graticule, rotating it to face North, and zooming in on Orion.

↳ Remix What's Out There

Mentally, it's sometimes easier to get started on a project when you don't have to do everything from scratch. Even though the code for the `D3.js` based sky map examples wasn't very long or intricate, I felt it was just the right thing for me to use as a base to start working from. Beginning from some basic code examples can help unblock you and allow you to build on something. And I don't mean just plug your own data into the base code and calling it finished! I mean to truly remix it and turn it into a new visual that takes the quirks of your data into account and has its own style. (Think "inspired by" instead of "cheap knock-off".)

[6] Star Map by Mike Bostock: https://observablehq.com/@mbostock/star-map and Sky by Matteo Abrate: http://bl.ocks.org/nitaku/9607405

As you can see in the images of Figure 11.7 and 11.8, I was already using the magnitude to scale the radius of each star; the brighter they appear to us, the bigger the circle. Now it was time to look at the *colors* of these circles. I started with a temperature-to-color-scale I'd already investigated and developed to be very similar to the star's actual perceivable colors for a previous astronomy-related project. For a while I played around with making the colors more vibrant, but that made the sky *way* too colorful.

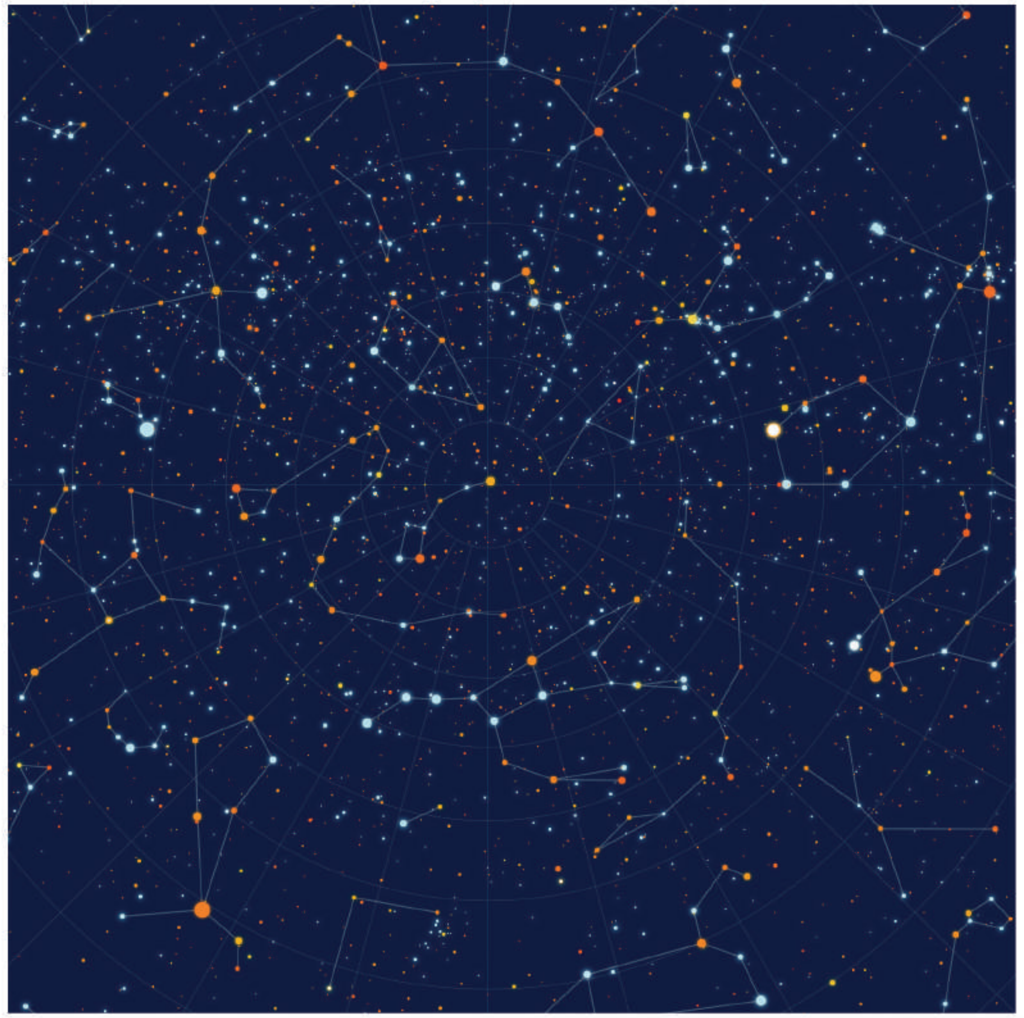


Fig.11.9

A much too saturated colorful sky.

But even with the more real and nuanced colors and adding a glow to each star (with canvas' useful `shadowBlur` property), the bigger stars looked a bit flat. In reality, our Sun looks a bit brighter in the center and dimmer around the edge. Luckily, canvas has the option to create radial gradients, which I could use to set up a unique gradient for each star: I'd have a base color for each star using a slightly lighter color in the center and a darker color at the edge (I used `chroma.js` to create the lighter and darker color from the base color). It took some experimentation to figure out the best settings. Figure 11.10 a, where I made the stars bigger to better assess the gradients, was *definitely* not correct.

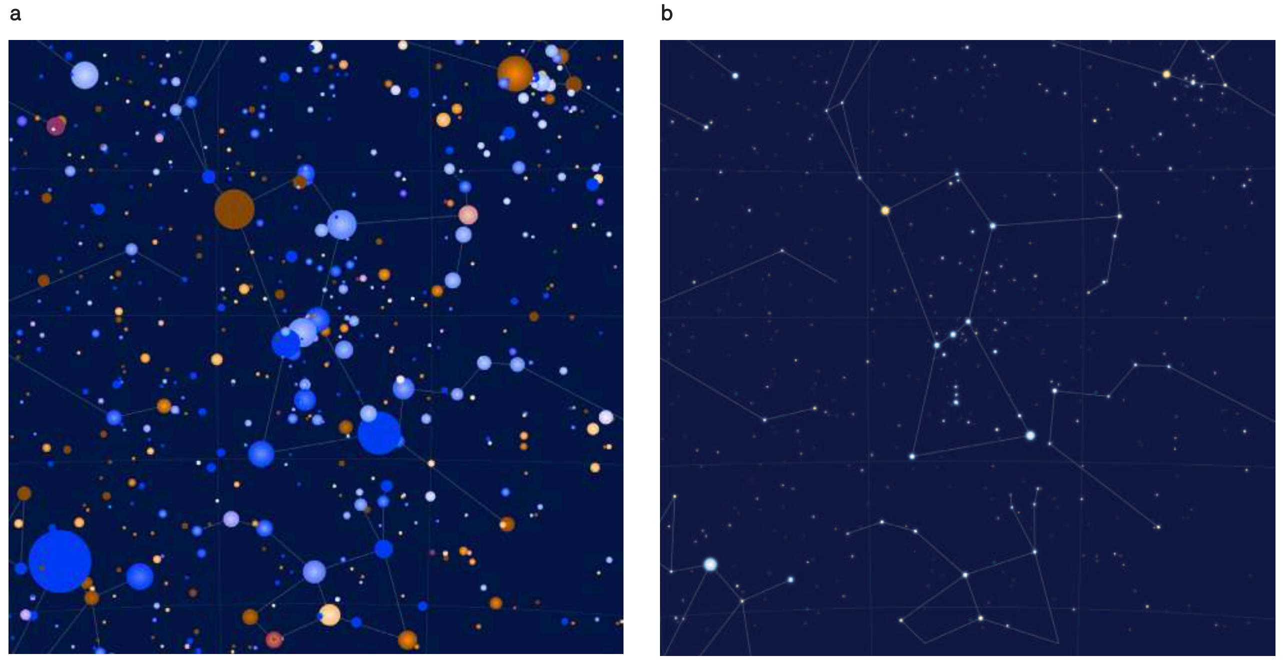


Fig.11.10 (a & b)

Trying to create a unique (sometimes tiny) gradient for each star

I clipped the sky map to a circle with a dashed line around it and added a separate dashed line inside the map to show the so-called *ecliptic*, for embellishment. This is the path that the Sun makes across the sky relative to us. Because our Earth's rotational axis is tilted by 23.44° with respect to the plane in which the planets go around the Sun, the ecliptic is not a straight line across the image (following the 0° declination line).

For the points where the background lines touch the sky map's edge I added the RA and declination degrees. I really liked the use of the zodiac signs in Alexander Jamieson's map from Figure 11.4 and replaced the degree number by the corresponding zodiac signs for 12 "major-RA" lines.

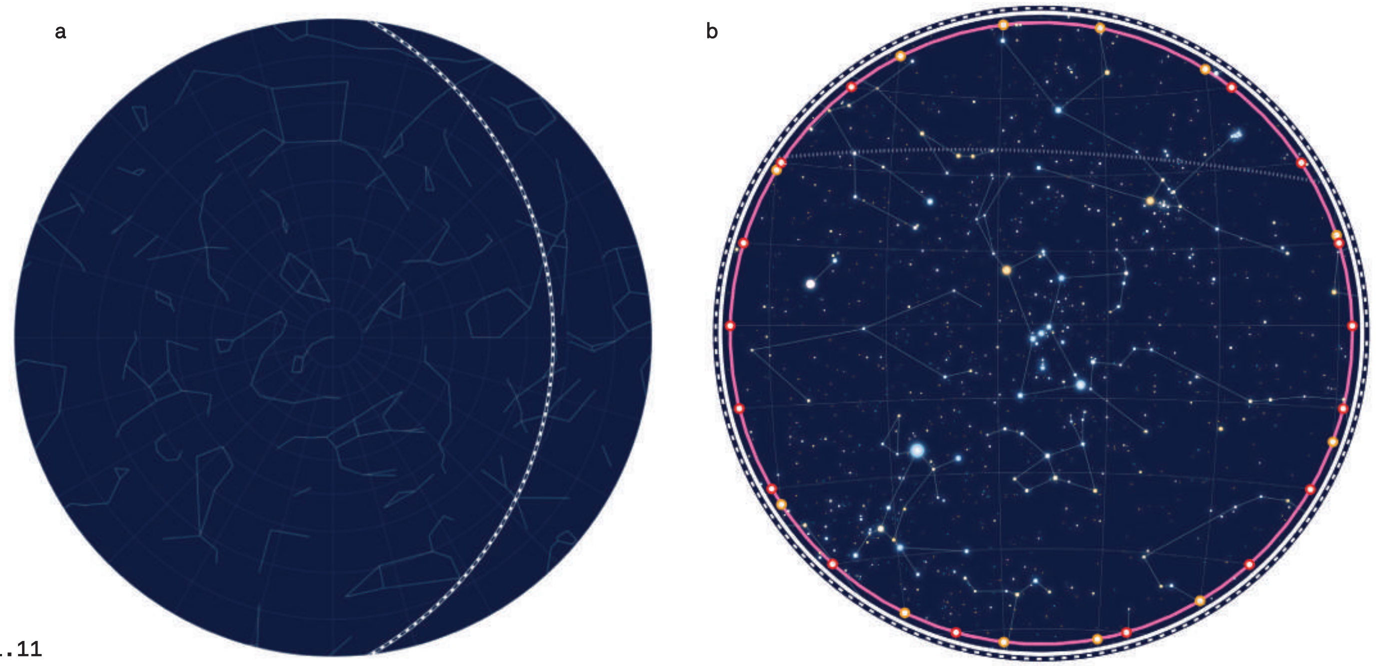


Fig.11.11 (a & b)

Clipping to a circle and adding the ecliptic path on the left, while calculating where the background grid touches the circle boundary for degree notations and zodiac signs on the right.

Having finished with the lines, it was time for the final big aspect that I had in mind for the sky map base: space itself. The background color ("space") up until this point was a dark blue. I really wanted to add some depth to each image by mimicking lighter colored streaks across the background, inspired by the gorgeous streaks of the Milky Way. Not long before, I'd seen some great experiments and bloopers from a friend that was using the contouring options of `D3.js` and remembered that it gave *exactly* the kind of feeling I was looking for with my "outer space." I started to experiment with the contour functionality and, after several iterations, built something I was happy with (see Figure 11.12)

A North and South pointer as the final ornamental element and my night sky base map was done! (See Figure 11.13) I was finally ready for the *actual* data visualization part of this project. (๑•̀ㅂ•́)و✧

Fig.11.12 (a,b,c,d)

Slowly building up a "swirly background" using the contour function of `D3.js`.

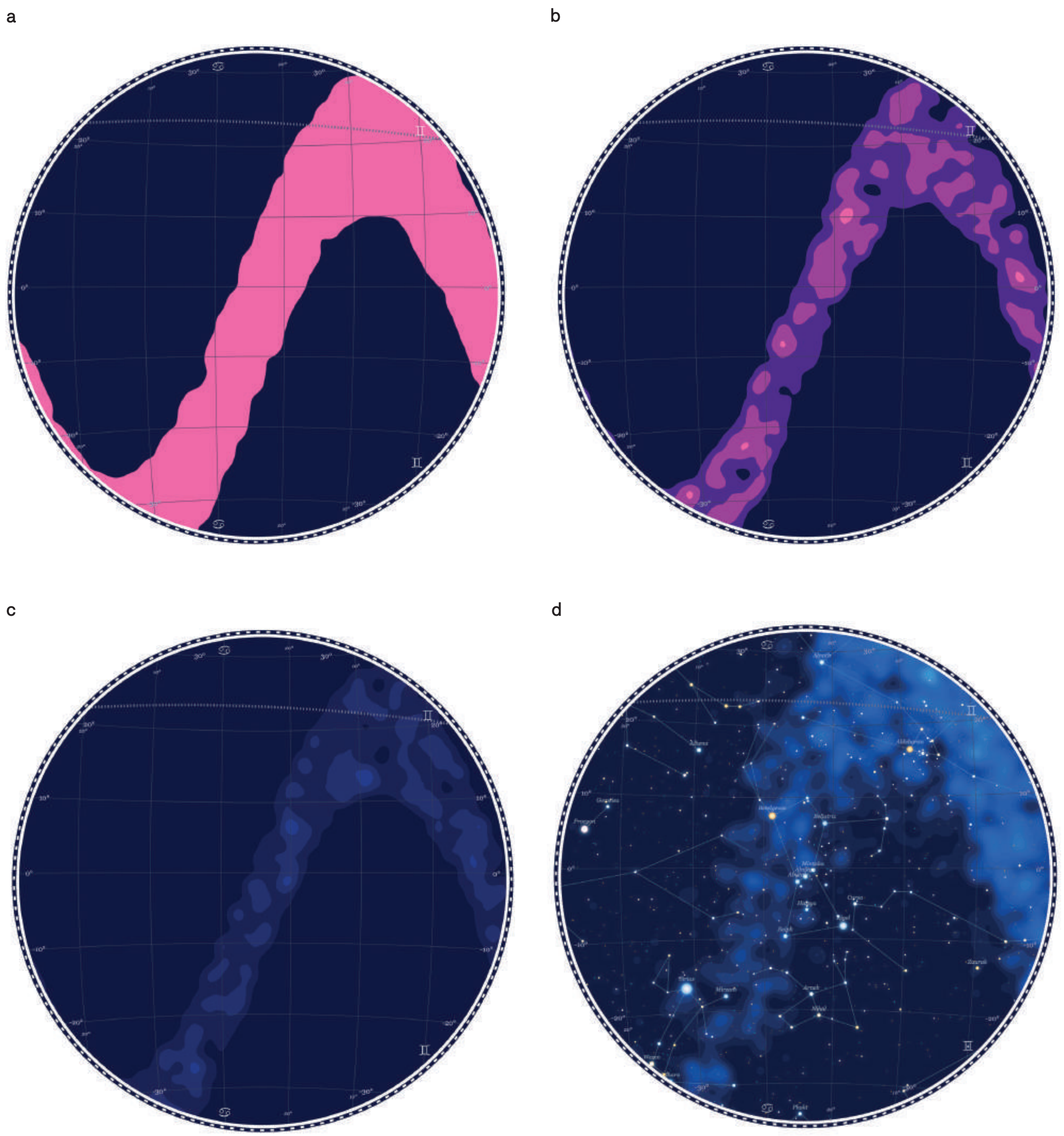


Fig.11.13

The final result of the base sky map. Now the data visualization would still need to be overlayed.

The end goal was to visualize how many constellations used each star. I started with creating a small donut chart around each star that was part of a constellation. Even though one star would be chosen to focus the visualization on, many neighboring stars would also be included in the different constellations. I first created simple donut charts featuring only white slices. When that was working and looked good I turned it into a colored version with rounded edges and a bit of padding.

I chose to switch and focus on the brightest star in the night sky, Sirius.

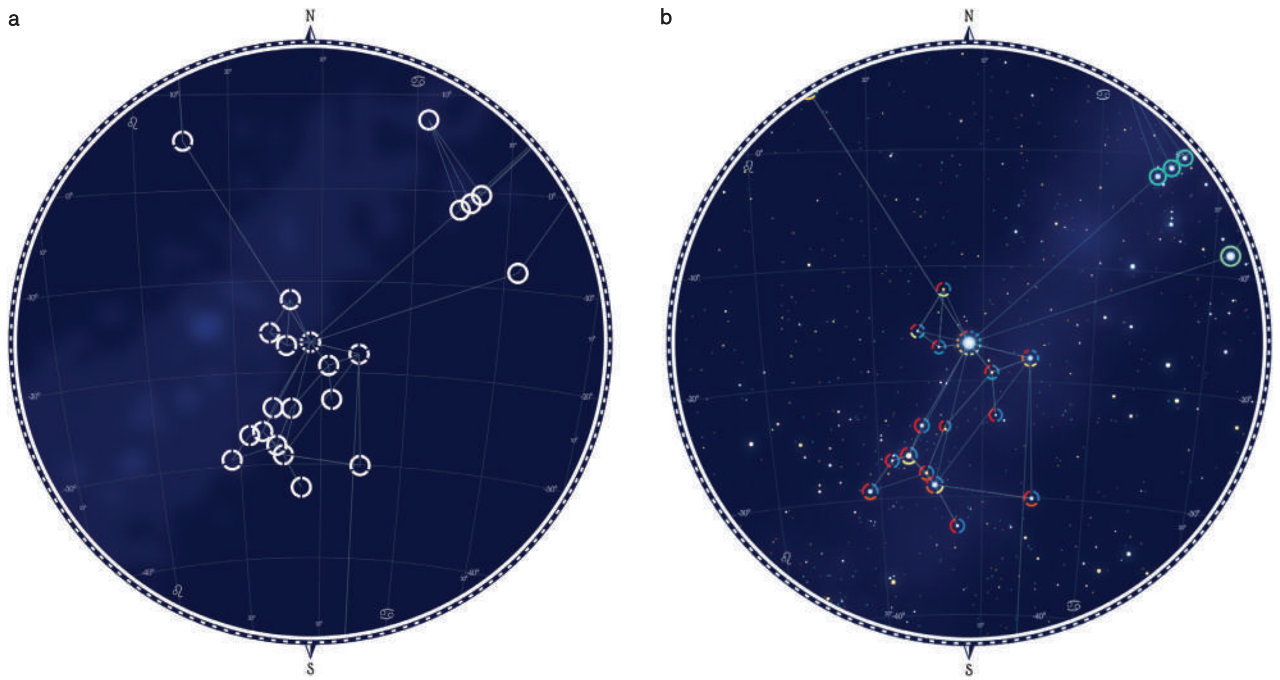


Fig.11.14 (a & b)

Creating a mini donut chart around each constellation star.

The lines in between the stars were a bit of a different story. I wanted these lines to be placed alongside each other, but I only had the exact center location of each star, so calculating the offset in the x- and y-direction that each extra line would need wasn't trivial, I thought. Until I finally remembered not to think in trigonometry, which created the wrong image of Figure 11.15 a, but in vectors and the *normal* vector (resulting in Figure 11.15 b).

You could already see my (mostly useless) math in the "Sketch" section earlier.

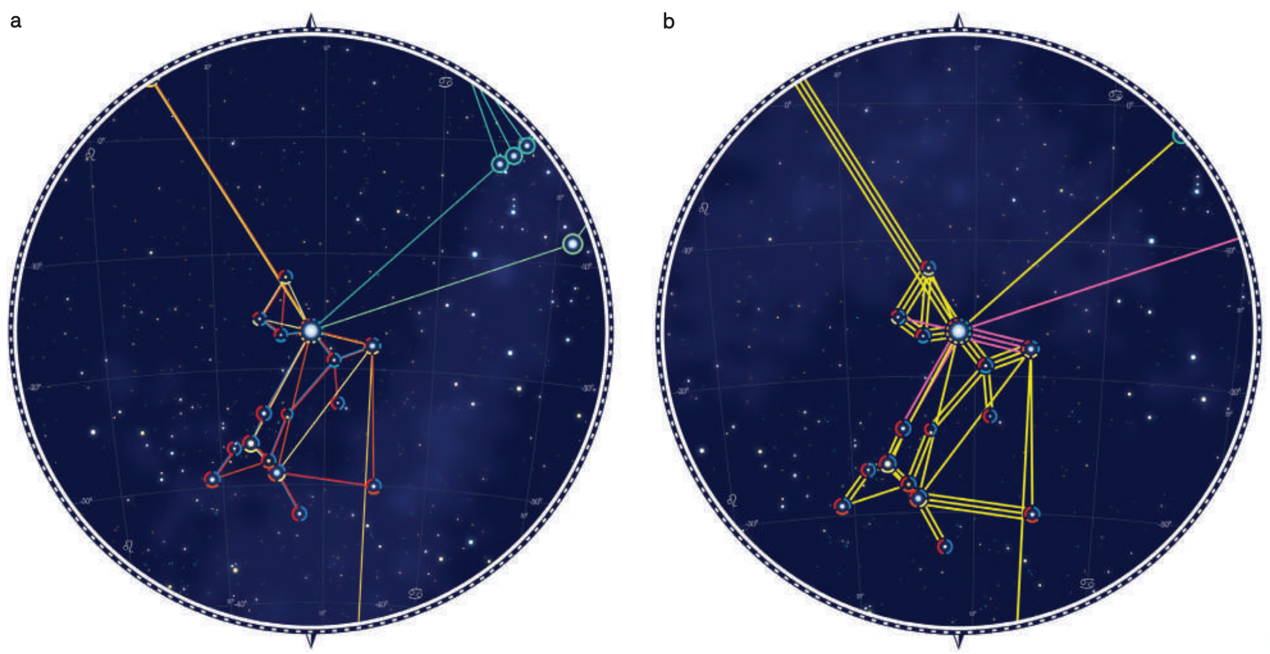


Fig.11.15 (a & b)

Positioning the lines to run parallel between the stars—doing it wrong at first and corrected eventually.

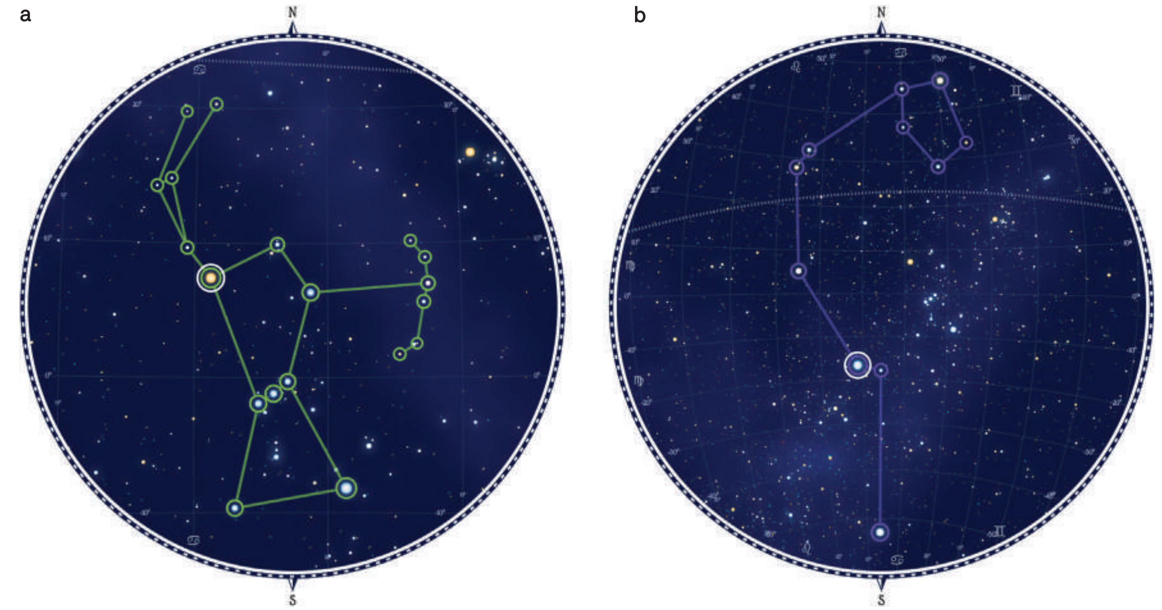


Fig.11.16 (a & b)

Focusing on (the white circle) Betelgeuse (in the modern constellation of Orion) and on Sirius in a star-line from Hawaiian culture.

Working with the different constellations showed me that using one particular zoom level and center would definitely not work to properly reveal each of the separate constellations. It took me a while to finally figure out the logic and to automatically calculate the optimum zoom level, rotation, and center that would nicely fit any constellation that I would give the program (see Figure 11.16).

To make it easier to select and see the full shape of each separate constellation (using the same star), I added all of the separate constellations in a ring around the version that showed all of the constellations at once.

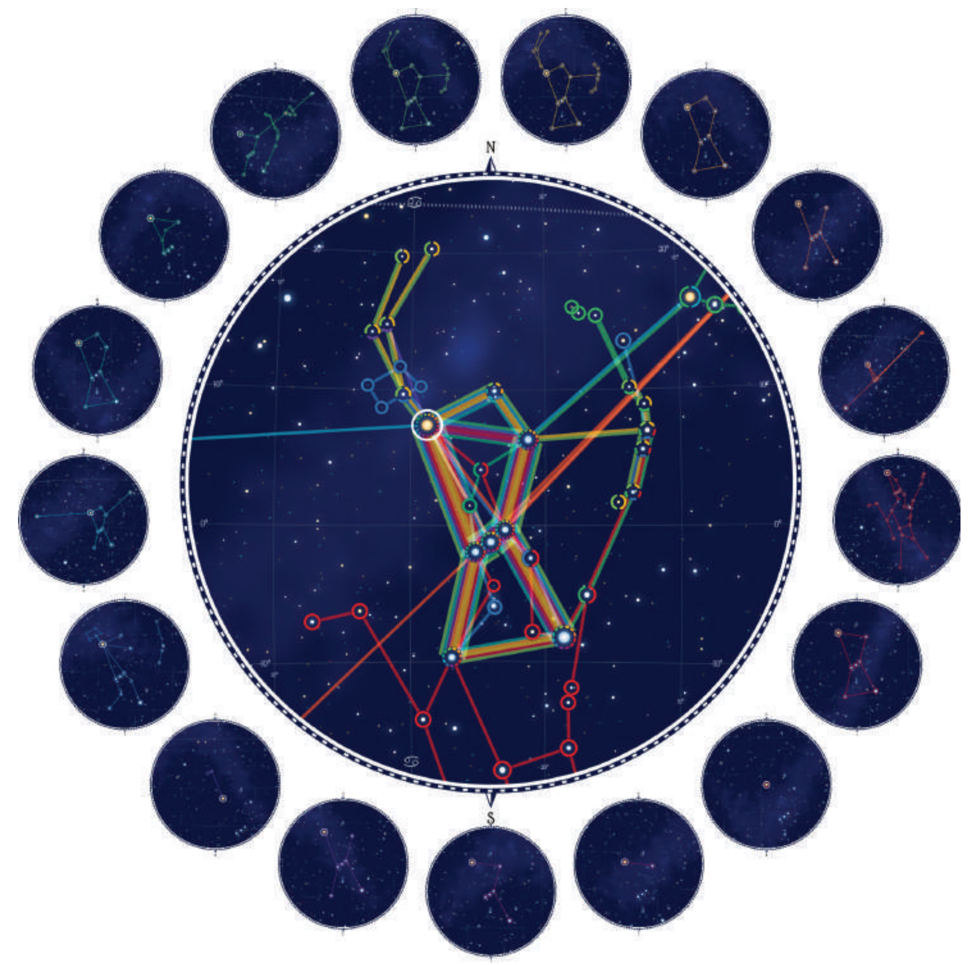


Fig.11.17

The separate constellations in a ring around the main sky map.

That immediately showed me two things. For one, this was *excruciatingly* slow! But also, the complete sky map on the mini circles wasn't needed at all. They were too small to really have any visual effect, and they were too distracting from the central map. Luckily, removing elements from the mini maps would also make them faster to load. The main thing to see were the constellation shapes anyway, which was the most performant part of the sky map's three layers: the glowing stars, the constellation lines and donut charts, and the entire background. After some fiddling around, I got it all working and ended up with the version from Figure 11.18. Finally, I added an interaction that allowed visitors to click any of the outside mini circles to see it drawn properly, and bigger, in the center (see Figure 11.30 for an example of selecting an outside mini circle).

I could've stopped there. The sky map was a complete visualization in its own right. But just showing the star Betelgeuse felt so incomplete! I had so much more data that I could use to tell a fuller and more interesting story. So even though I had already racked up way too many hours to get to this point, I decided that this project would become a complete article; with beginning, middle, and end. (In other words, *even more visuals*. (◉•‿•))

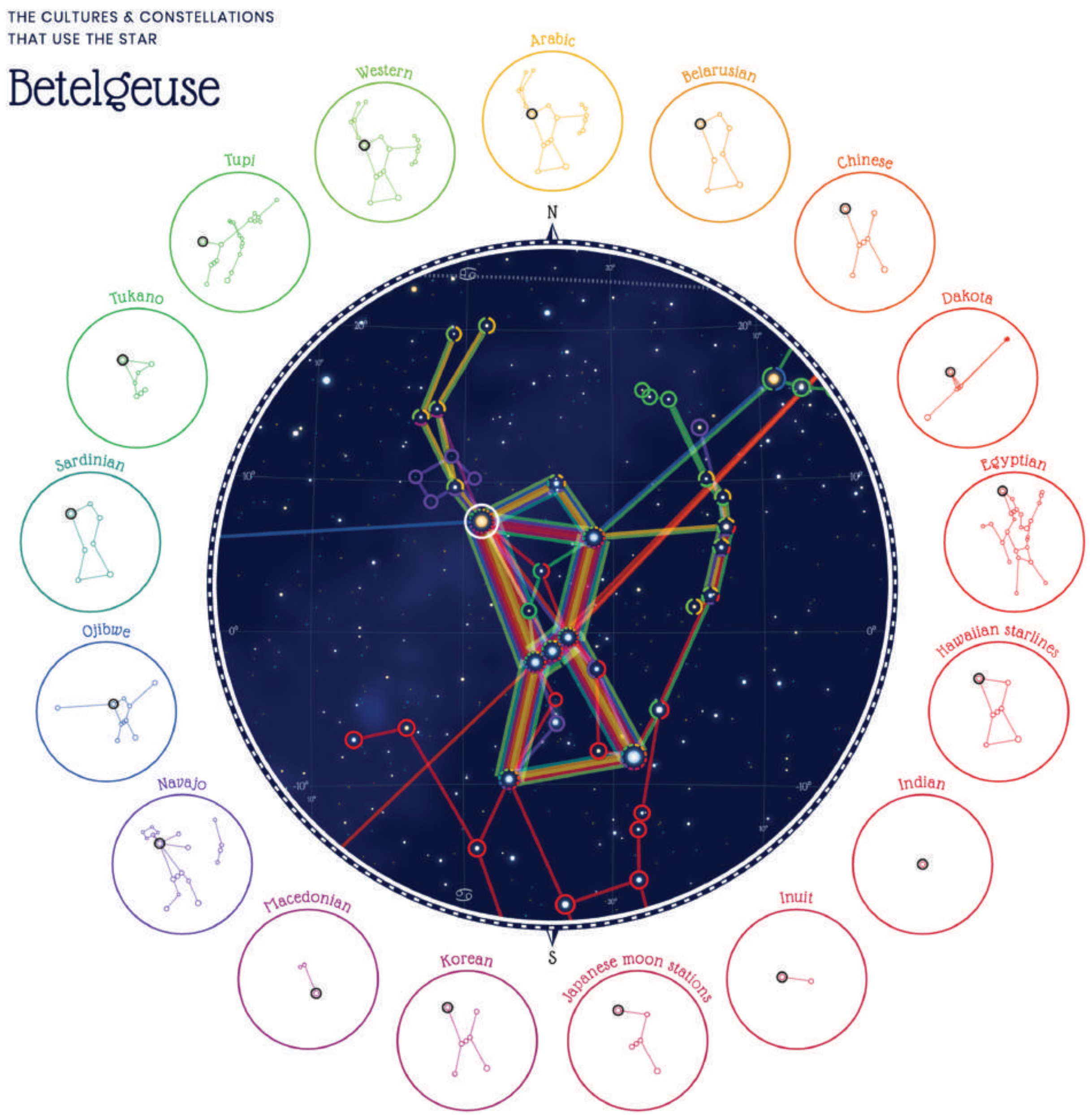


Fig.11.18

The final look of the circular sky map with the mini circles around it.

An (almost) full sky map that would show *all* of the constellations of one chosen culture was the first on my list. It would allow users to select a culture and view its constellations separately. Given how complete my circular sky map function was, setting up the base for this was quite easy. In essence, the only change I had to make was to adjust the projection from *stereographic* to an *equirectangular* one (while also using a different width/height and not clipping the visual to a circle). For this full sky map I made sure to have the background fuzzy patches follow the *actual* rough location of the Milky Way.

I was quite happy with the result, so I decided to use it for the header of the full article as well.

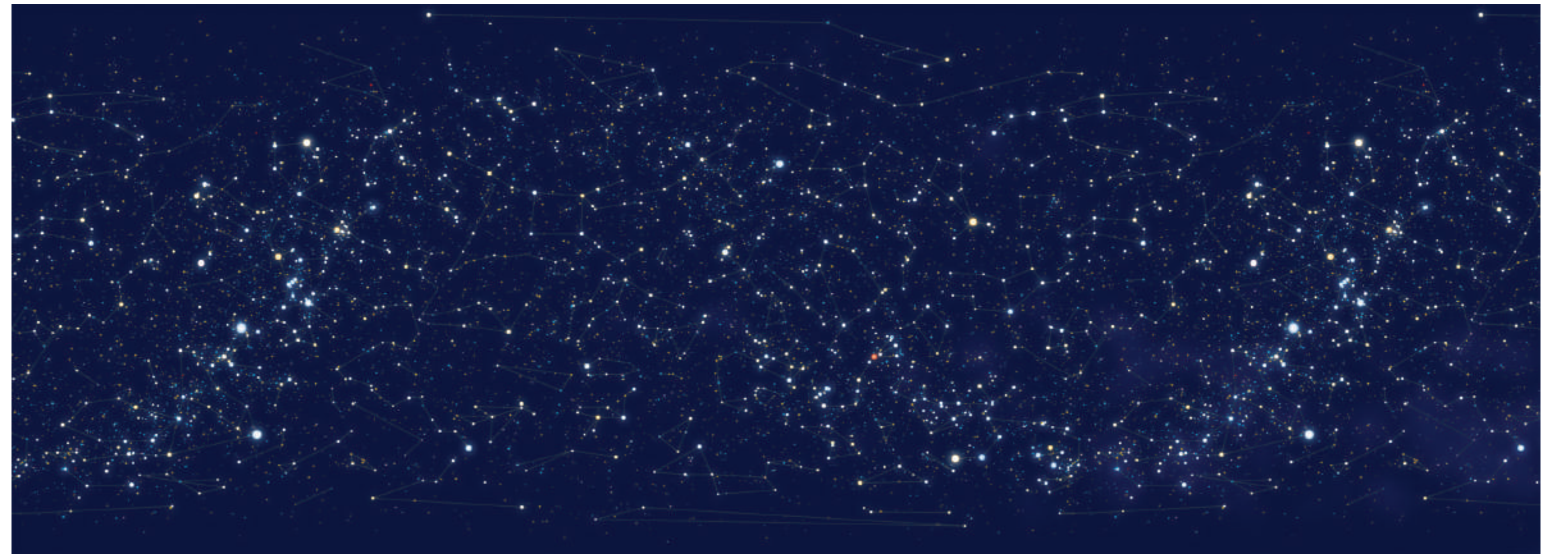


Fig.11.19

The full sky, now with the background fuzzy patches sort-of following the shape of the Milky Way.

Betelgeuse might be a fascinating star, but I wanted to reveal many more interesting stars and constellations! The function to create the full sky map with all of its constellations could be used for any star, thankfully. What *did* end up taking several hours was the exact design of these extra sky maps on the page and manually going through about a hundred stars and selecting the ±15 I thought looked the most interesting and diverse.

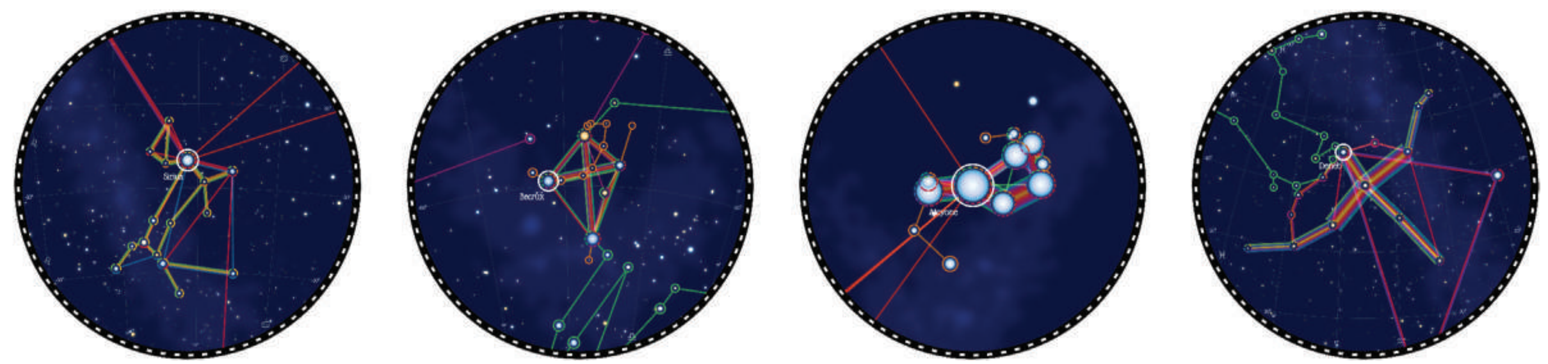


Fig.11.20

An early look of several smaller sky maps, each focusing on a different star.

The final visual pieces to add to the page were the statistical charts, starting with the scatter plot that displayed a star's brightness versus the number of constellations it was a part of. As ±2,200 stars were included in at least *one* constellation, I went for canvas as the base. However, I used a separate SVG on top for all the axes, text, interactivity, and annotations. Using canvas made it easy to reuse the same coloring of the stars as I had in the sky maps. However, with the white background those colors looked *much* too soft, and the gradient effect was too distracting (Figure 11.21 a). Removing the gradient and adding a multiply effect to darken any overlapping stars helped to make it visually more appealing (see Figure 11.21 b). However, I felt that the colors were still too soft. So I made them more vibrant, added a bit of "glow" around the edges, cleaned up the axes a bit, and the visual style of the scatterplot was done (see Figure 11.21 c). Eventually, I also added a mouse hover and textual annotations.

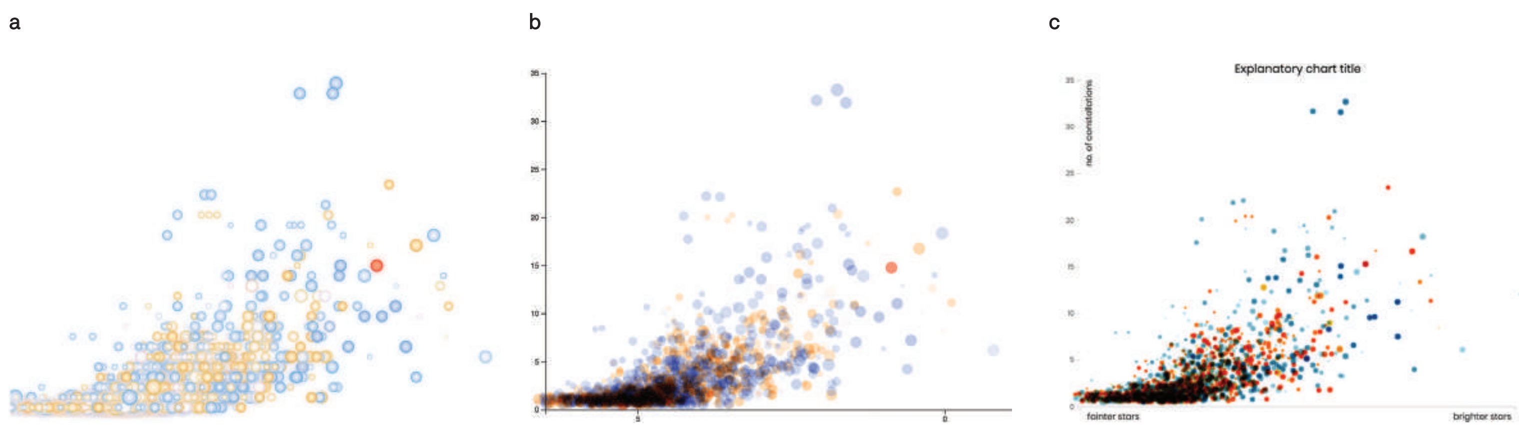


Fig.11.21 (a,b,c)

First using the same star colors and gradients as I did for the sky maps, but later going for darker colors and no gradient, and finally even more vibrant and glowing stars.

↳ Annotations Are of Vital Importance

Often overlooked, annotations are one of *the* best ways to make a chart understandable to an audience. Underutilized in many data visualizations, annotations are the ideal way to highlight *exactly* those things that you, as the creator, want the audience to pay attention to. My current go-to is the wonderful `d3-annotations` library. And I think the scatter plot from this project has the most intricate placement of annotations that I've ever applied (see Figure 11.26 later in this chapter for the fully annotated version).

At the bottom of the article I added a section that tells more about each culture. Selecting a culture results in the full sky map updating to show all of the constellations from that culture. And a bar chart that I had in mind with the average number of stars per culture eventually ended up as a small mini bar in each of the culture "boxes" (Figure 11.22).

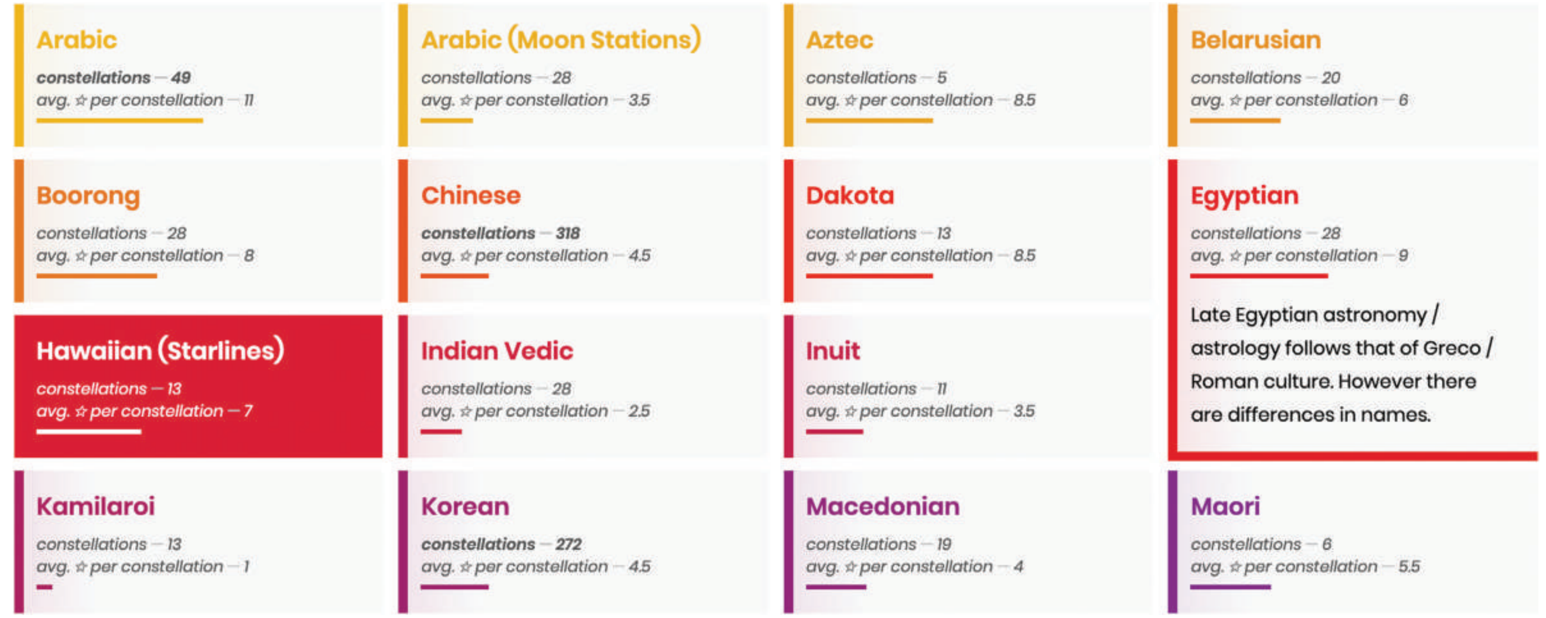


Fig.11.22

All the sky cultures in their own overview.

And then! Then I replaced as many of the visuals as I could with images. (●__●) Images are much easier to load than doing the heavy sky map calculation, and the sky maps wouldn't change anyway.

Adding in text between all the different visuals, and my second (and thankfully last) full article style data visualization was finally done!

Reflections

This was my longest project in terms of hourly investment. I clocked about 110 hours, but estimate I spent more than that, due to not always timing myself whenever I thought something would take five minutes to do, and suddenly, I was an hour in. Some parts took an unexpected amount of time to work on, such as setting up the functionality to create a base sky map that could handle any star and constellation combination. I am generally less enthusiastic about working on overall page layouts, but I spent extra time trying to perfect this layout since it was a vital part of the story.

"Beautiful in English" isn't far behind in terms of hours spent.

Even though it took so long, I'm super happy to have created a project that combines my love of astronomy with my passion for dataviz! Especially since this was, for me, my farewell to the creation of new visualizations as a part of *Data Sketches*. I'm amazed at all the things that I've learned about making data visual across the 12 topics. And it's fascinating to look back at my skills for the very first project and comparing that to the full-length article that my final project became. I'm exceptionally happy to have been a part of *Data Sketches*. It has opened doors to opportunities that I didn't even know I was looking for!

I ain't ever doing it again though. Damn, such work!! [¬°-°]¬

Figures in the Sky

↳ FiguresInTheSky.VisualCinnamon.com


Figures in the Sky

How cultures across the World have seen their myths and legends in the stars

No matter where you are on Earth, we all look up to the same sky during the dark nights. You might see a different section of it depending on your exact location, time & season, nevertheless the stars have fascinated humans across time and continents.

Our own creativity combined with stories about local legends and myths have created a diverse set of different constellations. And even though the stars don't change, people have found many different shapes in the same sky. From humans, to animals, to objects, and even abstract concepts.

The same sky, different figures

Let's compare 28 different "sky cultures" to see differences and similarities in the shapes they've seen in the night sky. Ranging from the so-called "Modern" or Western constellations, to Chinese, Maori and even a few shapes from historical cultures such as the Aztecs.

Take the star Betelgeuse. This red supergiant is one of the brightest stars in the night sky. In proper darkness, you can even *see* that it shines in a distinctly red color. It's part of one of the easiest to distinguish modern constellations known as Orion, named after a gigantic, supernaturally strong hunter from Greek mythology.

Fig.11.23

The start of the "Figures in the Sky" article.

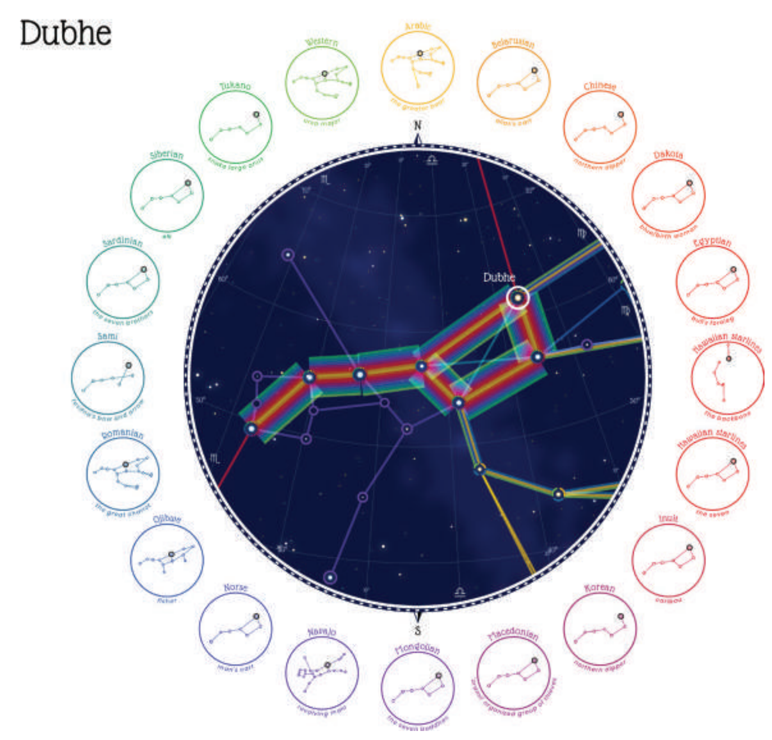


Fig.11.24

The many constellations that use the star Dubhe, part of the well-known Big Dipper.

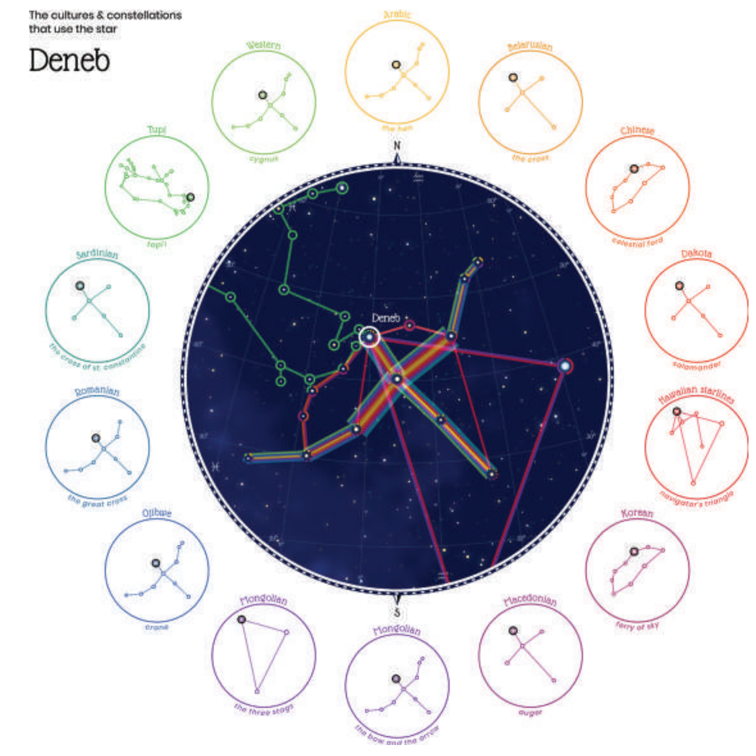


Fig.11.25

All constellations that are connected to the star Deneb. In Western cultures its constellation is known as Cygnus (the Swan), but it is also part of the "Summer Triangle," a very easy-to-make-out group of three stars in the high of a Summer night on the Northern hemisphere.

Fig.11.26

The full sky map showing all 88 "modern" (Western) constellations.

Fig.11.27

This beauty of a constellation comes from several tribes in South America and is called Veado (which Google tells me is similar to "deer"). I would say that it seems a bit too specific for a constellation that can "easily" be found in the sky, but that's perhaps my own bias of having lived in very light-polluted areas all my life.

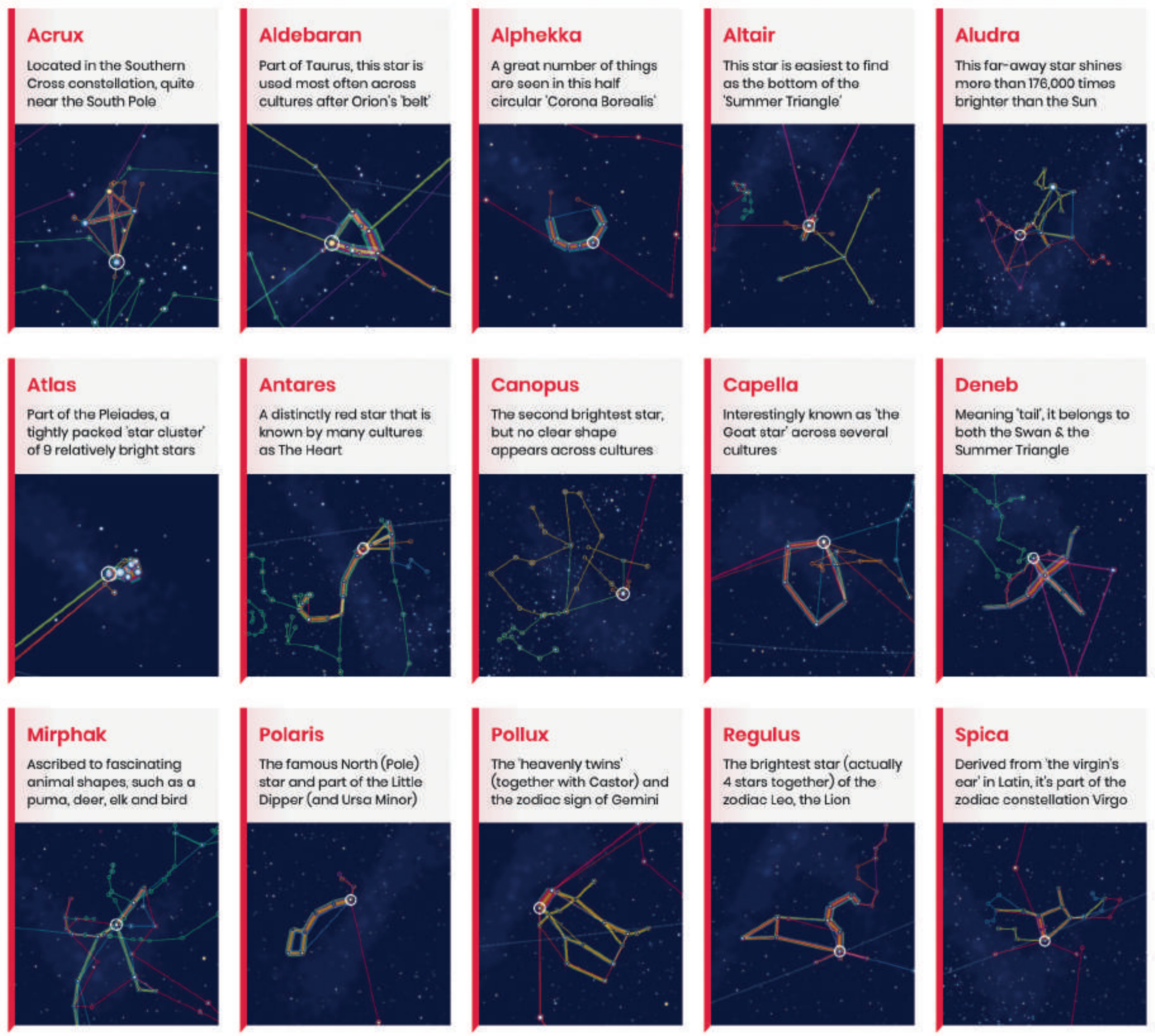


Fig.11.28

After highlighting the constellations of Betelgeuse, Sirius and Deneb, the article lets the viewer inspect 15 more stars by clicking on any of the mini images.

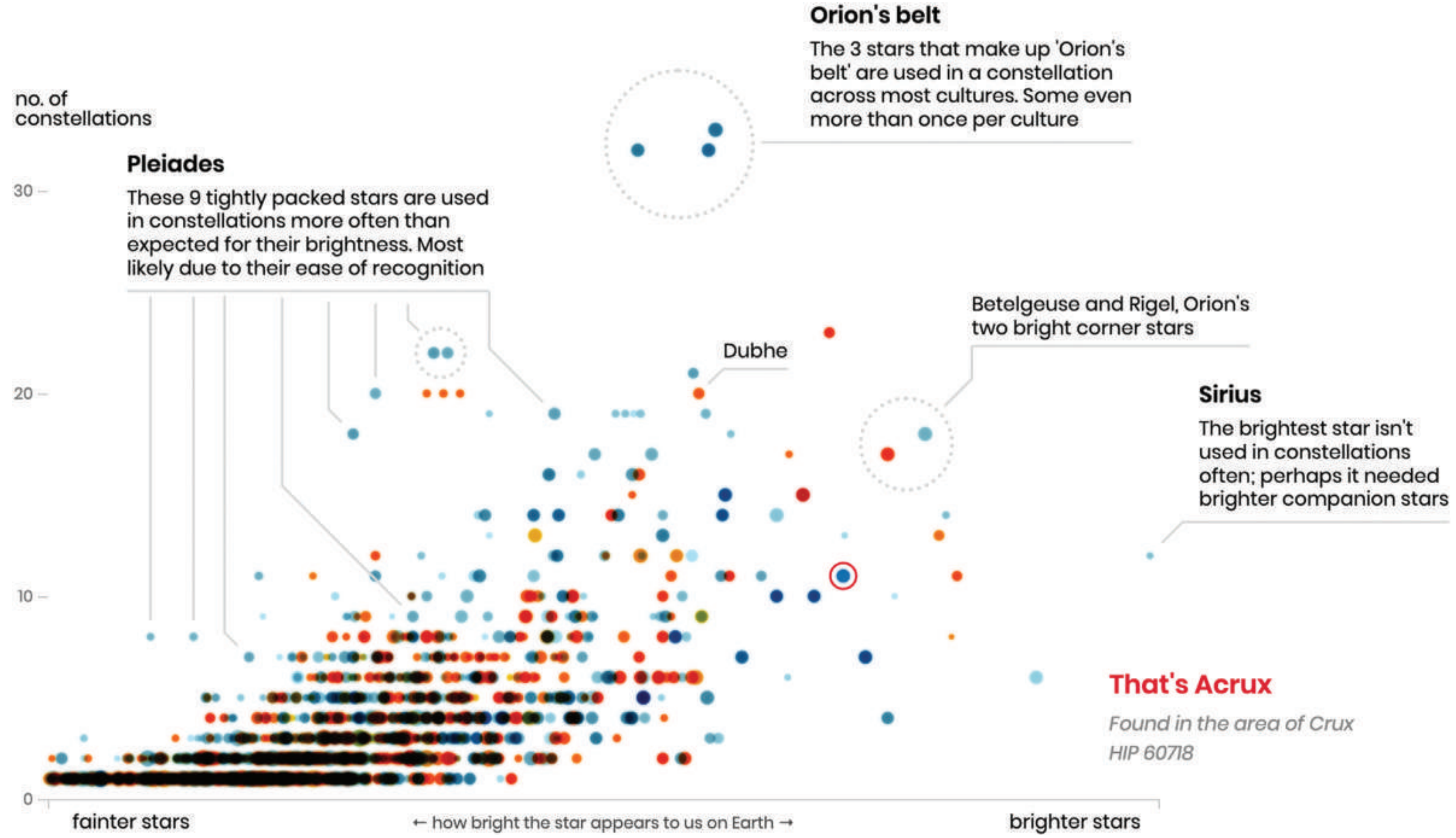


Fig.11.29

A scatter plot showing all ±2,200 stars that are included in at least one constellation.

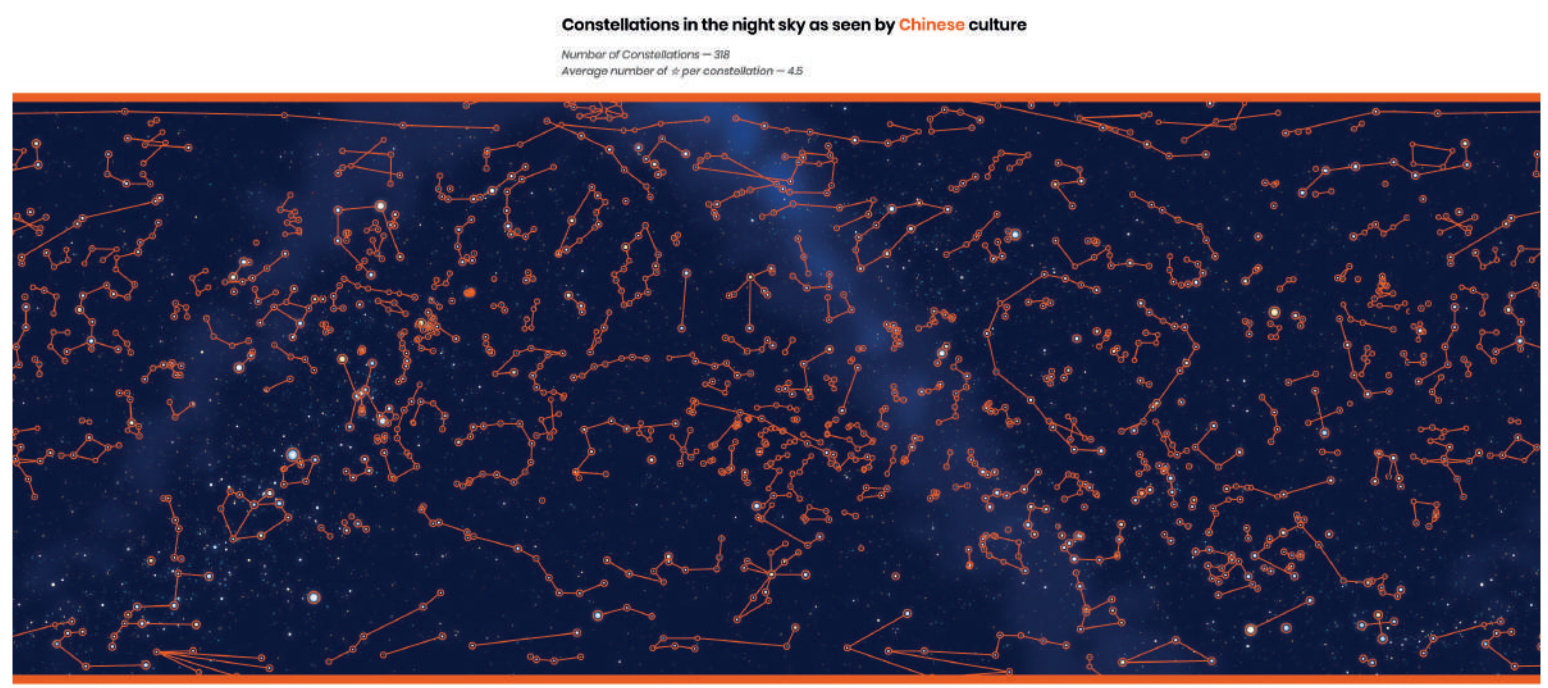


Fig.11.30

The full sky map of the 318 different Chinese constellations.

Legends

SHIRLEY

Just like Nadieh, it took me *forever* to decide on a good dataset and angle for this project. We chose “Myths & Legends” because it sounded like a great topic with a lot of potential, but the ideas I came up with either didn’t excite me much or were difficult from a data gathering perspective. I wanted to do something related to my Chinese background and bounced from Chinese and Asian mythology to classic Chinese literature to *Mythbusters* episodes.

Then, the idea came to me after watching *Crazy Rich Asians*. I loved Michelle Yeoh in the movie, but it wasn’t until I read more about her that I learned how accomplished and legendary she was. It made me wonder about all of the legendary women across history that I’ve never heard of, and the idea took shape from there.

Data

Once I had the idea about doing something with legendary women, the next step was figuring out how to get the data. I decided early on that Wikipedia, with its user-generated content, would be a great resource. I wanted to figure out a way to get the "top" women on the platform, but ran into a tricky problem: I didn't have any idea how to define "top." Would it be page views? Page length? Inbound links? And even though I had seen something similar on *The Pudding*—they were even awesome enough to write about their methodology[1]—I still wasn't sure how to go about adopting their approach for my own project.

The Pudding is the same awesome visual essay collective I published my "Hamilton" project on!

Then, it hit me: instead of trying to define "top" myself, I should just look for a definitive list. After a few Google searches, I ended up on a Wikipedia page with a list of the 51 female Nobel Laureates. They were all incredible women, yet I hadn't heard of most of them. It was the perfect dataset.

I copy-pasted the table—with information about the category (Peace, Literature, Physiology or Medicine, Chemistry, Physics, Economic Sciences) and year of their awards, as well as the achievements that led to the award—into a spreadsheet and did some light formatting and cleaning. I exported the spreadsheet as a CSV and used an online converter to get the data into JSON format.

This is one of the most important data gathering lessons I learned from Nadieh: spreadsheets are great for cleaning data. The extended lesson is to use the right tool for the job, instead of trying to use the same hammer (code) every time.

Because I also wanted to get data from the laureates' individual Wikipedia pages, I researched for ways to access the Wikipedia API. It was a little hard to navigate (I wasn't sure if I should use Wikidata or MediaWiki, which are the two options that came up when I searched "Wikipedia API"), but thankfully *The Pudding* had me covered. A quick dig through their repo led me to their code that used `wiki.js`, and I could interface with that Node.js package instead of the actual Wikipedia API.

After that, I wrote a script using `wiki.js` to programmatically get additional data from each woman's Wikipedia page, including basic biographical information, the number of links into their page ("backlinks"), and the number of sources at the bottom of their page.

Sketch

In early 2018, my friend and I pitched a design to a potential client. We didn't get the contract, but the core idea—to represent the individual data points as multifaceted crystals—really stuck with me. I pinned some gorgeous photos and paintings of crystals and mused how I could programmatically recreate them.

Fast-forward to fall 2018 and I had the idea of legendary women but wasn't sure how I wanted to visualize them. Then, one day as I was going through the "Information Is Beautiful Awards" shortlist and pinning my favorite entries, I came across the crystals again in one of my Pinterest boards. As soon as I saw artist Rebecca Chaperon's gorgeous paintings of crystals,[2] I knew I wanted to represent each of the legendary women as one of those bright, colorful crystals—because how beautiful would that be?

[1] The Pudding, "What Does the Path to Fame Look Like?": https://pudding.cool/2018/10/wiki- breakout/
[2] Rebecca Chaperon, "Crystals": https://www.thechaperon.ca/gallery/crystals

It wasn't long before I had come up with the other details. The size of the crystal would represent the number of articles linking back to a laureate's Wikipedia page (her "influence"). The number of faces on the crystal would map to the number of sources at the bottom of her page (because she's "multi-faceted"; get it? (*≧艸≦), and the colors would represent the award category.

The only thing that evaded me was how to position the crystals. For the longest time, I could only think to lay them out in a two-dimensional grid and have the reader scroll through them. But then I took Matt DesLauriers's "Creative Coding" workshop on Frontend Masters[3], where he taught (among other things) `Three.js` and WebGL. The workshop inspired me to try out a 3D layout, and I knew immediately that I would use the z-axis for the date they received their award: the closer to the foreground, the more recent her award.

For more explanation of `Three.js` and WebGL, see "Technologies & Tools" at the beginning of the book.

All of this came to me so quickly and naturally, that I didn't draw a single sketch of the idea.

Code

For years I wanted to make *something* physical, and in 2018, I made it a goal to create a physical installation. But every time I thought about it, I got stuck thinking about 2D projections (or TVs) on walls; I didn't know how to take advantage of all that *floor space*.

And then one day, it hit me: of course I didn't know how to think in physical spaces, I worked digitally in 2D all day long. So if I could teach myself to work in 3D digitally, then it should (hopefully) follow that I could think in physical spaces also. I put `Three.js` and WebGL at the top of my list of technologies to learn.

I took Matt's Frontend Masters workshop and learned the basics of `Three.js`, fragment shaders, and vertex shaders. I learned the "right-hand rule" to orient myself in WebGL's coordinate system: with my right palm facing me, use the thumb for the x-axis (increases going right), index finger for the y-axis (increases going up, which is the opposite of SVG and canvas), and the middle finger for the z-axis (increases coming out of the screen and towards us). I learned that WebGL's coordinate system doesn't operate in pixels, but rather in "units" of measurement that we can think of as feet or meters or whatever we like, as long as we're consistent.

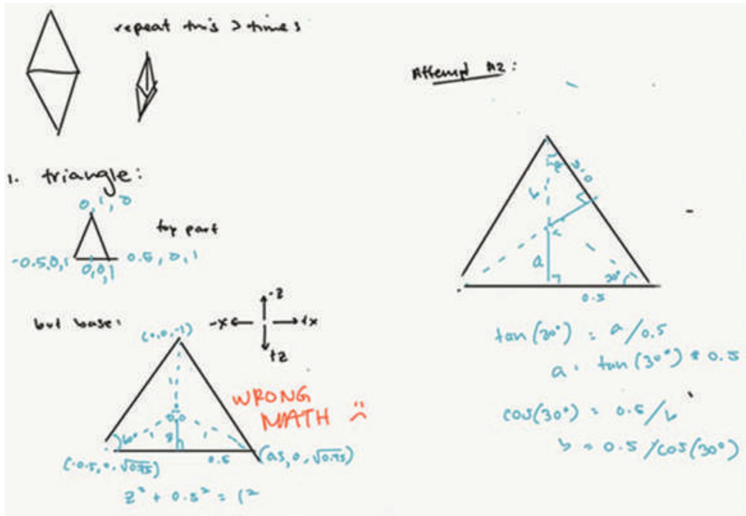
In mid-November, David Ronai asked me if I was interested in participating in *Christmas Experiments*, an annual WebGL advent calendar. I was hesitant to accept, since I had never worked with WebGL before, but David encouraged me to give it a try and promised to put me later in the month to give me more time. I agreed, knowing that the deadline would give me the motivation I needed to complete the project. I made it a goal to do a little bit each weekday starting December 1st, until I could get to something presentable on the 23rd—the slotted date for my Christmas Experiment.

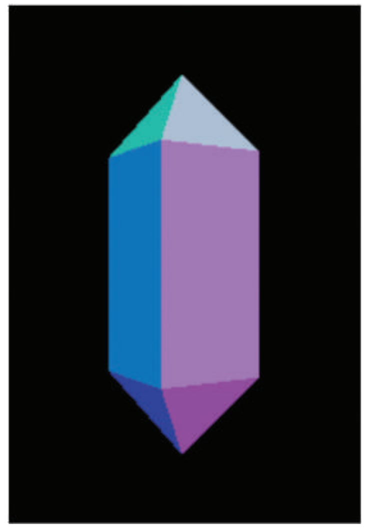
I started by reading the first two chapters of WebGL Programming Guide,[4] which taught me how WebGL was set up. I then rewatched the Three.js section of Matt's workshop so that I could see what heavy lifting it was doing for me. After the workshop, I created an Observable notebook to figure out the minimum amount of setup required to draw with Three.js.

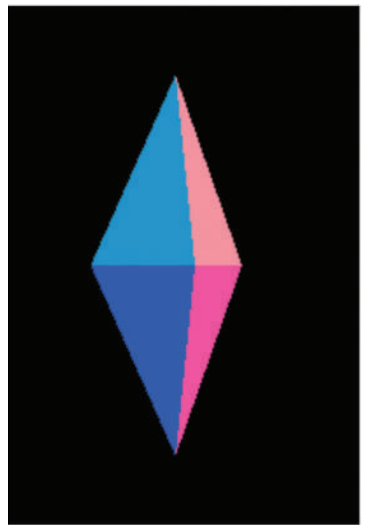
I always like understanding the most fundamental building blocks required to make something work—the core foundation of a technology or library.

[3] Matt DesLauriers on Frontend Masters: https://frontendmasters.com/teachers/matt-deslauriers/

After setting up the notebook, I wanted to create a crystal shape. I decided to use `Three.js`'s `PolyhedronGeometry` because I could define a set of points and specify which three points would make up each triangular face. On the first day, I only managed to create a triangle, but on the second day, I managed a 3D (rotating!) crystal as well as the crystal shape I had in mind (Figure 11.1). And even though I later found a better and easier way to create the crystals I wanted, I'm really grateful for the practice `PolyhedronGeometry` gave me to think through WebGL's *x*, *y*, and *z* coordinates.







`Fig.11.1`

Notes trying to figure out the math for where to position the points, and the two resulting successful crystal shapes.

Once I was satisfied with the shapes, I moved on to learning how to add color to them. I went back through the workshop's section on vertex and fragment shaders and played around with Matt's fragment shader code (Figure 11.2), which taught me some commonly-used GLSL (OpenGL Shading Language) functions.

Shaders were originally computer programs used for shading, but are now used in a variety of ways in computer graphics. In WebGL, fragment shaders are used to color a shape, and vertex shaders are used to manipulate the position of each point in a shape.

`Fig.11.2`

Experimenting with Matt's fragment shader code.

[4] Matsuda & Lea, "WebGL Programming Guide": https://sites.google.com/site/webglbook/

The next goal was to use the fragment shader to color the crystal shape, but I wasn't a fan of how it ended up looking, so I decided to read more into the different ways of outputting colors with shaders. This is when I turned more heavily to Patricio Gonzalez Vivo's *The Book of Shaders*,[5] and in particular, the chapter on "Shaping Functions." I learned about GLSL's built-in trigonometry and math functions, which take in a number (or a set of numbers) and output another. I also learned about the `mix()` function, where I could input two colors and not only get a new color between those two, but also individually mix the Red, Green, and Blue channels to make completely new colors that aren't between the two original colors. Once I combined those two functions, I was able to get beautiful gradients and shapes.

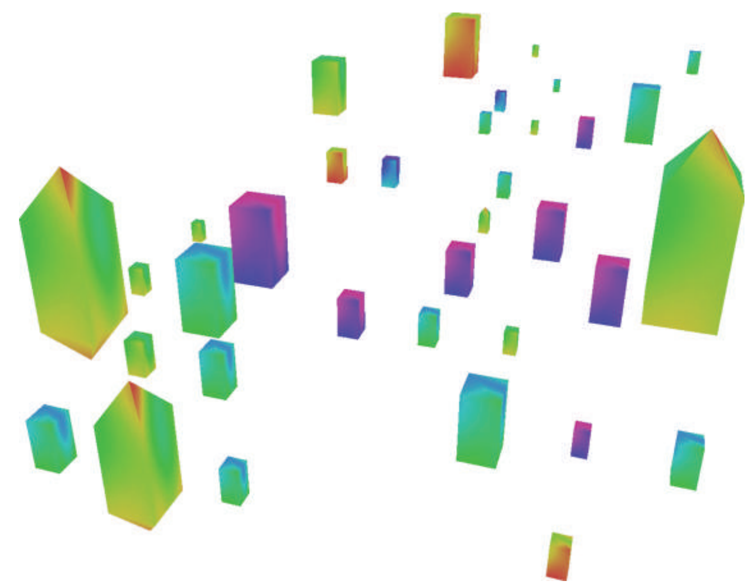
`Fig.11.3`

Gradients that were created by mixing blue and yellow, and tweaking the Red, Green, and Blue channels at each "pixel" with `power()`, `sine()`, `absolute()`, `step()`, and `smoothstep()`.

When I felt happy with the colors, I switched gears and started plugging in the data: number of faces for number of references at the bottom of the Wikipedia page, size for number of Wikipedia backlinks, and color for the Nobel Prize award category.

`Fig.11.4`

First attempt at the crystals with the data plugged in.



[5] Patricio Gonzalez Vivo, The Book of Shaders: https://thebookofshaders.com/

I really didn't like this output for two reasons:

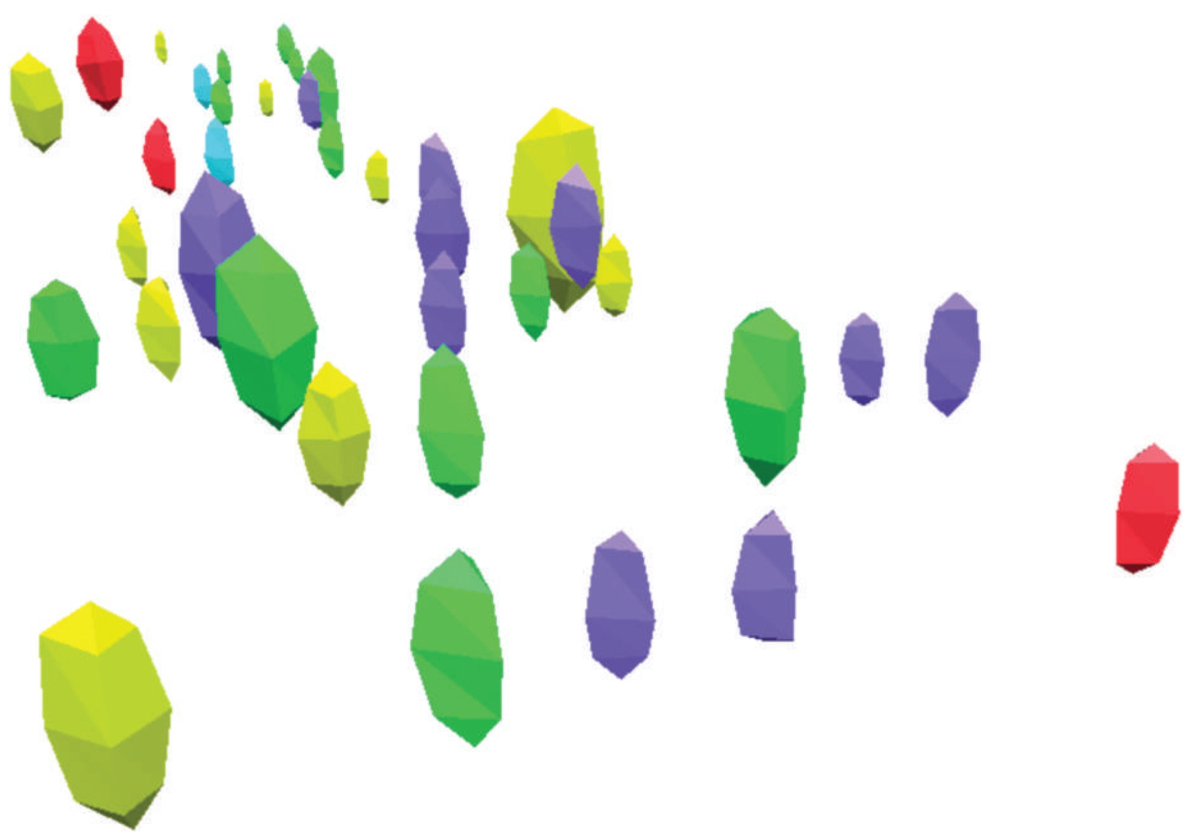
- I was hard-coding each shape, so I only had three distinct shapes: a rectangle, a rectangle with a triangle on top, and a rectangle with a triangle on the top and bottom. This meant most of the "crystals" were just rectangles.
- I also wanted one color per category for a total of six colors; so for each crystal I only passed in one color and programmatically manipulated that color with a shaping function. The resulting colors didn't look good at all.

It was around this time that I came across a demo for Bloom[6] (a post-processing effect that gives objects a glow around them), and in that example there were round, gem-like objects that looked quite similar to the crystals I wanted.

When I looked at the code, I learned that it used `Three.js`'s `SphereGeometry` but with the `flatShading` option on, which shows each distinct face of a shape instead of the default smooth, round surface. This made me realize that in addition to being able to manipulate and mix colors to get new ones, I could also manipulate the settings on an existing `Three.js` geometry to get a new (and completely different looking) shape—very cool! So I swapped out the `PolyhedronGeometry` with `SphereGeometry`, set the width and height segments to programmatically vary the number of faces depending on the data, stretched out the shape by setting the vertical scale to twice the horizontal scale, added jitter to the vertices so that each crystal looked a little different, and ended up with a much more interesting set of shapes:

Fig.11.5

The new crystals where I programmatically set the number of faces, instead of hard-coding every shape.



[6] https://vanruesc.github.io/postprocessing/public/demo/#bloom

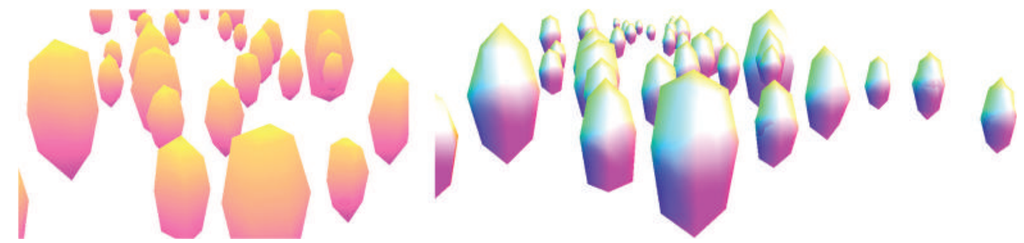
Once I had my new shapes, I went back to my fragment shader code to apply what I learned about color mixing and shaping functions (Figure 11.6).

I liked the new colors, but didn't like that I had lost the hard edges. Thankfully, Matt taught me I could get the definition back by calling `computeFlatVertexNormals()` on the geometry and adding it to each "pixel" color in the fragment shader. This not only made the edges really apparent, but also faked a light source, making the bottom and back of the crystals appear darker and in shadow (Figure 11.7).

From there, I played around with two sets of gradients: one for awards in the humanities (Peace, Literature, Economic Sciences), and another for awards in the natural sciences (Physics, Chemistry, Medicine) (Figure 11.8).

Fig.11.6

The crystals with the new fragment shader code applied.



I love how much the first one looks like sweet potatoes, which is seriously my dad's favorite food. (*≧▽≦)

Fig.11.7

The crystals with the edges defined.

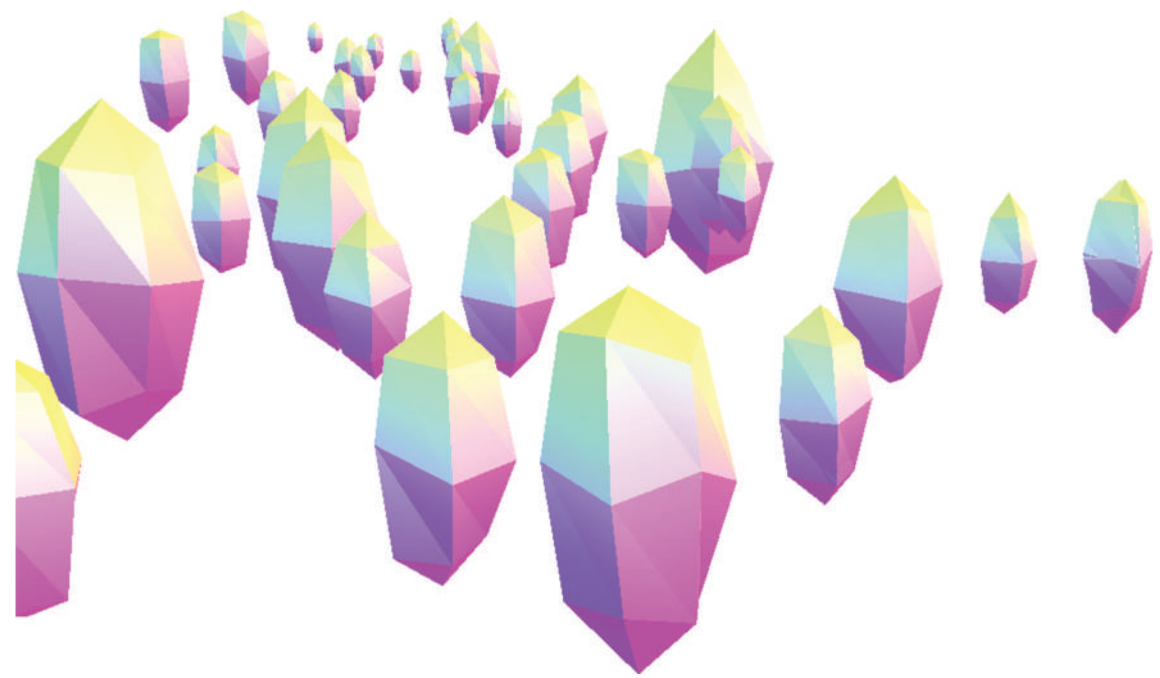


Fig.11.8

Experimenting with two different color schemes to distinguish awards in the humanities and natural sciences.

Two because I knew that I couldn't come up with two distinct gradients

Next came the background. I created the “floor” by using a `PlaneGeometry` placed below the crystals, dividing it up into a constant number of segments and jittering the *y*-position of those segments to create a sense of uneven ground. I created the “sky” by placing a huge sphere around the scene with the viewer inside of it. I experimented with three different kinds of lights: hemisphere lights and ambient lights to give the “sky” a nice sunrise glow, and directional lights to cast shadows from the crystals to the “floor” (Figure 11.9).

To finish the piece, I added “stars” to represent all the men who won Nobel Prizes in the same time period, as well as annotations for each crystal. It was a fun challenge trying to add text for the annotations: I tried to create the text using `Three.js`’s `TextGeometry` at first, but the page became completely unresponsive because `Three.js` was rendering each letter as a 3D object. After some Google searches, I found a solution to render the text within canvas, create a `PlaneGeometry`, and use that canvas as an image texture to fill the `PlaneGeometry`—a much more performant solution!

I was quite upset when I learned that there have been 866 male Nobel Laureates but only 53 women Nobel Laureates. There’s always a gasp of disbelief when I reveal to an audience that the stars are male Nobel Laureates, because there are so many of them compared to the crystals.

My favorite part of the visualization is how I decided to use the third dimension to represent time. The crystals are placed according to the decade the laureate received her award, so the closer to the front (and the viewer), the more recent the award. But the decades are only revealed when a user “flies up” to view the crystals from above; if they “walk through” the crystals at ground level, they will only see information about each woman. I did this because I really wanted the user to learn more about each laureate first, before they “flew up” for a holistic view.

To finish, I created a landing page with a legend for my legends (hehe), and managed the componentization and interaction with `Vue.js`.

Unfortunately, because the text is rendered in canvas, it is treated as an image and thus isn’t a11y (accessibility) compliant—something I only realized later on.

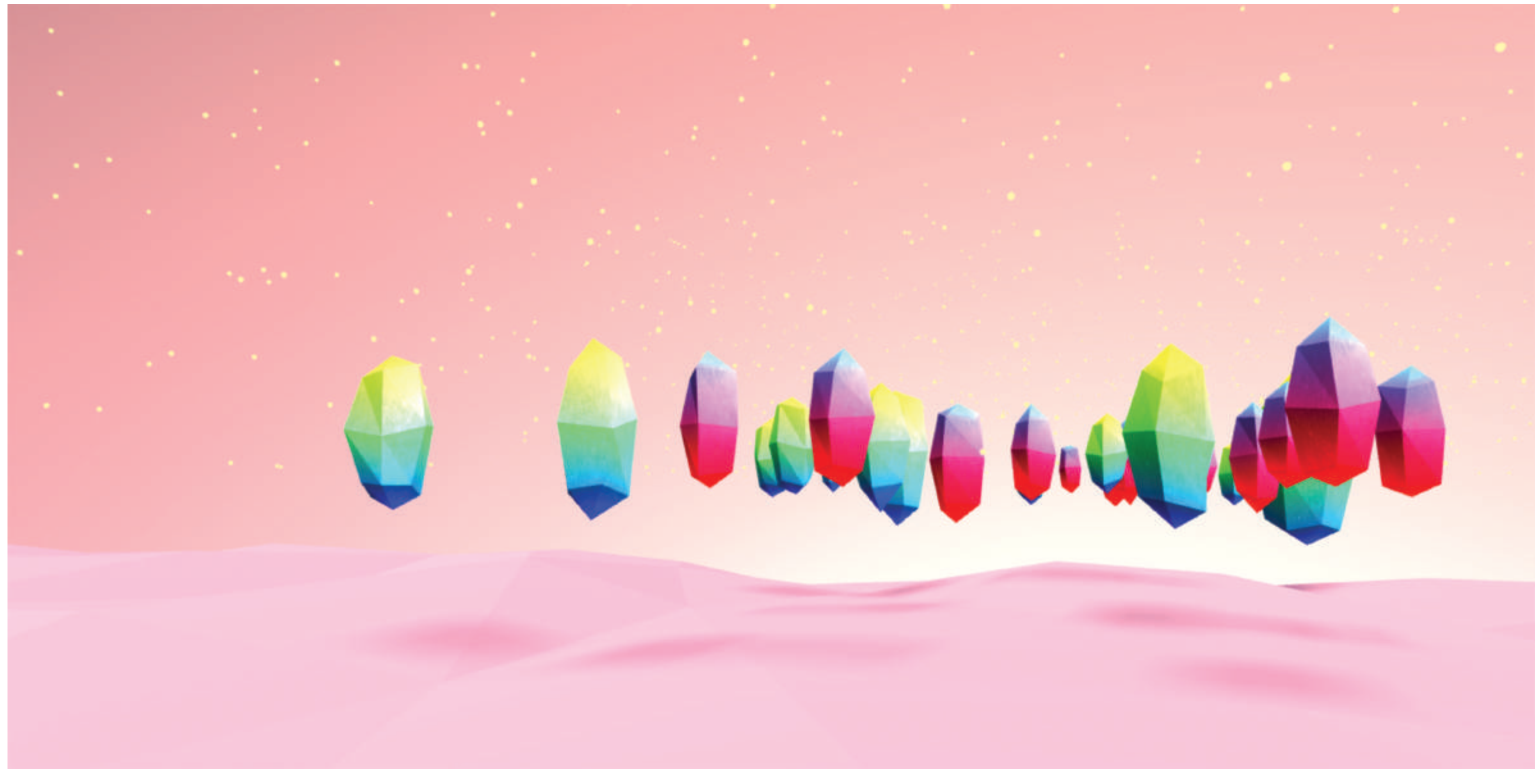


Fig.11.9

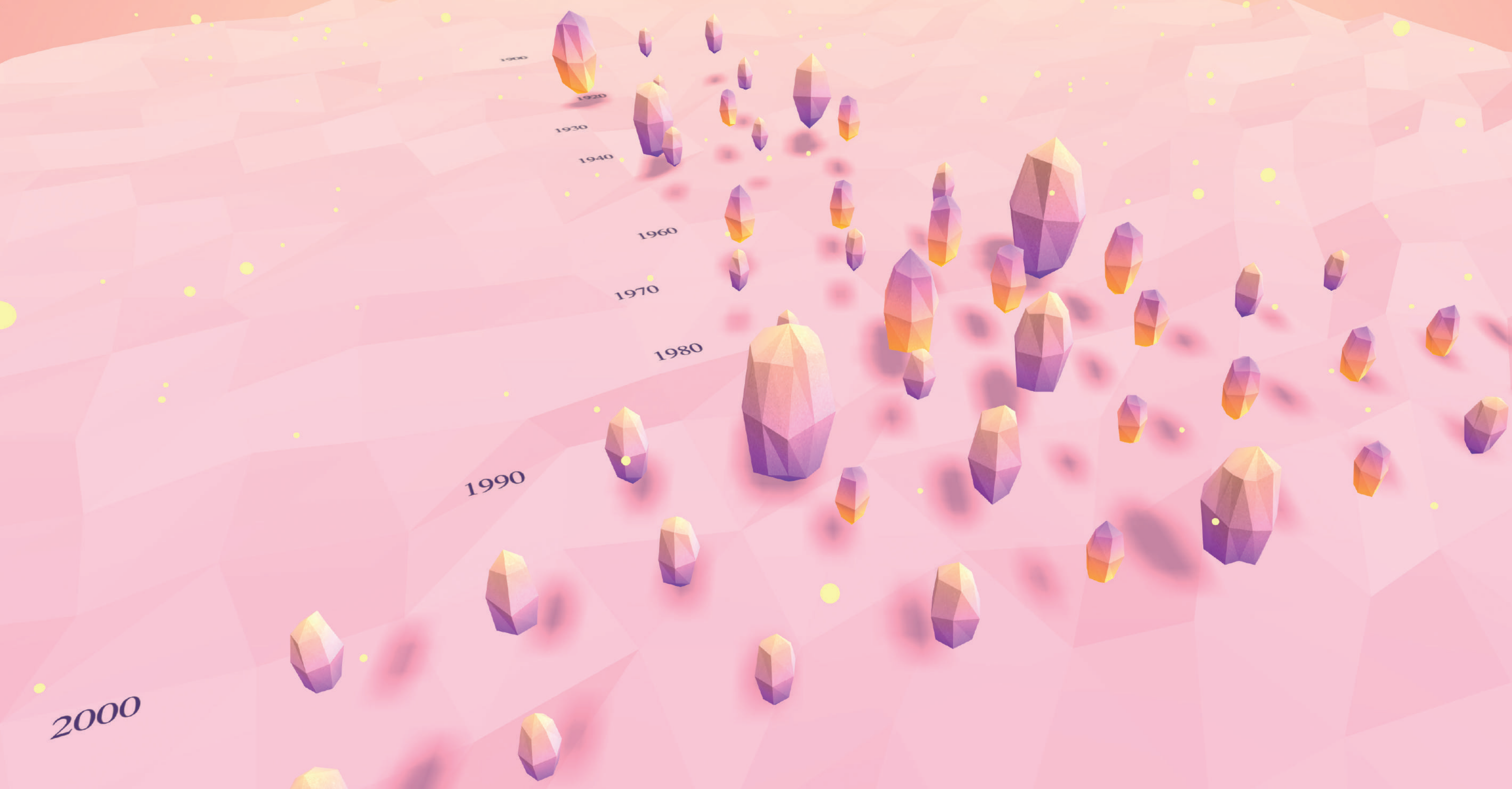
Background with a floor and sky added.

Reflections

This project was super fun, and I’m so proud that I was able to finish it in three weeks—something I hadn’t been able to do since the “Travel” project. I was also able to teach myself `Three.js` and a little bit of GLSL, which I wanted to do for a very long time. This project was a great opportunity to use the third dimension and also gave me the confidence to experiment with more 3D and physical installation projects in the future. But most importantly, I’m so glad I chose this dataset of women Nobel Laureates; it has since motivated me to work with and highlight datasets featuring underrepresented groups.

Legends

↳ shirleywu.studio/projects/legends



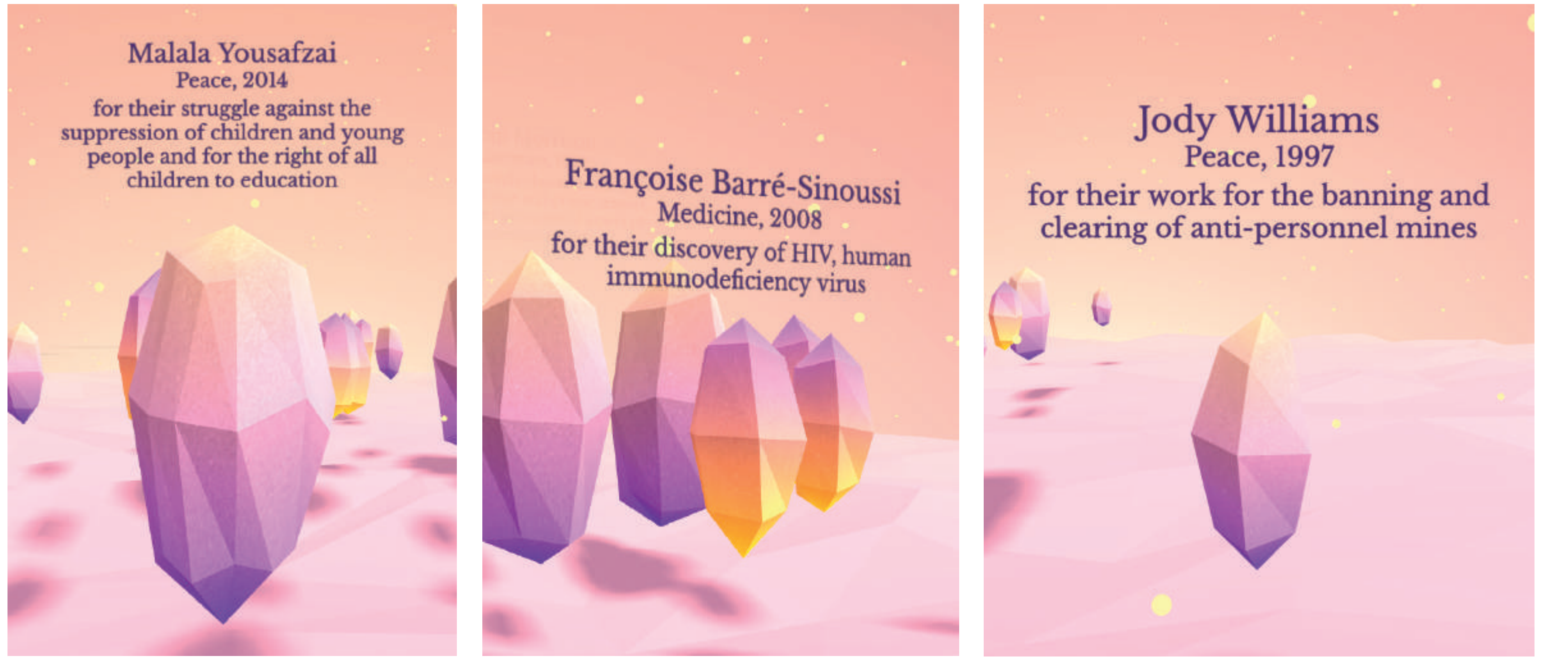


Fig.11.11

Recent women Nobel Laureates
I find incredibly inspiring.

Fig.11.12

I also turned this project into washi tape! But because each crystal wasn't distinct enough, I turned them into flowers instead.

Fig.11.13

My main goal with this project was to create a physical installation, and I got the opportunity when I collaborated with my studio-mate and illustrator-muralist-extraordinaire Alice Lee! It was a fun learning experience as I had to pay attention to details I never had to when working digitally, like physically organizing the pieces by prize category so we painted them the correct colors (top left), and then again by decade so that we hung them up in the right order (middle left).