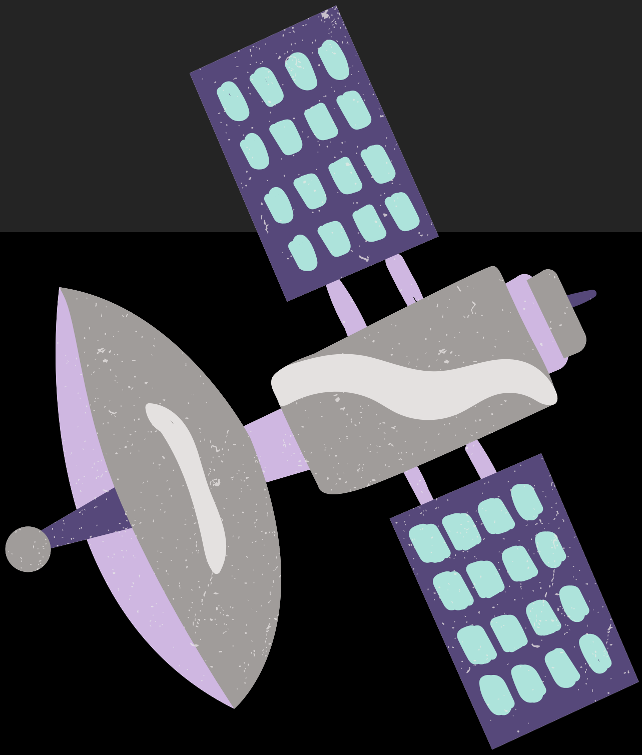
Astro
VIZards

Satellite Viewer

- Data Visualization -



Written By

Miguel Crespo
336122

Aybars Yazici
342664

Can Kirimca
353437

From idea to product

Many people know there are satellites orbiting around our planet every day. But not so many know exactly where they are or what they are doing.
In other words, the problem is that this information is not easily found by the general public.

This project presents a solution to this challenge in the form of an easy-to-understand visualization featuring all the information needed for the curious user.

Idea

Our way to achieve this goal is with a visualization primarily based on the position of the satellites. For that, we leverage a map (both 2D and 3D) which will allow users to understand the extent of the satellites above their heads.
At the same time, users can easily query information about them which is presented in a compact way.

Because we aggregate information, is easy for the user to check exactly what he needs, allowing him to answer queries as satellites owned by a specific country.

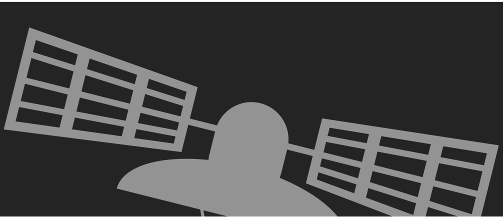
Furthermore, satellites are non static objects, so we focus also our attention in designing a visualization that accounts for that.

Target audience

This visualization has been focused on two types of users:

The first one is people without prior knowledge that wants to know more but is not interested in the specific details.
We provide map visualizations as an engaging way to increase their knowledge.

The second one is people that need to find information more detailed about the satellites.
For this use case, we provide filtering capabilities as well as curated information about the satellite



About our dataset

Our dataset contains information about the satellites currently in the Earth's orbit. We obtain this dataset by joining two sources of data:

- A dataset gathered from NORAD that contains a unique reference number and name for each satellite, and also its orbit information in TLE format. This information enables us to compute the coordinates of a satellite at any time instant. An example of a TLE is the following (Corresponding to the International Space Station):

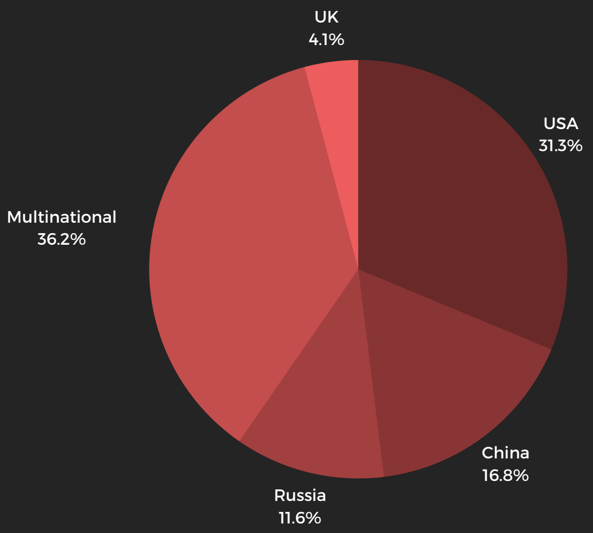
```
ISS (ZARYA)
1 25544U 98067A   08264.51782528 -.00002182  00000-0 -11606-4 0  2927
2 25544  51.6416 247.4627 0006703 130.5360 325.0288 15.72125391563537
```

- A Kaggle database containing additional information on the satellites such as:
  - The country and organization operating the satellite
  - Users of the satellite (Government, commercial etc.)
  - Purpose of use (Communications, observation etc.)

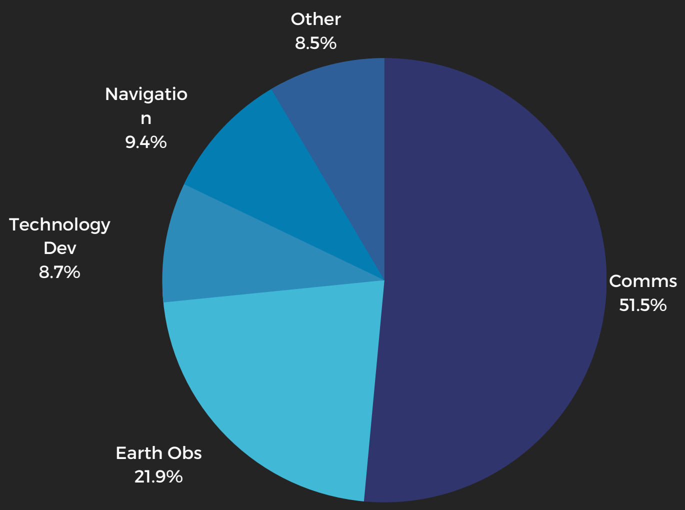
Statistics of data

- Total Number of Satellites: **939**
- Number of satellites in use: **239**

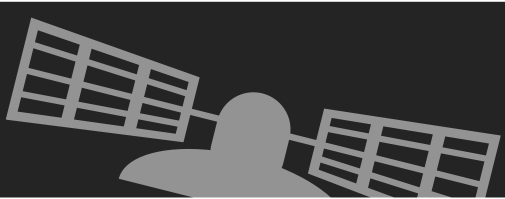
Operator Countries:



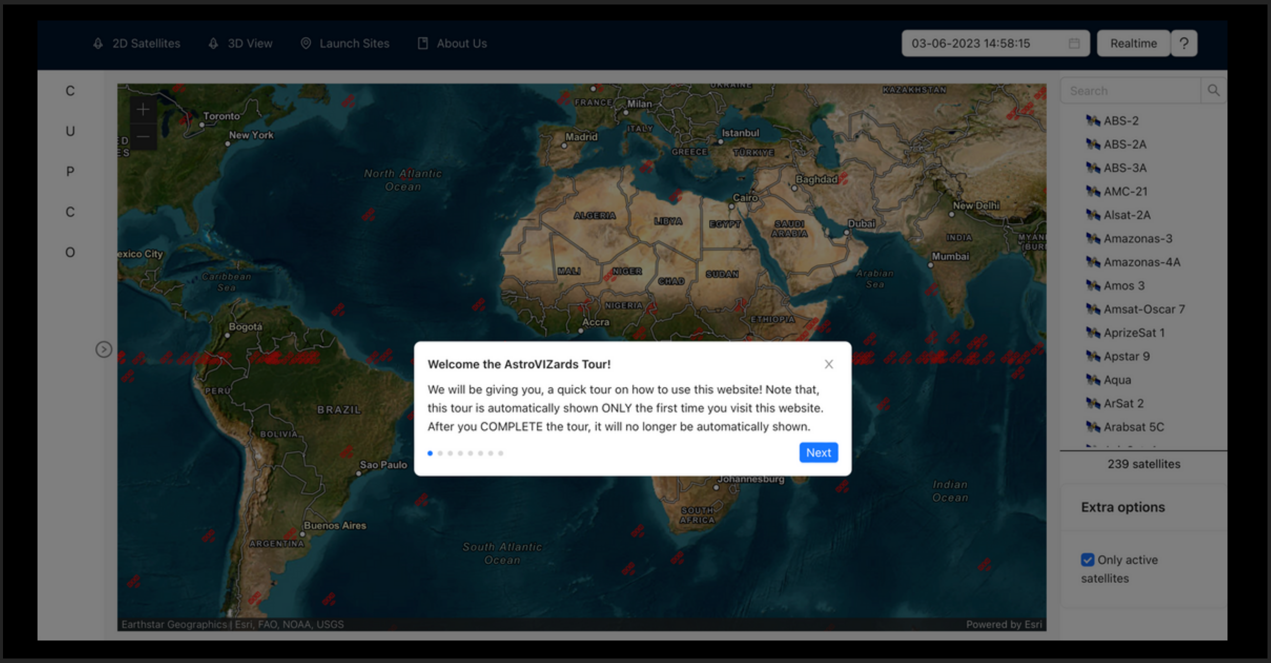
Purposes of Use:



Astro
VIZards



Our visualization



Welcome the AstroVIZards Tour!

We will be giving you, a quick tour on how to use this website! Note that, this tour is automatically shown ONLY the first time you visit this website. After you COMPLETE the tour, it will no longer be automatically shown.

Next

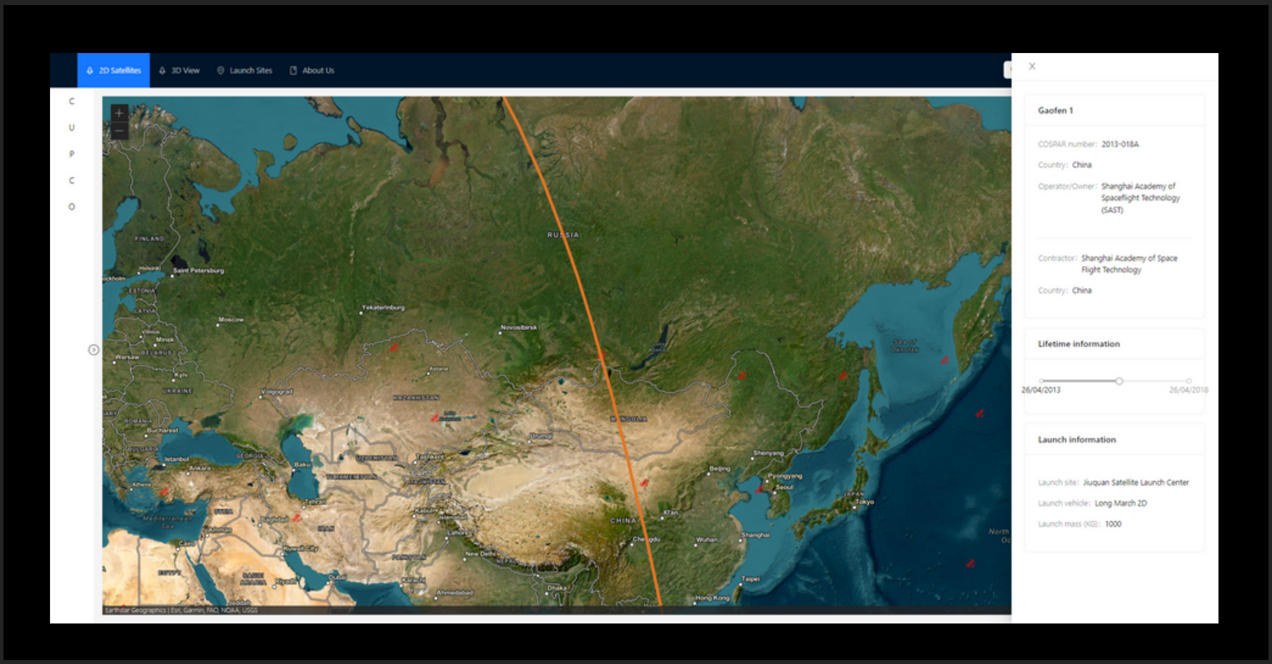
- ABS-2
- ABS-2A
- ABS-3A
- AMC-21
- Alsat-2A
- Amazonas-3
- Amazonas-4A
- Amos 3
- Amsat-Oscar 7
- AprizeSat 1
- Apstar 9
- Aqua
- ArSat 2
- Arabsat 5C

239 satellites

Extra options

☑ Only active satellites

The first visit to the website will guide the users to different sections as an overview of the capabilities of the visualization.



Gaofen 1

COSPAR number: 2013-018A

Country: China

Operator/Owner: Shanghai Academy of Spaceflight Technology (SAST)

Contractor: Shanghai Academy of Space Flight Technology

Country: China

Lifetime information

26/04/2013 26/04/2018

Launch information

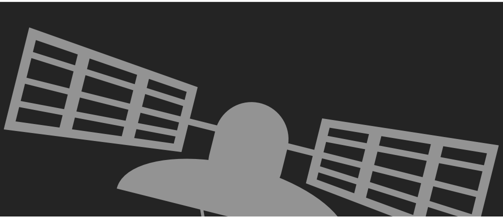
Launch site: Jiuquan Satellite Launch Center

Launch vehicle: Long March 2D

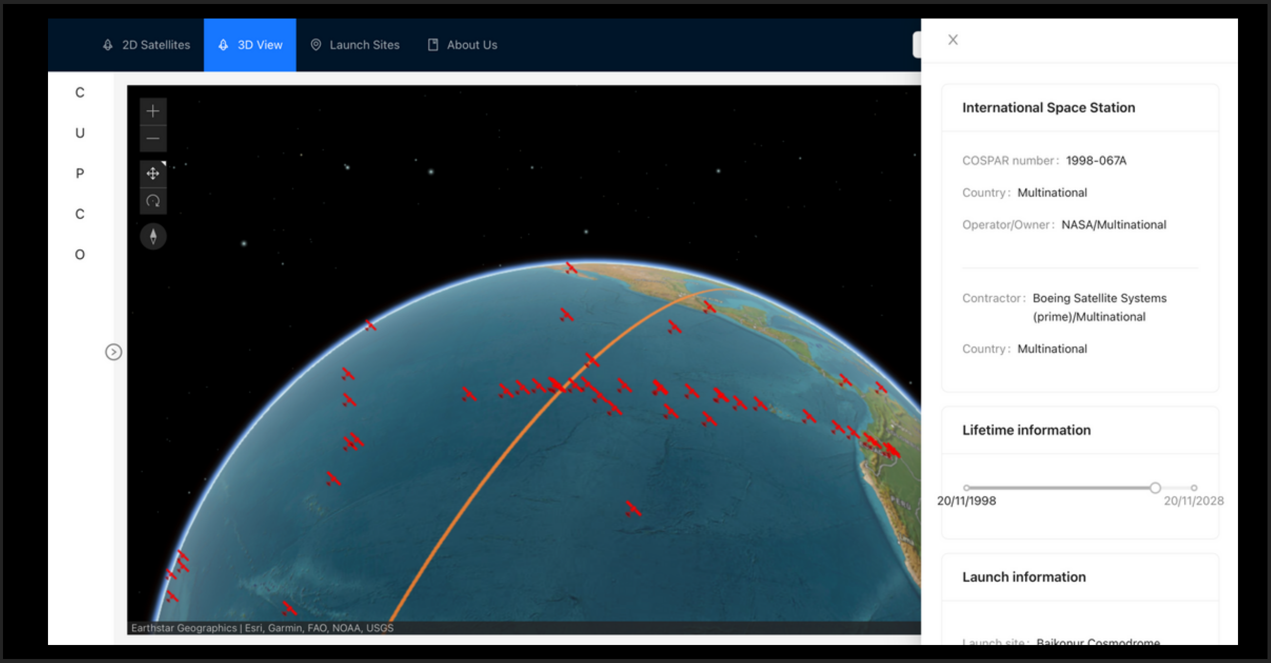
Launch mass (KG): 1000

In this first section, a 2D map will present all the satellites orbiting the earth. After clicking on a satellite icon, the orbit of the corresponding satellite is displayed. On the right side, one can see detailed information about the satellite, such as its owner, purpose and launch site.

Astro
VIZards



Our visualization II



International Space Station

COSPAR number: 1998-067A

Country: Multinational

Operator/Owner: NASA/Multinational

Contractor: Boeing Satellite Systems (prime)/Multinational

Country: Multinational

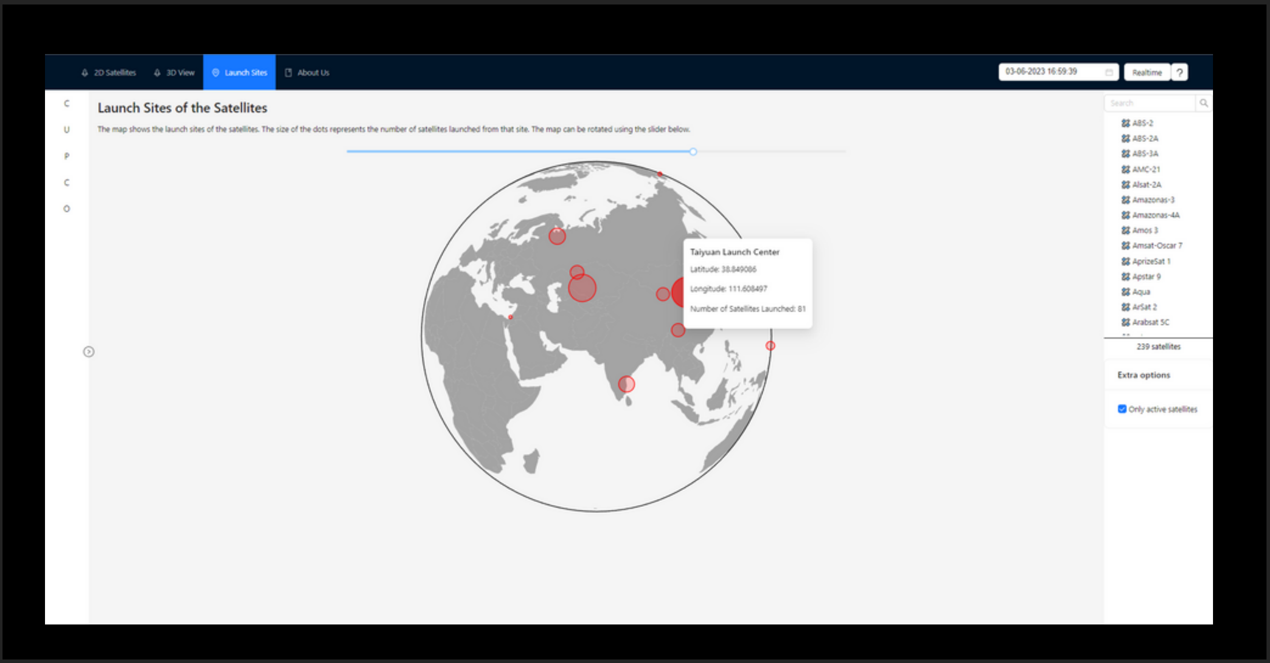
Lifetime information

20/11/1998 20/11/2028

Launch information

Launch site: Baikonur Cosmodrome

We also provide the option of viewing the satellites on a 3D map. Same as the 2D map, the user can view the orbit and detailed information by clicking on a satellite.



Launch Sites of the Satellites

The map shows the launch sites of the satellites. The size of the dots represents the number of satellites launched from that site. The map can be rotated using the slider below.

Taiyuan Launch Center

Latitude: 38.849086

Longitude: 111.608497

Number of Satellites Launched: 81

- ABS-2
- ABS-2A
- ABS-3A
- AMC-21
- Alsat-2A
- Amazonas-3
- Amazonas-4A
- Amos 3
- Amsat-Oscar 7
- AprizeSat 1
- Apstar 9
- Aqua
- ArSat 2
- Arabsat 5C

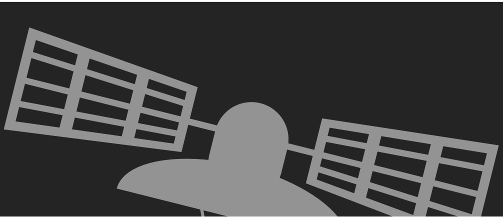
239 satellites

Extra options

☑ Only active satellites

On the launch sites page, we provide a projection of the launch sites on the globe. Each site is represented by a dot with radius proportional to the number of satellites launched from that location. By hovering the cursor over a dot, one can view information on the corresponding site.

Astro
VIZards



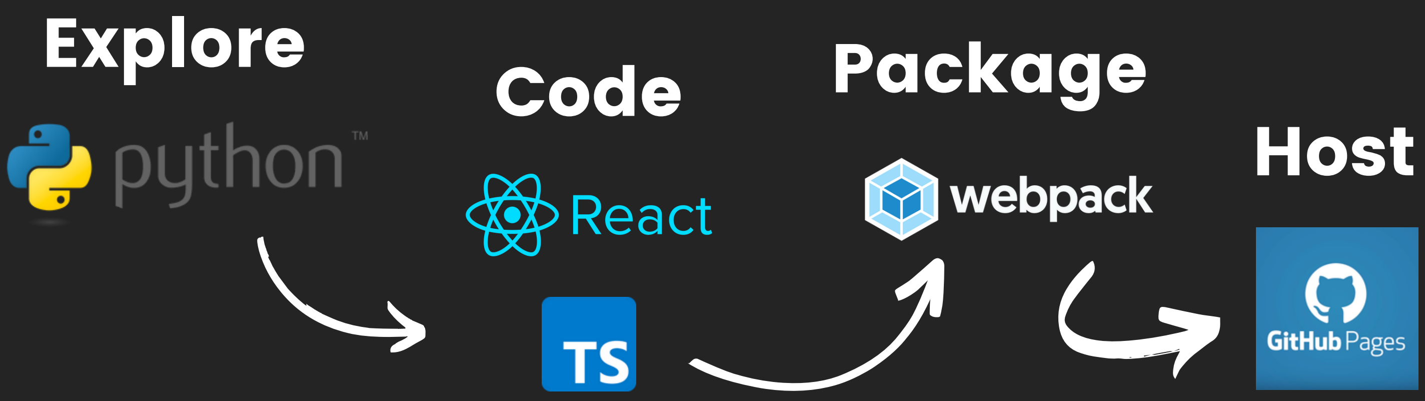
Technical implementation

Dataset preparation. We have implemented all the processing of the data locally using Python. The idea here was to relax the requirements of the user's machine when loading for the first time the website.
Quite important has been the detail of creating the relation between TLE data and the non-orbits dataset to later present that information together in the visualization. We have used Pandas as the state-of-the-art tool it is.

Website. The main requirement of our website is that it should run locally but present interactivity. We opted to leverage React to implement all the functionality. All the code has been written in TypeScript.
Given that visualization should be aesthetic, we decided to use Antd for the CSS framework, providing us with a playground that is responsive (a requisite for all visualization nowadays).

Map visualization. We have leveraged ArcGIS library to develop all the logic regarding maps. The most important feature is that it provides professional maps what we can use to plot our satellite's information. This provide a smooth experience to the user regarding tracking the satellites or zooming to know if an specific satellite is above his area.

Publishing the visualization. The last part is packaging and publishing. For improving speed, we decided to introduce the prepared data as a JSON that could be easily loaded without speed penalties in our visualization. Additionally, we leverage Webpack for joining all the assets in big chunks that are easier to download with the browser. This step also compresses the text if possible and is needed for providing the best experience.
Finally, our visualization is available to the world in Github Pages.

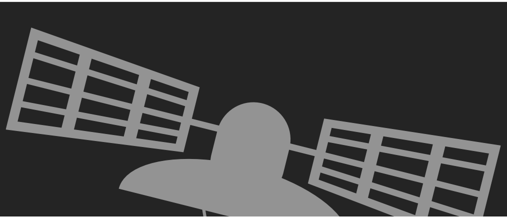


Explore

Code

Package

Host



Challenges and decisions

Frontend

Designing an interface that is intuitive and user-friendly posed a significant challenge. It required gaining a deep understanding of the users' needs,. Ensuring the availability of appropriate controls and tools to enable efficient exploration without making it too complex was definitely one of our most cruicial challenges. Hence our decision to add a tutorial for our website on the first visit of the users.

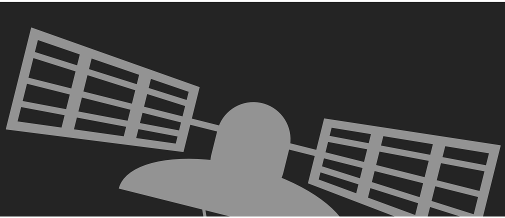
Choosing the right visualization techniques to effectively portray the satellite positions posed a yet another complex challenge. We carefully evaluated options such as employing 2D or 3D maps, overlays, and interactive graphics to accurately and intuitively communicate the satellite positions to users. A crucial aspect was finding the right balance between providing sufficient detail and maintaining simplicity in the visualization.

State Management

The Map visualization library we have used, ArcGIS, was more complex to integrate than we thought. In our initial design, we had excessive amount of states being passed as props to numerous components resulting in one small state change causing loads of rerenders, slowing down the application. Thus having a efficient yet simple state management structure was of utmost importance.

The state management structure had the goal of decoupling the various inputs of our app. We created 2 manager classes, MapManager, SatelliteManager. These classes only had the goal managing their respective data. For ex, all filtering & info fetching actions had to go through the Satellite Manager. Meanwhile, zooming, map click event handlers, drawings had to go through the Map Manager. This way, we had a quick and efficient method to identify bugs as there could be only one point they could occur.

The initilizations of managers were optimized through the useMemo hooks. The components then were connected to these states through the useState and useEffect hooks which have the satellite state as their dependency.



Related Work

This idea is not new, but prior tools in this field lack some features that we find quite interesting to have. Many users only want to explore what is above their heads in the most easy way.

- **Satellite Explorer** is one of such tools, but while being aesthetic, it suffers from being easy to use or present the information in a direct way.
- **In the Sky** features an amazing overview of the position of the satellites (similar to ours) but their visualization is a bit less aesthetic and the number of options that the user need to select to filter satellites makes it not usable.
- **Satellite Map** presents the best 3D globe visualization, but it is not clear how to filter or track down a specific set of satellites. Also, it does not show the orbits of them or a global overview of all satellites.

Our contribution to the community is an easy-to-understand visualization with advanced filtering capabilities that can help any enthusiast about satellites to understand and check their information.
We additionally paid a lot of attention to draw extra information as easy as possible.

Peer Assesment

Miguel Crespo. Research the visualization topic. Exploratory analysis and search of the different datasets to be used in this project. Development of the pipeline for loading the data and plotting exploratory results. Analysis of the TLE format. Implementation of the routines that compute the current position and orbit of the satellites. Implementation of the details of satellites, real-time tracking, and part of the layout. Writing of Report 1, 2, and 3.

Aybars Yazıcı: Implementation of the majority of the react app, including the responsive layout, routing, state management and the tutorial popup. Analysis and implementation of the ArcGIS SDK for JS: (1) Rendering the maps both 2D and 3D, (2) Drawings of graphics such as satellites (3) Connecting it to create-react-app state, for dynamic updates with respect to user events such as filtering and clicks. (4) Optimization of drawings. Writing parts of the report 1,2, and 3.

Can Kirimca: A small portion of data exploration. Processing the dataset for the launch sites plot. Implementation of the launch sites plot page containing the projection of the launch sites onto the globe and interactive events displaying site information. Implementation of the AboutUs page. Writing of Reports 1,2 and 3.