HOW ARE VIDEO GAMES DISTRIBUTED ACROSS DIFFERENT GENRES AND PLATFORMS, AND HOW MANY DEVELOPERS ARE INVOLVED IN CREATING THEM?

Process Book

Hugo Hof, Eric Kurmann, Camille Steiner

COM-480 Data Visualisation

Spring Semester 2024

Introduction

Beginning of the idea

At the beginning of the Data Visualisation course, we were hesitant to work on movies, Disney characters or video games. Looking at last year's project, we saw that many groups have worked on the film topic. And the Disney character was very complex to analyse. So, we decided to work on the theme of the video game.

By analysing the video games dataset, we saw that each video game is characterised by three main components: its genre(s), its platform(s) and the developer(s) that have created the game. So, based on this fact, we decided to show the diversity on the video game world with these three aspects and the relationships between them along the years from 1971 to 2023:

- A first visualisation will therefore aim to show that video games are characterised by many genres and that the most popular genres evolves over the years (popular genres => number of games developed in each genre).
- A second visualisation will show that video games are developed for many platforms. It will also show that the most popular platforms have evolved over the years.
- A third visualisation will focus on developer diversity. We will try to show which developers have released the most games each year, and what are the most popular games for each of these developers.
- Finally, a final visualisation will show the different links that exist between these three components.

Data Analytics

To create the different graphs, we needed to process the data using jupyter notebooks from the dataset we choose. In milestone 1, we cleaned the dataset csv into a new csv without duplicates, the correct types, removed unused columns, change the format of the date and at the end of this first data processing we plotted some basic graph to give us ideas of the different visualizations we could do.

Then, we still needed to process it to be able to create the graphs. For each graph, we created a JSON file that is then used to display it. We grouped the values by year then by key values. The JSON file is better for having a key value relationship

Choosing the visualizations

In our website, we would like to show the video games diversity at different scales. For that purpose, the three parts of our website show one after the other the genre's diversity, the platform's diversity and the developer's diversity that realized the video games. To conclude, we wanted to show in a last visualization the links that exists between these three cited diversities in an interactive diagram. To keep a consistent design style, each visualization is composed of a left part and a right part aligned horizontally. We will now present in more detail these four visualisations:

Bar chart diagram: Genre's Diversity

- ***Left Part:*** An interactive bar chart diagram that present the most popular genres in a year selected by the user. The user can also select the maximum number of genres he wants to see

on the graph. The genres are sorted in function of the number of games linked to each genre for the selected year. The user can also click on a play button such that the graph automatically evolves to show the most popular game's genre along the years. Finally, the user can click on a bar to see some information relative to the selected genre on the right part of the visualization.

- *Right Part:* A definition of the video game's genre selected, and a podium of the top-3 games defined by the selected genre and released on the selected year established in function of the criteria used by the user (average review grade, number of players, number of reviews made on this game, etc).

Treemap diagram: Platform's Diversity

- *Left Part:* An interactive Treemap diagram that presents the platforms on which video games have been released. The user can select a year from 1971 to 2023 on a slider bar to see the most popular platforms on this specific year. If he clicks on a play button he sees an animation of the popularity of the platforms through the years. If the user clicks on a platform, some details about it would be displayed on the right part.
- *Right Part:* On top, a list of all games released on the platform selected on a specific year sorted by a criterion the user can choose in a dropdown menu. On bottom, a line graph that represent the evolution of the popularity of the platform selected given a criterion given in a similar menu.

Circle Packing Diagram: Developer's Diversity

- *Left Part:* An interactive circle packing diagram that presents the developers that have published video games in the year selected by the user. The user can also select the maximum number of developers he wants to see on the graph. The size of a circle on a graph is dependent of the number of games it has released the selected year, the more the developer has released games in comparison to the other developers, the more the size of its circle on the graph will be important. The user can also click on a play button such that the display of the graph automatically updates to show the evolution of the developers that have published the most games along the years. Finally, the user can click on a circle on the graph to see information corresponding to the selected developer on the right part of the visualization.
- *Right Part:* Firstly, a list of all the games that the selected developer has released the selected year will be displayed and the user can sort this list in function of a criteria that he can choose (number of players, average review grade, number of reviews, etc). Secondly, there is a line graph that presents the evolution of the selected developers along the year in function of a criteria selected by the user (number of games released, number of reviews of the games released by the developer, etc).

Sankey Diagram: Interactive Diagram

- *Left Part:* An interactive Sankey diagram that presents the links between the genres, the platforms and the developers in the year selected by the user. The number of items to display can be chosen by the user for each of the three components. He can also change the order of the components displayed. Finally, the user will be able to select multiple items from any of the three components, which will display information on the right part of the visualization.

- **Right Part:** A list of games that correspond to the items selected on the left part of the visualization ordered in function of a criteria selected by the user. A graph that shows the global evolution along the years of the selected items in function of a criteria selected by the user.

Technologies

The programming language and libraries that we used during the project are listed in this part:

Python

We used the Python language to process the original csv data file and to draw some basic graphs to get an idea of what we will do with the data and have basic statistics (like the total number of games, developers, etc).

Panda Python's Library

We used the Panda library to create a data frame of the original csv file of the video game's dataset. This Python's library has facilitated the data processing in complement of the Python language to create the different JSON files containing the data processed used by the graph in the visualizations. It also has been used to facilitate the cleaning of the data during the milestone 1 of the project.

HTML/CSS/JavaScript

We used HTML language for the structure of the web pages and visualizations. We used the CSS style in complement with custom games font and pixelate font.The css style used for the pacman animation on the main page, the css style used for the mario animation on the introduction and about us page and the css style used for the sonic animation on the conclusion page.

FullPage.js

We used the FullPage.js Java script library to organize the pages of our website into multiple slides page. We can navigate between the different slide page with a menu located on the right of the web page.

D3.js

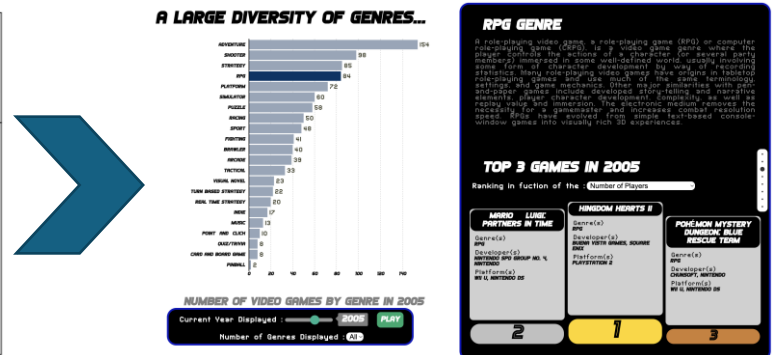We used the Java Script D3.js library to create our interactable graphs.

Implementation

Bar chart diagram: Genre's Diversity

On the left part of the visualization, a bar chart diagram is implemented. The elements in the graph are displayed in descending order in function of the number of games released for a genre each year. Each bar of the graph has on its left, the name of the genre associated to it and on its left, the number of games released for this genre. The bar chart graph contains also vertical bar grid to put separation step in the number of games released for readability. I have chosen to implement the graph with only one colour element for simplify the readability in comparison to the initial schema. By clicking on a bar (a genre), the colour of the bar has a greater opacity to indicate that this genre was selected and linked to the information on the right part of the visualization. Below, a toolbox is present to select the year in which the bar chart will display the genre's diversity and the number of genres to display. It also contains a play button such that the displaying of the graph is updated every second to show the genre's diversity along the year. Finally, the right part of the visualization contains a description and the top-3 game in function of a criterion that the user can choose in function of the year and genre selected before.
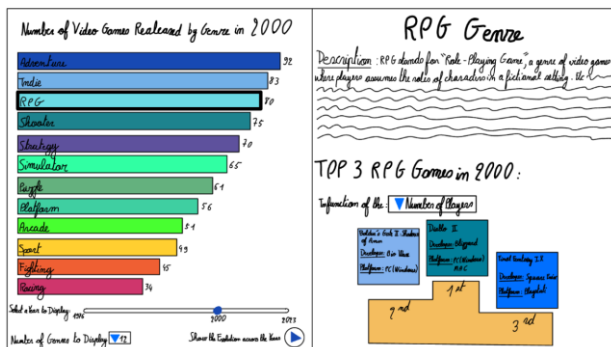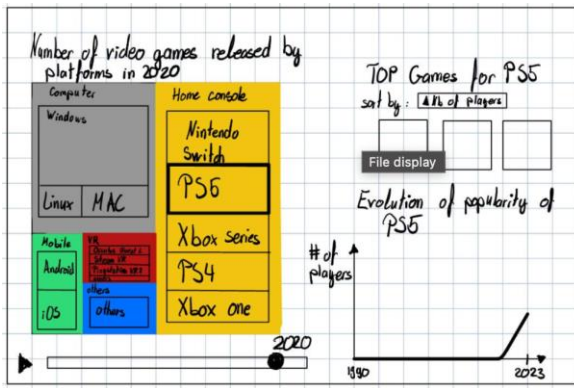

Initial schema of the Genre's Visualization   *Genre's Visualization Implementation*

Treemap diagram: Platform's Diversity

On the left side, there is a treemap diagram to help the visualisation of platforms popularity. To improve the visualisation, the platforms are grouped by category each having a specific colour. We searched which hardware category, out of ten, fitted the most each platform and linked them in a JSON. If you move the mouse over the diagram, a tooltip on which a text with the hardware category of the platform, the platform and the number of games released will be shown with respect to the selected year. When a platform is selected, the colour of its rectangle representation is darker. For readability reason, we also changed the text colour to white. The data are displayed depending on the year so the user can select the latter and play the animation to see the evolution of each platform through the years. On the right top side, there is a list of the games sorted by a criterion. On the bottom, there is a line graph showing the evolution of the popularity of the platform through the years given by another criteria. The two criteria are selected by the user in their respective drop-down menu.

*Initial schema of the Platform's Diversity*
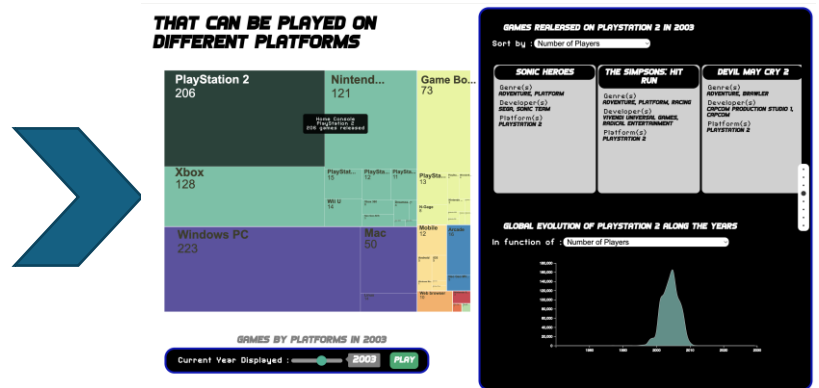
Platform's Visualization Implementation

Circle Packing Diagram: Developer's Diversity

On the left part of the visualization, a circle packing graph is implemented. Each developer is represented in the graph by a circle with a size depending on the number of games released by this developer. The size of the circles is also dependant on the element that has the maximum number of games for the year selected (to avoid that for example in 1971 there is only few very small circles elements) for readability. The user can also drag a circle on the graph and move its position. If a circle is too small to read the name of the developer correctly, then the user can hover over the circle and information corresponding to the hovered circle in the left-up corner of the graph will be displayed. In the toolbox below the graph, the user can also select the year when he wants to see the developer's diversity and the number of developers to display (showing many developers in the graph may create some latencies on the website). The user can also click on the play button to automatically redraw the graph every three seconds to show the developer's diversity along the years. The user can click on a circle on the right, then the opacity of the circle will increase and information relative to the selected developer will be displayed on the left of the visualization. Finally, the right part of the visualization shows the list of games realized by the selected developer in the selected year ordered in function of a criteria selected by the user. This right part also contains a graph that shows the global evolution of the selected developer in function of a criteria selected by the user along the year.
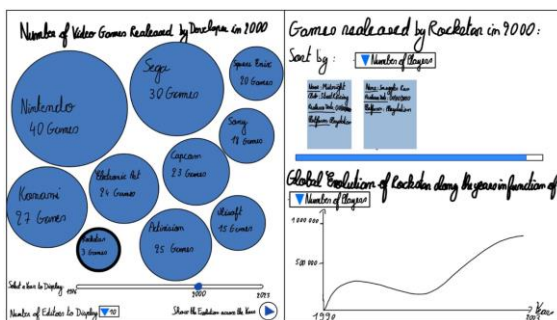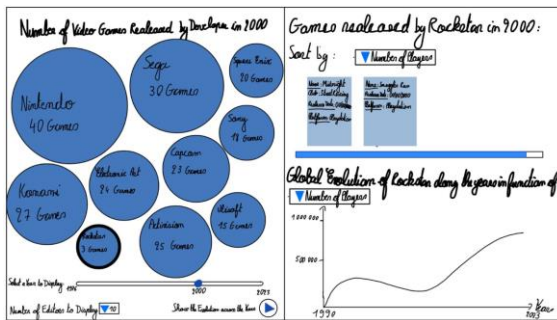

Initial schema of the Developer's Visualization
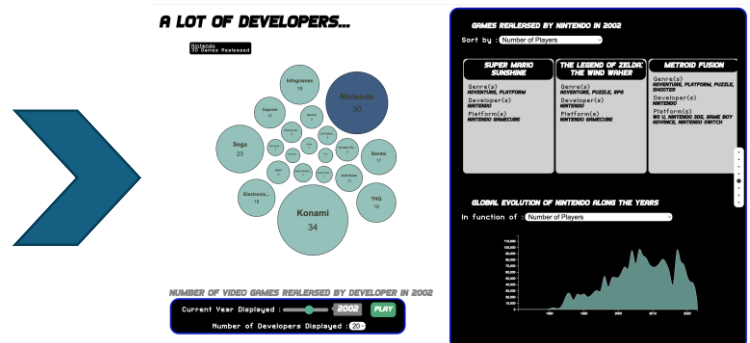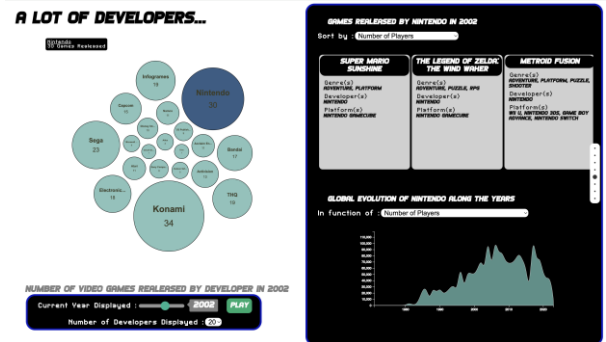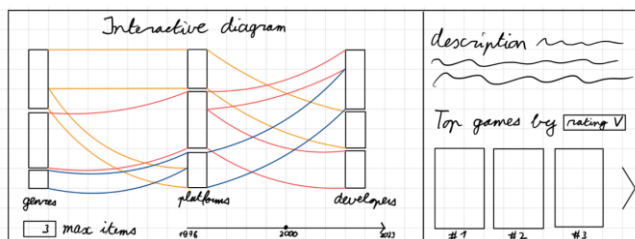

Developer's Visualization Implementation

Initial schema of the Developer's Visualization

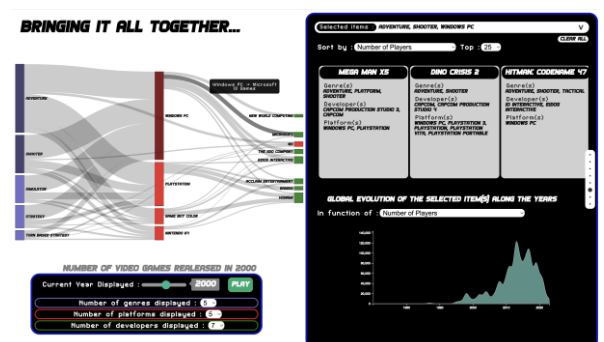

Developer's Visualization Implementation

Sankey Diagram: Interactive Diagram

On the left part of the visualization, a Sankey diagram is implemented. Multiple items can be selected and marked by a darker colour, and they can also be moved around. Hovering on the links or nodes will show more information about it. Below is a toolbox that can be used to change the year that is using a slider or a play button. The number of items displayed for each of the genre, platform and developer can be changed using the corresponding tools. These can be used to change the order of the links by dragging above or below. On the right side of the visualization, there is a dropdown menu that shows some of the items selected and clicking on it will make it show all the selected items along with crosses to delete them. Below, a list of game filtered by the selected items and sorted by a criterion is displayed. At the bottom of the right part, a graph is displayed that represents the selected items along the year based on a criterion. I chose not to have a description for each item as there was a huge amount of them.

Initial schema of the InteractiveVisualization

Interactive Visualization Implementation

Challenges

It was our first project using the HTML/CSS/JavaScript language.

Circle Packing Diagram

It was quite challenging to put in the correct scale the size of the circle in function of their importance in the graph. Furthermore, the name of the developer was sometime too long to enter in its corresponding circle, so we must split it in this format (developerTooLong...) and the number of words displayed should be chosen in function of the size of the displayed circle. And the font size and distance between the developer's name text and the number of games released should be adapted too. Furthermore, this different size should be adapted in function of the other developer's circle displayed such that if there is only one if for example there is only one circle displayed with open game released it is not displayed in a very small size and take sufficient place in the graph.

Responsive web page

A web page can be rendered on multiple types of screens and sizes. Our first version of the web site was not responsive to many types of window sizes. We needed to tweak a lot of CSS style and learn more about it to be able to make a more responsive UI. We managed to do it by having resizable graphs, better use of the divs with grid or flex display, sizes in percentage, etc.

Text in the platform treemap

We had some problems with the text on the platform treemap because it was not possible to automatically resize and put on multiple lines, so we had to do it manually by calculating some min and max functions with the height and width of each rectangles to decide the size of the font and how many characters we wanted to display.

Peer Assessment

Hugo Hof

Design and implement the Genre's and Developer's visualization and process the data for these visualizations. Designed the title and introduction pages and design the CSS styles of the website. Worked on all milestone reports, on the redaction of the process book and on the video.

Eric Kurmann

Design and implementation of the interactive Sankey diagram's visualization and process the data for the visualization. Worked on making the web site more responsive to different window sizes. Design the conclusion and about us pages. Worked on all milestone reports and on the redaction of the process book.

Camille Steiner

Design and implementation of the platform's visualization and process the data for the visualization. Worked on all milestone reports and on the redaction of the process book.

Repository link

Website-link

Thank you for your attention and we hope that you will enjoy our visualization.