



Joseph DePasquale
CSCI E-3 Final Project
Explore the X-ray Universe
Spring 2016

Project Description

This project is a web-based application that allows the user to explore press release imagery from NASA's Chandra X-ray Observatory in the context of the whole sky. It leverages the power of image metadata in the form a JSON database, combined with the JavaScript API of Microsoft's WorldWide Telescope (WWT) application to provide the user with a full-sky, planetarium-like viewing experience that shows the location of each press release on the sky. This facilitates drawing connections between different objects that are usually seen in isolation from one another. An example of such a connection can be seen in figures 1 and 2 below which show the Chandra press image of the Flame Nebula (shown in Infrared and X-ray light) and it's contextual representation on the sky in the WWT window. Seeing the optical image of the Flame Nebula in an interactive environment allows the user to explore further and make the connection that this source is located in the constellation of Orion and situated near the belt stars. This can of course be related in the text of the press release, but there is a more immediate satisfaction in the self discovery allowed by this tool.

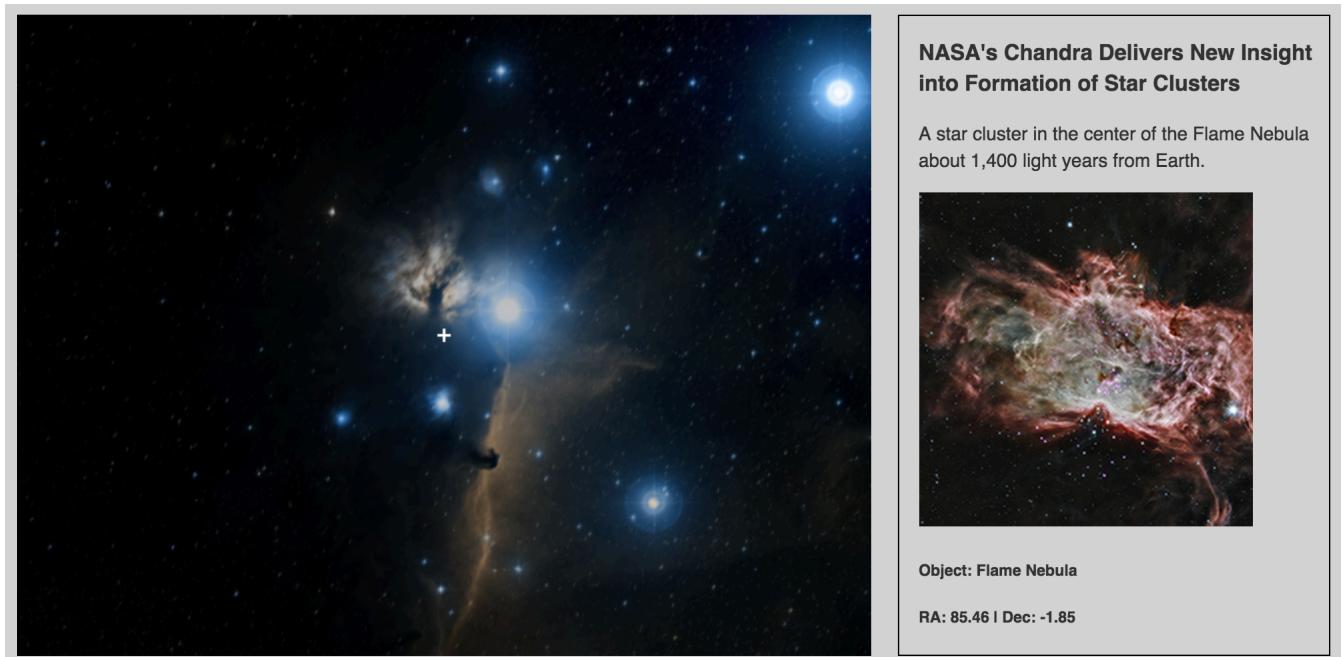


Figure 1: The Flame Nebula. On the left is the WWT view which uses optical data from the Digitized Sky Survey to provide the user with an interactive all-sky optical viewing experience. On the right is the Chandra press of this object.

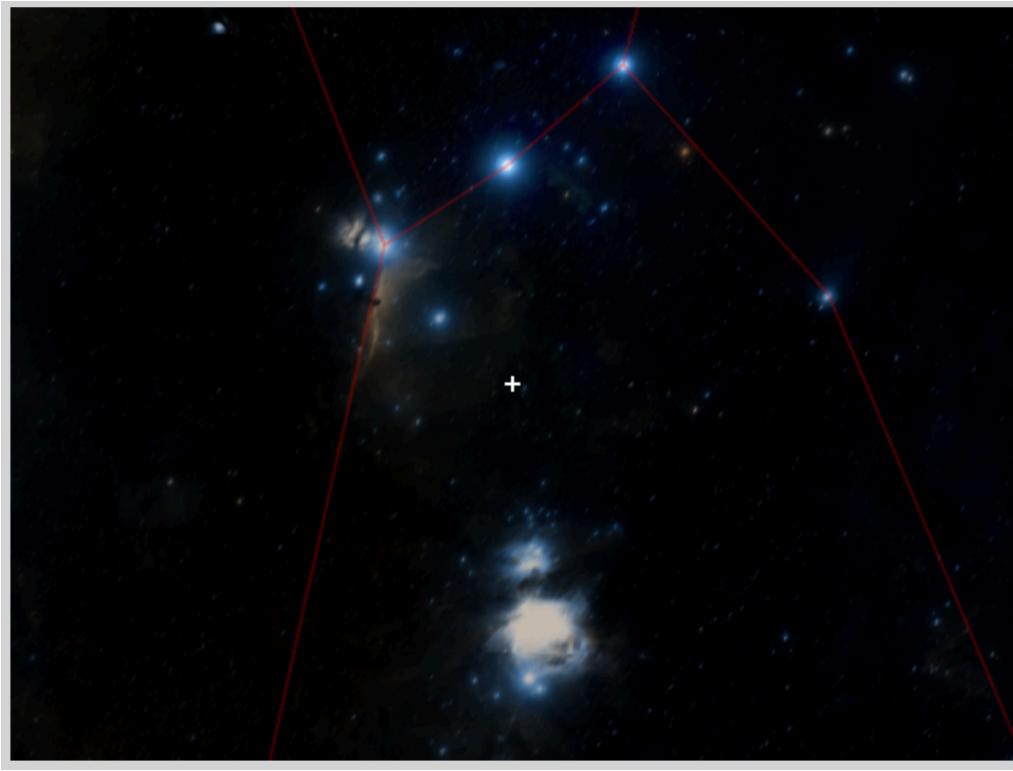


Figure 2: Zooming out further from the Flame Nebula shows that this source is actually next to one of the stars in Orion's belt, Alnitak, and located just above the famous Orion Nebula.

Project Specifics

WWT's JavaScript API provides the contextual backbone of this project. In the end, after setup and initialization, it really only takes one call to display the location of the current object on the viewing window, once the X and Y coordinates (known as Right Ascension and Declination or RA and Dec) are known. These data are obtained from a JSON-formatted data file, which is itself generated from image metadata and updated after every press release is published. Although the page serves up a random press release when first loaded, there are search and filter options available to the user at the bottom of the page. The user has the option of loading another random release or clicking the "more options" link to further refine the experience.

Under "more options" the user is presented with a form for doing name-based searches and/or filtering data by three broad categories: Stars, Nebulas, and Galaxies (see figure 3). There is also a table of all available press imagery, which dynamically updates according to the search and filter options and indicates to the user how many press releases are currently available as a visual cue that the searching and filtering are working. The results can also be sorted by date. Every

entry in the table is clickable and will load that release in the two main windows at the top of the page. The site uses list.js to handle the list creation as well as the search/sort and filtering options.

The screenshot shows a search interface titled "Search Options". On the left, there's a "Name Search" input field with "e.g. Cassiopeia A" placeholder text, a "Surprise Me!" button, and a "more options" link. Below it is a "Filtered Search" section with categories: Stars (checked), Nebulas (unchecked), and Galaxies & Galaxy Clusters (unchecked). On the right, under "Search Results [select one to load above]", there's a "Sort by date" button and a note that the current count is 525. Three results are listed: 1) N132D, described as the remnant of an exploded star in the Large Magellanic Cloud, released on 1999-09-01. 2) Crab Nebula, described as a supernova remnant and pulsar in the constellation Taurus, released on 1999-09-28. 3) PKS 0637-72, described as a luminous quasar in the constellation Mensa, released on 1999-08-26. Each result has a small thumbnail image.

Figure 3: The expanded search options interface.

Future Refinements

In addition to cleaning up and beautifying the user interface, there are a few challenging problems that were outside the scope of the project given time constraints. I hope to continue working on this and have it deployed on Chandra's public website.

[1] Making better use of the WWT interface

I am completely satisfied with the usability of the interface as is now but there are a lot of options to wade through in the API. I intend to take some time to dig deeper and see what other potentially useful features could be added to this part of the site. For example, I know it is possible to actually load a copy of the press image directly into the WWT window as an overlay with transparency. This would be ideal as the user could then really see exactly how the image relates to its surroundings.

[2] Filtering options

The current limited filtering set is great as a demonstration of what is possible. I would like to take it further by refining the category definitions and also providing

multi-dimensional filtering. I attempted to implement both category and distance based filtering but was not able to get the filtering working correctly in time for the project submission. This is definitely on my to-do list to make the project viable for the general public.

[3] list.js

list.js was a great way to get started with implementing fast and clean list generation and searching from of a fairly large dataset. However, it does have its limitations and I would prefer to look into other options or possibly code my own list handler from scratch.

[4] \$(window).load AND \$(document).ready

Due to the large amount of data loaded into the table, the page does take a while to load and I found that I had to implement both window.load and document.ready of to get it to behave properly. Specifically, the first random release will not load properly if the page is still stuck reading/loading JSON data. My workaround was to put the main page creation functions in document.ready and the JSON data handling in window.load. This way the main content loads without incident and the JSON stuff is handled in the background while the user is still looking that first image. This could definitely be streamlined and cleaned up.