Adam Filiz
Professor Pang
CMPS 161: Intro to Data Visualization
20 March, 2019

Exo: A Visualization of Confirmed Exoplanets

Abstract

My final project, Exo, is a web-based program that visualizes all exoplanets (planets outside the solar system) that have been confirmed by NASA and that are able to be mapped out in 3D. Exo obtains exoplanet data using the NASA Exoplanet Archive API, and translates exoplanets' celestial coordinates to cartesian coordinates. The goal of Exo is to make it easy for a user to determine where we are (Earth) in relation to confirmed exoplanets.

Introduction/Motivation

During the fall quarter of 2018, I took the class BME 18: The Scientific Principles of Life with Dr. David Haussler. Towards the end of the quarter, Dr. Haussler had UCSC astrophysics professor Natalie Batalha give a lecture which she called "The Search for Life in the Universe."

Dr. Batalha introduced herself as a former NASA scientist who worked on the Kepler mission—a NASA mission with the objective of discovering exoplanets similar to Earth. Throughout her lecture, Dr. Batalha brought up how far astronomers have come in discovering exoplanets and information about those exoplanets. The Kepler mission, which was conducted from 2009 to 2018, discovered and confirmed 2,338 exoplanets alone (out of the 3,925 exoplanets that NASA has confirmed).

She moved on, saying that the exoplanets that NASA has discovered vary greatly in properties. Exoplanets orbiting dead stars, exoplanets with oceans of lava, exoplanets with two suns—these are all real types of exoplanets that exist beyond our solar system.

I asked her after that lecture whether NASA publicly posted their findings regarding exoplanets. She told me about the NASA Exoplanet Archive[1], where they store and post all of their data involving exoplanets (both confirmed and unconfirmed). For my final project for CMPS 161, I wanted to create a program that could visualize that data and allow any user to get a gist of where we are in the scheme of the exoplanets we've discovered. I also wanted to let the user interact with the data by being able to compare exoplanets' properties to one another.

Thus, my final project is an exoplanet visualization, which positions exoplanets based on their celestial coordinate values, and visualizes their sizes and effective temperature values. On top of this, Exo allows the user to interact with the

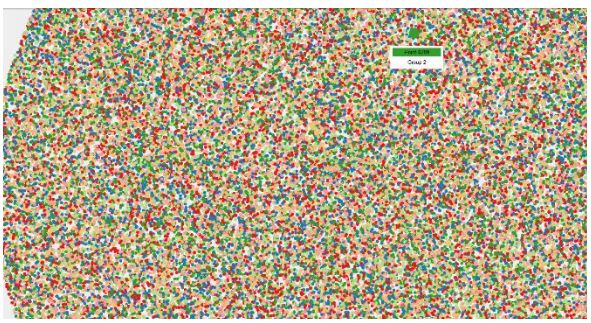
[1] NASA. *Confirmed Planets*, CalTech, exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=planets.

data of the exoplanets, as the user can click on exoplanets and see their data, and also select multiple exoplanets and get a comparison of the data properties of those exoplanets.

Technical Detail

The original idea for the program was to display exoplanets in a 2D space. I originally chose to use a 2D space because of my prioritization of user friendliness. If all of the planets are on a 2D scale, the user does not have to think about the positioning of the planets as much, and can focus primarily on the data itself.

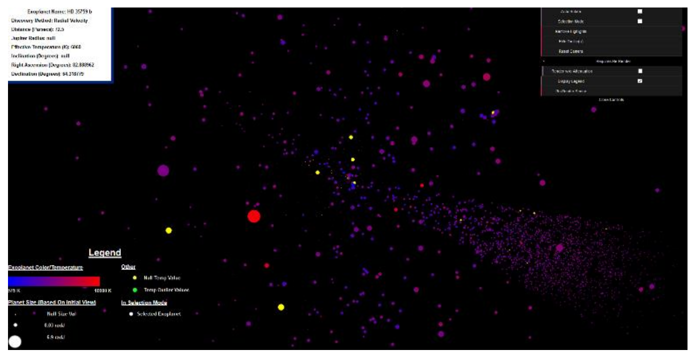
I started on this 2D implementation using a tutorial[2] on how to visualize points in a 2D space using three.js. While this 2D visualization of the exoplanet data was never fully implemented, the following image shows the result of the tutorial.



2D Points Representing Data

However, as one can see from the image above, it is quite difficult for the user to get a gist of where Earth is amongst the many points on the screen. Furthermore, it is slightly more difficult to deal with the cluttering of points on a 2D plane than on a 3D plane, where points have a third axis to be plotted on (thus less cluttering).

After talking to the professor about this development, I decided to go with a 3D implementation instead. I still used the aforementioned tutorial to learn how to plot points in three.js and how to create tooltips (pop-up window panes with relevant info/data). While the user may take more time analyzing the data and data points this way, the user can get much more context for the visualization in a 3D space. More on this topic will be discussed in the "results" section of this paper.



Exo and its 3D Visualization

The first step of developing Exo was to calculate the position of each exoplanet. After researching how to map celestial coordinates to Cartesian coordinates[3], I used the following algorithm, which takes an exoplanet's distance from Earth, their declination, and their right ascension values. Note that declination and right ascension values are provided in decimal degrees, while distances are in parsecs.

---

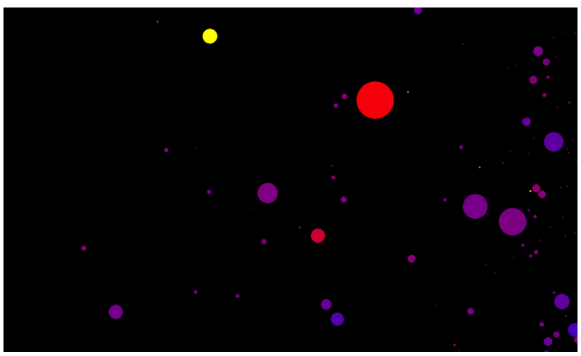
[2] Custer, Grant. "Using Three.js for 2D Data Visualization." *Observable*, Observable, 4 Aug. 2018, observablehq.com/@grantcuster/using-three-js-for-2d-data-visualization.

[3] Scalise, Greg. *SF: Putting Your Stars in Their Places*, 2003, fmwriters.com/Visionback/Issue14/wbputtingstars.htm.

rVect = cos(declination)

*x = rVect * cos(right ascension)*
*y = rVect * sin(right ascension)*
*z = distance * sin(declination)*

This formula gives (x, y, z) coordinates in a Cartesian space, with the scale of the space being in parsecs. However, it is impossible to map an exoplanet without a distance, declination, and right ascension value. Out of the 3,925 exoplanets in the confirmed exoplanet database provided by the NASA Exoplanet Archive, 5 exoplanets do not have all of these values. Thus, Exo visualizes all *mappable* exoplanets, and ignores those which are un-mappable.



Exoplanets of Dif. Size/Temp

After visualizing the positions of the mappable exoplanets, the next task was to visualize the exoplanets' size and temperature values. Size was visualized by scaling the size of each point (with one point representing one exoplanet) according to the Jupiter radius value of each exoplanet. Using these Jupiter radius values, for each exoplanet, the scale of the exoplanet is determining by interpolating the Jupiter radius value between the largest and smallest radius values found. It is important to note that one Jupiter radius unit is equivalent to the radius of Jupiter, or 69,911 kilometers, which is roughly 10.97 times that of Earth's radius. [4]

However, not all confirmed exoplanets in the confirmed exoplanet dataset have Jupiter radius values. With the goal of Exo being to visualize as many exoplanets as possible, it would be inappropriate to simply leave these exoplanets out of the visualization. Thus, exoplanets without Jupiter radius values are given size values that are much smaller than the size values of the smallest exoplanets. This allows the user to still see the color and position of an exoplanet while also see that there is no data on the size of the exoplanet.

The visualization of an exoplanet's temperature, or rather an exoplanet's effective temperature, was handled in a similar fashion. It is important to clarify that effective temperature is that of an exoplanet's star. It is different from actual temperature, and is usually much lower than the actual temperature. For the reader's knowledge, the definition of effective temperature is: the surface temperature of a star, calculated by assuming a perfect black body (hypothetical radiating mass) radiating the same amount of energy as the star.

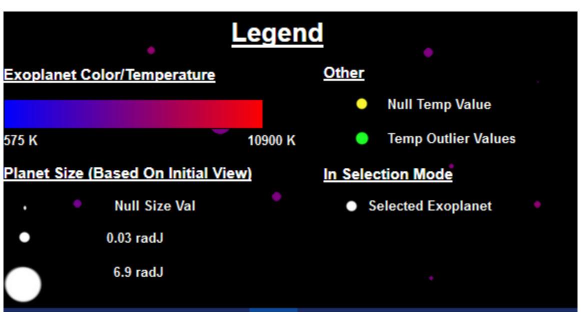
Exo visualizes an exoplanet's effective temperature using Andrew Noske's method of interpolating RGB values between two colors[5]. Each exoplanet is mapped to a color ranging from blue to red––blue being the coldest temperature, and red being the warmest. One problem that I ran into with this method of generating colors was that a subset of exoplanets had

[4] Redd, Nola Taylor. "How Big Is Jupiter?" *Space.com*, Space Created with Sketch. Space, 20 July 2018, www.space.com/18392-how-big-is-jupiter.html.

[5] Noske, Andrew. "Code - Heatmaps and Color Gradients." *Code - Heatmaps and Color Gradients - NoskeWiki*, 6 Feb. 2019, www.andrewnoske.com/wiki/Code_-_heatmaps_and_color_gradients.

abnormally high effective temperature values compared to the rest of the dataset. When I used Noske's interpolation method, all of the exoplanets ended up being blue (cold) when there should have been a variation of colors instead. To correct for this, I simply colored the outlier exoplanets a different color altogether (coloring them green instead). With these outliers outside of the max and min effective temperature range, the visualization had a nice range of colors from blue to purple to red.

Like sizes, not all exoplanets in the confirmed exoplanet dataset had effective temperature values. To correct for this as well, I colored these exoplanets yellow, which is outside of the color range of blue to red. Thus, the user is able to see the normally colored exoplanets (those with effective temperature values), the green outlier exoplanets (those with abnormally high effective temperature values), and the yellow exoplanets with no effective temperature data. Of course, a toggle-able legend is displayed so that the user can see what each size and each color means in terms of the data.



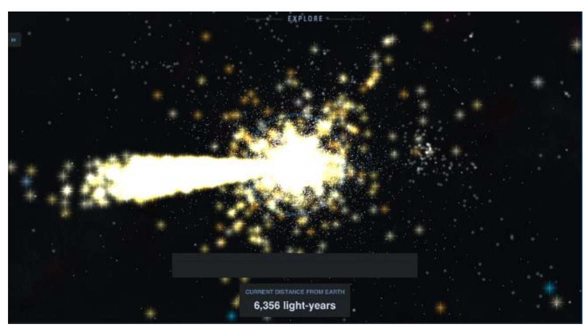
Exo's Legend (w/ Exoplanets in Background)

The final step of developing Exo was allowing the user to select an exoplanet and see its other data, as well as to compare the data between exoplanets. To do this, when the user clicks on the program, three.js employs a raycaster to detect which points/exoplanets are clicked on. In the case of overlapping points, the point closest to the mouse cursor is selected. Once a point is selected, a tooltip appears showing the data of the exoplanet associated with that point. This data is obtained from the JSON file that is retrieved using NASA's Exoplanet Archive API.

When the user uses the "selection mode" feature, the user is allowed to select more than one point. When two points are selected, two tooltips (one for each exoplanet) are displayed side by side so that the user can compare the exoplanets' data. In the case that more than 2 exoplanets are selected, the average of each numerical data element (Jupiter radius, inclination, etc) between the selected exoplanets is displayed in one tooltip. This allows the user to get an understanding of the typical/average properties of a subset of exoplanets.

Related Works

Eyes on Exoplanets – NASA[6]



Eyes on Exoplanets Visualization

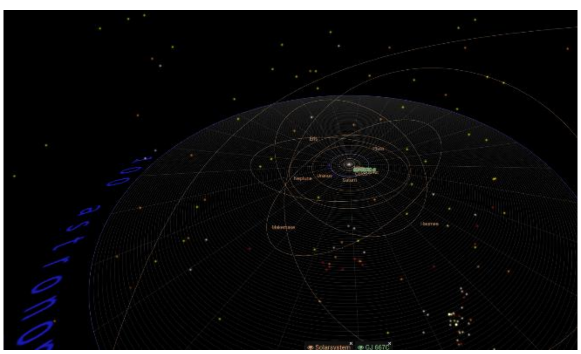
This application-based visualization is similar to Exo in that it visualizes all exoplanets in a visual space. The user can move through space just as in Exo and click

[6] "NASA's Eyes: Eyes on Exoplanets." *NASA*, NASA, 14 Dec. 2018, eyes.nasa.gov/eyes-on-exoplanets.html.

on individual exoplanets for more information. *Eyes on Exoplanets* also allows the user to take a "tour" of exoplanets, as it can zoom in on a particular exoplanet and see its orbit around its star.

However, the visualization itself does not visualize planet size and effective temperature from a distance very well—for these properties, only the color of each exoplanets' stars are shown, with size not being immediately discernable unless one zooms in all the way. The application itself is also quite demanding graphically, causing the framerate to slow down when zooming on slower computers. Exo's goal is to make it as easy as possible for users to get an overview of exoplanets' positions and properties, while also allowing the user to compare data between exoplanets. *Eyes on Exoplanets* accomplishes the goal of showing exoplanet positions, but strives for other goals different from those of Exo.

ExoPlanetSystems[7]



ExoPlanetSystems

ExoPlanetSystems is similar to Exo in that it is a web-based visualization. Like *Eyes on Exoplanets*, it visualizes both exoplanets and the stars that those exoplanets orbit around. However, with *ExoPlanetSystems*, it takes much more time to pan/zoom around the exoplanets using the mouse. It uses quite a few labels for each exoplanet and exoplanet(s)'s star, including labels for inclination, system planet, etc. *ExoPlanetSystems* does have a exoplanet comparison feature, but only compares exoplanets' sizes in shape form (larger circles = larger exoplanets and vice versa).

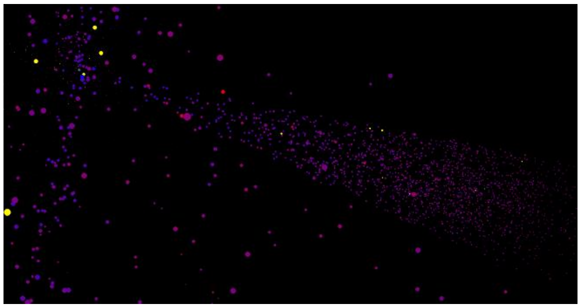
Overall, *ExoPlanetSystems* aims to provide quite a bit of data on each exoplanet, but makes the user explore the celestial space slower (perhaps to allow the user to appreciate the details involved in the program). It also only visualizes 1,795 exoplanets compared to Exo's 3,920 exoplanets. Exo aims to make it easy for a user to get a glimpse of the scheme of all exoplanets that have been confirmed by NASA in a short amount of time, while allowing the user to compare exoplanets' data if they want more details. Since *ExoPlanetSystems* focuses more on the details, it aims to solve a different problem than Exo's.

Results

There are a few observations to make about the visualization itself. For one, there is a notable cone shape of exoplanets, with the tip of the clone positioned closer to where Earth is at the center of the visualization. If one clicks on a decent number of these exoplanets in this cone, they would notice that their names all start with "Kepler." Indeed, almost all of the exoplanets in this cone were discovered by the Kepler mission's space telescope. Based on the way the Kepler telescope was always facing one direction during its 9 year mission while it was orbiting the sun, this

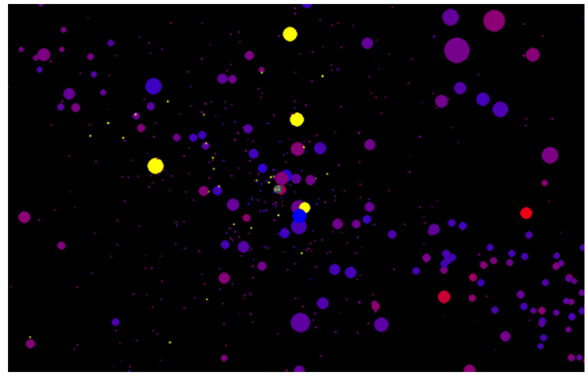
[7] Krüger, Tommy. "Loading Galaxy Data ..." *ExoPlanetSystems - A Visualization of Exoplanet Systems*, Mar. 2014, exoplanets.tommykrueger.com/.

cone had to be produced because of the telescope's orientation. Otherwise, the exoplanets in this cone would be scattered across the 3D space and not congregated in this cone shape.



Kepler "Cone" Shape

Another observation to make is that many of the exoplanets positioned closer to Earth don't have Jupiter radius values, and that many of the exoplanets without effective temperature values are also closer to Earth. I would argue that these exoplanets that are positioned closer to Earth were discovered with less sophisticated technology, and thus astronomers were not able to get exact Jupiter radius/effective temperature values for these planets. Unfortunately, I cannot support this guess with facts, as the NASA Exoplanet Archive does not include exoplanet discovery dates as part of the confirmed exoplanet dataset.



Center of the Visualization (Close to Earth)

Finally, if one looks very carefully at the exoplanets, one can see other smaller cones of exoplanets (one can also interpret these as lines). While the correlation between the exoplanets in these particular cones isn't as concrete as those found during the Kepler mission, it is possible to tell from Exo's visualization that other searches for exoplanets resulted in finding exoplanets in cone-shaped patterns (in terms of positioning).

Conclusion

In terms of achieving the goals of the visualization, I believe Exo did just so. Users can zoom out in seconds to see the scale of the program and to see the extent to which our search for exoplanets goes. On top of that, users have the option of going beyond just seeing the data visualized, and are able to interact with it as well by comparing specific exoplanets' data. Since Exo pulls the confirmed exoplanet dataset straight from NASA, Exo will always be up to date on its data. While it is a distant wish, I hope that users (me included) can use Exo in the future to learn more about the exoplanets we know of and where they are in the universe.