Siddhanth Sabharwal, 999229332

Philip Weeks, 805884447

George Zhou, 912391771

Tanner Embry, 913351817

ECS 163, Final Project Final Write-Up

Meteorite Landings Visualization

A description of the dataset and its relevance. That is, what you are trying to visualize, and why you want to visualize it.

The dataset we have chosen is "Meteorite Landings" from NASA's Data Portal. It is a comprehensive data set from The Meteoritical Society that contains information on all the known meteorite landings. It contains information such as location, time, type and mass of meteorite. This dataset is relevant to us because we are all big fans of space and we are interested in designing a visualization engine that would animate some part of it. It is very relevant also because meteorites are literally foreign bodies from outside space on Earth and we find that to be very interesting.

First, we are trying to visualize the meteors falling from the Earth. We want to visualize this because it is the actual action of the foreign body landing on Earth, and is the only thing that is time dependent. Motion easily grabs the audience's attention and is useful visual channel here. It expresses the landing positions and times of the meteorites well and we believe it is effective at it as well.

Second, we are trying to visualize how far away meteors are from a user chosen location. We want to visualize this because it gives the user the ability to see meteorite distribution from their point of view. This allows us to implement a user driven narrative with free interactivity.

Third, we are trying to visualize the classes of the meteorites because someone without domain knowledge will have an impossible time understanding the classification system used by scientists. This allows us to sort the data to expose patterns and the user to navigate through the dataset based on categories they are interested in.

How you initially proposed to visualize the data, and why you want to visualize it that way.

We initially proposed to visualize the data using a time-based animation, a radial chart with logarithmic scaling, and a force directed hierarchy graph.

We wanted to use a time-based animation on an interactive world map because it would allow the user to filter based on mass, filter by time range, and pan to a location and see the closest meteorites there. We wanted to use a radial chart with logarithmic scaling to have more room to show the meteorites as compared to the first view where they may be cramped together. We wanted to use a force directed hierarchy graph because it would allow us to have the meteorite classification chart so the user could see how the classifications work and be able to adjust the nodes of the chart if they were interested in seeing how nodes interact with each other.

How the visualization system changed throughout the design / implementation process, such as things you tried that just didn't work.

Our first challenge was the preparation of the Meteorite Landing dataset. At over 45 thousand records we had a lot to work with and first discarded any records with incomplete or missing information that we wanted to use, such as map location, mass, and type. To trim the dataset further so our animation didn't crash the browser, we removed older records before the creation of NASA in 1953. The data also contains data for meteorites as small as 0.01 grams, but in the interest of animation speed and making the markers less cluttered on the screen we set a baseline threshold at 10 grams, and preprocessed out everything below that. In practice, we found that even with that threshold the map was very cluttered and suffered from performance issues and later increased the default threshold to 100 kilograms, but we give the user a dropdown menu to select a lower threshold if they desire.

Our second challenge was implementing a slider so the user could filter by year range. The slider lets you select a beginning year and an end year. For example, if someone choose 1987 to 2001, he will be able to see all the meteorites that fell during that period. Since the only timestamp we have is year the meteorite fell, we were having issues with the slider being too dynamic and being able to select month and day ranges. To fix this, we made the slider snap onto the closest year to deal with rounding issue. Next, we wanted to make all our filters interact with each other and have an add-on effect, so we had to dynamically calculate the working set of meteorites after each interaction and pass that between our filters and interactions. The mass filter works along with the year slider, so that the visualization will show meteorites that fell during a certain period and within the mass range selected. Once we finished implementing the first visualization, we had to link it with the other views in our interface.

Our third challenge was gathering enough domain knowledge about meteorite classification to effectively implement the hierarchy. We solved this issue by adding a hyperlink for each meteorite that links to the entry for that meteorite on the Meteorological Society webpage. Now the user can click on those links and be directed to a page that shows more information on that meteorite without cluttering up our own visual interface. We thought this would be more effective than adding a glossary section in our interface.

Justifications and explanations for changes and your team's design decisions.

Making the default setting for the visualization the most restrictive in terms of mass filtering was done with performance in mind. We don't want the initial animation to be too long or too laggy. During testing we realized that since we are making requests to the Google Maps API to plot our markers, on a computer with poor rendering the markers would appear in a very laggy way. We also don't want the user to get bored and close the page if they have to wait too long for the animation to end and begin interacting with the system.

Making the slider snap to the closest year and not be fully dynamic was done so the user doesn't incorrectly think they are filtering by parts of the year. We also added the ability to clear the slider by clicking on an unselected region which then clears the time filter and shows all markers.

Instead of trying to define obscure classification terms ourselves, we decided it would be more accurate to simply link to the expert's description. While we wanted to try and use a visualization to make domain specific knowledge more accessible, with over 350 unique classes in our data set we found this to be too ambitious without being a domain expert. We could identify some of the larger branches and since many classifications only appear a few times we decided to use add a major class attribute to our data set and select some of the broader categories to group out data set into, as well as some of the more interesting ones such as meteorites with Lunar or Martian origin.

A description of the final visualization system, including implementation details, visual

encodings, and interactions.

The final visualization system is composed of three things: a time-based animation on an interactive map, a radial chart, and hierarchy chart.

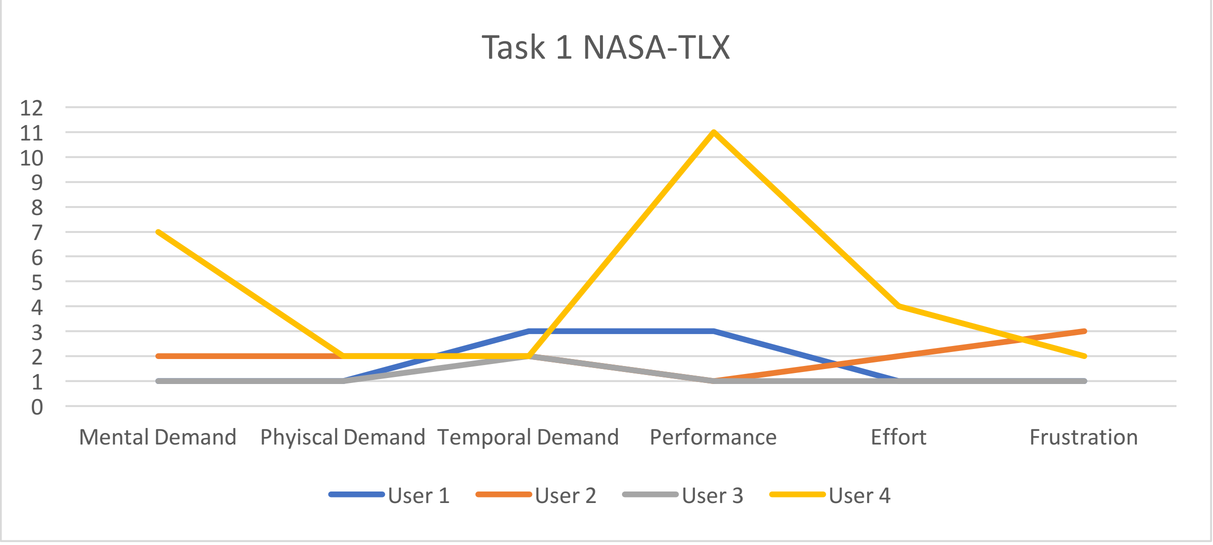
The time-based animation goes through our data and checks if the meteorite's mass is greater than our default level and if so makes a call to the Google Maps API to plot it on the map. During the animation the user can see progress through a marker in the upper right hand side of the screen that changes based on the year of the marker dropped. After the animation is complete, the user has a few affordances. First, they can replay the animation. Second, they can pan to a location based on the geocoder box. After the user types in a string and we pan to that location, we dynamically set the zoom level so at least 1 meteorite is visible. This was done so if the user searches for a location they are guaranteed to see something appear. Third, they can select the mass threshold and have more/less meteorites appear on the map. Fourth, the user can select a year range based on the slider below and filter the meteorites on screen. The filters work together so you can simultaneously adjust by year range and by mass. Fifth, you can pan and zoom the map using the buttons on the map itself. Sixth, you can open the radial chart after you type a location and have it centered to that point. Seventh, you can hover over any of the markers and see its class. Eighth, you can click on any marker and have its classification hierarchy appear. Ninth, you can change from satellite to map view based on a button on the map but that is default functionality provided by the Google Maps API.

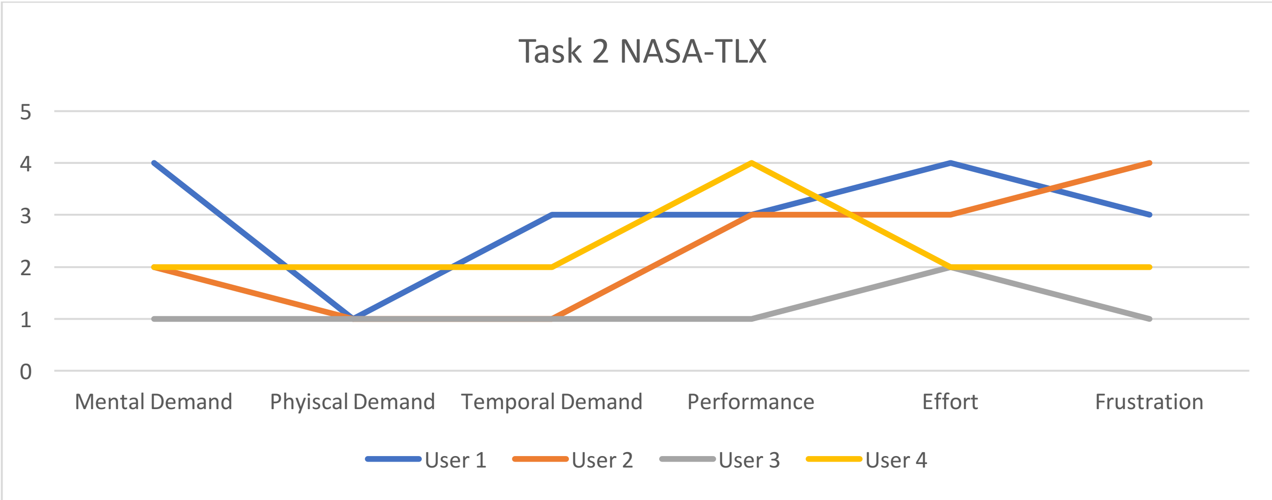
The radial proximity chart was intended to provide an alternative view of the data in a similar context to that of the map. When a point is clicked on the map a radial chart showing distance and bearing of all meteorites within 2500 kilometers is calculated and plotted on a logarithmic polar axis. By using a logarithmic scale we hope to display the same data in a different fashion, being able to see events close to the target spaced further apart so they can be examined in greater detail while still being able to see any regions with large amounts of clustering that might warrant further examination. To help support this usage we let clicks on target meteorites on the radial map navigate to and center the event on the main map. We originally allowed the entire data set to remain plotted on the radial chart even while the main map had filters with the idea that the radial chart would keep all information available even while playing with settings meant to adjust the animation, but the disparity between having things visible on one plot while missing from the main map could be confusing so this was dropped in favor of having the same filters for consistency.

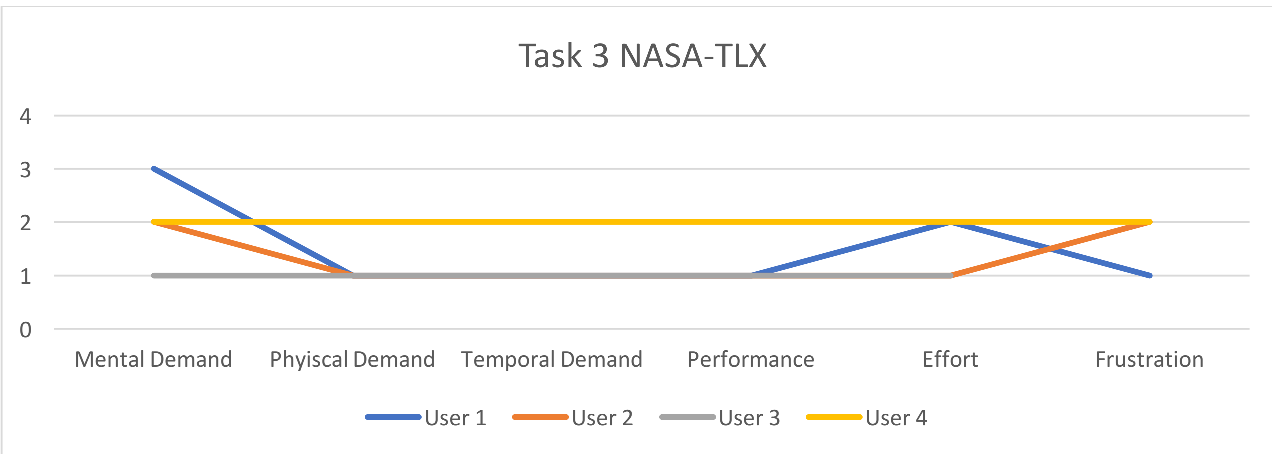
The classification tree chart was intended to let the user explore detailed information on any specific meteorite. Where the other visualizations focus more on providing a broad scope of information about the meteorites in general, this view shows the meteorite name and classification in a way that is easily understandable by a novice user that doesn't have extensive knowledge on the subject. With this visualization, the user can select any class of meteorite on the tree and all meteorites in the main world map view that fall under that class will be highlighted. From the tree overlay view, it is also possible to read more about the specific meteorite by directing the user to a database entry containing extensive information on it. Originally, we were going to have a much larger tree containing many smaller subclasses of meteorites but it ended up being too confusing for users and the highlighted groupings on the map would have far less entries to give any useful information. For these reasons, we grouped meteorites into the most common major classes and used these instead. This resulted in a much cleaner design and was easier to understand for the average user.

Your evaluation section.

We had another group evaluate our interface. The most common feedback we received were the affordances were not obvious so we added in tooltips and text labels to make the functionalities more obvious. The charts plotting their NASA-TLX results are below.







Apart from a user evaluating the performance of our first task as an 11 and the mental demand as a 7, no other feedback was higher than a 4 so we were quite pleased.

How tasks were divided among team members.

Siddhanth and George worked on visualization 1. Philip worked on visualization 2. Tanner worked on visualization 3. All members needed to interact their views with each other so there was quite a lot of cross-collaboration between the parts.