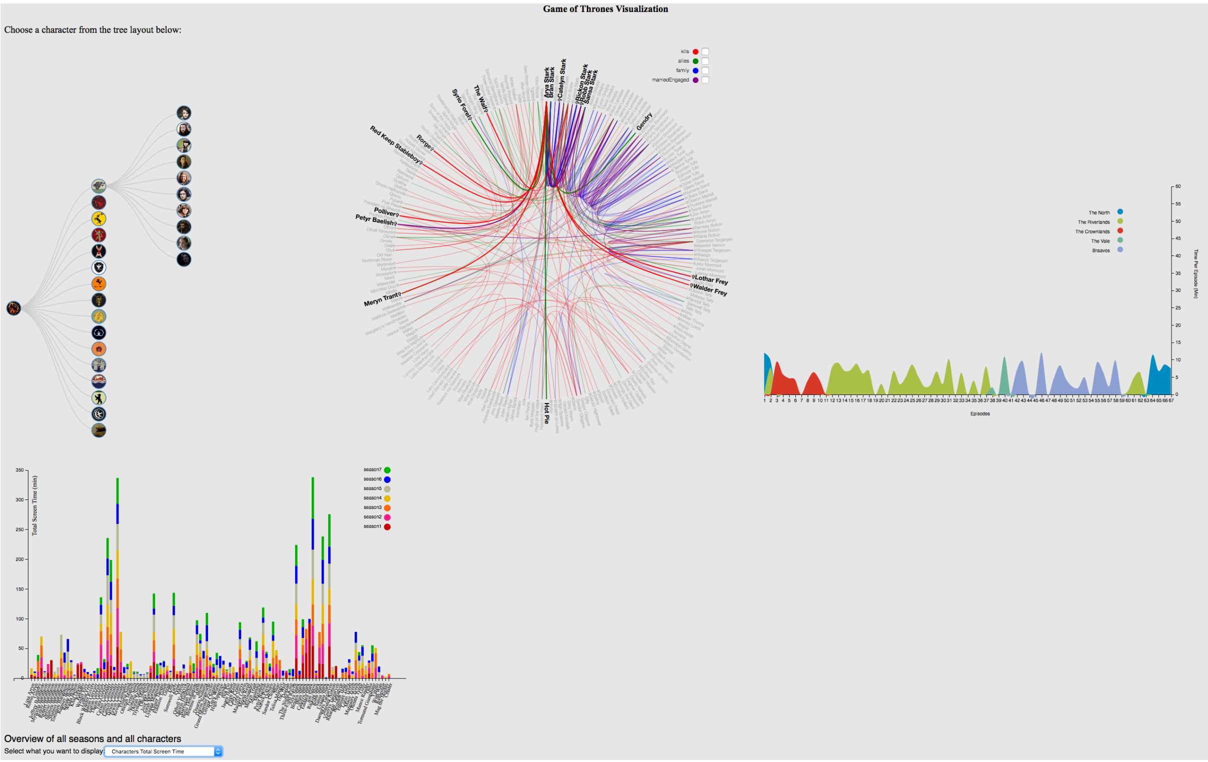
# Data Visualization Project: Game of Thrones

Name: Georgia Zarnomitrou, Student number:4724089
Name: Alexandros Stavroulakis, Student number: 4747933
Name: Sofia Tsoni, Student number: 4747941

December 21, 2017



# 1 Introduction

Game of Thrones is an immensely popular television show that has aired for seven seasons on HBO since 2011. The show is based on the epic fantasy novel series, "A Song of Fire and Ice", written by George R.R. Martin. The latest season, the seventh of eight in total, has regularly drawn in over 10 million viewers, with millions more watching the episodes illegally online across the world. Our decision to visualize data about this series is based on the complexity of the story, the show's popularity in social media and the great amount of fan theories it generates. During its seven seasons, more than 500 characters appear and the interactions between them are numerous. In this paper we will focus on data concerning most of the major characters in the show (specifically 103 characters), their screen-time accross all seven seasons, the fictional locations the characters travel accross and the relationships beteween the characters (four kinds of relationships are depicted: 1.killings, 2. familial relationships, 3.spousal relationships and 4.alliances).

Our project mainly focuses on the second variant of the assignment using D3.js [1] a javascript library for manipulating documents based on data. However due to the complexity of the structure of the data we found, we also put an effort in pre-processing them. Our visualization design consists of a combination of a collapse tree, a bar chart, a stacked bar chart, a relationship graph and a streamgraph. In the second section of this document we present the data selection and processing procedure we followed with our Game of Thrones data. Next we present the dimensions of our design space. Afer that follows a section where we show and offer justification for our visualization and design decisions. Then there is a small section mentioning the code used and the most important functions in it. Next there is a section dedicated to the observations we can make based on our visualization techniques. Lastly, we have a section mentioning our individual work.

# 2 Data selection and Pre-processing

Through various meticulous online searches we were able to gather a great amount of information about the show. This process proved somewhat of a challenge since we had to wade through numerous sites with incomplete or erroneous data. In addition, we had to sift through the data we found in order to keep those pertaining to the show and not the book (since there are many differences between the two). However in the end we were able to find multiple sources of data on different aspects of the series. Following is a reasoning of our selection of the data we visualize:

1. "Game of thrones" is the complex story of nine family houses trying to rise to power and claim the iron throne through alliances and battles. So, given the amount of characters portrayed in the series we decided to look for data that would help us to separate and categorize each character according to his/her house or the army they served. These data helped us construct the collapse tree, which was used as our navigation menu. Through a search in the web, we found the appropriate data for this purpose [4]. The dataset that we used from the previous source was the "characters-houses.json". However, we had to add some more parameters in the file, such as images, so that they could be displayed in the nodes of the tree.

2. The thing that separates this series from other shows is the fact that there is no clear protagonist in the story. The show follows the story of various characters and anyone of them can be killed unexpectedly. Therefore, in an effort to decipher which characters and which houses have the greatest importance in the series fans have gathered data on different characters screen time. So, we decided it was important to show the screen time afforded to these characters and how that has changed over the course of the show. For this purpose we gathered data from two sources [2, 3] and we used the file "ScreenTimePerHouse.csv" that contains the screen time of each house in the series per season and the file "ScreenTimePerSeason.csv" that contains the screen time of each character per season.

3. Another important aspect of the show are the alliances and relationships that exist between the various characters. We used the same source where we found data for the tree, and we used the dataset "characters.json" [5] which contained a lot of information about the relationships of the characters. However, the dataset was not complete, which means that for main characters some really important connections were missing, thus we had to add a great number of relationships between characters. All the improvements are stored in our "characters.json" file.

4. The show takes place in the medieval period in a fictional world made up by many locations. Throughout the seasons we follow our characters traveling across these locations so we decided it was important to depict the locations visited by a character in each episode. Thus we found a very useful dataset [6], which contains a lot of information, as well as locations, for every scene in the show. However, as we will mention in the following sections, we had to conduct some pre-processing steps in order to extract all the necessary data for our graph.

### 2.1 Pre-processing

The first great challenge we faced was finding the appropriate data for the information we wanted to visualize. Since Game of Thrones is extremely popular, lots of people have created datasets that represent the world of Game of Thrones. However, since we decided to create 4 different visualizations, we faced the problem that the available datasets were restricted and incomplete for our visualization purposes. Thus we had to pass through a pre-processing step in order to generate the data we needed to use in our visualization. The greatest effort was put in the pre-processing of the file needed for the StreamGraph.

The initial dataset, in which the StreamGraph was based, included information about every scene in every episode. Specifically, each entry in that Json file presented information about an episode and all the scenes were analyzed. In turn every scene contained information such as: the starting and ending time of the scene, which characters were involved in the scene and in which locations it took place. Since we decided to present the location of a character in each episode and his screen time in the StreamGraph, we had to write some programming code in order to extract these data from the initial dataset. This was necessary due to the fact that the information was presented in great hierarchical depth inside the Json file.

Furthermore, some pre-processing was required in all the other datasets we found. The majority of these datasets had limited data on information such as the relationships between characters. Thus we had to add a lot more data in the already existing dataset in order to present overall information in our visualizations.

Beside the addition of data, the other pre-processing steps were mostly implemented in Javascript. Entire sections of code needed for this pre-processing were added in the final file where all our visualizations are implemented. In addition, we used some of excell's features for the pre-processing of our csv files. Specifically, we used excell's summarization and sorting functions as well as its text to data feature. Lastly, we would like to note that since each graph makes use of more than one dataset, it was necessary to do the entire pre-processing keeping that in mind.

## 3 Task Description

In this section we will present a definition of the tasks suited to our data. For this purpose we follow the design space dimensions as they are mentioned in the paper by Schulz et al. [7].

### 3.1 The Goal

Firstly, we present why we pursued these specific visualization tasks, meaning what was the goal we were hoping to achieve. Our visualization tasks mostly had the goal of confirmatory analysis and in a lesser extent that of presentation. Specifically, the bar chart, the stream graph and the relationship graph's aim besides presentation of known data, is to test our assumptions about which characters or houses hold a key role in the series, which characters have moved most across the fictional world of the show, which character has the most alliances or killed the most people. However, the tree and relationship graphs are also used to describe and present in an easily understood and aesthetic way known data about the characters of this show.

### 3.2 Means

Next we present how our visualization task is achieved. Our visualizations were carried out in the three following ways: 1. Navigation: Our collapse tree graph serves as the navigator for our visualizations since you can browse the different houses that exist in this fictional world and search for a specific character. 2. Re-organisation: Both our relationship graph and our bar chart offer the user the opportunity to reduce the data presented by extracting certain information. In the case of our relationship graph this is done by the accompanying check boxes that let the user filter the kind of relationship they wish to see. In our bar chart this is offered by the drop down menu where a user can select to see an overview of all the data or a season by season transition. In the bar chart the user is also offered the opportunity to click on a single bar and extract information about a specific character. 3. Relation: The stacked bar charts and plain bar charts we present give the user the opportunity to make comparisons between the screen time of different characters or houses.

## 3.3 Characteristics

In this section we present what kind of data characteristics our visualization tasks seek. Two of our visualization tasks seek low-level data characteristics whereas the other two seek high-level data characteristics. Specifically, our bar charts and our collapse tree are used to derive simple observations about our characters or obtain the screen time that corresponds to that character.On the other hand our relationship graph and our stream graph offer a different type of information. The relationship graph offers correlations between our characters and our stream graph shows the complex combination of which character was at which location in a specific time and how long he is shown there.

## 3.4 Target

Next follows the question where in the data do our tasks operate, meaning which part of the data is the target of our visualization tasks. Our visualization techniques target different aspects of the data objects. Specifically, our bar charts and our stream graph target attribute relations since they link our characters to temporal values (screen time in bar charts and duration in a scene in the stream graph) and spatial values (fictional location where a character was found in a particular episode). However our relationship graph and our tree graph link our characters with one another thus here our target are structural relations. As far as the relationship graph is concerned this connection is the product of familial and spousal bonds or alliances and killings. In the tree graph our characters are linked o one anoher based on the house they belong in.

## 3.5 Cardinality

Here we consider the scope of our visualization tasks, meaning how many instances of the targeted data are considered in our visualization tasks. In our bar charts all instances of our data are considered but when you click on a specific bar only one (single) instance is taken into account. In our tree graph the visualization task takes into account multiple instances since we can click on a house and see a certain number of instances. In the tree graph we are also concerned with single instances since we can hover over a character and get data only about him. The relationship graph is concerned with all the instances when no specific character has been chosen through the tree graph or bar chart. But it also considers multiple instances since it shows the relationships between a group of people when a specific character has been chosen. Lastly our stream graph considers only a single instance since it depicts information about one character at a time.

## 3.6 Who and when

In this section we consider the order in which our tasks appear and the type of user they are addressing. As far as the order of our tasks is concerned first the user must pick a character from the collapse tree and that adjusts the information already displayed in the relationship graph and the bar chart so that it is character-specific. At that point the stream graph appears depicting information about the character the user has chosen. In addition we offer the user the freedom to pick a specific character from the bar chart (we are refering to tha bar chart shown when he selects the option "Characters per season" from the drop down menu) and that adjusts the information displayed by the relationship and stream graphs. Our visualization project mostly adresses the fans of the show who have difficulty keeping track of all the information and relationships about the characters due to the series complex nature. These visualizations could also be used by someone who has not seen the show but he has to have some knowledge about what it is about and interested in learning more.

# 4 Visualizations

As mentioned above in order to depict the data we collected we chose to use a collapse tree, a bar chart, a streamgraph and a relationship graph. At this point, we will present in more detail our decision making process and justify our final decisions on the visualization techniques we used.

## 4.1 Collapse Tree

We wanted to have a navigation element in our page, so that we will be able to find the data that we were looking for, for that purpose we decided to create a collapse tree. Because of the complex relationships among the characters, we realized that the most clear classification is based on the "House" they belong in. During the seven seasons there have been a lot of changes among the alliances and too many betrayals, so the best way to separate the characters was their race or house name, which from now on will be called "House" of the character. Since this feature seems to be hierarchical we chose to represent it using a tree. The choice of a tree as the navigation menu was decided because the characters are the main data for our Game of Thrones visualization and they are easily accessible through a hierarchical structure.

As it can be seen, the root node of the tree is a characteristic image of the whole series. The root node has 15 children, which are the most important Houses, and one more category named "Others" that contains the rest of the characters who do not belong in a House. Hovering over each node/image we can see the name of the House, by clicking at it, all the characters that belong to that House appear. Here we also have the hovering feature, but this time it provides more information. Specifically, a small window appears with some main data (Dead or Alive, Actor Name, Number of appearances etc) for every character, which are not presented in any other graph. The topology we chose for the tree is easily understandable and probably the most expected for the user, so that he/she will be able to follow the preferable path without confusion.

Furthermore we chose to use a images for our characters as a better way to find the preferable one, since most viewers of this series are more familiar with the looks of a character rather than his/her name. In this way it is quite easy for the user to find the character he seeks and by choosing it all the other graphs will be adjusted in order to present all the available information we possess about the specified character.

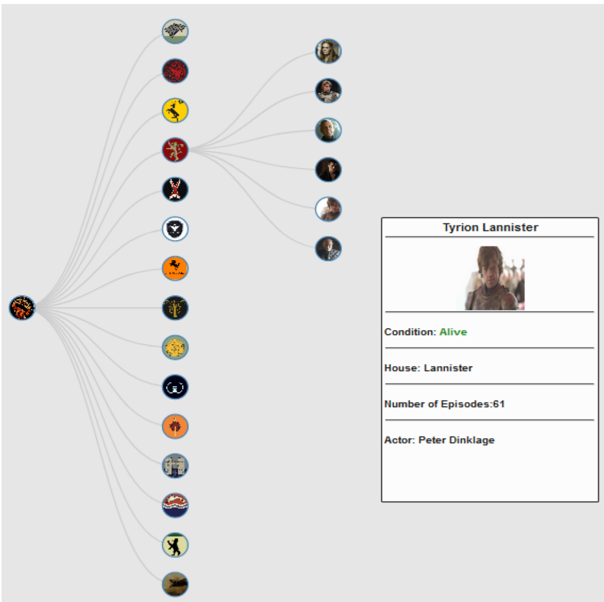



Figure 1: Collapse Tree as navigator menu,specifically when hovering on Tyrion Lannister

## 4.2 Bar Chart and Stacked bar chart

In order to present the screen time of the characters and the houses throughout the seven seasons we chose to use two representations for our two datasets (ScreenTimePerHouse.csv and Screen-

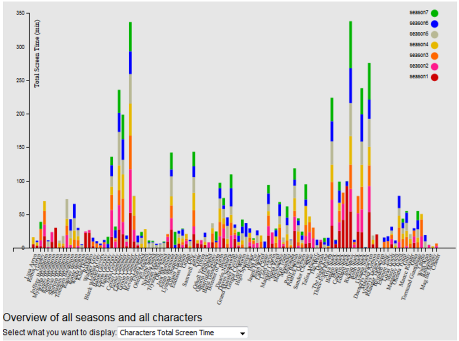



Figure 2: Stacked bar chart for total screen time of characters

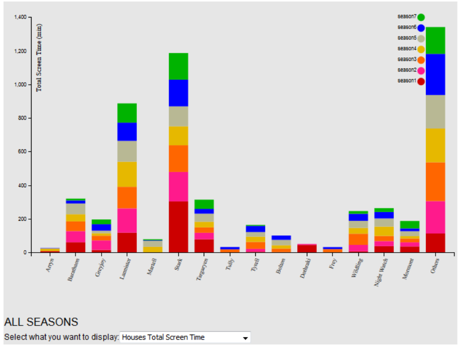



Figure 3: Stacked bar chart for total screen time of houses

TimePerSeason.csv). The user is given a drop down list from which he can choose one of four options:

1. Characters Total Screen Time
2. Characters Screen Time Per Season
3. Houses Total Screen Time
4. Houses Screen Time Per Season

The first visualization technique we chose was a stacked bar chart since we had to display one quantitative attribute (screen time) and two categorical attributes (character names or houses depending on the dataset used and seasons). At this point we would like to note that for the purposes of this visualization we treated the ordinal attribute seasons as if it was a categorical one. For our bar charts we chose to create two axes the first one is linear (Y-axis) and used for our quantitative value and the other is ordinal (X-axis) and used for our character names (house names in the case of the dataset ScreenTimePerHouse.csv), while we treated the ordinal attribute by visualizing it with the use of different colors. This decision was mainly taken due to the large number of characters in our dataset which would make the use of color to differentiate them an impossibility. This bar chart gives the user the opportunity to compare the total screen-time between characters and observe how the screen time of each individual season makes up the total screen time. In addition, it gives the user the opportunity to compare the screen time of different characters in different seasons. This is especially helpful in our case since new characters are introduced in the series each season.The two stacked bar charts displaying total screen time for the characters and houses are shown in figures 2 and 3 respectively.

The second representation was chosen due to the large number of characters displayed and the small screen time associated with some of them. This makes it very difficult for the user to distinguish the screen time of some characters in the stacked bar chart. For this reason we decided to create another representation where we show a bar chart using only the attributes character name (or house name) and screen-time. In this representation each season is displayed separately and we give the user the opportunity to click the button next season to see and compare characters screen times for each season separately. This change between seasons is displayed with the transition of each character bar. It is really important to note that in this bar chart the characters are grouped by the house they belong in (and sorted in alphabetical order of the houses). Each house is assigned a color chosen based on the houses emblem (these emblems are depicted in our collapse tree). The user is also given the opportunity to see a single character's screen time per season (again as a bar chart) if he clicks on that characters bar in the "Characters Screen Time Per Season" chart which is an option of the drop down menu. The same thing occurs when a user clicks on a house's bar in the "House Screen Time Per Season" option, meaning that the user can see the screen time of the specific house he clicked for each season. In figure 4 and 5 we display two such bar charts depicting all the characters and houses screen time for season 1.

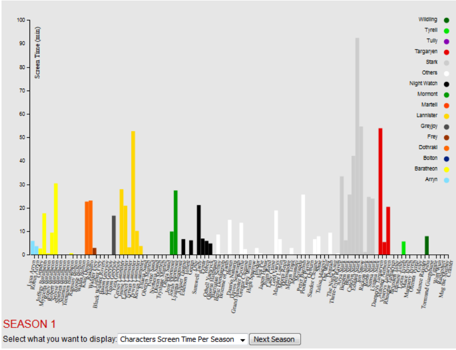



Figure 4: Screen time of all characters for season 1

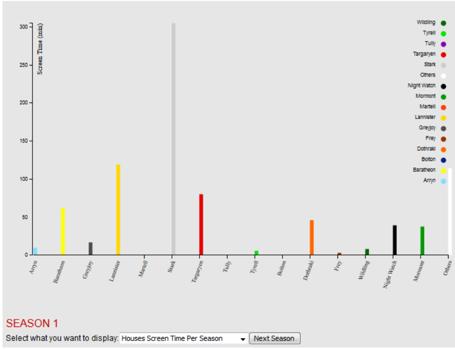



Figure 5: Screen time of all houses for season 1

### 4.3 Relationship Chart

Figure 6: Relationship Chart when clicking Arya Stark in the Tree, for all types of relations

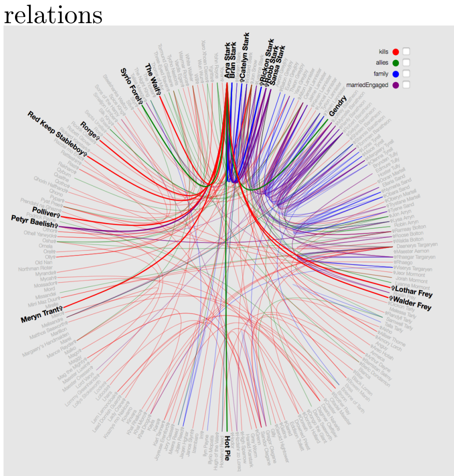
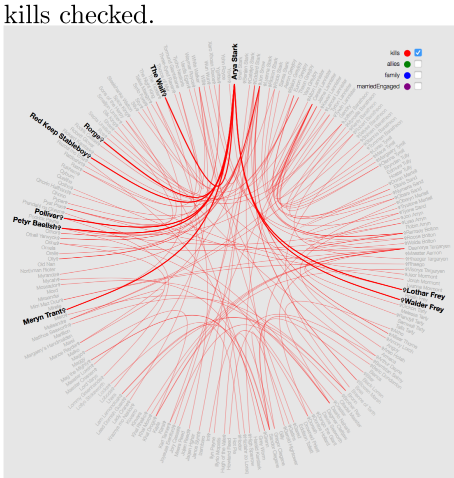



Figure 7: Relationship Chart when clicking Arya Stark in the Tree and checkbox of kills checked.




What made Game of Thrones popular, is its plot. In each episode we see many interactions between the characters and many times these interactions are conflicting. Children killing their fathers, allies becoming enemies, killed characters coming back to life and many more. We were searching for a way to represent some of these interactions and relationships between the characters, but because of the huge amount of data that we possessed, every visualization seemed to be unclear. What was clear is that the relationships should be presented as a network between all characters. Another fact that was clear was that the relationships should be filtered in a way.

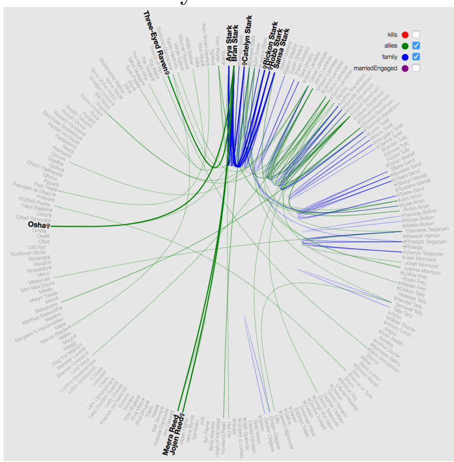
For all these reasons we chose a relationship graph, where characters will be the nodes. Nodes shouldn't overlap and thus we put them in a circular shape. The links between the nodes represent specific relationships between the characters. We believe that the most important relationships in the Game of Thrones are the family bonds, the killings (which means that one of the nodes has killed the other), when a marriage or engagement has taken place (which in GOT happens for some characters more that once) and a friendship bond (which is equivalent to allies in our case). In addition, we have added a small cross next to every dead character, which becomes darker when hovering over it.

Every relationship as mentioned above is presented by a color, which is relevant to its meaning. For example killings are represented by the red color, allies by green, family bonds by blue and engagement by purple. We wanted to get an idea of the links just by the color between the characters. Red is the color of fire and blood, so it is associated with energy, war, danger etc and that is why i was chosen for the killings, green is the color of nature. It symbolizes growth, harmony, freshness, and fertility. Green has strong emotional correspondence with safety, so we decided to symbolize allies with that. Blue is the color of the sky and sea. It is often associated with depth and stability. It symbolizes trust, loyalty, wisdom, confidence, intelligence, faith, truth and it seemed the most appropriate for the family, because it is the most stable bond even in GOT.

Purple combines the stability of blue and the energy of red. Purple is associated with royalty. It symbolizes power, nobility, luxury and we thought about visualizing the engagement or marriage with that.

The challenging part, in the whole procedure, was to avoid edge crossing, which unfortunately was not possible. We adjusted the nodes in an specific order, so that all the family bonds will be next to each other (figure 8) and the killings as spread as possible (figure: 7). To achieve that we sorted the data based on the House they belong. In addition to add we add checkboxes, so the user is able to choose just one kind of links to visualize between the characters. Last but not least, in order to make the connections more clear we added the hover feature, one can hover to the character he/she likes and the nodes connected with him/her will become bold as long as the edges between them.

Figure 8: Relationship Chart when clicking Arya Stark in the Tree and checkboxed allies and family are checked.




## 4.4 Stream Graph

An exciting characteristic of Game of Thrones is that they have created a whole world with its own map. Each House in the Game of Thrones has conquered a different part of this world, but this doesn't restrict the characters from traveling in many different places through the seasons. What we wanted when we decided our subject, was to create a map in d3js, where we would present the different locations that every character visits in each episode. However since the world map of Game of Thrones is a fictional map, we weren't able to design it in d3, because it would have required too much time. For that reason we had to search for the second best solution in order to visualize the locations. What we all agreed on is that we wanted to visualize for each character every location that he visits and how much time he/she spends there throughout the whole show. That meant that we had three attributes to visualize, one categorical (locations), one ordered (episodes 1-67) and one quantitative (the time). So Stream Graph seemed to be the best choice after the map.

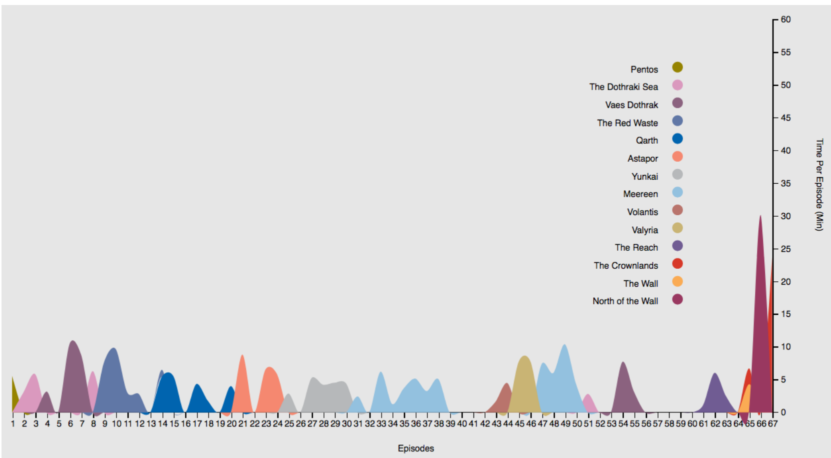



Figure 9: Locations for Jorah Mormont through the whole show

More precisely in the Stream Graph we present the episodes that a specific character plays in, his screen time in every episode and of course the locations where the character has appeared at. Of course it is also important to mention that a character can visit more than one location in the same episode.

Another point that we have to mention is the different colors for every location. Since there is no "specific" color for any location in the map, we decide to color all the locations arbitrarily.

However we were careful to choose colors that are easily distinguishable when we present the locations of a character.

Last but not least we have also implemented a hover function in the Graph. When the user wants to concentrate on a specific location, he can hover on that and see in which episodes the character is appearing in that location, while all the other layers with the locations disappear temporarily.

# 5 Code

Our project consists of the main code which is in the file **project.html** and is written in Javascript. Also we have a css file with all the needed css code to improve the appearance of our page. We have 5 datasets in json format and 2 in csv. Also we have downloaded the code for a tooltip which consists of two files **jquery.tipsy.js** and **tipsy.css**. For the requirements of our graphs we needed both version 3 and version 4 of the d3, so we had to add the file with version 4 manually that is why we have a **d3v4.js** file. The files for the tooltip and the 4th version of the d3 are not written by us but are existing libraries we used. All the other files contain code written by us.

As far as the main file with our code is concerned, we have all the functions which create the graphs together. The basic functions are:

1. **initializeTree();** With this function we create the tree. This method is called only once at the beginning of the code.
2. **initializeRG();** This function contains the required code to create the Relationship Graph, which presents all the characters and all the connections between them.
3. **selectRepresentation();** This function is used to see which bar chart visualization the user wants to see. Using this function the main function that creates the axis and bar charts is called. The first time this function is called we have elected it should display a Stacked Bar Chart with the Overview of the total screen time of all major characters in all the seasons.
4. **drawStreamgraph(selectedPerson);** Using this function the Stream Graph is generated, but this function is called under certain conditions. Specifically, this function is called only when a character is picked either through the tree or the bar chart (by clicking) and this is because the function needs the selectedPerson variable as an argument .

# 6 Results and Further Improvements

This visualization project presents to the user the main characteristics of the show in order to understand the plot and the reason why this show has gained so much popularity. Since this show is too complicated and it is extremely difficult to remember every detail, we have tried to represent, through our visualizations, the most important aspects of it. With the combination of all the visualizations we have created, we allow the user through the use of a visualized navigation menu to see the relationships of all the characters in the show (this is the most complicated part of the show) and other interesting facts about each character individually e.g their screen time on the show or the locations they visited.

## 6.1 Observations

Since a very high number of characters appear on the series, we decided to focus and present the most important ones. We made this deliberate choice in contrast to the complicated nature of the show in order to make it more easily understood by every user even the ones that are not passionate fans but casually interested. Looking at our visualizations a user can make several interesting observations about the characters of this show and their importance. Specifically, we can determine a character's or house's importance by the total screen time afforded to them. For example we can see that Jon Snow, Tyrion Lannister and Daenerys Targaryen are the most important characters in the show and could be thought of as the protagonists of the story. As far as the houses are concerned based on their total screen time the most important ones seem to be House Stark, House Lannister, House Baratheon and House Targaryen (for this question we do not take into account the group others since it contains characters with minor roles). What is more we can see a connection between the numer of members belonging in a House and that Houses importance. Specifically, Houses Lannister, Stark and Baratheon are the ones with the most members ha number is 6, 10 and 9 respectively. However House Targaryen is an obvious exception since with only 4 members holds the fourth highest total screen time.

Another observation we can make by looking at these visualizations is that with the exception of Tyrion Lannister and Sansa Stark the characters with the highest screen time are the ones with more kills in the relationship graph. It is also apparent that the members of the most important houses (indicated by their total screen time) are the ones for which more relationships appear in the relationship graph. By observing the stream graph we can also see that Daenerys Targaryen and Jorah Mormont are the two characters that visited the most locations (fourteen in total). These are only a few of the connections and observations someone can make through our visualization of these aspects of Game of Thrones.

### 6.2 Further Improvements

We really enjoyed our time working with this project, not only because we liked the subject, but also because there were so many interesting things that we were learning each day. Unfortunately due to the lack of time, we focused on some things and neglected others. If we had more time we would have dealt more with the visual part of the whole page, for example work more with the css and maybe change the fact that all the graphs are in one page.

Another thing we would have liked to add, is a more complex searching mechanism. We faced some difficulties in making a clearer visualization of what we wanted because of the huge amount of data we found. What we chose to do is reduce the characters and the relationships we visualized in order to make everything easier. What we would like to do in the future, is provide some kind of search among the data and visualize only those specified by the user.

## 7 Individual Work

Generally through the design and the decision-making phase of this project we all worked together. When we knew what we wanted to do and how we would approach it, we decided that each of us should work individually on a specific task and we would help each other as it became necessary. However, we met at least once per week in order to keep track of what the others were doing, and also to solve problems that may have appeared. All of us wanted to deal with every aspect of the project and that is why each one created a different visualization. In this way everyone dealt both with pre-processing of data and coding in D3. Lastly, we would like to note that although each of us wrote a specific part of the report the content was approved and decided by all of us.

### 7.1 Georgia Zarnomitrou

The individual task I was assigned had to do with the implementation of the bar charts and the stacked bar charts. As we mentioned above all the decisions about the visualization tasks we implemented and the data we used were group decisions. And though a large part of my time was used to my assigned task I also had some input on the implementation of the stream graph. In addition, I participated in the process of combining all the visualization tasks together and I had some design input in the final result. As far as the report is concerned I wrote the part pertaining to the bar charts, the task description, the observations and part of the section "Data and pre-processing" specifically the part that is relative to the data used to implement the bar charts. Lastly, I would like to mention I did some pre-processing of the data in the csv files by adding or deriving new data.

### 7.2 Sofia Tsoni

The main part that I had to implement was the Relationship Graph. For the Relationship Graph I had to adjust the existing data and select which characters will be included in it (GOT has more than 500 characters). Also I participated in the creation of the StreamGraph, were most of the work was about the correct structure of the data. We knew what kind of information we wanted to visualize, but the data needed a lot of work. Last but not least I collected all the codes and linked the graphs between them in order to have a coherent form. As far as the report is concerned

I wrote part of the introduction, my part about the datasets, the relationship graph, the code and the further improvements. Last but not least, I did the screen recording explaining our project.

### 7.3 Alexandros Stavroulakis

The first responsibility I took in the project was the building of our general navigation menu. Thus I created the Collapse tree in order to serve this purpose. In the tree we wanted to present as much information as possible for each character separately, so I had to collect information from all the dataset to present it at an information window for each character. After that I started working on the Streamgraph while Georgia and Sofia were finishing their visualizations. Since the creation of StreamGraph for our specific case required a lot of effort, we worked all 3 of us in order to finish it. As far as the report is concerned, I wrote about the pre-processing steps, the detail information about the collapse tree and StreamGraph. Another important task that I took responsibility for, was the integration of both versions of D3 in our code. At the beginning of the project there was a misunderstood and we started working in both v3 and v4 versions of d3. Since it was difficult to adjust our codes in order to work with one version, we decided to use both.

## Bibliography


1. Data-Driven Documents: https://d3js.org/
2. https://github.com/Preetish/GoT_screen_time/tree/master/datasets
3. https://www.shortlist.com/entertainment/tv/game-of-thrones-characters-screen-time-ranked-seas 320419
4. https://github.com/jeffreylancaster/game-of-thrones/blob/master/data/characters-houses.json
5. https://github.com/jeffreylancaster/game-of-thrones/blob/master/data/characters.json
6. https://github.com/jeffreylancaster/game-of-thrones/blob/master/data/episodes.json
7. Hans-Jörg Schulz, Thomas Nocke, Magnus Heitzler, and Heidrun Schumann. 2013. A Design Space of Visualization Tasks. IEEE Transactions on Visualization and Computer Graphics 19, 12 (December 2013), 2366-2375. DOI=http://dx.doi.org/10.1109/TVCG.2013.120