

---

# Creating with Data: Technical and Evaluation Report

**Aimee Holdsworth**  
**Department of Computer**  
**Science and Creative**  
**Technology**  
University of the West of England  
Coldharbour Lane  
Bristol, UK  
[author1@uwe.ac.uk](mailto:author1@uwe.ac.uk)

submitted 12/04/2018

## **Data Collection**

This term I created my own dataset as I couldn't find a dataset that provided the information I wanted. I decided I wanted to create a new visualisation as this would be the best way to challenge myself and settled on the topic of music, specifically the 100 most listened to songs of 2017. I found data online and compiled this into a CSV file as I was going to get everything working offline more moving it to an online database(DB). I found a Music API online called MusixMatch that would provide my visualization with lyrics to songs. This would be useful during the later stages of my project including online data.

## **Data storage and transfer**

I originally worked with my data offline on my computer in a CSV file. This was for ease whilst I was creating the bulk of my visualisation. However, once I had it nearly all done I moved the data to an online DB to try and connect everything up online so it's accessible. I had a few issues with connecting to my DB, as it would connect successfully and show this in the console, however the data would not be drawn out. At this time, I am unable to fix this issue so had to revert to a CSV file.

## **Data analysis and cleaning**

As I had created my own dataset by compiling different data from online my data set didn't need any cleaning as

```
function toggle_visibility(id) { //function to show and hide the lyrics
  var e = document.getElementById(id);
  if(e.style.display == 'block') //shows the data
    e.style.display = 'none'; //hides the data
  else
    e.style.display = 'block';
}
```

Figure 1: An example of a function being used to toggle the visibility of lyrics

only the necessary data was included. However, the API I found needed a lot of cleaning when drawing in the information. The API I chose was great because it could provide me with lots of information about songs, however, I didn't need all this information. As to not slow down the load speeds of my visualisation or create a messy visualisation with useless information I cleaned this up. I did this by getting specific information from the website. By only getting small modules of the data bandwidth is also saved. I created variables to store specific data such as the song lyrics and then used functions to display this onscreen.

### Data visualization

As my visualisation progressed so did my ideas for how it should look, this came from realising different strong and weak points in my capabilities and adjusting my plan so that I could achieve my goals. Originally, I was planning on creating a bar chart to show my data, however, I didn't feel like this was the best way to represent the data. I carried out more research and discovered bubble charts which would be perfect for my data. I first explored a few different bubble chart examples to try and get my head around the code before creating my own. I found that it was quite different to my last visualisation so took a bit of adjusting to, however once I got going everything began to fall into place. For this project I had many different HTML pages, this was so lots of information could be shown in a simple way, for example a page for each for different songs.

To make my chart interesting I added in interactivity. This included drag behaviour that meant that the user can zoom in and out and move the chart around. Using D3s drag library I was able to implement this easily with a few lines of code. I also applied a force layout to the chart so

that users could move individual bubbles if they wanted to play with the chart, but they would then snap back into place with a transition streamline this movement.

I used functions throughout my visualisation as they were very useful in implementing different interactive features. For example, using buttons linked to functions the user can change the view of the chart to show in assorted colours depending on how the data is sorted.

On the song pages I used the web audio API to visualise songs and then API data from MusixMatch to draw in the lyrics and song information. I used HTML to create a toggle function which meant that the user could decide if they lyrics were shown. (Figure 1) I had errors with Cross Origin Requests with my API and had to add code that would allow these so that the API could correctly get the data.(Figure 2)

The most difficult feature to implement on this page was resizing the audio visual and making sure that it was responsive. I wanted the page to look nice and the lyrics and visualisation to line up, to do this I needed to resize the visualisation, this took a bit of playing around with the code. The tricky part of making it responsive, was to make sure that it didn't override the resizing code. By making sure the code was linked to the right part of the visualisation I made sure both elements worked properly.

### Evaluation

I think that this project went very well, I am proud of the visualisation that I have made and think that it has vastly improved the skills I developed on the first assignment. I challenged myself to include difficult elements and keep working on them when I thought it was going to be too hard, for example clicking on the top 10 bubbles and being

taken to a new, specific page. I worked for a long time on this before I had a breakthrough! I like that this visualisation is more interactive with many different states and the user being able to click, drag and show/hide elements. I think that I have done well including all these different elements.

If I had more time though I would very much like to carry on working on this project and add in more interactivity. I enjoy adding in these elements that engage the user and pushing myself to try coding new features.

I am particularly proud of how my knowledge of D3 has grown, instead of having to google the answer every time I found that I was able to think of the solution myself and code it. I look forward to being able to do this more and more in the future.

Part of the project that I found hard was linking my visualisation to the online database. My PHP skills are still limited and I struggled to understand all the code. This part took me a while to complete as I had errors fetching the data and then even though all the PHP was correct, and the data was printing in the console my visualisation would no longer show. Unfortunately, I couldn't get this to work and had to go back to using a CSV file, I think the error was lying within the way bubble charts are set up in D3 as the data was loading correctly into the console. It is a shame I could not get this to work, however, I still developed skills learning how to connect to an online DB with PHP and overcame many problems with this part of the project before getting to this final problem.

## References

- Appiah, L. (n.d.). *d3.js bubbles with zoom & click to select - JSFiddle*. [online] JsFiddle.net. Available at: <http://jsfiddle.net/LeslieOA/stjxjxt8/> [Accessed 9 Mar. 2018].
- Blockspring (2017). *D3 Wordcloud*. [online] Bl.ocks.org. Available at: <https://bl.ocks.org/blockspring/847a40e23f68d6d7e8b5> [Accessed 16 Mar. 2018].
- Chartio. (n.d.). *How to Show Data on Mouseover in d3.js*. [online] Available at: <https://chartio.com/resources/tutorials/how-to-show-data-on-mouseover-in-d3js/> [Accessed 6 Mar. 2018].
- Developer.musixmatch.com. (2018). *Musixmatch Developer API*. [online] Available at: <https://developer.musixmatch.com/> [Accessed 3 Apr. 2018].
- D3Noob (2013). *Update d3.js Data Dynamically (Button Click)*. [online] D3noob.org. Available at: <http://www.d3noob.org/2013/02/update-d3js-data-dynamically-button.html> [Accessed 9 Mar. 2018].
- D3Noob (2016). *Show / hide elements on mouse click with d3.js*. [online] Bl.ocks.org. Available at: <http://bl.ocks.org/d3noob/5d621a60e2d1d02086bf> [Accessed 9 Mar. 2018].
- D3Noob (2018). *d3.js tool tips with links in data*. [online] Bl.ocks.org. Available at: <http://bl.ocks.org/d3noob/2e224baa7472c5e9e0e8> [Accessed 4 Mar. 2018].
- Graphla, T. (n.d.). *Data Storytelling with Word Clouds in D3*. [online] Graphla.com. Available at: <http://www.graphla.com/portfolio/data-story-telling-word-clouds.html> [Accessed 9 Mar. 2018].
- Guerra, J. (2016). *Bubble Chart d3 v4*. [online] Bl.ocks.org. Available at: <https://bl.ocks.org/john-guerra/0d81ccfd24578d5d563c55e785b3b40a> [Accessed 5 Mar. 2018].

- Hobbelt, G. (2018). *d3.js: force layout; click to group/bundle nodes*. [online] Bl.ocks.org. Available at: <http://bl.ocks.org/GerHobbelt/3071239> [Accessed 4 Mar. 2018].
- Imura, T. (n.d.). *Bubble Chart with D3.js using Realtime Data*. [online] CodePen. Available at: <https://codepen.io/girliemac/pen/qFGtw> [Accessed 5 Mar. 2018].
- Jhin, C. (2016). *D3-Force: Split Categorical*. [online] Bl.ocks.org. Available at: <https://bl.ocks.org/cjhin/077ecb0fd03f0abd576d39c70563a724> [Accessed 28 Mar. 2018].
- Jsfiddle.net. (n.d.). *D3 Bubble chart - JSFiddle*. [online] Available at: <https://jsfiddle.net/r24e8xd7/9/> [Accessed 5 Mar. 2018].
- Karatus, O. (n.d.). *D3 DropDown*. [online] CodePen. Available at: <https://codepen.io/tarsusi/pen/reovOV> [Accessed 25 Mar. 2018].
- Liu, W. (2018). *D3 Mouse Events*. [online] Bl.ocks.org. Available at: <http://bl.ocks.org/WilliamQLiu/76ae20060e19bf42d774> [Accessed 6 Mar. 2018].
- Mesquita, D. (2017). *A gentle introduction to D3: how to build a reusable bubble chart*. [online] freeCodeCamp. Available at: <https://medium.freecodecamp.org/a-gentle-introduction-to-d3-how-to-build-a-reusable-bubble-chart-9106dc4f6c46> [Accessed 5 Mar. 2018].
- Packt (2015). *Rapid D3.js Tutorial: Creating a Word Cloud* | *packtpub.com*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=1KEiTiu0k44> [Accessed 16 Mar. 2018].
- Pignatelli', M. (2017). *Limit panning/zooming in d3 v4*. [online] Bl.ocks.org. Available at: <https://bl.ocks.org/emepyc/7218bc9ea76951d6a78b0c7942e07a00> [Accessed 10 Apr. 2018].
- Phuong (2018). *D3 Bubble chart*. [online] Bl.ocks.org. Available at: <http://bl.ocks.org/phuonghuynh/54a2f97950feadb45b07> [Accessed 5 Mar. 2018].
- Preziotte (2016). *preziotte/party-mode*. [online] GitHub. Available at: <https://github.com/preziotte/party-mode> [Accessed 8 Mar. 2018].
- Schmuecker', R. (2018). *D3.js Drag and Drop, Zoomable, Panning, Collapsible Tree with auto-sizing..* [online] Bl.ocks.org. Available at: <http://bl.ocks.org/robschmuecker/7880033> [Accessed 19 Mar. 2018].
- Singh, S. (2016). *d3.js | Web Audio API*. [online] Bl.ocks.org. Available at: <http://bl.ocks.org/eesur/6ad4ee84c81b664353a7> [Accessed 8 Mar. 2018].
- Soma, J. (2016). *Making an interactive bubble chart with D3 v4*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=lPr60pexvEM> [Accessed 9 Mar. 2018].
- Vallandingham, J. (2018). *Creating Animated Bubble Charts in D3*. [online] Vallandingham.me. Available at: [http://vallandingham.me/bubble\\_charts\\_in\\_d3.html](http://vallandingham.me/bubble_charts_in_d3.html) [Accessed 23 Mar. 2018].
- Viz Data (2016). *d3 color scales & colorbrewer (v3)*. [online] Bl.ocks.org. Available at: <http://bl.ocks.org/emmasaunders/52fa83767df27f1fc8b3ee2c6d372c74> [Accessed 6 Apr. 2018].
- White, C. (n.d.). *Simple D3.js Dashboard*. [online] Colinwhite.net. Available at:

<http://colinwhite.net/Dash2.5/> [Accessed 26 Mar. 2018].

www.tutorialspoint.com. (2018). *D3.js Requests API*.  
[online] Available at:  
[https://www.tutorialspoint.com/d3js/d3js\\_requests\\_api.htm](https://www.tutorialspoint.com/d3js/d3js_requests_api.htm) [Accessed 16 Mar. 2018].