VI Project

Bruno Bastos 93302
DETI
Universidade de Aveiro

Leandro Silva 93446
DETI
Universidade de Aveiro

***Abstract*—Games Data visualization it's a hot topic. This report shows different approaches to Pokemon Data visualization providing interactive and useful data representations, that allow the user to not only create new strategies and obtain better results competitively but also explore interesting facts and curiosities about Pokemon.**

***Index Terms*—Data Visualization, Pokemon Dataset, Bar Chart, Graph, Scatter Plot, Radar Chart, Heat Map, Line Chart**

I. Motivation and Objectives

A topic that is not so widely approached is the visual representation of data from games. Not only there are parts of the data that are handmade by the company or the players, meaning that it reflects their choices and ideas, but there's also some data that can be used for statistics and data analysis. Opting for the representation of data from games, gives us more room to explore different types of representations and allows us to see their benefits from a player/user perspective. The representations should provide better insights of the game for the different types of players. It can be used to assist the player improve its skills or performance during the game play or/and can help other types of users who like to obtain general knowledge and curiosities of their favorite game. The main focus of the chosen representations is to be able to fulfil those two necessities. After all, games are a product and a product needs to have some way to keep their clients excited and integrated. The visual representations of the information can help the user to embrace the project.

The game that was chosen by us, was Pokemon. An old franchise with a huge community around the games, that are focused on strategic battles using different Pokemon with unique stats and types. The data is widely available and can be used to help the players develop their strategies and be able to perform much better, achieving the results with less effort and with more fun. It also helps the player get to know the universe of the games and be excited about next products.

II. Users and Questions

Depending on the game and on the knowledge about that game, there can be different types of users/players that have different objectives. Some want to be better at the game, others just want to learn and have fun playing it. To better understand the players' needs, it is first important to understand the game. On a general note, in a Pokemon battle, two trainers use a team of six Pokemon where only one can be on the field at the same time. Pokemon attack using moves that can deal damage that is calculated according to the stats of each Pokemon in the field. The stats can be Attack, Defense, Special Attack, Special Defense, Speed and Health. Also, a Pokemon can have one or two types. Types represent the weaknesses of a Pokemon, increasing the damage it takes from a type that is strong against it.

The representations that we aim to obtain, are intended to fulfil the needs for the two types of players described previously.

A. Competitive Player

The first one, that wants to better understand the game mechanics in order to outperform and win against other players. Its goal is to be competitive and use the best strategies to win more battles. The main focus, according to the game, should be on choosing the best team of six Pokemon to have as less weaknesses as possible when comparing to the opponent's team. This leads us to several questions that this type of player/user might want to be answered by our representations.

There is, however, an important aspect when researching for the optimal strategy. If the player knows beforehand the opponent's team, it can choose a strategy that will increase its probability of winning against that specific team, thus the set of questions that it needs to be answered will differ slightly from when it is not possible to obtain that information before the battle.

The player should start by seeking information about a specific Pokemon of the opponent's team. Leading to the question:

How can it find and see the information about a specific Pokemon? Information can be its stats, types and respective weaknesses.

Supposing that after viewing the information, the player finds a strategy using an advantage obtained from that information, this strategy will generally consist in choosing a Pokemon that has either good stats that exploit the enemy weaknesses or a type that is strong against the enemy. So the questions should be:

How can the user find the most suitable Pokemon based on the stats and/or types?

However, there can be multiple choices for that criteria. Many options have already been eliminated with the previous question, but one still remains.

How can the user compare information about 2 Pokemon in order to decide which one fits its team better?

Since the player team can consist of a maximum of 6 Pokemon, it will most likely repeat this process a few times. After its full is fully decided it should ask:

How can it compare its team with the already known enemy team?

The previous questions were made assuming that the user has access to some information beforehand. However, this will not always be the case and the user needs a way to predict what the opponent's team consists of. There is a way of knowing if the picked is good against a specif opponent, but it is possible to pick that is good in general. So the question rises:

How can the user understand and see which Pokemon or set of Pokemon are more predominant and more likely to be chosen by an unknown opponent?

B. Casual Player

As previously mentioned, there are some other players who do not care about being competitive and just want to learn more about the games and this universe. Although some questions can be similar to the ones of the competitive player, there are others which are aimed towards a more curious type of user and are not really need in order to perform good in battles. Usually, a more casual player might not be interested in buying every single game of the franchise. The difference in games comes from the generation that they are representing. Each generation introduces not only a new map but also a different set of new Pokemon. Since, the dataset, that will be mentioned afterwards, only contains information about the Pokemon and not the games in specific, the information visualized can only show interesting aspects that distinguish Pokemon across generations.

How can the user understand and see the differences between Pokemon games and the characteristics of those Pokemon across the games that they are inserted into?

The user can also be willing to search for interesting facts about the tendencies of the company behind the games. Things like types, stats and their correlation can sometimes tell much about how the company perceives the game and how they will most likely behave in the future.

How can the user understand the relation between Pokemon and their types? How can stats be related with their respective types? And how can stats be related with themselves?

These should be the principal questions that the users want to be answered by the representation of data. There's a bit more focus on the Competitive user because the data allows to be explore this area and there is room for more than statistical representation of data.

III. Dataset

The dataset [1] used in order to populate the representations has the information about every Pokemon. There is however, some information that is not going to be used in the representations, because we believe that it is as important or less suitable for our users. So, the dataset was altered, some features have been removed. The final dataset consists of:

- **Name**: the English name of the Pokemon
- **Pokedex Number**: the national dex Pokemon number, essentially the id of the Pokemon
- **Generation**: a number corresponding to the game where the Pokemon was released
- **Stats**: some features that represent how apt a Pokemon is to given task (attack, defense, special attack, special defense, speed, health)
- **Pokemon Types**: a Pokemon can have 1 or 2 types.
- **Type Relations**: some types are strong against others. The dataset has the relation between types and how much extra damage they do to each other.

IV. Visualization Solution

This section presents the visualization solutions that answer the questions that the user needs, as well as the process of reaching those final solutions.

A. Low fidelity prototype and user feedback

The earlier versions of the prototype consisted of a Ven Diagram with the Pokemon spread across the types, a Chord Diagram that had the dominance relationship between types, a Radar Chart that could be used to compare two Pokemon and two Heat maps that showed the relation between types and stats.

User feedback allowed us to rethink some of the representations and lead us to problems related to said visualizations.

Initially, in order to answer the question "How can the user understand the relation between Pokemon and their types?" and partially the question "How can the user find the most suitable Pokemon based on the stats and/or types?" our solution consisted of a Ven Diagram where each type is represented by in a circular shape and the relation between types was obtained in the interception between the shapes of both types. However, before the Visualization solution was implemented it was possible to predict that it wouldn't be viable because there are too many types and relations between them. The user would most likely have a hard time understanding the type relations.

The Ven diagram, however, focused on understanding the relation between types by comparing the number of Pokemon between one type and the other, and since Pokemon can have 2 types, it would also reflect that relation when the shapes intercepted. Furthermore, there is another way of seeing the relation between types. As it was previously mentioned, types can be strong or weak against other types. At the beginning, the idea was to use a Chord Diagram to represent those advantages between types. Chord diagrams are displayed in a circumference which is divided in multiple parts, where each part has a size that is respective to some measure. One of the problems was to define this measure, as it could the number of types that a type counters, or the number of types that counter it. It was not obvious, and also it could be a bit confusing for the user to understand a basic relation. There is also another problem, because if a Pokemon had two types, the user would need to see the weaknesses for each of the types and try to understand the resulting weaknesses.

Since at the time, it was only a low fidelity prototype, it was hard to see further problems, so those were the main issues of our initial data representation.

B. Functional Prototype

The functional Prototype was then developed taking into consideration the user feedback from previous stages of development.

The main problems were mostly due to the high difficulty to understand the Ven Diagram because of the numerous relations between types. The Chord Diagram lacked consistency and wasn't very useful when a Pokemon had to types. Aside from the problems, there were questions that weren't well formulated in the initial versions of the prototype and were now reconstructed to be the final ones, which were described in the previous sections. This led to the need of newer forms of visualization that were added to the prototype.

"How can it find and see the information about a specific Pokemon? Information can be its stats, types and respective weaknesses." To answer this question the representation used a Bar chart with the stats of the Pokemon and had a table that showed how much damage the Pokemon would take to a specific type. This table representation is much simpler to look at than the Chord Diagram and solves the problem of when a Pokemon has two types. Figure 1 is an example of the representations can look like.

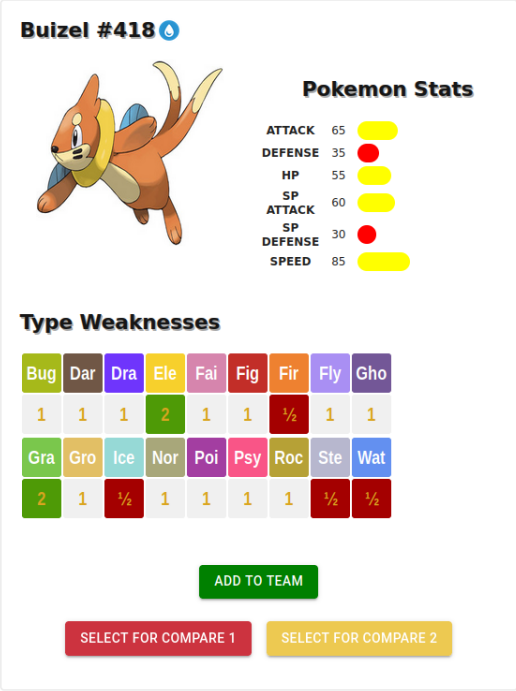

Fig. 1. The Pokemon Information Card.

Before talking about other representations, it is important to understand the available filters. The filters allow to select which generations of Pokemon are represented in the data, and also allow to filter the types of Pokemon selected. Related to the types, there is a select that allows the user to set the types to be inclusive or exclusive, which means that the Pokemon need to have all the types selected or can have at least one of the types selected, respectively. The filters are global, meaning that they impact multiple representations at the same time. Filters also allow to search a Pokemon by the name. Figure 2 left side shows how the filters look like.

"How can the user find the most suitable Pokemon based on the stats and/or types?". This question required two different visualizations, one to find the Pokemon based on types and the other to find the Pokemon based on the stats. In order to see them using types we used a Graph where the nodes represented one or two types. The single types would then be linked to the two type nodes that had that type. For example, there would be a node with the type "Fire" and one with the type "Dragon", then if there is a Pokemon that had both types, a new node would be created designating all the Pokemon with "Dragon" and "Fire" as types. When applying filters like the selected generations or the name, the number of Pokemon will change, and the Graph will be different. However, when applying a filter based on the type, the Graph will highlight that type making it easier to see which types were selected. Figure 2 illustrates that better.

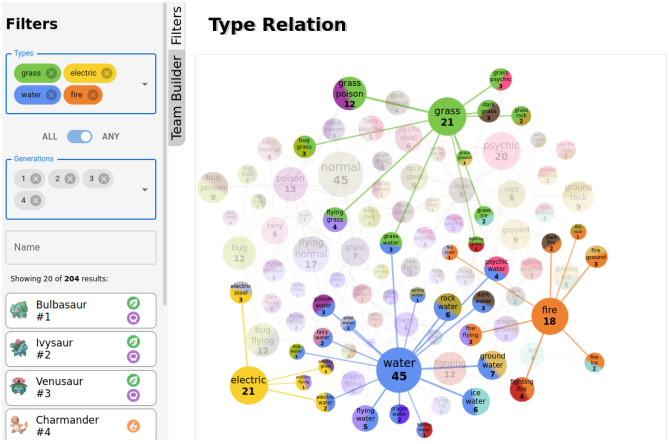

Fig. 2. The filters' sidebar alongside the synchronized type relation graph.

When it comes to the relation between stats, our approach uses a Scatter Plot where each point represents a Pokemon and the axis represent a Stat. The user can change the axis by selecting a different stat. When clicking on a point, a card similar to Figure 1 will appear. This way, the user can see which Pokemon are better fitted for a given stat. The global filters are applied to Scatter Plot, so it only shows points of the Pokemon that match that filter. Figure 3 is an example of how the Scatter Plot can look like.

Until now, it was only possible to compare Pokemon based on a single stat or according to the type. "How can the user compare information about 2 Pokemon in order to decide which one fits its team better?". To answer this question, however, it is required a new form of visualization. A Radar Chart can plot the stats of two Pokemon and make it easier for the user to compare both stat wise. In the card information of

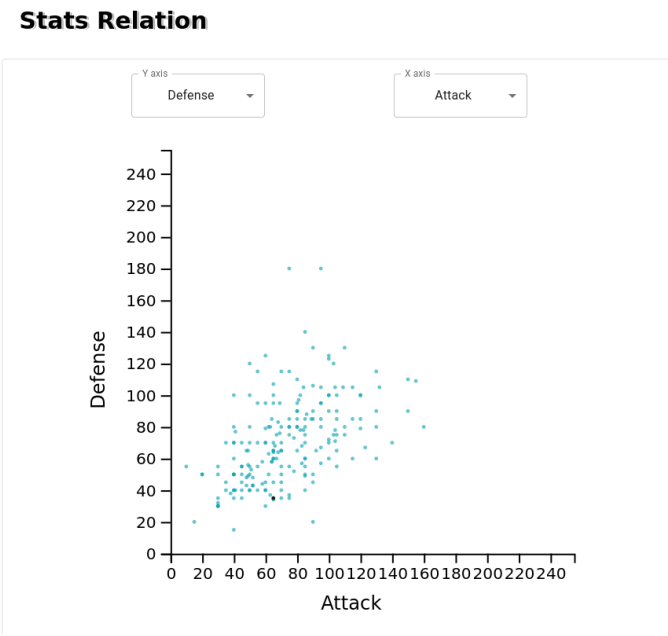

Fig. 3. The Scatter Plot Chart used to represent the relation of 2 stats axes.

the Pokemon (Figure 1) there are two buttons related to this feature, ”Compare1” and ”Compare2”. When clicking on one of them, the selected Pokemon will take the given position in the Radar Chart. This way, the user can select two Pokemon of choice and be able to compare them on every stat to see which of them fit its requirements better. Figure 4 shows a possible example of the Radar Chart.

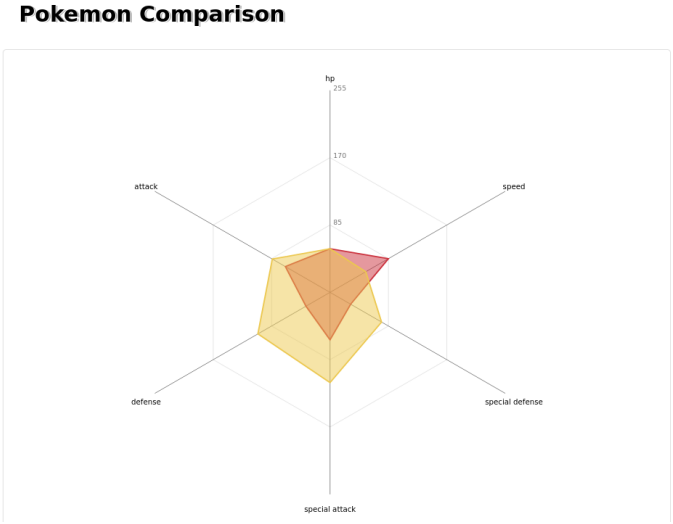

Fig. 4. The Radar Chart used to compare Pokemon side by side.

”How can it compare its team with the already known enemy team?”. This is answered using the Team Builder feature, which allows the user to add a Pokemon to its team. It then shows information like the average stats and type coverage(which is the types that the team is strong against). This way, the user can see if its stats are better than the opponent’s, or it has an answered for the every Pokemon in the opponent’s team. Figure 5 presents the Team Builder feature and it’s representations.

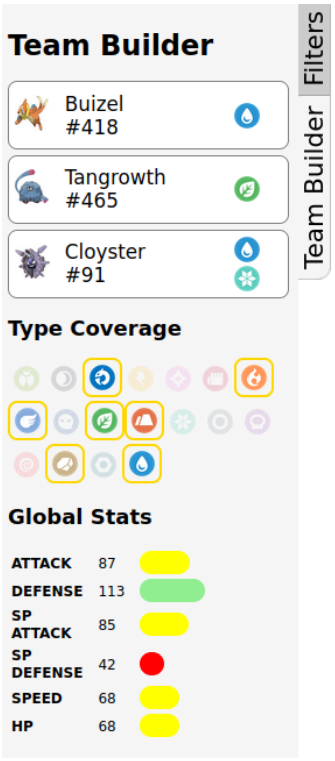

Fig. 5. The Team Builder sidebar.

”How can the user understand and see which Pokemon or set of Pokemon are more predominant and more likely to be chosen by an unknown opponent?”. Although this question is not fully answered by a single representation, the user can understand the likelihood of a Pokemon type by analyzing the Graph with the types and can choose a team with good average stats that has small counters.

For the questions of the Casual Player, ”How can the user understand and see the differences between Pokemon games and the characteristics of those Pokemon across the games that they are inserted into?” can be answered with the help of a Bar Char with the number of Pokemon per generation and with the help of a Line Plot with a line for the evolution of a stat per generation. Figure 6 and Figure 7 represent those representations, respectively.

As for the last set of questions, ”How can the user understand the relation between Pokemon and their types? How can stats be related with their respective types? And how can stats be related with themselves?”, they were answered using two Heat Maps. The first has the relation between types and stats of a Pokemon, showing that there are tendencies used when selecting the stats or types of a Pokemon. As an example, a Pokemon of the type ”Steel”, in general, has high defense and low speed. Figure 8 shows that Heat Map.

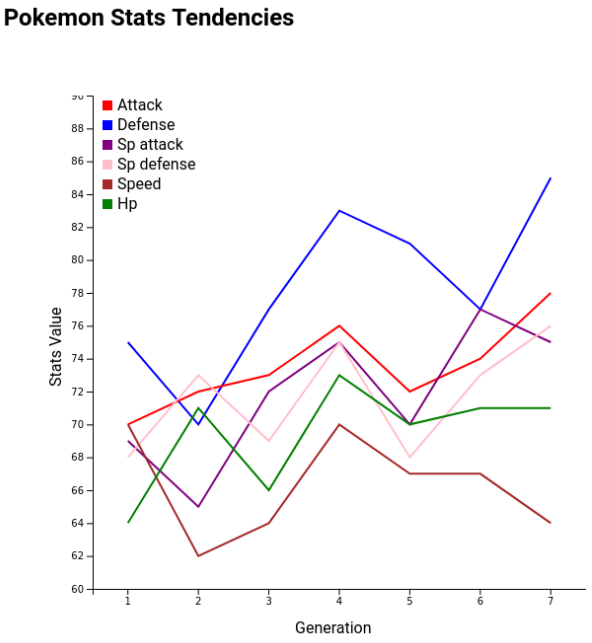

Fig. 6. The Line Plot used to show the Pokemon stats tendencies by generation.

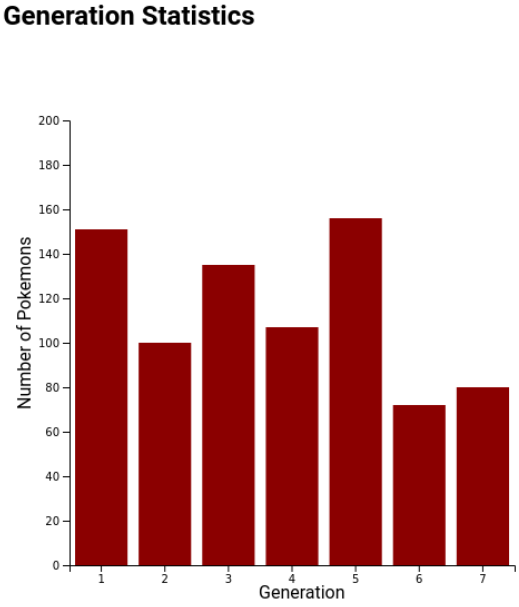

Fig. 7. The Bar Plot used to show the number of Pokemon per generation.

The way stats relate with themselves is observed in the next Heat Map. Similar conclusions can be achieved through this representation, for example, defense and speed are not correlated. Figure 9 presents that Heat Map.

C. Implementation Challenges

During the development of the functional prototype, we had very few problems using the D3.js library. The library

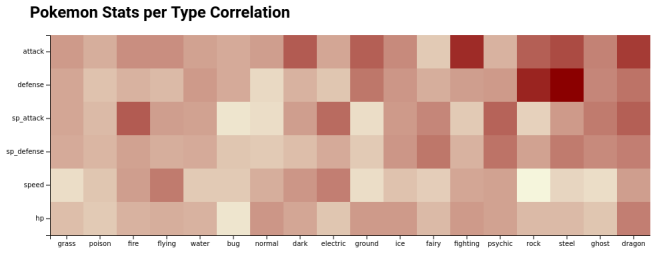

Fig. 8. The Heat Map used to show the Pokemon stats per type correlation.

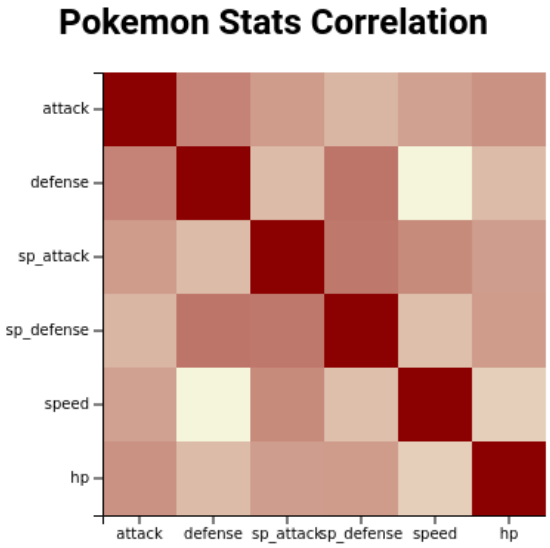

Fig. 9. The Heat Map used to show the Pokemon stats per stat correlation.

is very intuitive and accessible, since there are many available examples that are easy to adapt. Nevertheless, there were some challenges that we had to face while using this library.

The main difficulties appeared in the making of the type relation network graph, which was made with the force simulation API from D3.js. We wanted it to have a fluid interaction and very little delay when updating the filters parameters. At a beginning stage, the graph was always resetting whenever a parameter was updated, which was very cumbersome and a poor usage of the D3.js library, which was to design to handle dynamic data. Once we implemented the update patters, the graph response to updates became faster and more visually elegant, but this was not enough. When the simulation was recreated with the new selection, the nodes were always starting at the same position, at gradually moving towards a stable position, which could take some considerable time. This could be avoided by overwriting the initial positions given by the force simulation with the current positions of the nodes that should still remain present in the new selection. For the nodes that didn't exist before the new selection, which were always nodes with 2 types (the nodes with 1 type are always present, even if no Pokemon of one type exits in the selection), their

initial position was calculated considering the middle position of the nodes at which these nodes would be connected to. All of this to reduce the simulation activity while changing the selection parameters. Finally, we gave the simulation initial positions for the first time loading the page, so that the users could enter the page without ongoing activity.

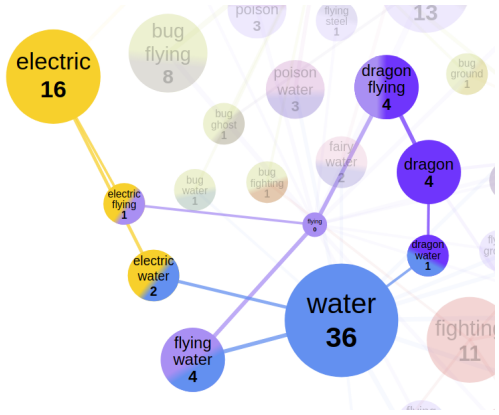

Fig. 10. Adjustable color orientation on the nodes with 2 types.

Another detail in the force simulation that presented to be challenging was maintaining the orientation of the 2 types' nodes, so that the edges of a type are all connected to the right side of the nodes' corresponding color type. However, this was more of a math problem than a difficulty in the library usage. The only requirement was, for every node of 1 type, to store a reference to an object given by the force simulation API, which contained their current position. With the access to the position of the nodes from which the 2 types' nodes were connected, the orientation of the color separation was possible to find. Figure 10 shows an example of the correct color orientations. Notice how the node flying is presented despite not having any element in it, which we considered to be convenient for the user.

V. Conclusion and Future Work

We were satisfied with the end results of the functional prototype. It proved to answer the questions that we raised, thinking on the users' needs and on the assessment given by our colleagues. Nevertheless, we recognize that there is many space for improvement, specially when it comes to: (1) system feedback; (2) non-misleading information; and (3) understandable tools.

The first point can be exemplified with the "Add to Team" button on the Pokemon information cards, since we perceived that the user could not tell immediately what that button was altering if the "Team Builder" left sidebar was not active. We could toggle this sidebar every time the user adds a Pokemon to the team, but it could be bothersome if the user was currently utilizing another tab in the sidebar. The ideal then should be a visual indication that it had successfully added a Pokemon without altering the system current state.

The second point concerned the Pokemon information cards. It has a field with the Pokemon's Type Weaknesses that stipulate which types are good against that specific Pokemon complemented with a color scale: green (very weak) and red (very strong). It was done this way so that the user could find easily the best Pokemon to fight against a known enemy team. However, some of our colleagues confused the green boxes with the meaning "This Pokemon is good against this type", when they should perceive "This Pokemon is weak against this type", despite the title "Type weaknesses". Thus, for future work, we think that we should find the opinion of more users to find if it is indeed misleading and the best solution to adjust this.

The final point was brought because of the "Type coverage" on the "Team builder" sidebar. It is meant to represent all the types that the user's team can have an advantage against for the fact of having at least one Pokemon that has that advantage. However, our colleagues didn't comprehend immediately its meaning, so we think that we need something alike an information tooltip to make it more understandable.

Despite all these problems, we think that they are easily fixable and that this functional prototype has altogether useful tools that combined prove to be a very convenient and coherent product for the end user.

VI. Appendix

Project contribution percentage:
Bruno Bastos - 50%
Leandro Silva - 50%

References

[1] *The Complete Pokemon Dataset.* (2017, September 29). Kaggle. https://www.kaggle.com/rounakbanik/pokemon

[2] Bostock, M. (n.d.). *Data-driven documents*. D3.js. Retrieved January 12, 2022, from https://d3js.org/