ANIVIS

The Missing Anime Categorizer

Table of Contents:

Proposal

Basic Info

Our Crew

Hanzhou Shi
hanzhou87@gmail.com
20336980

Juan Yescas
jcyesc@gmail.com
XXXXXXXX

Kannan Chinnasamy
kchinnasamy@dons.usfca.edu
20324509

Github Repository

https://github.com/hanjoes/vitches

Project page

http://hanjoes.github.io/anivis/

Background and Motivation

What are the feelings when you have just finished you beloved anime series? You will have a lot of feelings like pleased, touched, sad, etc. But if the anime is really good, 90% of the time the only thing you know is that you **want more.**

Alas, although we have those great anime databases like MyAnimeList, but it's just so much mental burden to get use to the user interface and to filter out the information we don't want from the website.

Wouldn't it be great if someone can collect all the **similar animes** and **visualize** them such that we can easily find the related ones by only a few clicks? And we will present **ANIVIS,** the missing anime categorizer**.**

Project Objectives

We have two main objectives:

1. The user of this visualization will be empowered to find the animes interests him/her through the easy to understand, easy to use interfaces.
2. Not only the users will know the detailed information for the current anime he/she has chosen, the user will also be able to know animes that have correlation to the current anime showing in various aspects.
3. Users can also explore the visualization to find anything that they will like via some attributes of the animes, for example, genre.

By working on this project, we, as developers, will gain several benefits:

1. We will be able to test our muscles in **d3.js**, in a project that's more close to production quality.
2. We will get involved in the whole loop of data acquiring, data cleanse, data visualization which should introduces a lot more **complexities** than other simple academic projects.
3. We will practice **team work** both in terms of human interaction and in terms of utilizing those tools (like **git**) which means a lot if eventually we want to work in the industry.

Data

We are getting data from **Anime News Network**.
It's a similar anime database that has reasonably comprehensive anime data.

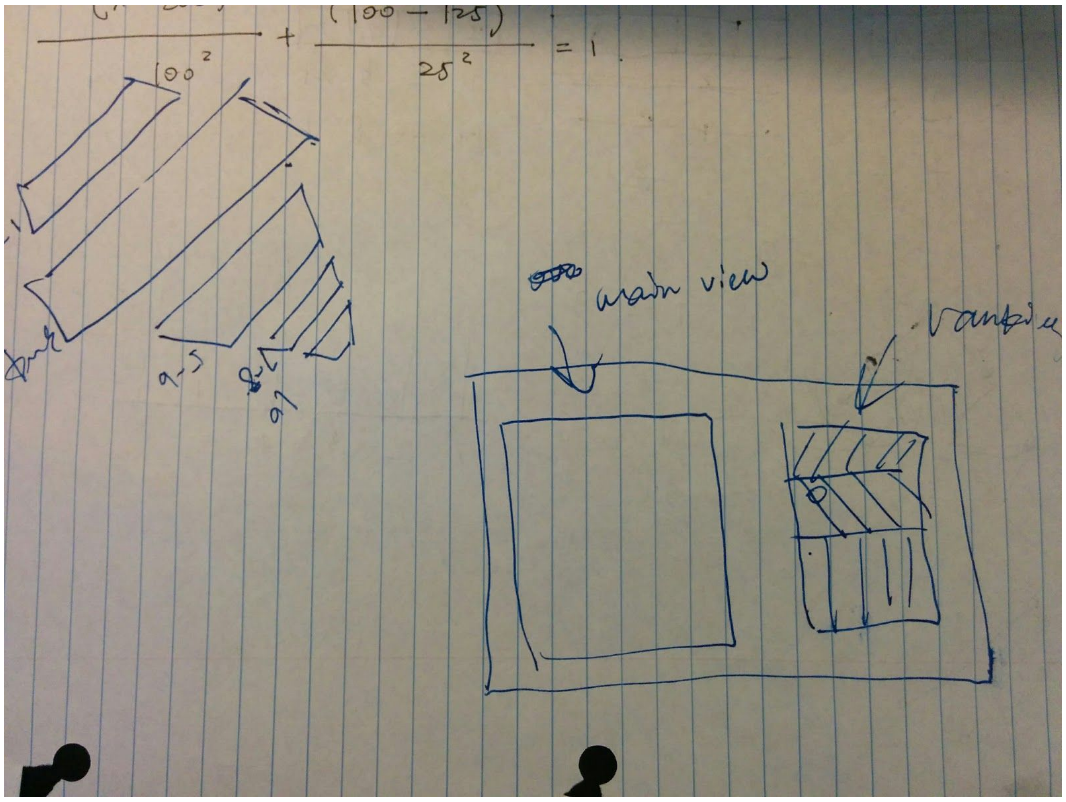
Data Processing

The data for this project is a list of all anime movies and TV series with title, author, genre etc. in a JSON. We used Anime News Network to extract all those information. The data obtained from the site are in XML format, but we intended to use JSON data, so it needs some formating.

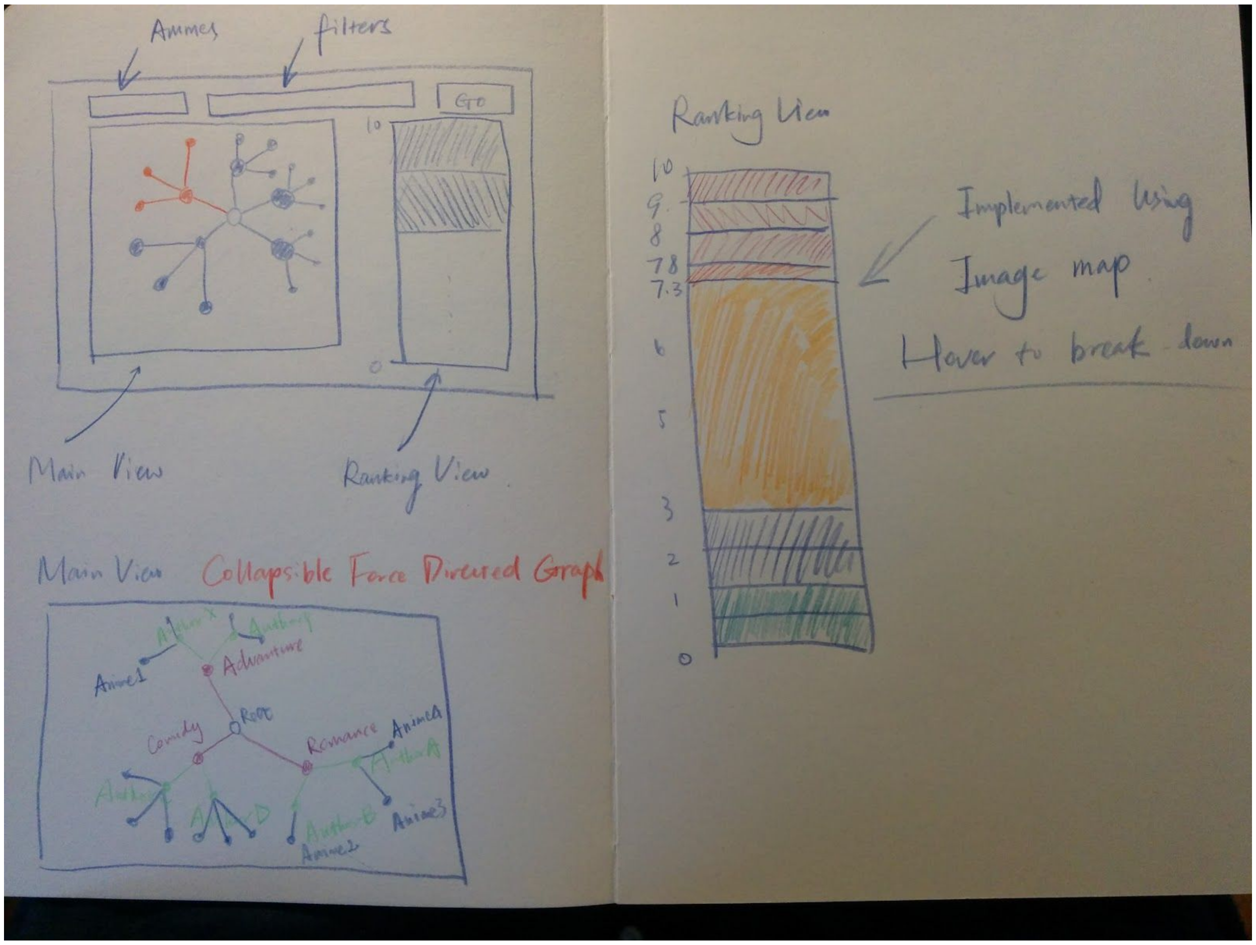
We have a python scripts that get the data from the "Anime News Network" API's and save it as an XML file. This data is transformed into the required format by using another python script which reads the XML file and save it as a separate file.

Visualization Design

Sketch



Finalized prototypes

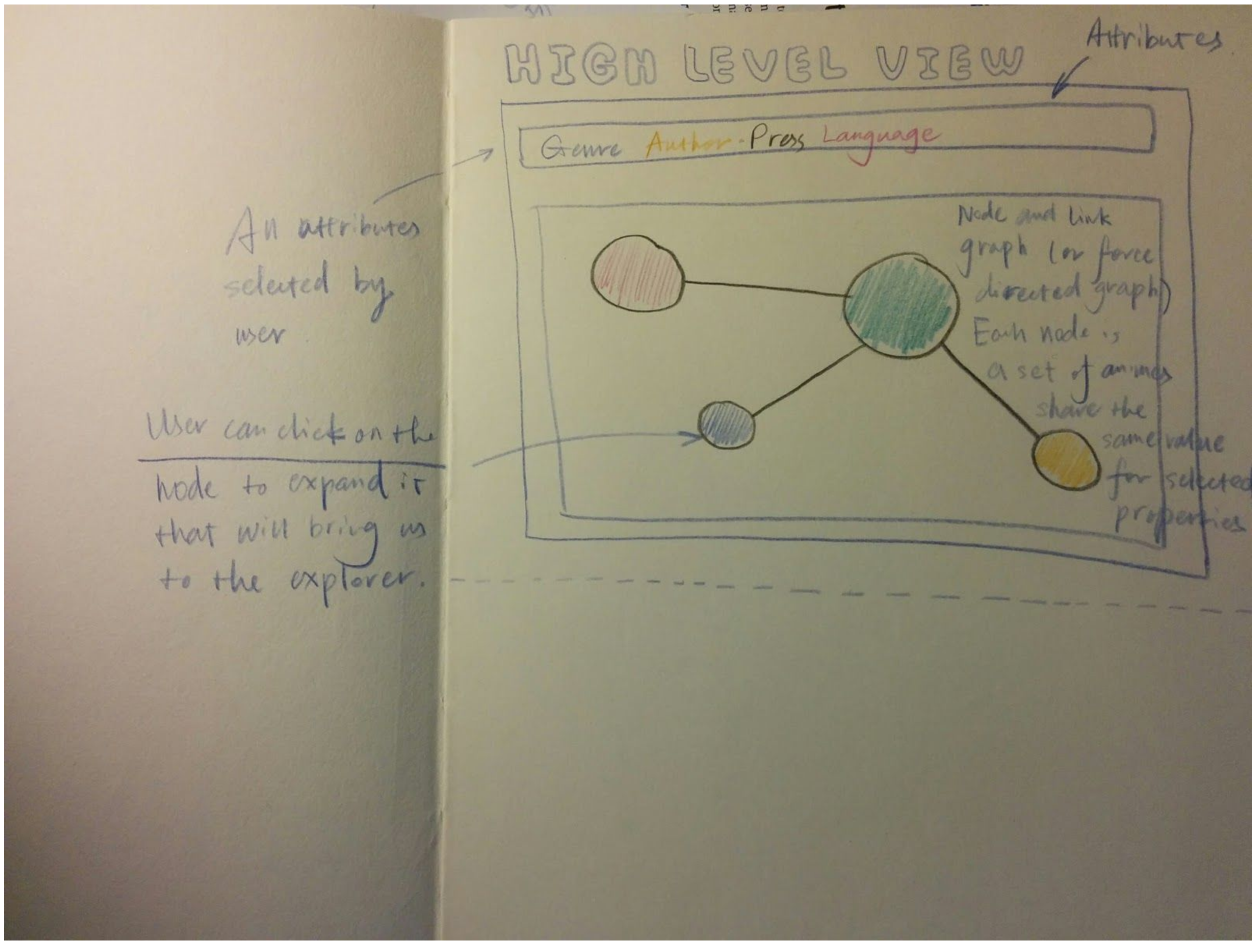


This is the appearance of our visualization. The whole scene is divided into two main parts. The main view (overview view) and the ranking view. The main view uses collapsible force directed graph to demonstrate the tree structure selected by user. And the ranking view uses a breakable gauge that uses a similar “break-down” technique in d3’s Image Map[1] example to show ranking information.

The sketch on the right shows the ranking view. As we can see, this is a broken down ranking view which has several segments in it. Each segment represent a range of ratings. This gives us an overview of how is the rating distribution of the data inside a specific range. And user could hover to see containing animes and click to break down the segment. As shown in the example, the biggest portion (denoted in orange) contains animes rated between 3 to 7.3.

[1] http://bl.ocks.org/nswamy14/df13d67b6efeb19eb640

Also the rankings shown is related to the selected subtree in the main view.


HIGH LEVEL VIEW
Attributes
Genre Author Press Language
All attributes selected by user
Node and Link graph (or force directed graph) Each node is a set of animes share the same value for selected properties
User can click on the node to expand it that will bring us to the explorer.

This is the overview view, or our main view. We are going to implement this part of our visualization using collapsible force layout. Each non-leaf node represents an aggregation of animes, and user can click on the nodes to collapse it down into sub-nodes. The level of the tree is determined by the attributes selected by the user in the filter. Each leaf node in this view represents a single anime. We can also collapse a whole subtree directly by typing the anime's name to locate that anime. In this case, the collapsed subtree can easily reveal all related animes in terms of the selected attributes.

Must-Have Features

List the features without which you would consider your project to be a failure.

1. A collapsible force directed graph to demonstrate the categorized animes.
2. A ranking view gives not only detailed info about anime ranking but also an overview of the ranking distribution.

3. User interface that allows users to query for a single anime.
4. User interface that allows users to filter using different attributes (genre, author, etc.).

Optional Features

1. Navigation history.
2. Animation.

Project Schedule

Week 1 (Mar. 20 th - Mar. 26 th)	Data Processing
Week 2 (Mar. 27 th - Apr. 2 nd)	Finalize the Visualization
Week 3 (Apr. 3 rd - Apr. 9 th)	Basic visualization, Main view
Week 4 (Apr. 10 th - Apr. 16 th)	Alpha Release.
Week 6 (Apr. 17 th - Apr. 23 rd)	Ranking view.
Week 7 (Apr. 24 th - Apr. 30 th)	Beta Release
Week 8 (May 1 st - May. 7 th)	Animation and more tweaks, text query.

Process Scribe

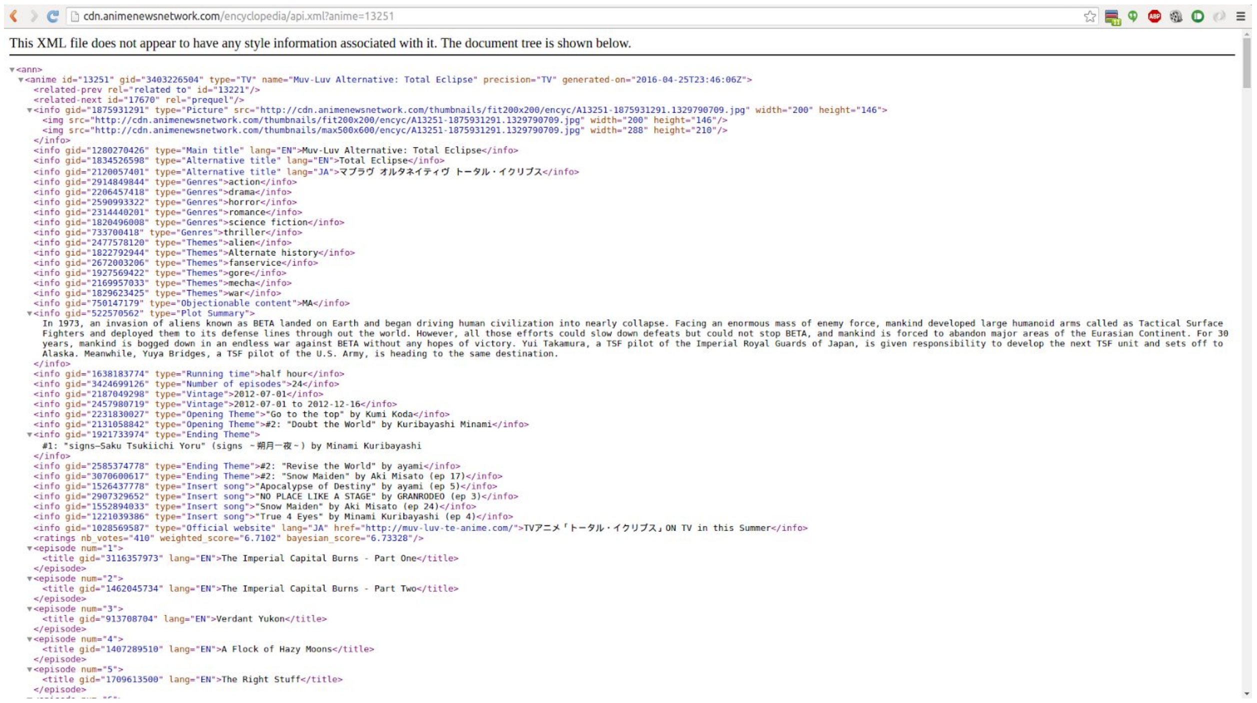
Outline

The ANIVIS project consists of the following phases.

Data Gathering

The data for the project is solely from AnimeNewsNetwork. It provide an API which can be queried to get the XML data for the given Anime. We use a python script that queries the API get the details of all the Anime's. Below is a screenshot source data. Here we wanted to get the information about anime with ID "13251". Now python script would query the AnimeNewsNetwork "*http://cdn.animenewsnetwork.com/encyclopedia/api.xml?anime=13251*", this will return an xml data which is saved locally and used as an input for the data transformation script.

Data Processing

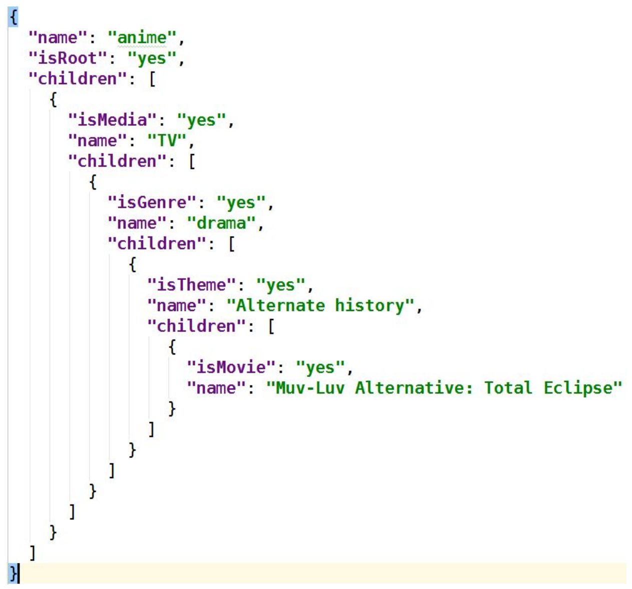

cdn.animenewsnetwork.com/encyclopedia/api.xml?anime=13251

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<ann>
 ▼<anime id="13251" gid="3403226504" type="TV" name="Muv-Luv Alternative: Total Eclipse" precision="TV" generated-on="2016-04-25T23:46:06Z">
   <related-prev rel="related to" id="13221"/>
   <related-next id="17670" rel="prequel"/>
  ▼<info gid="1875931291" type="Picture" src="http://cdn.animenewsnetwork.com/thumbnails/fit200x200/encyc/A13251-1875931291.1329790709.jpg" width="200" height="146">
     <img src="http://cdn.animenewsnetwork.com/thumbnails/fit200x200/encyc/A13251-1875931291.1329790709.jpg" width="200" height="146"/>
     <img src="http://cdn.animenewsnetwork.com/thumbnails/max500x600/encyc/A13251-1875931291.1329790709.jpg" width="288" height="210"/>
   </info>
   <info gid="1280270426" type="Main title" lang="EN">Muv-Luv Alternative: Total Eclipse</info>
   <info gid="1834526598" type="Alternative title" lang="EN">Total Eclipse</info>
   <info gid="2120057401" type="Alternative title" lang="JA">マブラヴ オルタネイティヴ トータル・イクリプス</info>
   <info gid="2914849844" type="Genres">action</info>
   <info gid="2206457418" type="Genres">drama</info>
   <info gid="2590993322" type="Genres">horror</info>
   <info gid="2314440201" type="Genres">romance</info>
   <info gid="1820496008" type="Genres">science fiction</info>
   <info gid="733700418" type="Genres">thriller</info>
   <info gid="2477578120" type="Themes">alien</info>
   <info gid="1822792944" type="Themes">Alternate history</info>
   <info gid="2672003206" type="Themes">fanservice</info>
   <info gid="1927569422" type="Themes">gore</info>
   <info gid="2169957033" type="Themes">mecha</info>
   <info gid="1829623425" type="Themes">war</info>
   <info gid="750147179" type="Objectionable content">MA</info>
  ▼<info gid="522570562" type="Plot Summary">
     In 1973, an invasion of aliens known as BETA landed on Earth and began driving human civilization into nearly collapse. Facing an enormous mass of enemy force, mankind developed large humanoid arms called as Tactical Surface
     Fighters and deployed them to its defense lines through out the world. However, all those efforts could slow down defeats but could not stop BETA, and mankind is forced to abandon major areas of the Eurasian Continent. For 30
     years, mankind is bogged down in an endless war against BETA without any hopes of victory. Yui Takamura, a TSF pilot of the Imperial Royal Guards of Japan, is given responsibility to develop the next TSF unit and sets off to
     Alaska. Meanwhile, Yuya Bridges, a TSF pilot of the U.S. Army, is heading to the same destination.
   </info>
   <info gid="1638183774" type="Running time">half hour</info>
   <info gid="3424699126" type="Number of episodes">24</info>
   <info gid="2187049298" type="Vintage">2012-07-01</info>
   <info gid="2457980719" type="Vintage">2012-07-01 to 2012-12-16</info>
   <info gid="2231830027" type="Opening Theme">"Go to the top" by Kumi Koda</info>
   <info gid="2131058842" type="Opening Theme">#2: "Doubt the World" by Kuribayashi Minami</info>
  ▼<info gid="1921733974" type="Ending Theme">
     #1: "signs—Saku Tsukiichi Yoru" (signs ~朔月一夜~) by Minami Kuribayashi
   </info>
   <info gid="2585374778" type="Ending Theme">#2: "Revise the World" by ayami</info>
   <info gid="3070600617" type="Ending Theme">#2: "Snow Maiden" by Aki Misato (ep 17)</info>
   <info gid="1526437778" type="Insert song">"Apocalypse of Destiny" by ayami (ep 5)</info>
   <info gid="2907329652" type="Insert song">"NO PLACE LIKE A STAGE" by GRANRODEO (ep 3)</info>
   <info gid="1552894033" type="Insert song">"Snow Maiden" by Aki Misato (ep 24)</info>
   <info gid="1221039386" type="Insert song">"True 4 Eyes" by Minami Kuribayashi (ep 4)</info>
   <info gid="1028569587" type="Official website" lang="JA" href="http://muv-luv-te-anime.com/">TVアニメ「トータル・イクリプス」ON TV in this Summer</info>
   <ratings nb_votes="410" weighted_score="6.7102" bayesian_score="6.73328"/>
  ▼<episode num="1">
     <title gid="3116357973" lang="EN">The Imperial Capital Burns - Part One</title>
   </episode>
  ▼<episode num="2">
     <title gid="1462045734" lang="EN">The Imperial Capital Burns - Part Two</title>
   </episode>
  ▼<episode num="3">
     <title gid="913708704" lang="EN">Verdant Yukon</title>
   </episode>
  ▼<episode num="4">
     <title gid="1407289510" lang="EN">A Flock of Hazy Moons</title>
   </episode>
  ▼<episode num="5">
     <title gid="1709613500" lang="EN">The Right Stuff</title>
   </episode>
```

The data from the source is in XML format, which is not useful for the d3 visualization used in the project. A python script is used to transform the data into the required format for the all the visualizations used. Python script load the XML file and process just the required information like anime name, media, genre, theme and rating and generates an intermediate json file. The intermediate json file is processed and create the required folder structure and the respective anime.json, rating.json for the Forced layout and Ranking view. Below is the sample Json file.

Force Layout data

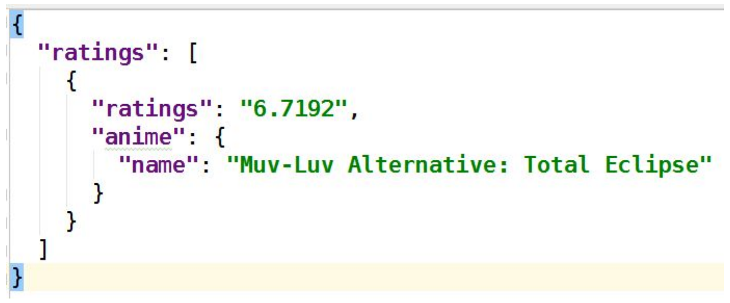


```
{
  "name": "anime",
  "isRoot": "yes",
  "children": [
    {
      "isMedia": "yes",
      "name": "TV",
      "children": [
        {
          "isGenre": "yes",
          "name": "drama",
          "children": [
            {
              "isTheme": "yes",
              "name": "Alternate history",
              "children": [
                {
                  "isMovie": "yes",
                  "name": "Muv-Luv Alternative: Total Eclipse"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

It uses tree structured json format where the leaf nodes are the anime that satisfice the user selected criteria and parent node forms the tree path of the criteria.

Query Url : *anime/TV/drama/Alternate history/anime.json"*

Ranking View Data



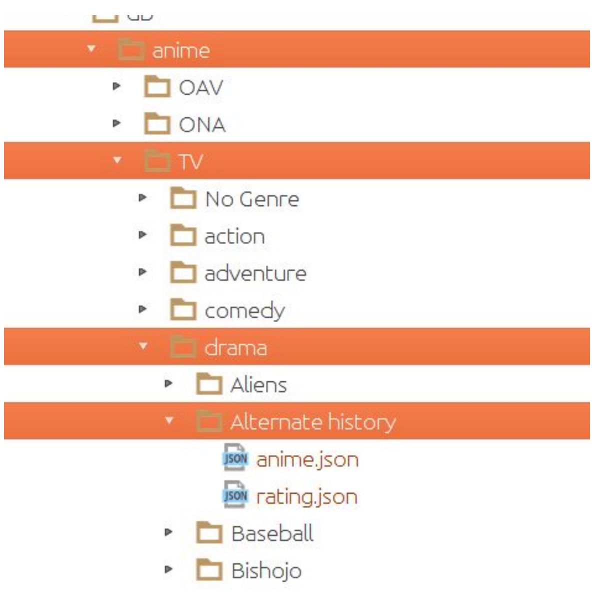
```
{
  "ratings": [
    {
      "ratings": "6.7192",
      "anime": {
        "name": "Muv-Luv Alternative: Total Eclipse"
      }
    }
  ]
}
```

It uses simple json array to save all the animes with name and ranking, that satisfies the given criteria, by the user.

QueryUrl: *"anime/TV/drama/Alternate history/rating.json"*

Build Database

The dataset used in ANIVIS is large, if tried to load the data into d3 in one shot makes the visualization sluggish. So we developed a Database structure using folder hierarchy. When the user changes values for filter criterias, the respective visualization would update the file that it load the data from. Below is a section of the database structure.



The Image show a data structure that is used for querying the data for the visualizations. The highlighted path will retrieve the data matching the filters used by the user.

In this case *"anime/TV/drama/Alternate history/anime.json"* for the Force Layout visualization and *"anime/TV/drama/Alternate history/rating.json"* for the Ranking visualization.

Visualization

The goal of this visualization is to display similar animes. And user could use different filters to narrow down the range of display. (There will also be a search box to quickly locate one anime if time allows). The vis will be organized into two main parts, the Ranking View and the Tree View.

For the **Ranking View**, there will be a breakable rectangle to show ranting distribution of animes. Allowed user interactions are hovering to show detail of a range, click to break down a range. And we use color interpolation to encode density of a specific range. This part will give users not only an overview of **how the ratings distribute** but **more detailed information** of what are the animes in the range.

For the Tree View, we decided to use a **Collapsible Force Layout**. It gives us a tree structure but at the meantime it's more flexible than a traditional collapsible tree. We can reorganize nodes at will to have some **interesting while useful visual effects**.

Challenges

Data Transformation

On of the most challenging part in ANIVIS is the data transformation. Since we used 2 different visualization it is hard to finalize on a single data format. So we had to do multiple iteration on the data and try out the best optimal. Structure.

Database

Since the data source was huge we need to come up with the different techniques for getting data dynamically into the visualization. The regular database requires lot of other feature like setting up the server and other stuff. So we create a folder structure with all possible index that are to be used and query them using the folder path.

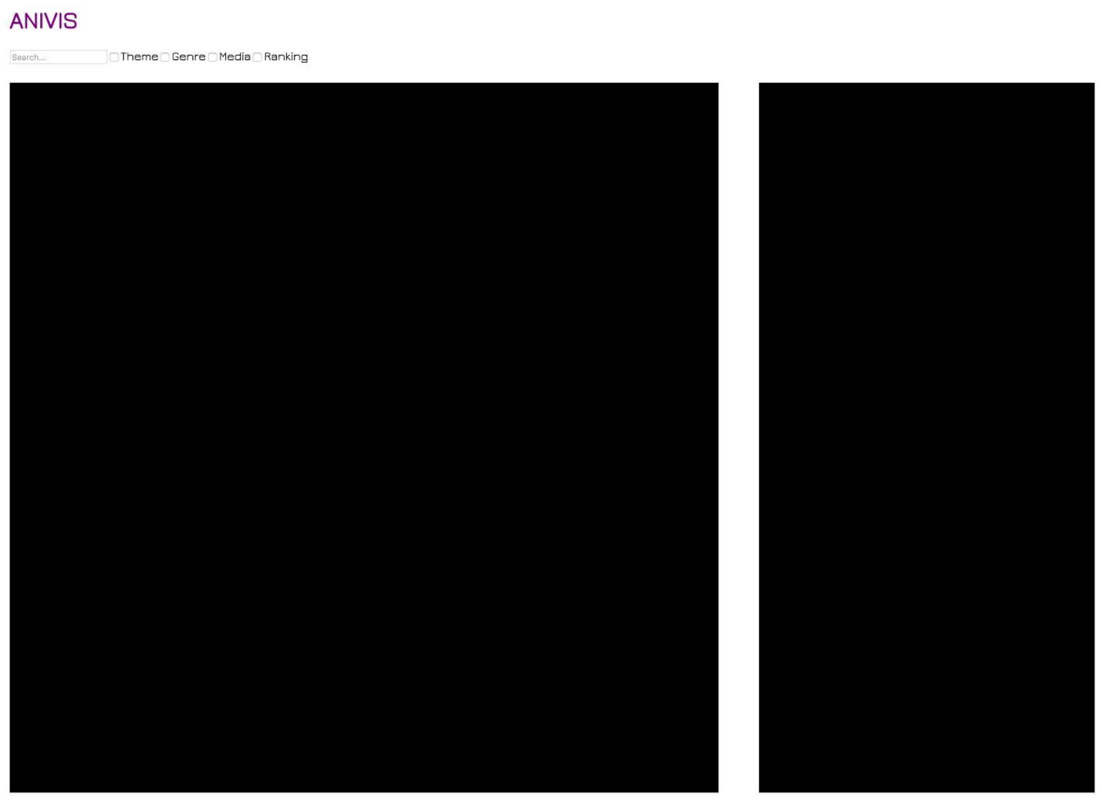
Evolution

March 19th - Joe

I decided to start the project early. So I called Kannan to hack with me in down town. Today's goal is to locate the data and grab it to local for future process. The process is simple, I created a python script which uses gevent to create multiple threads to concurrently fetch data from our poor data provider. I did the math that it could take up to half an hour if I did not do it concurrently. Unfortunately, my first attempt failed because they have some sort of access control on their web server, and my request frequency was too high. After lower the concurrency of my script, I successfully fetched data from their API.

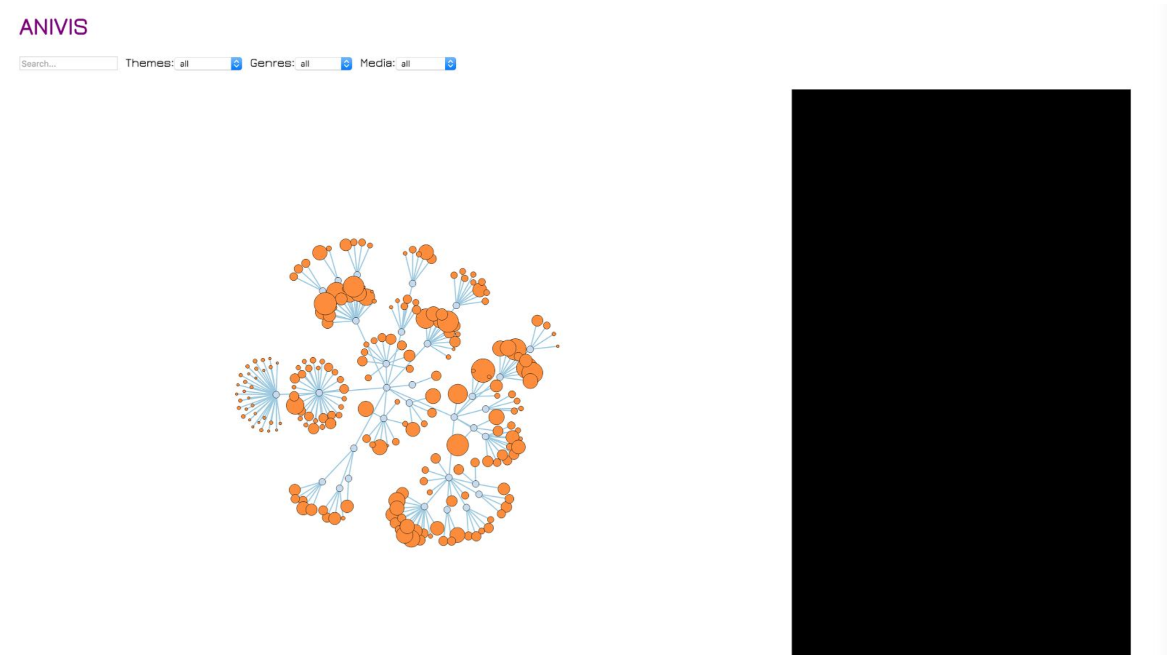
April 10th - Joe

Today I started the web page construction for ANIVIS. I decided to use npm for package management and React to build user interface. The reason I choose React is the internal mechanism which is used by React to handle state transition of it’s user interface is so similar to the underlying idea of d3. Check out this article if you are curious. And npm just makes dependency management trivially easy, we can include packages just like we are using Java or other programming languages that runs on server. After several hours I was able to build up a web page that has basic interactions, though the visualization parts are now empty.



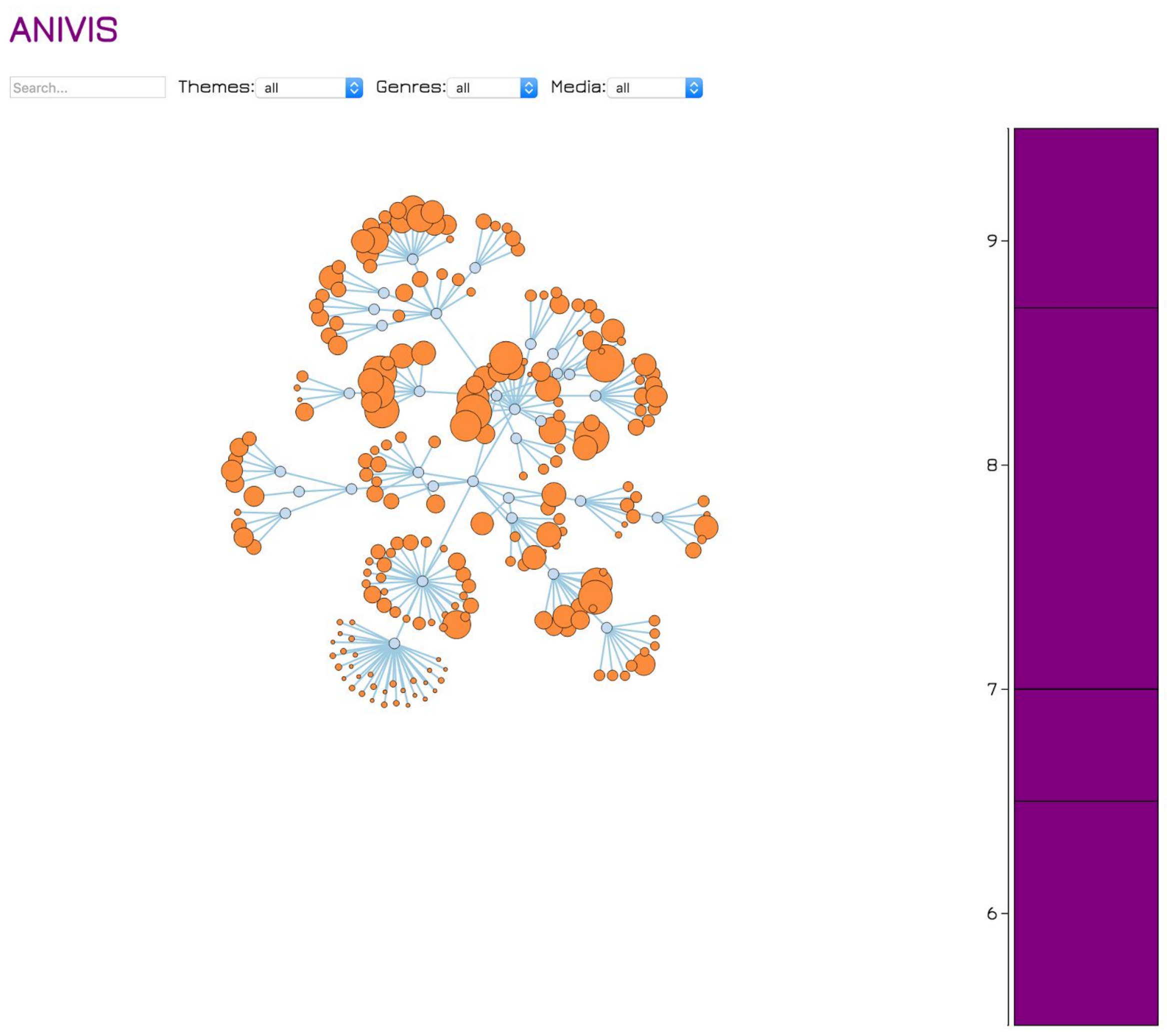
April 17th - Joe

Today I added the Collapsible Force Layout example to the prototype page to get data integration started, but Juan is still writing the code for it. So the force layout you see from the picture is not bound to any data. Besides, after some discussion with my teammates, we decided to use **drop-down lists instead of checkboxes for the filter interaction.** This makes sense because user may want to have finer control over the data displayed.



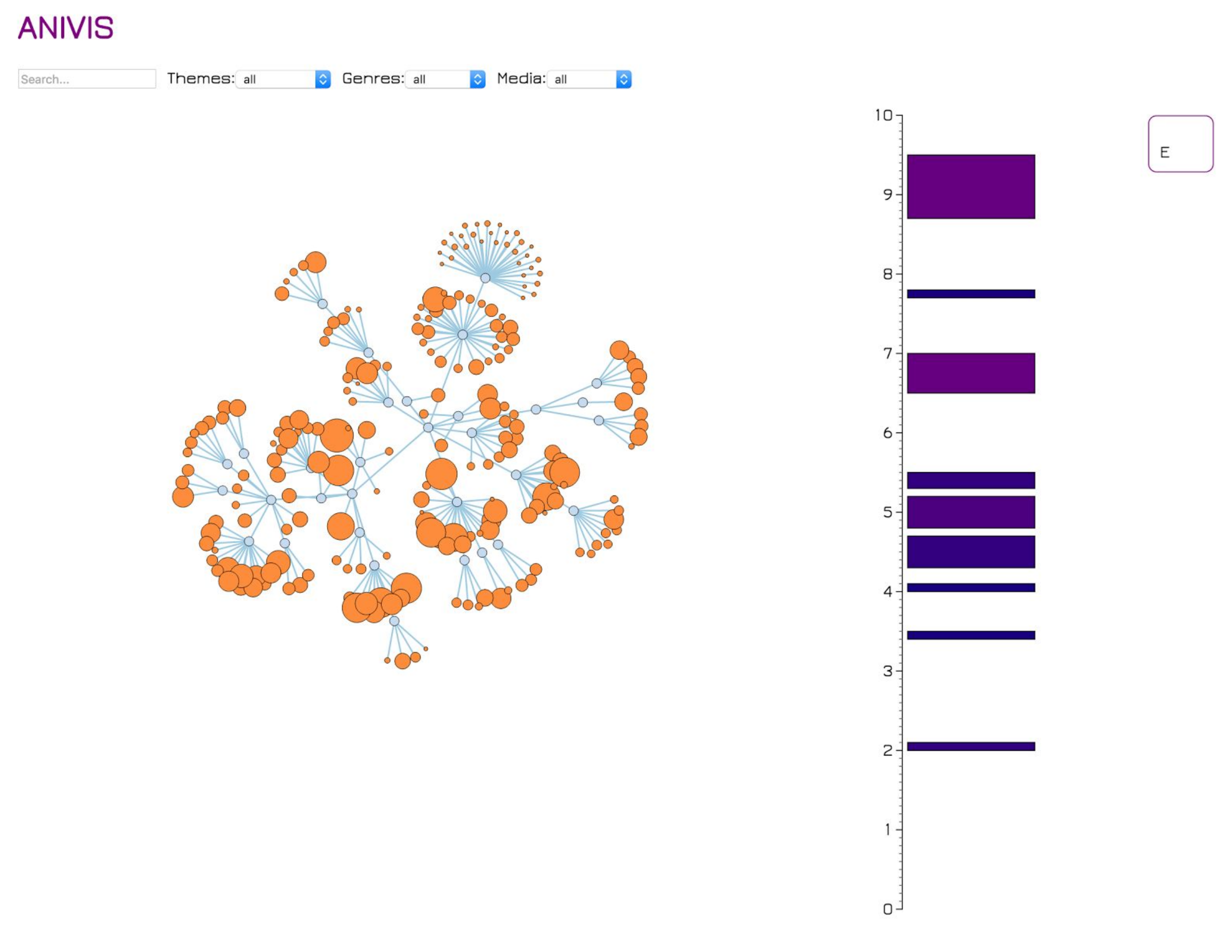
April 18th - Joe

Today I added a simple "Ranking View" that can be broken into pieces when clicked. I was planning to use something similar to Image Map, simply because it's cool. But after some digging around I found that it might not fit our use case. Since we don't really want to establish the connection between the data and some image. What I really want is just to break things into pieces, such that user could have better idea of the ranking distribution.



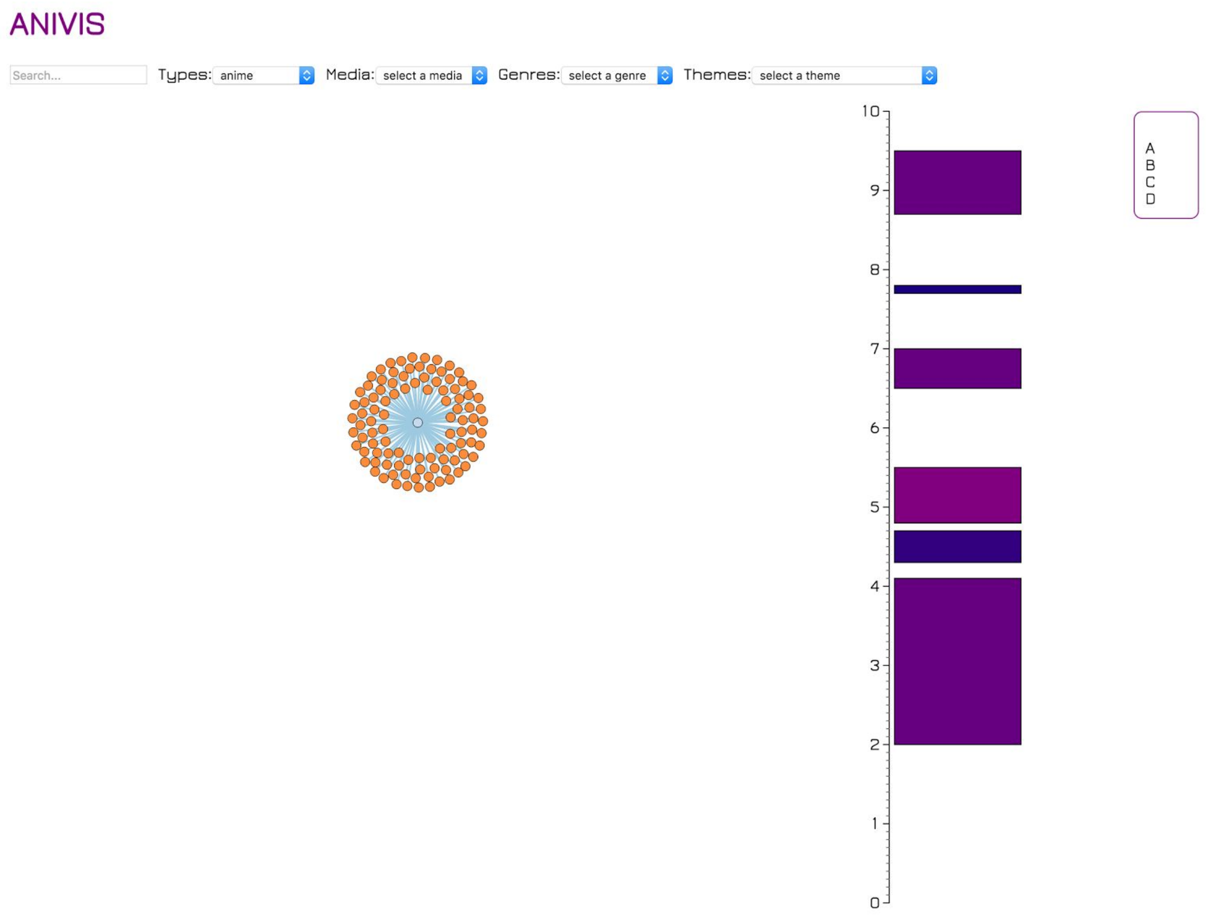
April 24th - Joe

I am keeping iterating on the Ranking View. And now the ranking view can be divided more correctly. Also, I added logic to interpolate the color of each rectangle shown in the Ranking View, the more red in the color, the more anime is in that ranking range. Besides, I added a Detailed View, which is the little round rectangle to the right of the Ranking View. My plan for it is to display anime details in the future. And maybe I can add more interaction to it such that our product can be more interesting.

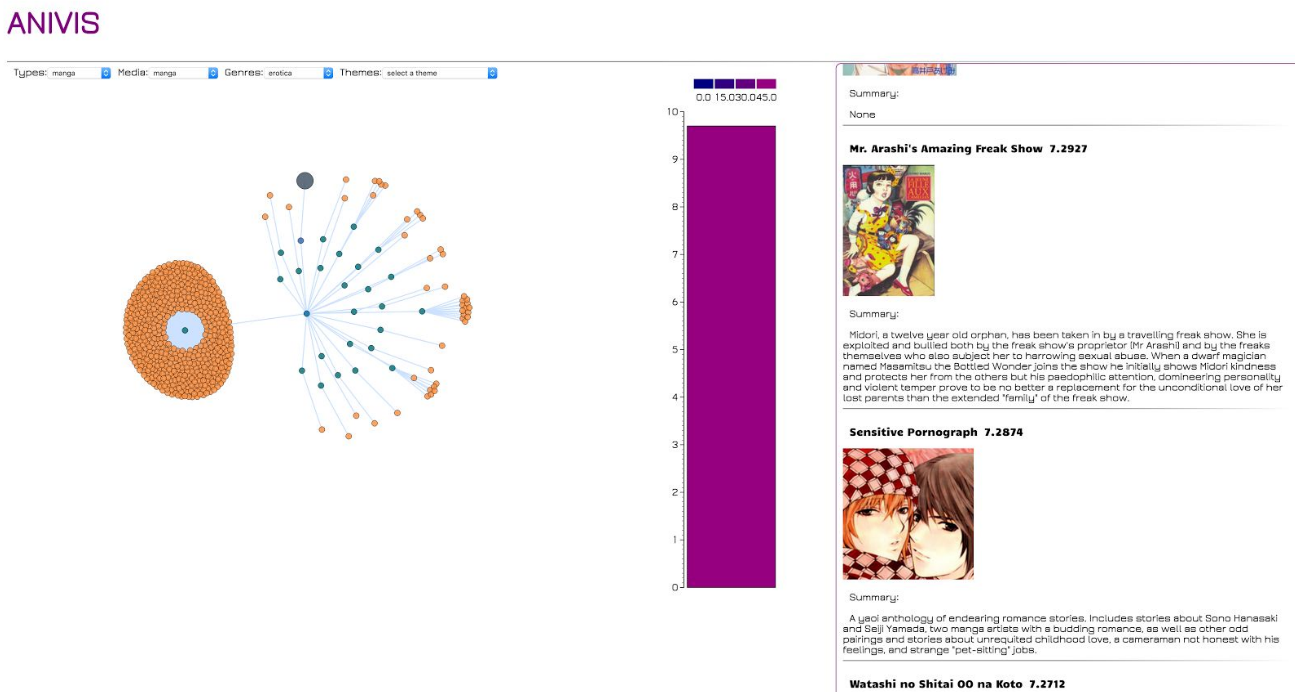


April 25th - Joe

Today, I’ve integrated the Collapsible Force Layout into our web page. There was not a lot of coding, but some figuring-out was needed to make it work nicely with React. And the graph shown below is what i got when I only selected the highest level filter. The visual encoding is not ideal because what we really like to see is a tree structure. So we need to modify the data to make it contain the tree structure. Kannan will be working on it.


ANIVIS
Search...
Types: anime
Media: select a media
Genres: select a genre
Themes: select a theme
10
9
8
7
6
5
4
3
2
1
0
A
B
C
D

May 10th - Joe



Summary:

None

Mr. Arashi's Amazing Freak Show 7.2927

Summary:

Midori, a twelve year old orphan, has been taken in by a travelling freak show. She is exploited and bullied both by the freak show's proprietor [Mr Arashi] and by the freaks themselves who also subject her to harrowing sexual abuse. When a dwarf magician named Masamitsu the Bottled Wonder joins the show he initially shows Midori kindness and protects her from the others but his paedophilic attention, domineering personality and violent temper prove to be no better a replacement for the unconditional love of her lost parents than the extended "family" of the freak show.

Sensitive Pornograph 7.2874

Summary:

A yaoi anthology of endearing romance stories. Includes stories about Sono Hanasaki and Seiji Yamada, two manga artists with a budding romance, as well as other odd pairings and stories about unrequited childhood love, a cameraman not honest with his feelings, and strange "pet-sitting" jobs.

Watashi no Shitai OO na Koto 7.2712

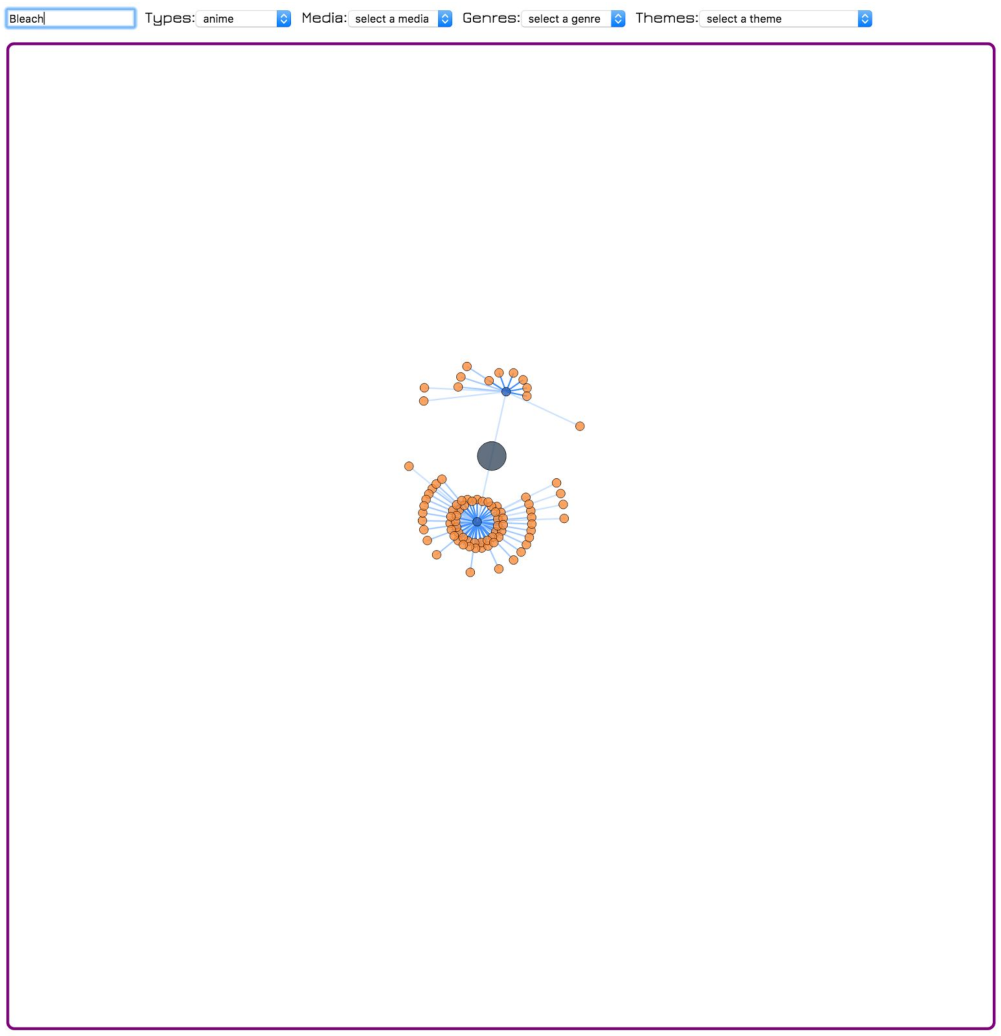
Today I added quite a few new features. You can see from the left side of the picture, the bigger node is the root of the whole tree and it is ponding. The reason behind this is sometimes it's easy to lost track of the tree given bigger datasets. And with a ponding root, we can easily find where the tree originated from. And it makes the vis more live. And also I completed the detail view to include more details (pictures, recommendations or what not). And the third feature I added is the search functionality, with searching enabled, we now can easily look up anime we are interested in. (search box is not shown in the picture above).

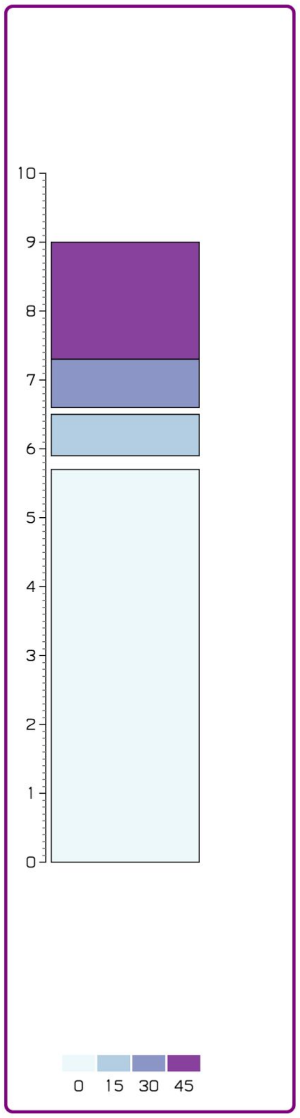
May 12th - Joe

After the final presentation, Alark had given us several suggestions, including:

1. Tweak the link length of the force layout for the similarity view.
2. Using color brewer's color for the ranking view.
3. Make the ranking view not transition from black (which is confusing)

I've implemented all these suggestions, and included screenshots below.


Bleach
Types: anime
Media: select a media
Genres: select a genre
Themes: select a theme



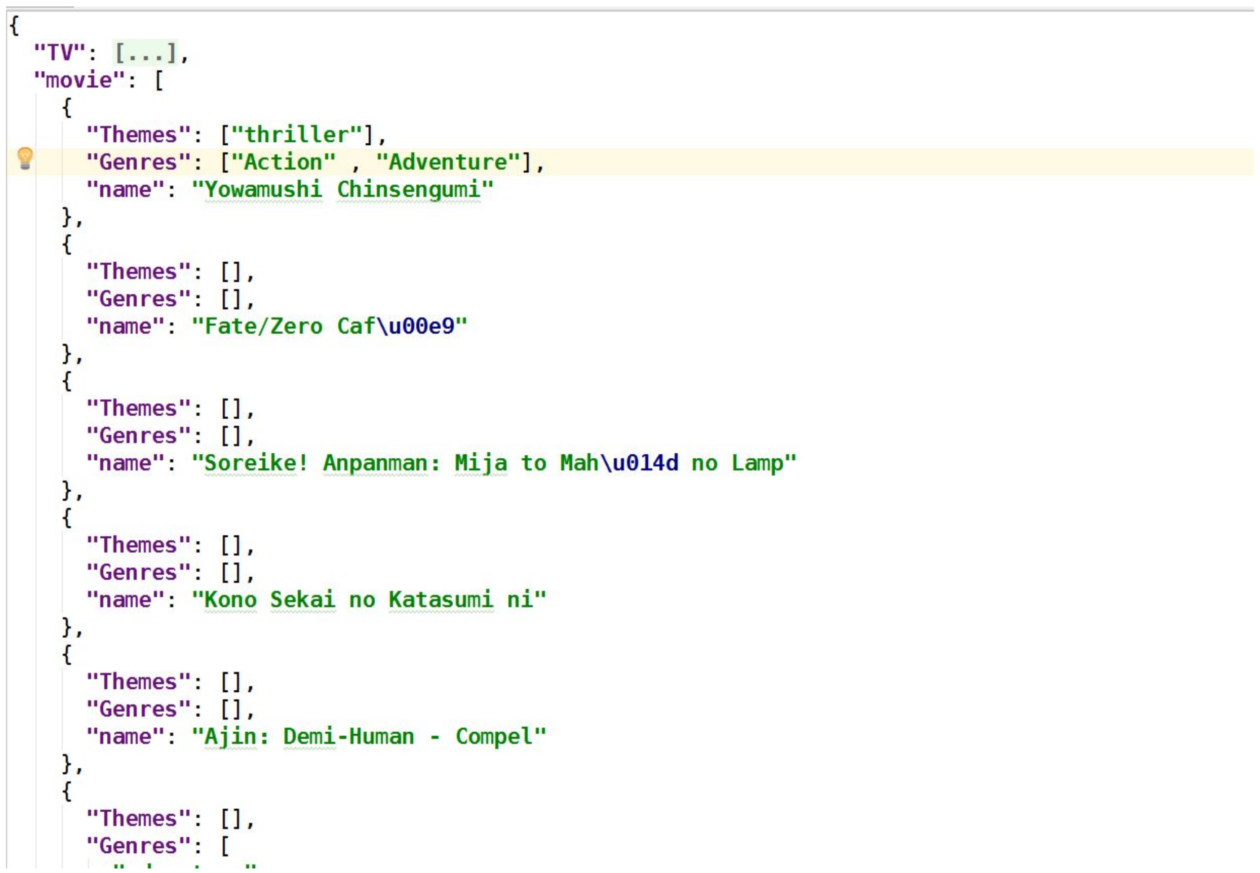
BTW, the ranking view looks really good with the brewed color :)

March 19th - Kannan

We want to start early. So Joe called me up to hack with him at cafe. The task for today is to get the basic of data gather and formating ready. While Joe was working on the python script for data gathering form the web API , I was working on a script that converts the gathered XML file to a Json File. So I wrote a script to that generated an intermediate JSON file.

April 5th - Kannan

After we have finalized on the forced layout visualization, Juan gave me an initial version of JSON that he need for the visualization. The goal for the next few day is expand the script so that it can use the intermediate JSON file and generate the data required for the Visualization. The finally got a one file that had all the information about the anime in the required format. .

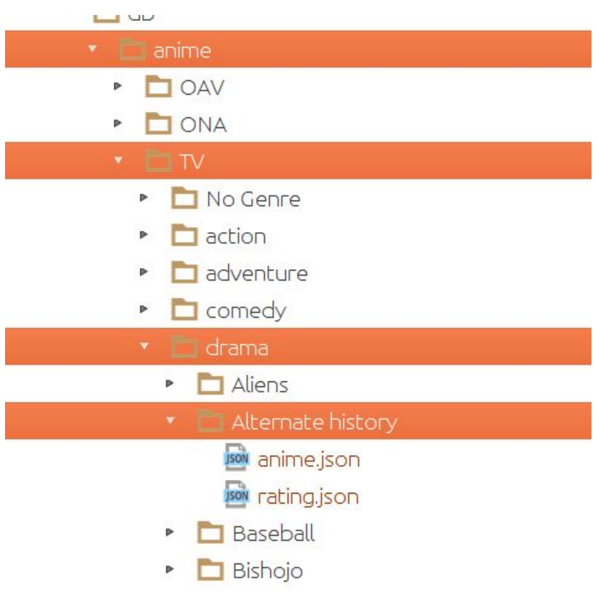


```
{
  "TV": [...],
  "movie": [
    {
      "Themes": ["thriller"],
      "Genres": ["Action" , "Adventure"],
      "name": "Yowamushi Chinsengumi"
    },
    {
      "Themes": [],
      "Genres": [],
      "name": "Fate/Zero Caf\u00e9"
    },
    {
      "Themes": [],
      "Genres": [],
      "name": "Soreike! Anpanman: Mija to Mah\u014d no Lamp"
    },
    {
      "Themes": [],
      "Genres": [],
      "name": "Kono Sekai no Katasumi ni"
    },
    {
      "Themes": [],
      "Genres": [],
      "name": "Ajin: Demi-Human - Compel"
    },
    {
      "Themes": [],
      "Genres": [
```

April 10th - Kannan

After we tried to integrate the data into the visualization we realized that the was way to huge to be able to load it browser without slowing the visualization. So I had to come up with a database design which would allow the client to query manageable amount of data when required.

I thought I would go with the traditional database design but the overhead for setting up and coming up schema was time consuming. So I thought to create a directory structure which contains the data that matches the client query. So create folder for each query parameter and a file called anime.json would have the data for the query.



April 16th - Kannan

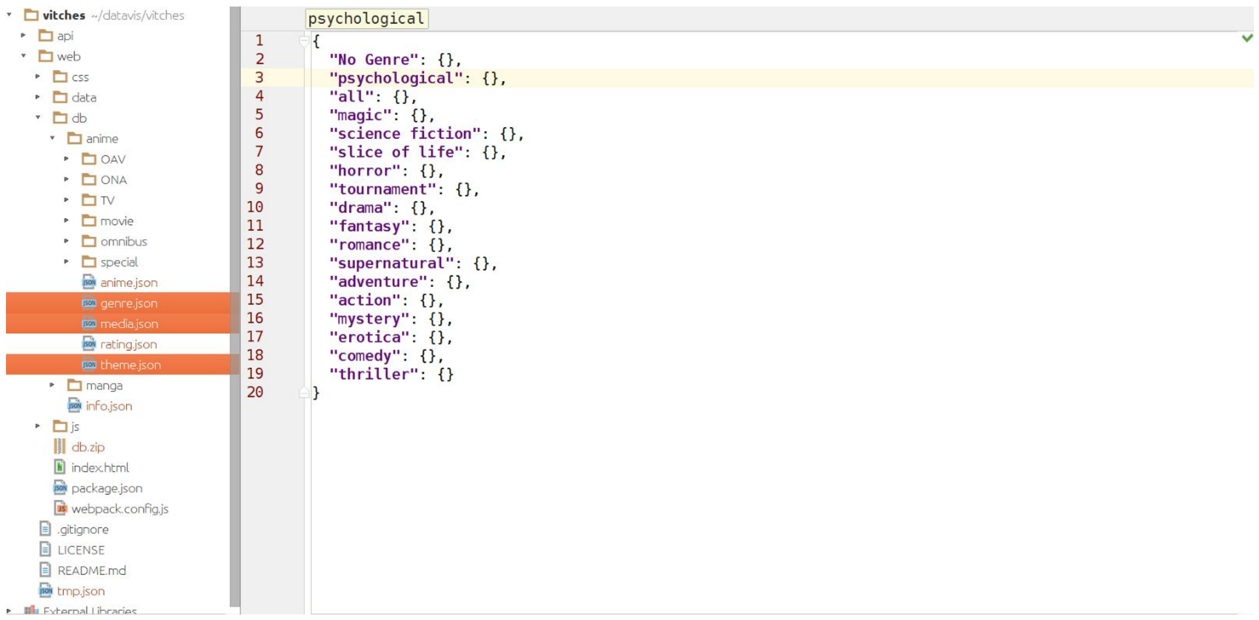
As I feared the data for the Force layout changed. So I need to change the script to make sure the data is as required worked on that for few days and came up with a solid data format.

April 20th - Kannan

Now we had a amazing new view that was created by Joe named “Joe View” (for obvious reason :P). Joe jave me a requirement for the data format for his visualization. Initially I thought I would create a new script for that data, and I thought to myself I’m a program I can do better, so I modified the existing script to generate the data for Joe View. It followed the same query structure to generate the limited amount data. Lucky the data requirement for this view didn't no change as much.

April 26th - Kannan

We has some Issues when integrating both the ranking and forced layout view. Where we went through few more iteration of the Folder structure by including list of media, genre and theme as a JSON file.



```
{
  "No Genre": {},
  "psychological": {},
  "all": {},
  "magic": {},
  "science fiction": {},
  "slice of life": {},
  "horror": {},
  "tournament": {},
  "drama": {},
  "fantasy": {},
  "romance": {},
  "supernatural": {},
  "adventure": {},
  "action": {},
  "mystery": {},
  "erotica": {},
  "comedy": {},
  "thriller": {}
}
```

May 2nd - Kannan

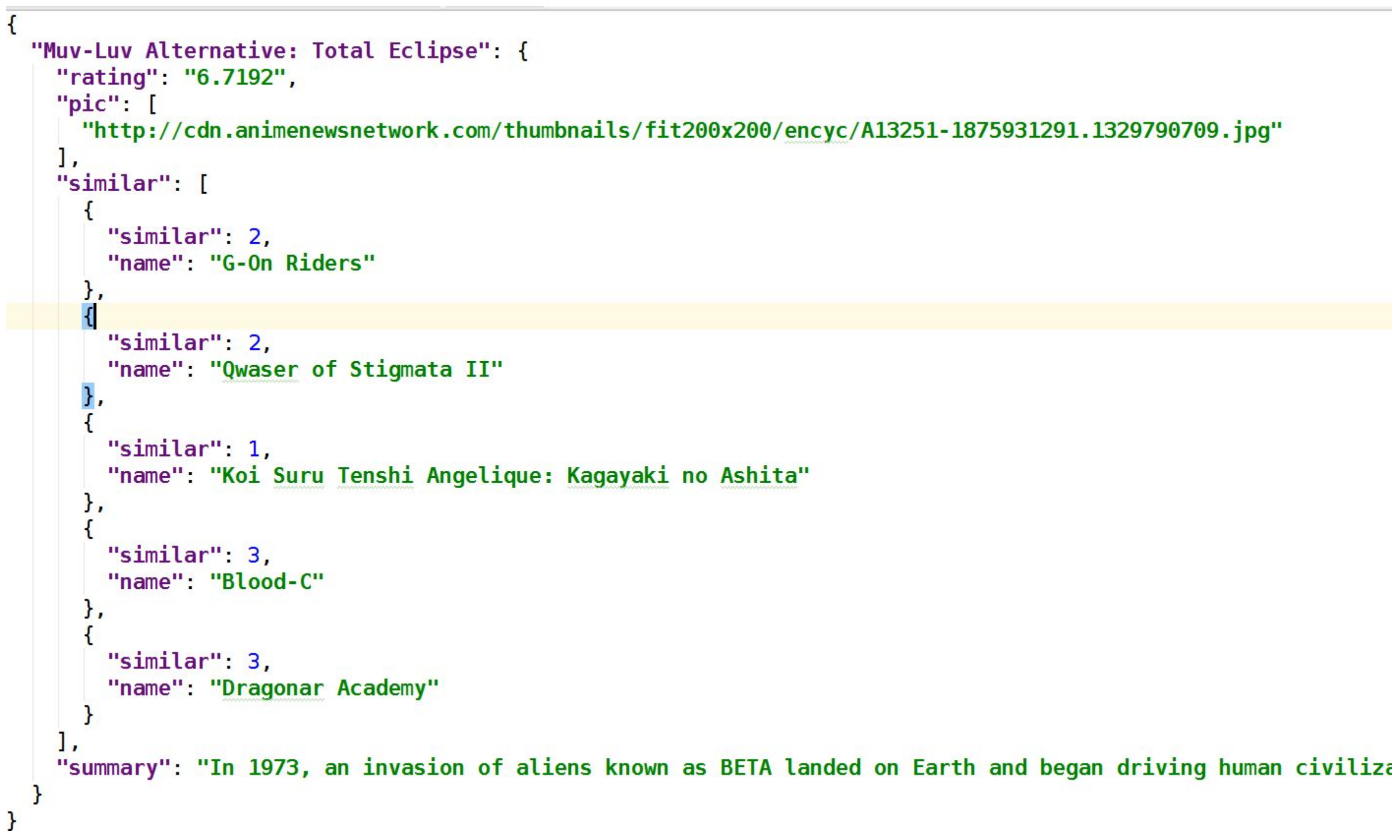
After Alark's input during the Beta Release, we wanted to improve on the visualization so planned to add more information about the visualization. Naturally I went back to the script to modify it to extract the information that are required and modify the existing data in the Folder structure to include these new information.

May 5th - Kannan

Now we wanted to show similar animes. I wrote a small algorithm that figure out the similar anime for ever other anime that we have. It basically check for the anime that satisfies most criteria like media, genre and theme and have better ranking than the current one.

May 10th - Kannan

After the integration of the data we found out that add these information blew up the visualization making it very slow. I create a new file structure which is info.json which contains info for all the anime. Even then the file was way too large for system to handle. So I tried to tweak with the data to reduce the size,



```
{
  "Muv-Luv Alternative: Total Eclipse": {
    "rating": "6.7192",
    "pic": [
      "http://cdn.animenewsnetwork.com/thumbnails/fit200x200/encyc/A13251-1875931291.1329790709.jpg"
    ],
    "similar": [
      {
        "similar": 2,
        "name": "G-On Riders"
      },
      {
        "similar": 2,
        "name": "Qwaser of Stigmata II"
      },
      {
        "similar": 1,
        "name": "Koi Suru Tenshi Angelique: Kagayaki no Ashita"
      },
      {
        "similar": 3,
        "name": "Blood-C"
      },
      {
        "similar": 3,
        "name": "Dragonar Academy"
      }
    ],
    "summary": "In 1973, an invasion of aliens known as BETA landed on Earth and began driving human civiliza
  }
}
```

March 20th - Juan

Because we were exploring the possibility of creating a visualization using force layouts, I started doing some research about that and discovering how they worked.

April 3rd - Juan

After finishing the initial version of the forced layout visualization, I sent Kannan the first draft of the JSON document that is needed in order to render the nodes in the visualization. I also started checking how we can query the anime data easily.

April 10th - Juan

Today I started creating a basic skeleton for our website, creating the css folder, the lib folder, organizing some dependencies and making sure that the website contains everything to be deployed successfully.

April 15th - Juan

I started researching about React and how it is used in our visualization. I worked with Joe in the integration of the forced layout view and React.

May 3rd - Juan

Once that the first draft was presented, I started making some polishing to the code and improving the force layout. I worked with Kannan to modify once more the data format in order to present the information.

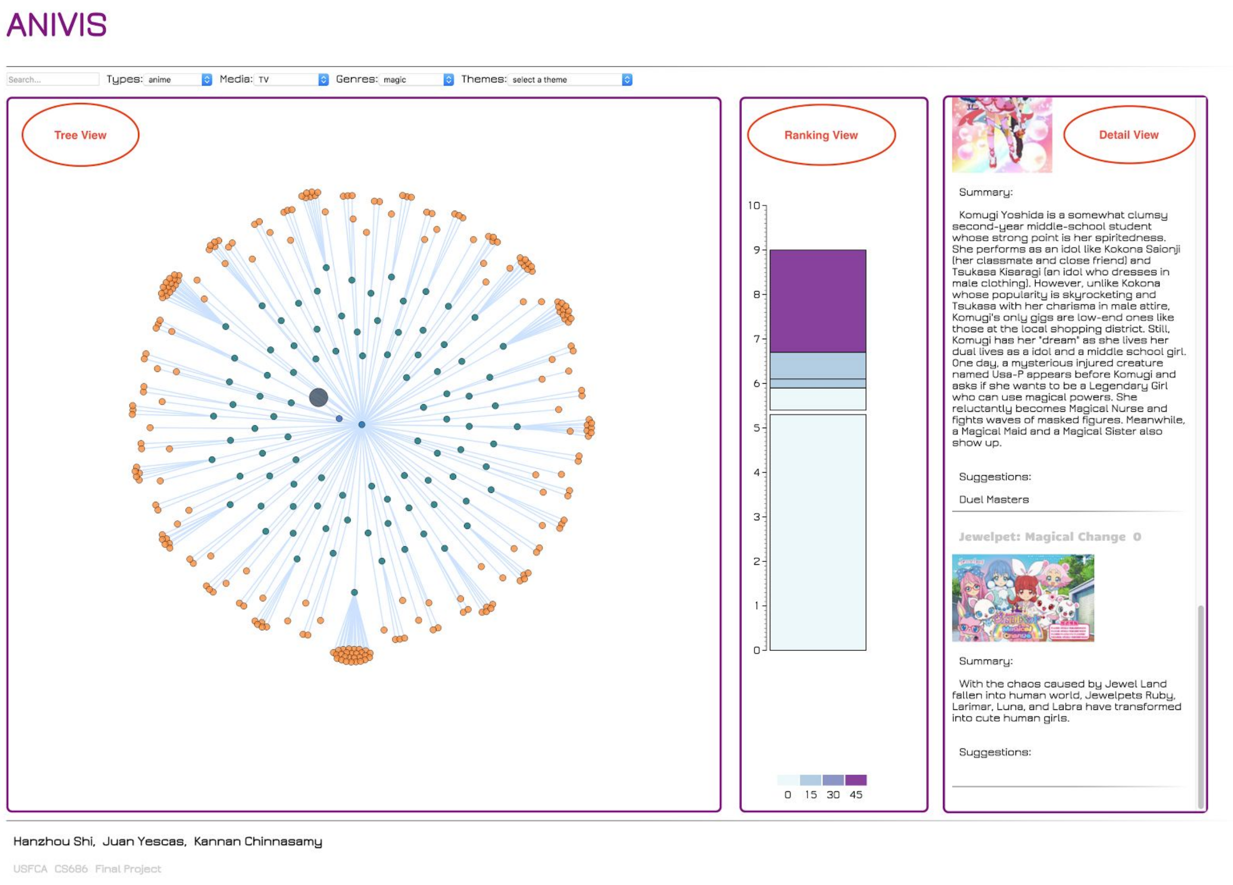
May 7th - Juan

Once that we had a stable version of our visualization, I started creating the presentation slides for our project. I incorporated a background about our project, technologies used in it, main visualizations and challengues.

Showcase

The following three pictures shows a common workflow when using ANIVIS.

First I use filter to filter out animes and try to find animes are of specific genre, has a specific theme and played on a specific media. Or click on the ranking view to divide the rankings to find either higher ranking animes or lower ranking animes. Also we can scroll up and down in the detail view if any of the anime shown worth some more attention.



Summary:

Komugi Yoshida is a somewhat clumsy second-year middle-school student whose strong point is her spiritedness. She performs as an idol like Kokona Saionji (her classmate and close friend) and Tsukasa Kisaragi (an idol who dresses in male clothing). However, unlike Kokona whose popularity is skyrocketing and Tsukasa with her charisma in male attire, Komugi's only gigs are low-end ones like those at the local shopping district. Still, Komugi has her "dream" as she lives her dual lives as a idol and a middle school girl. One day, a mysterious injured creature named Usa-P appears before Komugi and asks if she wants to be a Legendary Girl who can use magical powers. She reluctantly becomes Magical Nurse and fights waves of masked figures. Meanwhile, a Magical Maid and a Magical Sister also show up.

Suggestions:

Duel Masters

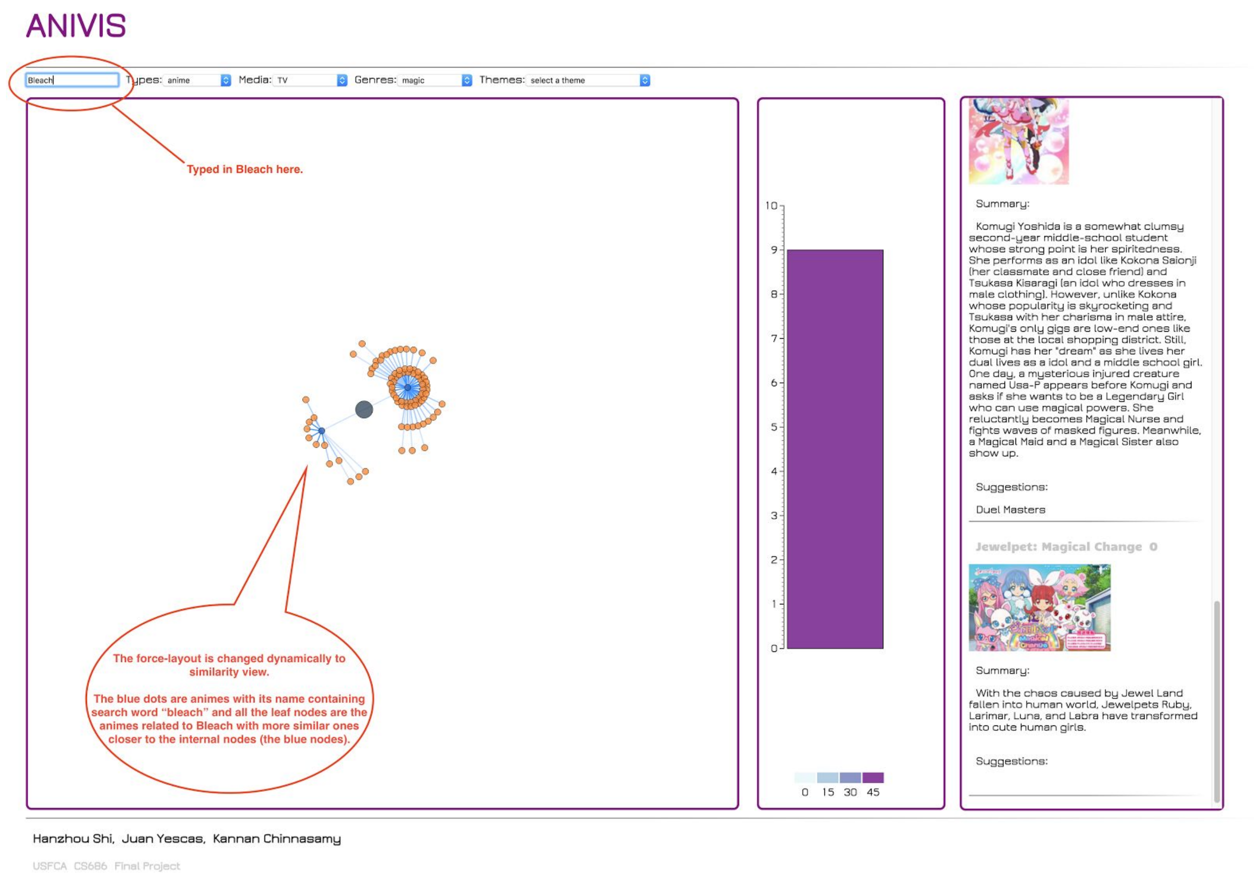
Jewelpet: Magical Change 0

Summary:

With the chaos caused by Jewel Land fallen into human world, Jewelpets Ruby, Larimar, Luna, and Labra have transformed into cute human girls.

Suggestions:

Whether or not I find the anime I'm interested in, I can start typing in the search box for some name related to animes, in this case "Bleach". And we can get those animes matching this search word and all the animes related to those search results.



ANIVIS

Bleach | Types: anime | Media: TV | Genres: magic | Themes: select a theme

Typed in Bleach here.

The force-layout is changed dynamically to similarity view.

The blue dots are animes with its name containing search word "bleach" and all the leaf nodes are the animes related to Bleach with more similar ones closer to the internal nodes (the blue nodes).

Summary:

Komugi Yoshida is a somewhat clumsy second-year middle-school student whose strong point is her spiritedness. She performs as an idol like Kokona Saionji (her classmate and close friend) and Tsukasa Kisaragi (an idol who dresses in male clothing). However, unlike Kokona whose popularity is skyrocketing and Tsukasa with her charisma in male attire, Komugi's only gigs are low-end ones like those at the local shopping district. Still, Komugi has her "dream" as she lives her dual lives as a idol and a middle school girl. One day, a mysterious injured creature named Usa-P appears before Komugi and asks if she wants to be a Legendary Girl who can use magical powers. She reluctantly becomes Magical Nurse and fights waves of masked figures. Meanwhile, a Magical Maid and a Magical Sister also show up.

Suggestions:

Duel Masters

Jewelpet: Magical Change 0

Summary:

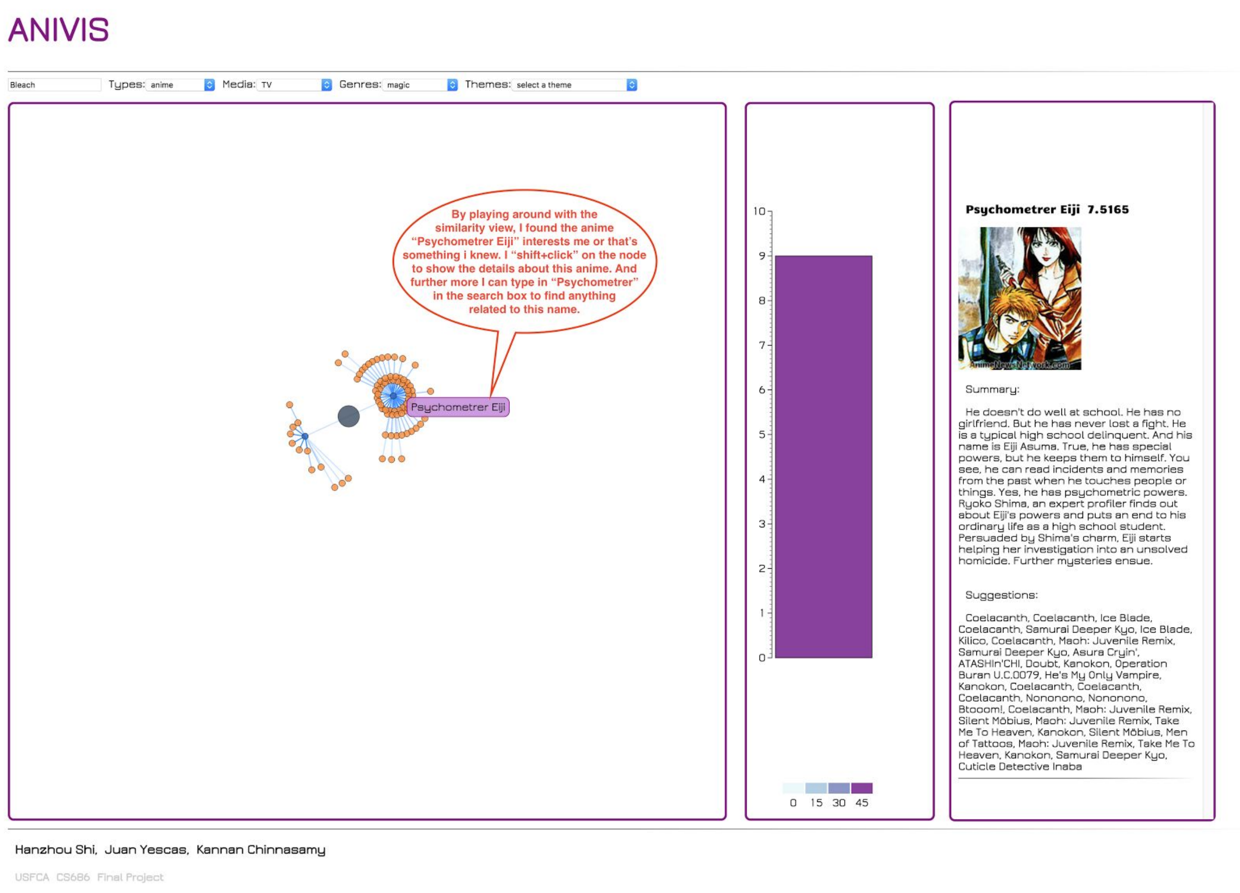
With the chaos caused by Jewel Land fallen into human world, Jewelpets Ruby, Larimar, Luna, and Labra have transformed into cute human girls.

Suggestions:

Hanzhou Shi, Juan Yescas, Kannan Chinnasamy

USFCA CS686 Final Project

Then I can start exploring the similarity view (which is the tree showing the similarity animes), and whenever something catches my attention, I can shift-click on that node and show the detail, and furthermore, I can use that anime to find other animes related to it. And keep going…



By playing around with the similarity view, I found the anime "Psychometrer Eiji" interests me or that's something i knew. I "shift+click" on the node to show the details about this anime. And further more I can type in "Psychometrer" in the search box to find anything related to this name.

Psychometrer Eiji 7.5165

Summary:

He doesn't do well at school. He has no girlfriend. But he has never lost a fight. He is a typical high school delinquent. And his name is Eiji Asuma. True, he has special powers, but he keeps them to himself. You see, he can read incidents and memories from the past when he touches people or things. Yes, he has psychometric powers. Ryoko Shima, an expert profiler finds out about Eiji's powers and puts an end to his ordinary life as a high school student. Persuaded by Shima's charm, Eiji starts helping her investigation into an unsolved homicide. Further mysteries ensue.

Suggestions:

Coelacanth, Coelacanth, Ice Blade, Coelacanth, Samurai Deeper Kyo, Ice Blade, Kilico, Coelacanth, Maoh: Juvenile Remix, Samurai Deeper Kyo, Asura Cryin', ATASHIn'CHI, Doubt, Kanokon, Operation Buran U.C.0079, He's My Only Vampire, Kanokon, Coelacanth, Coelacanth, Coelacanth, Nononono, Nononono, Btooom!, Coelacanth, Maoh: Juvenile Remix, Silent Möbius, Maoh: Juvenile Remix, Take Me To Heaven, Kanokon, Silent Möbius, Men of Tattoos, Maoh: Juvenile Remix, Take Me To Heaven, Kanokon, Samurai Deeper Kyo, Cuticle Detective Inaba

Annotated Bibliography

Bostock, Mike. "Data-Driven Documents-D3. js." (2014).

The author implemented a framework for data-driven visualization in Javascript. It is one of the most widely used visualization libraries among practitioners. The authors of ANIVIS are also using d3 to implement their ideas.

Bostock, Mike. "Force-Directed Graph" (2016). [LINK]

The author implemented a force-directed as one of the d3.js examples. This graph is demonstrate a basic idea to display node-link data with force-directed layout. This inspired the author of ANIVIS to use some kind of force-directed graph to visualize relevance between animes.

Bostock, Mike. "Collapsible Force Layout" (2016). LINK

The author implemented a collapsible force-directed graph in this d3.js example. In addition to an ordinary force-directed graph, the author made each node collapsible. This layout is more suitable to display data has categorical attributes. The authors of ANIVIS decided to use this example as a starting point to implement their main view.

Bostock, Mike. "Force Layout" (2016). LINK

As a part of D3 reference, the author has given detailed information about how to use the force layout feature built inside d3. This will be a valuable resource for ANIVIS authors when they are building their visualization.

Holten, Danny, and Jarke J. Van Wijk. "Force-Directed Edge Bundling for Graph Visualization." ***Computer Graphics Forum*****. Vol. 28. No. 3. Blackwell Publishing Ltd, 2009.**

The authors of this article presented a technique called edge bundling which can be used in node-link diagrams to reduce edge clutter and in addition to reveal edge patterns. The authors of ANIVIS used this article as a reference when they are implementing their force-directed

nswamy14. "Image Map" (2016). LINK

The author implemented a simple algorithm to map an image to SVG circles. This inspired the author of ANIVIS that they can use a similar technique to implement the ranking view. The ranking view initially gives an overview of

anime rating distribution while after being broken down into smaller pieces, user will have more detailed information about the actual rating information.

Facebook React (2013). LINK

React is a javascript library used to build user interface. The ANIVIS team is also React to implement the user interaction parts, especially the data passing of their product.

Npm, Inc. npm (2014). LINK

Npm is the package manager for Javascript. The ANIVIS team is using npm to manage their packages.

wonfu (2015). LINK

Wonfu is a well-known visualization made by Shirley Wu. I adopted a lot of idea including code organization from that project.