

Information Visualization – G13-A – Checkpoint V Report

Francisco Campaniço

IST

Lisboa, Portugal

francisco.campanico@tecnico.ul
isboa.pt

João Rafael

IST

Caldas da Rainha, Portugal

joaolucasrafael@tecnico.ulisboa
.pt

Rodrigo Oliveira

IST

Lisboa, Portugal

rodrigo.domingues.oliveira@tec
nico.ulisboa.pt

INTRODUCTION

Our project is related to electronic sports, more commonly referred to as Esports (eSports is a common way of spelling this as well, but it's an incorrect one). Esports differ from “normal” sports because instead of being a primarily physical activity; while any sport will have a component of tactics behind it, there's also a much higher requirement for physical prowess. The “electronic” component refers then to the fact that this kind of sport is related to playing video games in competitive tournaments, which will, of course, change the necessary requirements for success (there's a lesser reliance on more “broad” physical conditions, but a higher one in hand-eye coordination, reflexes and tactics (and, depending on the game being played, additional requirements may include a good aim, or being able to manage several characters both effectively and quickly, for example).

Depending on the game being played, the team sizes vary from one to usually five. In larger size teams, there's a captain who will not only play in the team but also make the bigger tactical decisions, and a coach who isn't an active player, but advises team members in both training and mid-game. Outside teams, there are always (usually two) *casters* commenting on the game, and a panel of commenters who, between games, give some opinions on team performance and the current *meta* (that is, what characters or weapons are most picked).

Esports have become a prevalent medium of entertainment in the past decade, as they combine the current generation's rising interest in videogames with the already established interest in sports (for example in Portugal, where football fandom is prevalent). They combine the appeal of videogames with the stories that can come out of sports, such as the up-and-comers defeating big established teams, the newest star who rose to the top in their first year, the dismaying loss of a highly praised team in only the first games of a tournament. Our primary motivation is that we both enjoy playing videogames and watching Esports tournaments, as they provide great entertainment.

While it's easy to find quite a lot of information for Esports, this information is either not organized in a particularly appealing fashion for quick assessment (one cannot conclude many correlations by browsing several player or team pages in wikis) or the only thing that's able to be assessed is the games which award the highest prize pools, or the highest earning players. It's not possible to correlate

the number of players in a country with its more general statistics (such as, for example, urban population, which would be interesting to do in this case due to the electronic nature of Esports, meaning it requires access to the Internet).

Tasks

Our approach allows the assessment of more interesting statistics related to Esports besides simpler information such as “highest earning teams”. In Checkpoint I, we decided that our visualization should have the following tasks:

- Analyzing how Esports have evolved through the years;
- Searching for the countries with the highest player earnings;
- Identifying at what ages players earn the most;
- Comparing different teams;
- Comparing different games;
- Comparing at which months most tournaments were held.

Our visualization allows all these tasks. Moreover, it specifies some of them further, and adds some new ones:

- One can not only identify at what ages players earn the most, but one can also filter this information per the desired country;
- It's possible to compare different games and teams through both monetary statistics (prize pool sum for games, earnings for teams) and number of tournaments hosted/played;
- One can correlate a country's number of players or their earnings with some of the country's statistics;
- It's possible to see what teams won tournaments, and how much they earned in each month, ranging from 2000 to 2017.
- It's possible to see how many tournaments of each game were played and how large the prize pools were in each month, ranging from 2000 to 2017.

RELATED WORK

We didn't find much related work in the field of Esports. It's common to find infographics (images displaying several interesting stats) such as [this one](#) for a particularly popular tournament, or “top 100” rankings for earnings or prize pools, such as the ones found on [esportsearnings.com](#), but we didn't take an inspiration in any “in-depth” visualization for the theme. As we stated in the

introduction, there weren't visualizations that went beyond very game-specific statistics (such as the infographic above) or comparing prize pools.

THE DATA

Our data was obtained from two websites' APIs: esportsearnings.com for esports-related data and worldbank.org for more general country statistics. The statistics for *WorldBank* included urban population, unemployment, GDP and education expenditure. Sadly, the education expenditure statistics were too incomplete for us to use them.

Issues

The *EsportsEarnings* API didn't display all the information available on the website, and since the data relied on user submissions, we were unable to use data for additional tasks. For example, while something such as the location where tournaments are hosted would be interesting, and the data was available in the API, due to it being user submitted there was too much variation in location format, such as "California, USA" (state, country) and "Greater Warsaw, Warsaw, Poland" (metro area, city, country), and we couldn't use it.

Another problem with the API is that each user is allowed only one query every two seconds, and the queries are very limited. This proved particularly annoying when we decided to add some interactivity where you can see what teams won (and how many tournaments they won) in a month of the user's choosing. To get all the tournaments (which we already had to do), we could only query the "100 most recent" with an offset which allowed us to move back to the first tournament in the website's data (which took place in 1996, for the curious). Since the website had around 32 thousand tournaments, this required 320 queries, and if one does the math, 10 minutes to perform all necessary queries.

This isn't a lot of course, but the thing is this query doesn't provide information for who won the tournament. To obtain that kind of information, a query for every tournament is required. This adds up to about 17(!) hours of leaving an automation script. We put it running on a Google server (thankfully IST provides free server uptime in these services) and we only remembered to check back a week after.

When trying to add functionality for selecting a country and seeing at what ages players earn the most in that country (we didn't have that data when we delivered Checkpoint II, but we ended up wanting to add that feature), we also ran into the problem that Pentaho DI (the recommended program for data processing) doesn't work well with nested JSON tables. The way we had this organized was that for each player, there was a sub-table with age values and what they earned for each age. We managed to make it work with the global data, but for each country, the solution was to just return this data for each country and... join it in a file

by hand. This resulted in about three hours of heavily repetitive work (the factory scene in Charlie Chaplin's "Modern Times" where he does extremely repetitive and mechanical actions came to mind) and a quite starved IST student. And the data we wanted, of course.

Compromises

We had to make several compromises in the data shown as well. The 32000 tournaments ranged from 1996 to 2018, but the 1996-1999 range only included about 10 tournaments, which wouldn't be very interesting to display as only roughly one fifth of months in those four years would have a tournament. Aside from that, since 2018 isn't over yet, it wouldn't be right to display data related to that month as well, so we ended up cutting those years out.

Also related to tournaments, the data for what teams won tournaments in a selected month was sadly quite incomplete. However, we felt that this information, and the interactivity it provided to our visualization, was too interesting to pass on, and so we ended up leaving it as-is.

For the "earnings by age" statistic (what ages players earn the most), we also had incomplete data, as some players didn't have specific age-related data. This data was not even acquired through an API and was instead scraped from each player's *EsportsEarnings* age-related page. As above, the information and interactivity given is far too interesting to pass on, and we left it as-is.

Several data required the use of Python scripts for us to get a usable table. For example, we had to add some IDs that the *EsportsEarnings* API used to some tables that didn't have them, as they were required for some interactivity actions.

Resulting data

We ended up with 15 MB of data, including around 4000 players, 400 games, 700 teams and 32000 tournaments. Since there were a lot of attributes that we didn't need in these tables (players' nicknames, real names, and tournament locations for example), we ended up filtering a lot of data that was unnecessary for our visualization, which resulted in only slightly above 2 MB of data.

With d3.js, this proved to not have scalability issues: The framework is probably built to handle much larger amounts of data, so this didn't really prove to be an issue. Our final data still included the same number of teams and games, but since we didn't need individual players or tournaments, we were left with derived tables such as the number of tournaments in each month, or the number of players in a country.

Aside from the feature where one selects a country and see earnings/age distribution for that country, which required us to revisit data processing in Checkpoint IV, and adding a few IDs to make some interactivity work, we didn't feel the need to restructure much of the data that we had already acquired in Checkpoint II.

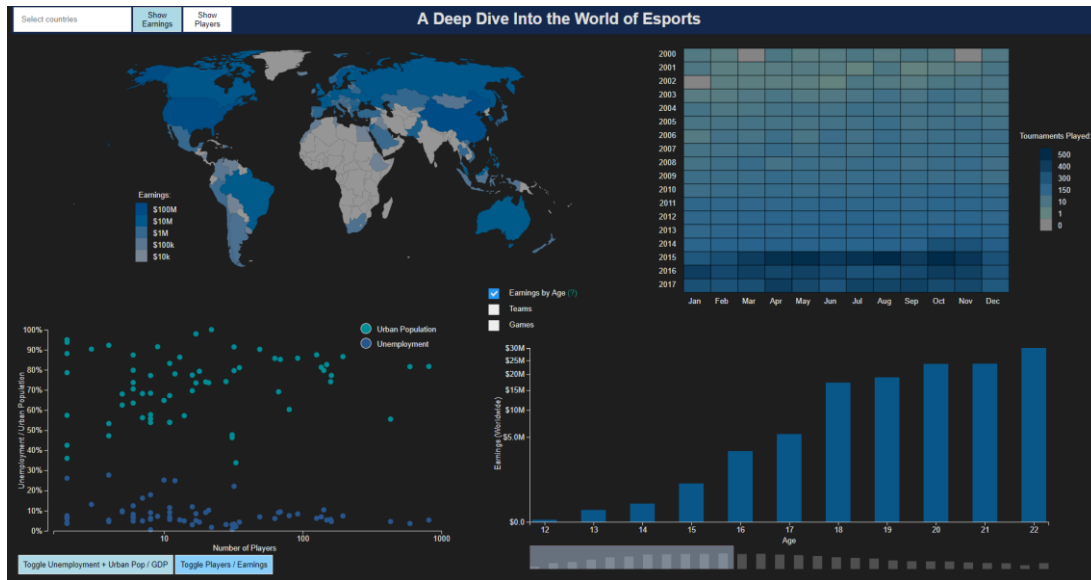


Figure 1. Overall layout for the visualization.

VISUALIZATION

Overall Description

The visualization contains four different idioms. The top left choropleth map displays the number of players or player earnings in each country. The top right heatmap displays the number of tournaments played for each month and year. The bottom left scatter plot relates player earnings or number of players in each country with more general statistics such as unemployment, urban population and GDP. The bottom right bar chart varies quite a lot, and it can display “earnings by age” (The earnings every player won at a certain age), teams and how many tournaments they played or how much they earned, and different games and how many tournaments were played for those games, or the sum of their prize pools.

It’s possible to filter the earnings/age statistic per country, or see what teams won tournaments in a specific month of the user’s choosing.

Choropleth Map



Figure 2. Choropleth Map and its search bar.

The choropleth map displays, through color saturation, player earnings by default, or number of earnings should a user click the “Show players” button. These statistics have slightly different hues so as to make it clearer that the map

changed, as these statistics have similar distributions by design (countries with more players will surely have higher earnings, exceptions notwithstanding).

Should a particular country prove difficult to find, the user can search for it in a search bar. We made not only countries with zero players non-selectable, but also one-player countries, since we also removed them from the scatter plot (after feedback from the professor).

Hovering on a country shows its name and both its earnings and number of players. Clicking in a country changes the scatter plot and bar chart (the details for each change in their idioms’ respective sections), and it will change its color. It’s possible to select multiple countries, but only the last one selected can change the bar chart (it also has a different color, in order to highlight that it’s currently the last one to have been selected).

Scatter Plot

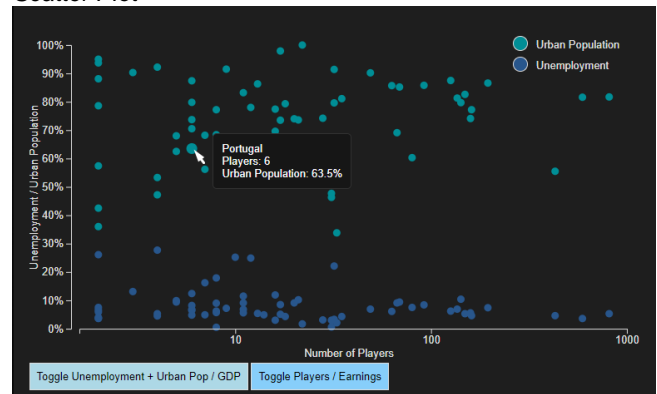


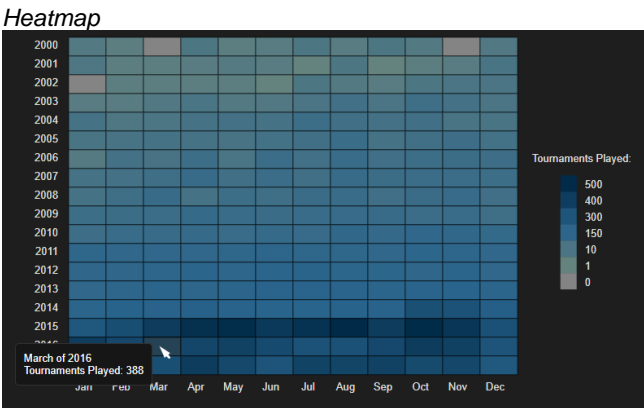
Figure 3. Scatter Plot.

The scatter plot, by default, displays two dots for each country; they both have an X axis value corresponding to the number of players the country has, but the Y axis is

either the country’s unemployment or urban population percentages (these have different colors and there’s a legend to distinguish which is which). Hovering over a dot will display the corresponding country’s name, and the stats its axis is currently displaying.

The user can change either axis by clicking buttons below the plot; it’s possible to change the Y axis to show only the country’s GDP, or the X axis to show the earnings instead of player count.

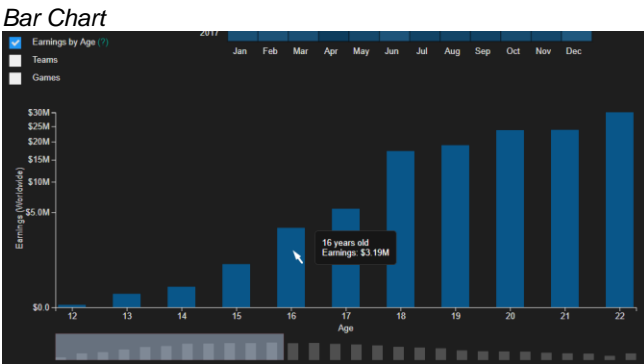
If the user has selected countries in the choropleth map, all unselected countries’ corresponding dots will heavily decrease in size as a way to highlight the selected countries.



The heatmap displays, for each month and year, how many tournaments were played in that timeframe through its color saturation.

Hovering a rectangle shows the month/year it corresponds to, as well as how many tournaments were played that month.

By clicking on a rectangle, the user can change the bar chart (the details for this change is in the bar chart’s section).

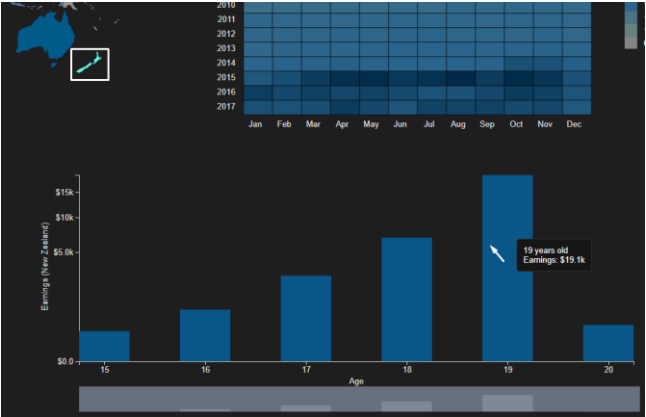


The bar chart displays, by default, “earnings by age” (since this name might be difficult to understand properly, there’s a tooltip stating that this statistic is “How much players earned at a specific age”).

There’s a smaller version of the bar chart below it, with a selection rectangle. The rectangle can be dragged to act as a scroll bar for the main chart.

Through the checkbox present above the chart, the user can change it to display several other statistics: teams and how much they earned, or how many tournaments they won, or games and the sum of their tournaments’ prize pools, or how many tournaments were played for each game.

If the user selected a country in the choropleth map, the bar chart will change to show the “earnings by age” statistic, filtered for the selected country.



If the user selected a month/year rectangle in the heatmap, the bar chart will change to display the teams that won tournaments in that month, and how many they won. The user can then click several items in the checkbox to see how much the teams that won tournaments earned that month, and check the games that were played in that month (more specifically, how many tournaments per game, and the prize pool sum for each game).



In the above cases, if there’s no available information for the month or country the user specified, the hovering tooltip for the country or heatmap rectangle will also show a “No data available” line below the usual information.

Rationale

For visual encoding channels, we used color saturation and hue, size, position on X/Y and height.

Potential

Following the tasks specified in the introduction:

Analyzing Esports throughout the years

We can see this in the heatmap, from the very low tournament counts in the early 2000s, to regularly reaching the hundreds in each month after 2010, to usually over 300 tournaments each month in 2015 and after.

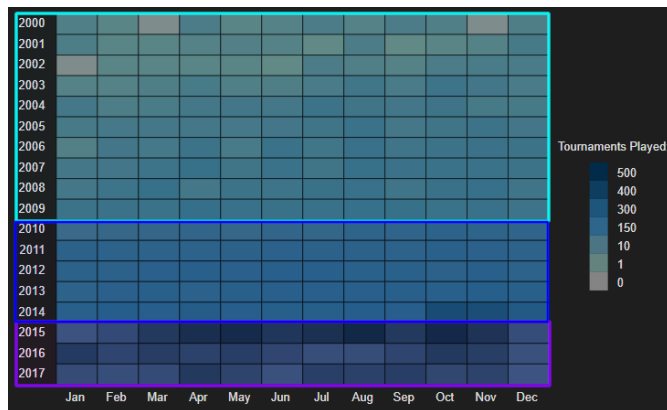


Figure 8. Analyzing tournament counts in the heatmap. We can see that tournament counts barely reach the hundreds in the light blue area (2000-2009), then have 100-200 in the dark blue area (2010-2014), and finally have over 300 regularly in the purple area (2015-2017).

Searching for the countries with the highest player earnings

We can either see what countries look darker in the choropleth map or change the scatter plot to display player earnings on the X axis, check the dots that are farther to the right, and hover on them to know what countries they correspond to.

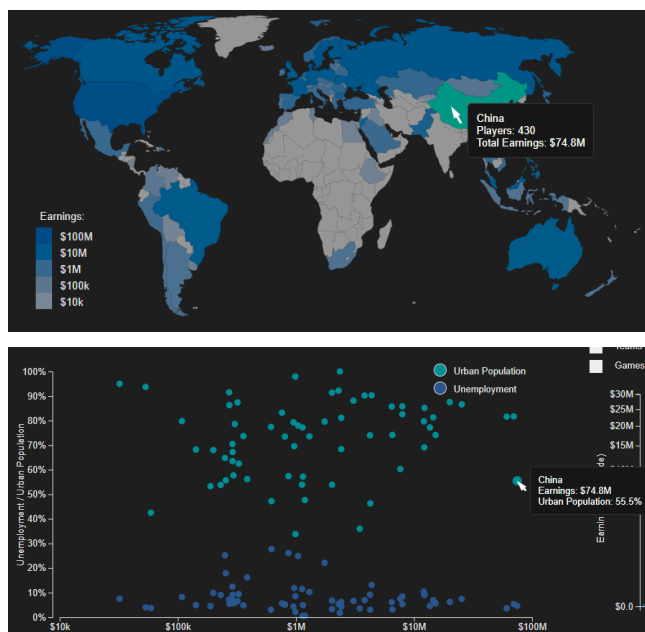


Figure 9. Checking countries with the highest player earnings.

Identifying at what ages players earn the most

This can be done using the bar chart. By default, it already shows this information, but we must scroll right to find out that 22 is the peak age globally.

If we want to find this information for a specific country, we'll have to select it in the choropleth map, and then check the bar chart again (For example, if we check the United States, we can see the peak is 20 years old).

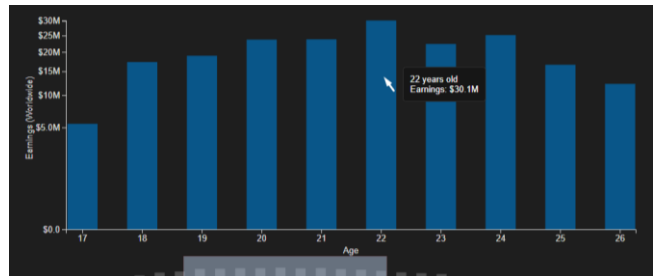


Figure 10. Checking the peak age for earnings in the bar chart.

Comparing different teams

We can do this using the bar chart, by clicking the “Teams” button in the checkbox. We can then see the different teams sorted through earnings. If we want to see how many tournaments they played, we can also click the “Tournaments” button.

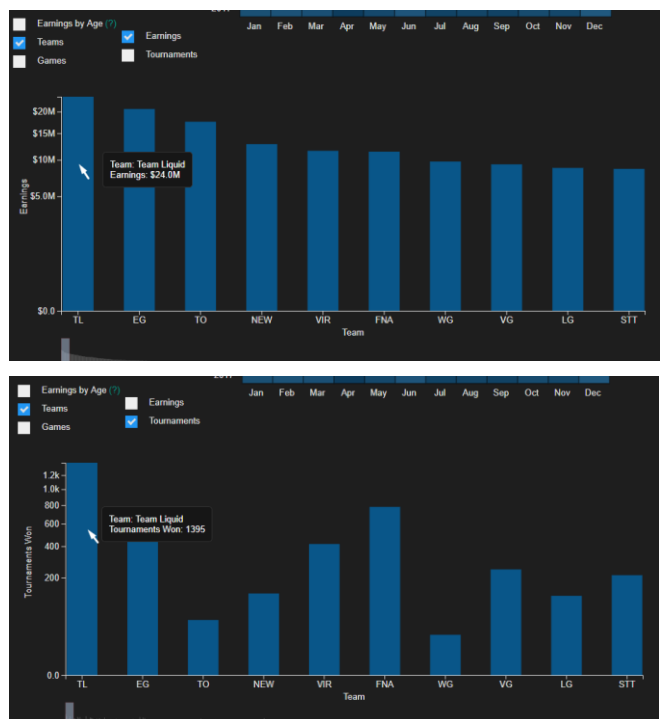


Figure 11. Checking what teams earned the most, and how many tournaments they won.

Apart from that, we can also compare teams in a specific month, by clicking in that month in the heatmap and

checking what teams won tournaments that month, and how much they earned.

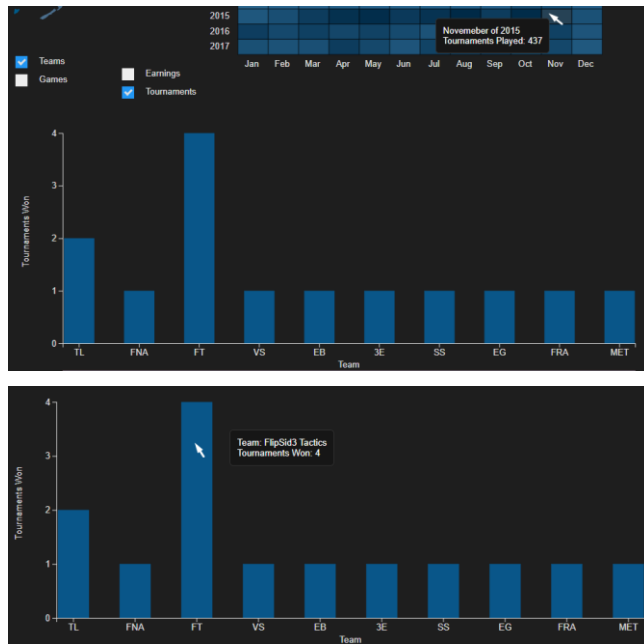


Figure 12. Checking what teams won in a specific month.

Comparing different games

This is done using the bar chart, by clicking the “Games” button in the checkbox. We can then see games sorted through earnings. If we want to see how many tournaments were played for each game, we can also click the “Tournaments” button.

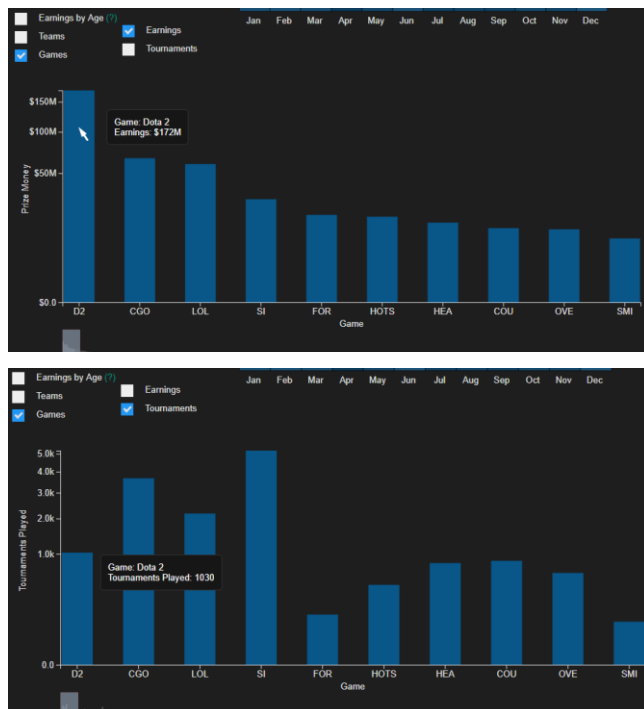


Figure 13. Checking what games have the biggest prize pools, and how many tournaments were played for each game.

Apart from that, we can also compare games in a specific month, by clicking in that month in the heatmap and checking what games had tournaments take place that month, and how large the prize pools were for each game.



Figure 14. Checking what games were played on tournaments in a specific month.

Comparing at what months tournaments are most held

It’s possible to do this using the heatmap. We can see that the lighter colored month overall is December, and that October seems to have the darkest colors overall throughout the years, meaning it’s the month where most tournaments are held.

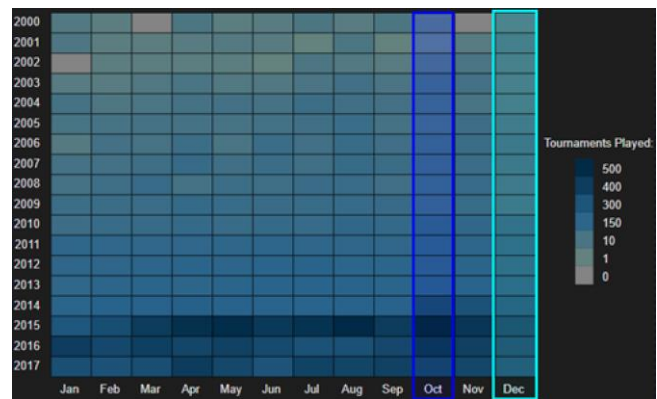


Figure 15. Analyzing peak months for Esports in the heatmap. December (light blue) is the lowest, while August (dark blue) is overall the peak.

Correlate the number of players in a country with its other statistics

We can do this using the scatter plot, since it already shows this information. For example, if we want to see how GDP in countries relates to the number of players, we can toggle the Y axis by pressing the “Toggle Unemployment + Urban

Pop / GDP” button, and then we can easily see that there’s a trend for higher GDP countries to have more players.

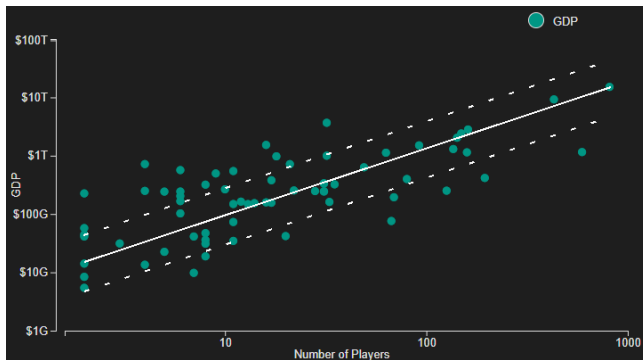


Figure 16. Checking the correlation between GDP and player counts in the scatter plot, we can see the trend for higher GDP to mean more players.

IMPLEMENTATION DETAILS

Aside from d3.js, we used jQuery to ease some interaction such as tooltips.

Challenges

While d3 does have quite a lot of features, the documentation is sub-par at times (consisting of a lot of disjointed GitHub repositories), so it took us some time to find decent scale types, such as the power scale, which ended up being more useful than a logarithmic scale simply because it also allows displaying different orders of magnitude effectively (unlike a linear scale) but it has a customizable exponent property, which solves the logarithmic scale problem where a lot of higher values are near-indistinguishable, since one can set the exponent at a lower value to allow both low and high orders of magnitude to have detail.

Another important function in d3 was the formatting. It’s possible to format numbers with quite a lot of customization, and we used that for very different things. We usually display large numbers with an SI prefix (k for thousands, M for millions, et cetera). The formatting is also useful for displaying percentage and dollar signs without messing around with strings too much.

In the bar chart, team and name games would look too verbose in the X axis. Therefore, we automatically reduce each name to a tag, and the user can hover over it to see the full name. This works well in order to make the axis less cluttered, but it’s limited, since it just tries to get each word’s first letter (resulting in problems such as *StarCraft II* showing up as SI). However, since it’s very easy to just hover on the tag and see the full name, we feel that small problems such as the above one are a good compromise.

Interactivity

The link between the choropleth map and the scatter plot is quite simple: We check what countries aren’t selected and decrease their radius from the default 4.5 to 0.75.

Interactivity between the choropleth/heatmap and the bar chart isn’t extremely complicated as well. Since we already had the code for changing both the bar size/number and its scrolling bar counterpart, it was just changing that code to use different data. What we do here is removing any bars that we don’t need (for example, if we change from viewing all teams to only the ones that won in a certain month, we surely don’t need 700 bars!), adjust scales and axes and change the bar width (in case there’s only a few bars), and do the same things for the sub-bars in the scroll bar.

We took advantage of promises, a functionality provided by Javascript, to ensure the data is completely loaded before starting to draw the corresponding idiom. This also means all the data an idiom uses must be loaded together in order to avoid problems with missing data (synchronous task).

Techniques

The first step for implementing the choropleth map was to adapt the d3 version on the example to a newer one (which was a lot more difficult than expected). Using a normal array, we keep track of the selected countries to interact with the dropdown and other idioms. In order to make it look prettier we added hover events and a “pulsing” effect on clicking a country, using transitions between color changes to increase responsiveness and give the user the actual feeling of clicking something.

The base for the scatter plot is mostly from the course’s tutorial (that is, just drawing circles and setting their positions). To make the animation where Unemployment and Urban Population dots “merge” into the GDP dots, we actually just move all pairs of country dots into the same position (meaning each dot on the GDP chart is actually two dots in the exact same place).

The heat map implementation was straightforward; it started with an example from the internet that was already in the version we are using in the project. To get the desired interaction with the bar chart, we added click events to each heat map rectangle and mouse over events to achieve hover effects.

Finally, for the bar chart, since we have several datasets with a considerable size, we wanted to find a solution that played well with such data. After searching for it, we came upon a simple scrollable bar chart that gave us the flexibility we needed. Due to its simplicity (it only worked with one dataset and had a version older than the one we are using), we had to implement a couple more features to get where we wanted. For toggling between different datasets, we included checkboxes to change one or both axes of the chart as well as updating the bars which corresponded to the new data. Other features include the creation of tags and interactivity between idioms which were previously mentioned.

CONCLUSION & FUTURE WORK

From making this project, we not only learned to work with tools such as D3 and Pentaho, but also scraping websites

and organizing data in general, as well as figuring out not only the clearest, but also the most interesting ways to display data in an interactive fashion.

We managed to not only address all tasks we set out to do in Checkpoint I, but also to add more possible tasks with extra functionalities, such as comparing different country stats or having more filters for months and countries.

Had we started over, we think we'd add a line chart for the earnings by age statistic, since it would make the visualization clearer. This statistic sort of behaves like a Bell curve (since it tends to follow normal distribution), so it would probably look better and clearer on a line chart.

Given 3000€ and another month to work on this project, we could maybe show tournament locations (with that time and money we could probably get around the different time format issue) and make the heatmap more interactive (filtering the tournaments played by teams or games).

We could also add the ability to see different countries' earnings by age in the same chart, which would lead to some interesting comparisons.