The Interactive Exoplanet Encyclopedia

Jaromir Camphuijsen

June 26, 2015

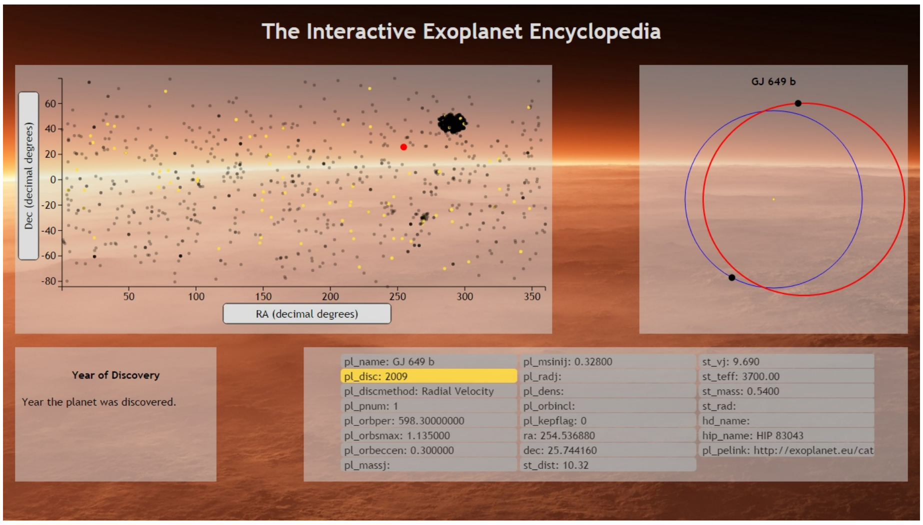

Figure 1: The Interactive Exoplanet Encyclopedia visualization

Abstract

Contents

1 Introduction **3**

2 Goals **3**

3 Data **3**

3.1 Source 3

3.2 Cleaning 4

4 Abstraction **4**

5 The design process **4**

5.1 First stage 4

5.2 Second stage 5

6 The code **5**

7 validate **6**

1 Introduction

Since the beginning of history the idea of life in the unknown has been attractive and frightening to us at the same time. Whether it is our own species on another continent, some weird jellyfish in the depths of the ocean or extraterrestrial life. This last category has not been found yet except in our own imagination and art. However the past few decades we have been able to explore the universe more and more and at this point about two thousand planets orbiting some star other than our own sun have been discovered. We call them exoplanets. Not all of them are able to accommodate life as we know it, however some may and this is one of the reasons why research concerning exoplanets is so exciting. In 2009 NASA launched the Kepler mission with the sole purpose of finding as many exoplanets as possible. Currently NASA maintains a database with 1854 confirmed exoplanets and more than 4500 candidates observed by the Kepler mission (and still counting). Having an astrophysics background and having spent a bunch of cold nights in the Anton Pannekoek Observatory in Amsterdam to observe an exoplanet (τ Bootes b) myself, I thought it would be interesting to make this great amount of data interactively visual.

2 Goals

The idea of this visualization did not originate from a scientific question but more a problem with the accessibility of the exoplanet data. There are some archives to be found on the Internet of which the NASA archive is only one. However NASA only offers a single planet light curve viewer and the available multi planet visualizations on the Internet are difficult to use [1] or cover only a small amount of data dimensions. The goal of the Interactive Exoplanet Visualization (IEE) is to give users the possibility to view all currently known exoplanets in a pleasant environment and discover correlations or connections between their different properties. A website like exoplanets.org [2] has this interactive plotting and single planet information, however the single planet information opens in a new window. It would be nice to have a visualization which is more attractive to the public. This means less, or better explained information, all functionality on one screen and an attractive look with smooth animations. The interface should be intuitive, clear and easy to use.

3 Data

3.1 Source

All used data was obtained from the NASA exoplanet database [3]. This choice was made because of the great amount of data present in the archive in combination with the accessibility. Even though some other archives like [4] have more consistent data, the NASA archive offers an API to obtain the data via url queries. Though not used in the current version of this visualization, this enables the visualization to always stay up to date with the archive. This in contrast to the case the data should be downloaded every time an update was made to the archive. In the current version the data is downloaded in csv format and loaded into the visualization. Since there was only a proper description of the data dimensions in a table on the NASA page, as part of the documentation, a python based DOM scraper was used to grab these values. This data was also saved as csv and imported into the visualization to be used alongside the actual data.

3.2 Cleaning

The raw data did not need a lot of cleanup, only the deletion of a descriptive header. When one of the two values of a data point in a two dimensional plot would be missing, the data point would not be shown. Another solution to missing values was considered, where the data point would take still be in the plot when one of the dimensions was missing. The value of the present dimension would be shown by drawing the point just below its axis to standout from the rest. Doing this to all data points seemed to clutter the axis, but the principle was applied to the single selected and highlighted data point. The small dataset with data dimensions and their descriptions needed some cleaning up as there were dagger signs appended to some of the labels and the descriptions were full of spaces and newline characters. These values were cleaned up in the scraper.

4 Abstraction

As intuitive interactivity was one of the main goals, this was also the first step in the process of translating the two main tasks (compare and explore), to an easy to use application. It led to the multiple windows, each showing the data on their own way. One window would show all data to give the possibility to compare and to serve as the main controller. Single data points should be selected to change other parts of the visualization to show detailed information of the selected object. This detailed information should be explained somehow in the same screen and being short in screen space this could mean another layer of interactivity. Together with the possibility to change the main view to different settings, this would satisfy the explore task. Much later in the process, the importance of feedback of the last interactive window to the first one was discovered, strengthening the comparing task of the visualization.

5 The design process

5.1 First stage

The general idea of the design was clear from the start. The outline of the present day visualization can already be seen in the first sketch: fig 2. The data is shown as a scatter plot where respectively x and y position are the data value encoding. The main interaction is the basis for the rest of the visualization: clicking on a data point in the main window, also called Multi Planet View or mpv, will select it and trigger the other window to show detailed information and a small model of the selected planetary system. In this model, called the Single Planet View or spv, the size, shape and velocity of the orbit are compared to the earth orbit and a properly scaled companion star is drawn in the middle. Color encodes the type of orbit: red for exoplanet and blue for earth. Two planets (not proportional) orbit the sun with proportional orbital periods. Other info

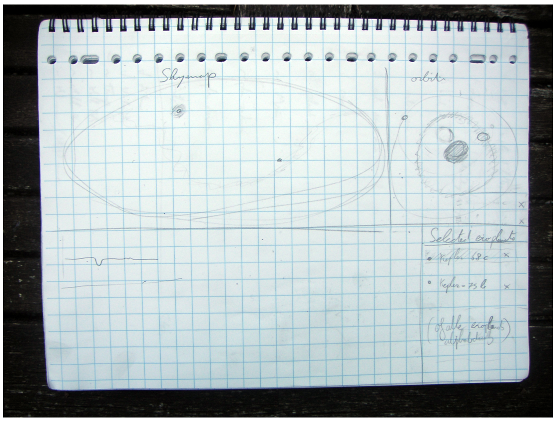

Figure 2: First sketch of the visualization

about the selected planet is shown on the bottom of the page in the info bar. This setup was also seen on the second sketch (fig 3)in which the background was set to black. The design with black background and light data is used a lot in astronomic visualizations to resemble the night sky look.

5.2 Second stage

In the second step of the design, drop down menus to change the scatter plot were added. They appear on clicking on the axis title and contain a selection of available dimensions. Selecting a dimension in the menu changes the corresponding axis and transitions the data to its new location. The single planet information bar shows entries with data keys and values which are clickable. This action will show the description of the chosen dimension in the small description screen. All selection actions are color encoded: hovering a clickable item turns the item orange, selecting a single planet turns this planet red and bigger in the scatterplot and selecting a dimension in the info bar turns this selection yellow and all the data points in the scatter plot with the same value for this dimension as well. The classic astrophysics black with white design was abandoned and a picture of the surface of mars was put in the background as it is a soft color and the picture is not disturbing the data too much. The background adds a pleasant view and feeling to the visualization. The final design as seen in fig 1 uses

6 The code

All of the functionality was build using the javascript D3 library [5]. This library can be used to bind data points to elements in the DOM. It enables you to manipulate those DOM elements simultaneously while manipulating the data and therefore keep track of the data points. For example, when the data is loaded it consists of a list of javascript objects. Using the d3 databinding with a keyfunction to assign a key to each object/element pair:

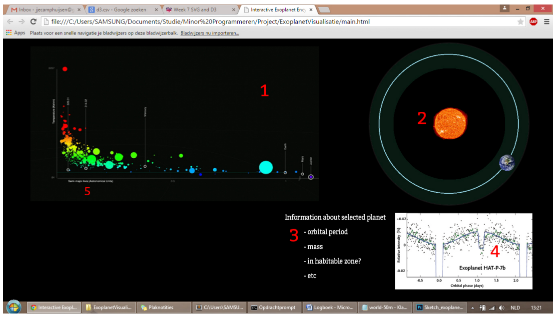

Figure 3: Second sketch of the visualization

```
// data-join with pl_name as keyfunction
    var points = plotContainer.selectAll(".dot")
        .data(selData, function(p) {return p.pl_name});

// enter new data
points.enter()
        .insert("circle")
```

All menus are constructed in a similar way using small hand picked subsets of all the available dimensions. These subsets are bound to svg ¡g¿ elements and can be translated and manipulated easily. The single planet view is also built using d3 databinding which makes it possible to easily expand the functionality to show more than one exoplanet in the future. First a new dataset (called "model") is constructed consisting of an exoplanet object corresponding to the selected object in the scatterplot and an earth object. These two objects contain all the information to draw and animate their own orbits. The d3.timer function starts on loading the page and never stops. The only thing that changes on selecting another planet in the mpv is the binding of the svg elements to the data. The descriptions of the data dimensions were not included in the dataset but could be found on a separate documentation page of the NASA website. The descriptions were scraped and cleaned up using a python script and the pattern.web library for searching the DOM.

7 validate

When looking at the rough structure of the visualization, the final product as shown on the title page resembles the initial sketches quite well. There is a Multiplanet view and a Single Planet View. However some functionalities, like a skymap with a different coordinate mapping and showing the raw observational data like transit light curves are not implemented. In exchange, some new functionality has been added. The selection of a certain dimension with a description and highlighting of all planets with the same value. This is a very important addition to the original concept and can be very useful when used on normative dimensions like the discovery method. A scientific flaw in the visualization is the orbit animation. The animated planets follow simple harmonic movements instead of Kepler orbits. This is merely a time issue and can be resolved later. There is one functionality which is absent and actually should not be. This is the possibility to zoom in on the scatter plot. The upgrade scatter function can be called with a different set of x and y scales so it is potentially zoomable, however again due to lack of time, the function was not fully implemented.

The visualization was shown to different people with and without scientific background. Most of them found the application interesting and fun to explore. One person remarked about the absent zoom function. Overall the feedback was positive which encourages to further develop the application. Upon orientation at the start of the project I came upon the visualization by Nadieh Bremer [6] and thought her visualization was really interesting. It would be interesting to see if the two visualizations could be combined into one.

References

[1] T. Hands, "exovis," 2014.

[2] C. P. S. consortium, "exoplanets.org," 2015.

[3] NASA, "Nasa exoplanet archive," 2015. [Online; accessed 11-June-2015].

[4] T. exoplanet team, "exoplanets.eu," 2015.

[5] M. Bostock, "D3 javascript."

[6] N. Bremer, "Exoplanets."