The PokéChart

Carl Melin (carme003), Edvin Nordin (edvno177), Philip Lam (phila527)

Abstract—
To be able to find the strongest pokémons you need to know the power-level for each one. The PokéChart is a tool to aid trainers by showing and comparing the stat values for every pokémon. When dealing with multivariable data it can become overwhelming. However, by using different visualisation techniques the user gets a chance to comprehend the data without the feeling of being overwhelmed. The PokéChart project was done to help understand the different techniques used in a information visualization. Using scatter plots and a parallel coordinate plot to visualize pokémon data from Kaggle in order to see connections, differences as well as the strengths and weaknesses of all pokémon. The resulting visualization displays the stats of 980 different pokémon, enabling comparison between them all in the two different plots. By using a filter function the user can chose the relevant pokémons to compare, reducing the cluttering, and to find connections on their own.

1 Introduction

The aim for this project is to create a interactive visualization to ease the search for stronger pokémon by comparing them to one another with respect to stat values. In order to make this possible techniques introduced in the course information visualization will be used.

2 Background and Related Work

The world of Pokémon is a vast one filled with many fantasy creatures called pokémon. Trainers around this world both catch and train these pokémon in order to become the greatest trainer. Pokémon battle each other by using abilities and stats with various strengths and weaknesses. In order to become the greatest pokémon trainer, they would want to know which pokémon is the strongest.

Since the project is to visualize the stats of pokémon, analysis of other pokémon stat visualization where done. As most of the visualizations are non-intractable they tend to be overwhelming and confusing. To combat this problem, this project would need filtering options as well as brushing and linking. This way the users can visualize and compare many pokémon but at the same time will not become confused and lost[2].

As the pokémon genre already have specific colors for specific groups of pokémons, these colors where chosen to be implemented in order to further help the users understand the visualization as these colors are already determined and understood.

3 Data

The dataset that was chosen was the *Complete Pokemon Dataset*[1] made by Mario Tormo Romero and was taken from the website kaggle. It contains information about all the pokémon from generation 1 up to generation 8 which is 980 different pokémon. The database was downloaded as a *Comma-separated values (CSV)* file, which makes it easy to load into the programming file. Since the database contains a lot of information only some of the attributes was actually used in the project.

The data that was used from the database was the name, pokédex entry number, generation number, typing and the stats of each pokémon. The stats contains the HP (hit points), attack, special attack, defense, special defense and speed.

The images were first planned to be gathered from another database found on Kaggle. However, this data set were not entirely complete or lacked data which were needed and could not be used properly. Instead, the images were collected from pokémons own webiste [2] where the pokémon images are connected through the same URL with the difference being their pokédex number. With this URL the images could easily be linked together with the stats in the programming files.

4 Method

To start, a rough plan of the design and type of visualization was sketched to get a fundamental idea what the final product could result in. This sketch includes a scatter plot with one dynamic and one static axis. This way, the user can compare the total amount of stats with a specific stat chosen in the dynamic axis. As the scatter plot displays all the pokémon according to their stats the user can hover over any point in order to see what pokémon that point represents as well as a bar chart of that pokémons stats.

Using the sketch, the scatter plot was implemented with a slight change. Instead of having one dynamic axis, two static axes was implemented along side six different tabs. These tabs would all have a different scatter plot with a different stat attribute, giving the impression of the static axis being dynamic.

Since a lot of information would be hidden and prevent an overview perspective by only using the scatter plot and hover function, a parallel coordinate plot was added. The parallel coordinate plot gives the user a different view to compare the stats of the selected pokémon. It can also handle larger amount of data as long as Additionally, the two plots were linked together via a highlighting function so the user can easily see the connection between the two plots when hovering over either a line or a point. Using linking in this visualization improves the understanding of the plots, making the data easier to interpret[1]. Utilising the hover function, images were added along side the name and pokédex entry number in order to help the user more clearly identify the selected pokémon.

As the data set contains a lot of pokémon to plot and compare, filtering options was added to help reduce the amount of data that is being visualized. Since the data set already had boolean columns, for example is-legendary which indicates whether the pokémon is legendary or not, the choice of what type of filters where clear. The other filtering options used where the typing, generation, sub legendary, mythical and normal. These filtering options where chosen

- *All authors are with Linköping University, Sweden, e-mails: carme003@student.liu.se, edvno177@student.liu.se, phila527@student.liu.se.*

[1] Kaggle: Data set with raw pokémon information, Mario Tormo Romero, accessed March 8, 2022, https://www.kaggle.com/mariotormo/complete-pokemon-dataset-updated

[2] Pokémon: Official pokémon home website, The Pokémon Company, accessed March 7, 2022, https://www.pokemon.com/us/pokedex/

due to being the more common way to filter through pokémon, be it either in the games, the pokédex or other visualizations.

5 Implementation

The implementation was done in the programming language python with the IDE *Pycharm*. The version handling tool *Github* was used in order to work together more easily on the project. Bokeh [3] and Pandas [4] are the two libraries that was used in the project. Bokeh creates html and JavaScript files that can be view in a web browser. Pandas data analysis and manipulation tool was used to import the dataset from a CSV file. The imported dataset was turned into a *ColumnDataSource(CDS)*, which is what Bokeh pulls data from to make it into different types of visualizations.

To make full use of Bokeh, instead of using multiple small *CDS*, only one was used and expanded on in the code. This makes the linking automatic between the scatter plots and the parallel coordinate plot. All scatter plots are always rendered but only when clicking on its corresponding tab is it shown and the other scatter plots are hidden. The hover function was styled using custom *CSS* code and uses attributes in the dataset to get the URL to the image, stats, name and pokédex number. Depending on the first typing of the pokémon the color of the points and lines are set with hex color code.

To filter the dataset, Bokehs class *CDSview* was used to only show a subset of the dataset. When a filter button is selected the program checks which ones are active and changes the CDSview to only show the pokémon fitting the critera. To be able to use the filtering buttons the program must be run on a Bokeh Server [5] which enables the use of JavaScript on the webpage.

6 Results

The result of the project can be seen in figure 1. The project can be divided into four different sections; the six scatter plots, the parallel coordinates plots, the hover function and the filtering options. The scatter plots and parallel coordinates plot are linked together so they will show the same information but from different perspectives. By hovering over either a line or a point the other will be highlighted as well.

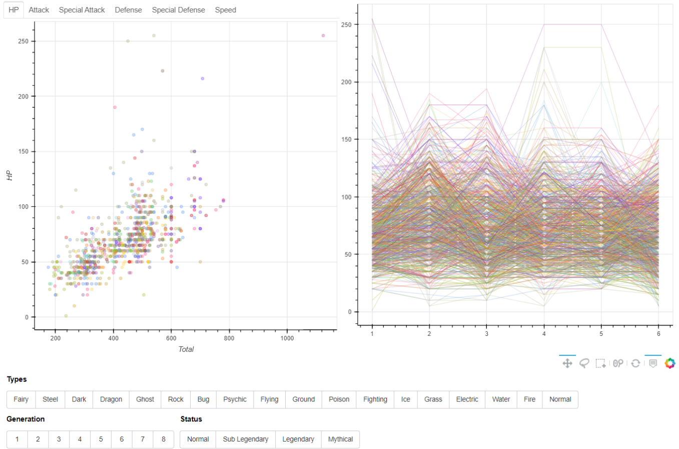

Fig. 1. The final result of the visualization.

The scatter plot visualizes all the pokémon with their stats total values compared to one single stat, in this case being HP and can be seen in figure 2.

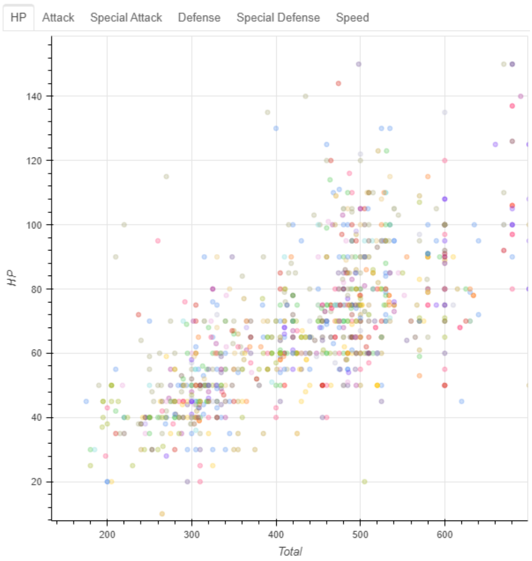

Fig. 2. Scatter plot of all the pokémon comparing total stats to HP

Above the scatter plot are six different tabs that each contain a unique plot for a different stat. The user can click on these tabs to display that specific scatter plot. As the X-axis is will display the stats total values the points in the plots are only going to change in the Y-axis.

The parallel coordinates plot shows the filtered pokémons with each stat compared to each other and can be seen in figure 3.

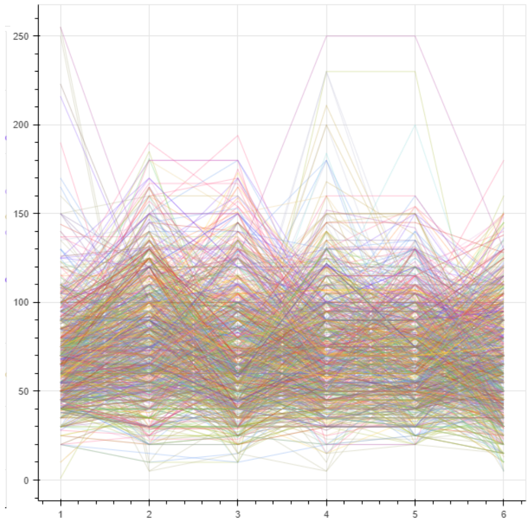

Fig. 3. Parallel coordinates plot comparing all stats at the same time.

When a pokémon is selected it will be highlighted in both the scatter plot and the parallel coordinates plot. In the parallel coordinaters plot the selected pokémon can be compared to other pokémons based on stats. The difference between the scatter plot and the parallel coordinates plot the ability to compare all the stats of a pokémon

[3] Bokeh: Python library for interactive visualization, Bokeh Development Team, accessed March 7, 2022, https://bokeh.pydata.org/en/latest/

[4] Pandas-dev/pandas: Pandas, The pandas development team, accessed March 10, 2022, https://pandas.pydata.org/docs/

[5] Running a Bokeh server, Bokeh Development Team, accessed March 10, 2022, https://docs.bokeh.org/en/latest/docs/user$_guide/server.html$

instead of a single stat with the stat total values.

The individual images are seen either when hovering over a point in the scatter plot or a line in the parallel coordinate plot. It contains an image, the name of the pokémon as well as a corresponding bar chart displaying the stats which can be seen in figure 4.

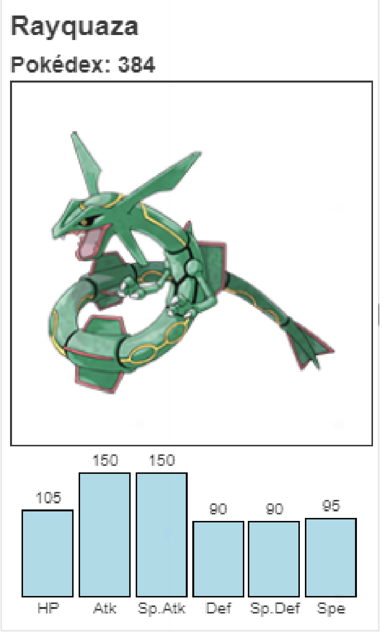

Fig. 4. Individual image with corresponding bar chart of their stats.

These images helps the user to instantly identify the pokémon in the scatter plot and parallel coordinates plot since without these, the user would have to use a separate tool to know which pokémon it is. Here the user can also get the exact numbers of the pokémons stat values presented as bars for each stat that uses height to show relative difference, which in turn describes the strengths and weaknesses of the pokémon.

The filtering options exist to help the user reduce the amount of pokémons that are visible on the screen and can be seen in figure 5.

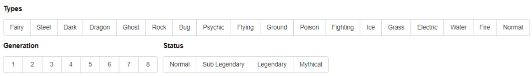

Fig. 5. Filtering options for the typings, generations and status of pokémon.

As mentioned earlier, the filtering options are a key feature to reduce the points and lines displayed in the plots. Without the filtering options, there would be to much information presented and the plots would become to cluttered to be of any use. As a user selects a filter, only the pokémons with the selected attributes will appear in the plots. Altering the filter by selecting additional attributes will add the pokémon with that attribute along side the prior ones.

7 Conclusions and Future Work

In conclusion, the project result did reach our goals in what we wanted to create as well as what we wanted to learn. It is easy to hide information in the hover pop-up function compared to showing as much as possible without making the visualization feel crowded. As the focus in this project was to visualize the stats and not to become another ”wiki”, we had to rethink parts to maintain the visualization aspect.

Noteworthy findings of this visualization was that many of same type have a similar pattern which are visible in the parallel coordinate plot. One example of this finding is that the fighting type pokémon all have similar pattern. The majority of fighting pokémon have an increase in attack and a decrease in special attack.

The result we got is what we sought out to accomplish both planning and programming wise. However, as we got started later than planned we feel the project got a bit rushed and that some corners were cut. The parallel coordinate plot for example could have more functionality where the axes could be moved in order to compare different axes with each other. We thought to reuse the filtering for each typing as a legend for better understanding of the chosen colors. However, as we got to the filtering implementing part we realized that the colors would not appear despite our efforts. This issue made us loose both the legend part as well as a way to create a more intuitive and colorful visualization. Another way to improve this project is to add a way to see all the pokémon pop-up view when the lines or points in the plots are located in the exact same place. Right now the images will stack upon each other creating a long column of information and too many of these makes the images disappear outside the screen. One implementation that fixes this is to add a function that displays all the images around the points that are stacked upon each other. This way, the user can see how many points are in the stack as well as the extra information such as the image, name and stats values.

References

[1] A. Kerren. 4. interaction techniques. In *Information Visualization*, pages 181–186. Linköpings Univeritet, 2022.

[2] R. Kosara and H. Hauser. An interaction view on information visualization. *Eurographics Digital Library*, 2003.

8 Contributions from Each Project Member

Edvin Nordin: My main role in this project was to handle the interactive portion of our visualization. I added the different tools usable on the plots: hovering, zooming, aswell as the brushing with box selection and lasso selection tools. Using Bokeh meant I did not have to implement the linking as this was done automatically. The filtering was also my responsibility. I created the buttons that changes the which pokémon are visible in the plots. At the beginning of the project my time was spent looking up different techniques and creating the parallel coordinates plot. I also helped the others in different areas when it was needed.

Carl Melin: My primary role in this project was to handle the data set that was used in our visualization. This role was given after I found the pokémon data set from Kaggle and the decision for this specific subject. My tasks were to import the data set by utilising the CSV file that Kaggle provided which was done by using the pandas and bokeh libraries. Since the data set containing images for all pokémon were not up to our standards we collectively search for alternatives and finally found the images on pokémons own website. I was also tasked with gathering the images from the website. Additionally I implemented the CSS for the layout in the hovering function. Lastly, it was my task to implement the colors both for the scatter plot as well as the filters. However, I was unfortunately only successful on the scatter plot and not the filters.

Philip Lam: My main role in this project was to structure the visualisation. This stared when we created the rough sketch for our design. I was assigned the task to create the scatter plots as well as the tabs to seamlessly swap between the plots. We had to reconsider the layout of our design to not hide away any information or options. I decided to place all the different tabs and buttons so they are visible by default in order to give a better overview.