

Data Description

The data I chose was taken from [Kaggle](#) and it was acquired by web scraping the information on vgchartz.com, a site used as a video game database for many games in the past 40 years. In 2018, vgchartz released a statement describing the difficulty in providing accurate information on video game sales data a result of discrepancies between retailers sales figures and actual copies sold. Here is a quick description of each field provided in the dataset:

img : URL slug for the box art at vgchartz.com
title : Game title
console : Console the game was released for
genre : Genre of the game
publisher : Publisher of the game
developer : Developer of the game
critic_score : Metacritic score (out of 10)
total_sales : Global sales of copies in millions
na_sales : North American sales of copies in millions
jp_sales : Japanese sales of copies in millions
pal_sales : European & African sales of copies in millions
other_sales : Rest of world sales of copies in millions
release_date : Date the game was released on
last_update : Date the data was last updated

The primary fields that I will be using are genre, console, developer, total_sales, and release_date as these will be the best categories to use since I would like to visualize a stacked area chart that tracks the total sales of a given genre over time. Additionally, this will allow me to create filters that are text-based and easy for users to understand, allowing users to further interact with the data.

Use Cases and Insights

1. Understanding sales trends by genre over time
 - a. The utilization of a stacked area chart allows users to think about how overall game trends evolved over time and how it has grown or declined
 - b. Clicking on a specific genre would allow users to see trends within that genre, without the visual clutter of additional genres
2. Exploring the rise and fall of consoles
 - a. Filtration by console would allow users to see how well certain consoles performed compared to others
 - b. Additionally, many games in this dataset are actually the same game, but on differing consoles, which allows users to think about how the platform a game is released on influences its sales figures
3. Analysis of developer success

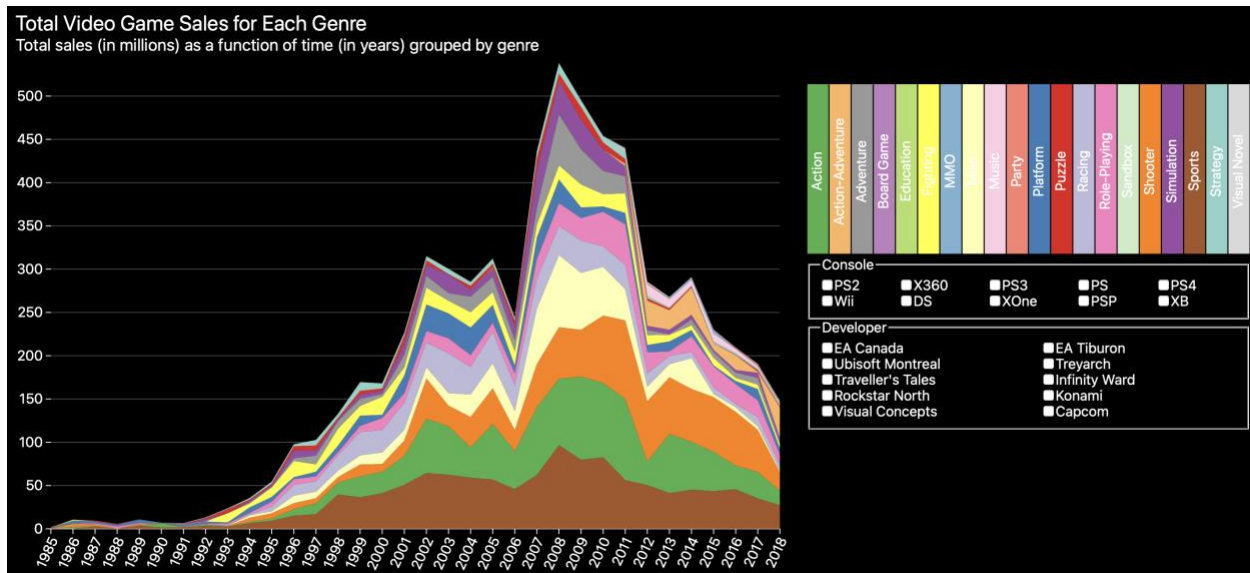
- a. While the talent of developer does not lie in their ability to sell their game, it would be interesting to see how certain developers are more monetarily successful than others. Is it because they focus on specific genres? Are the most monetarily successful ones able to branch out of their genre and explore other types of games, while still maintaining their sales figures?
4. Consumer behaviors and patterns
 - a. There have been many “booms” in the gaming industry, and it would interesting to see what the spikes in total sales can be attributed to
 - b. By seeing which genres maintain monetary success, we can use this information to formulate new and innovating topics by potentially create games that belong in multiple genres
 - i. Such games have seen large amounts of success including Rocket League, Legend of Zelda, Mario, etc.

Storyboard and Interaction Justification

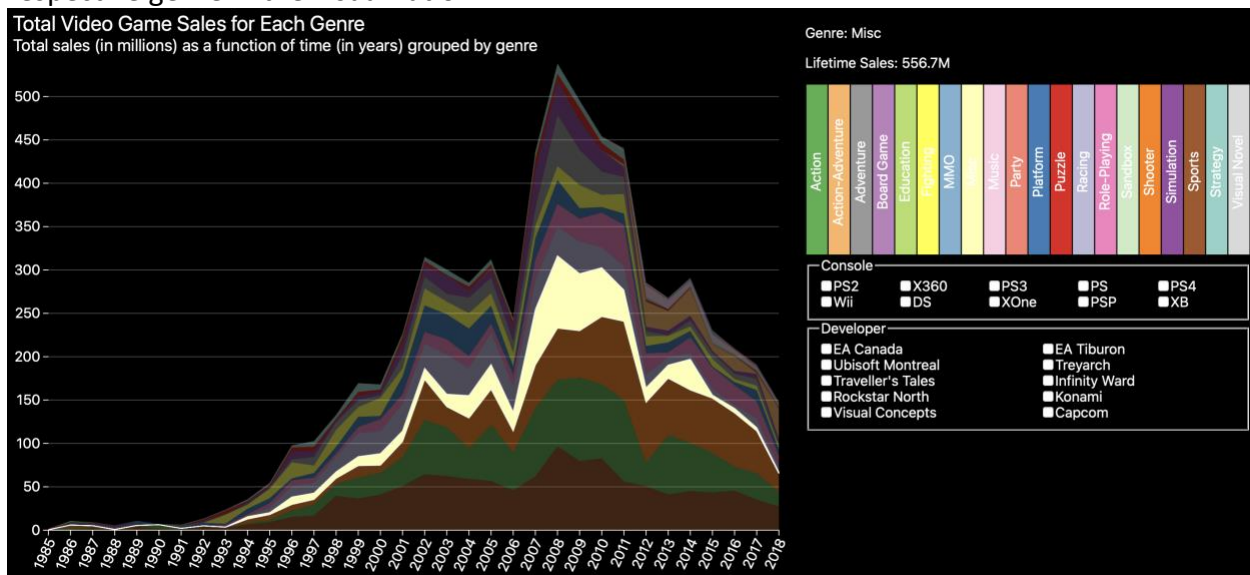
Scene: Landing Page	Scene: Hover Functionality	Scene: OnClick Functionality
The landing page will show the chart, filters and color legend of the chart.	Upon hovering over a stack/section, it will display some information about the genre.	Upon clicking on a genre, it will only zoom in on that specific genre, and hide others
Scene: Return to all genres	Scene: Filters	Scene:
When clicking on the stack again, return to the unzoomed view	When filtering by certain consoles/developers, it will dynamically update the data	

The usage of the hover functionality allows users to focus on one genre at a time without forcing them to zoom in. The other genres will be dimmed and the hovered one will be at full opacity (cannot see in storyboard as it is hard to draw). This dimming feature prevents distractions by the other genres and allows them to focus in on specific trends within that genre. However, if they do want to see more information, they can click and it will zoom in on the genre, furthering the analysis of specific trends. Clicking again would allow users to return to the original view, and the lack of usage of a back button makes the design feel more inviting and less cluttered. The filtration on the right side allows for intuitive filtration through the usage of checkboxes. When a user selects one of these boxes, the data will dynamically change according to their criteria.

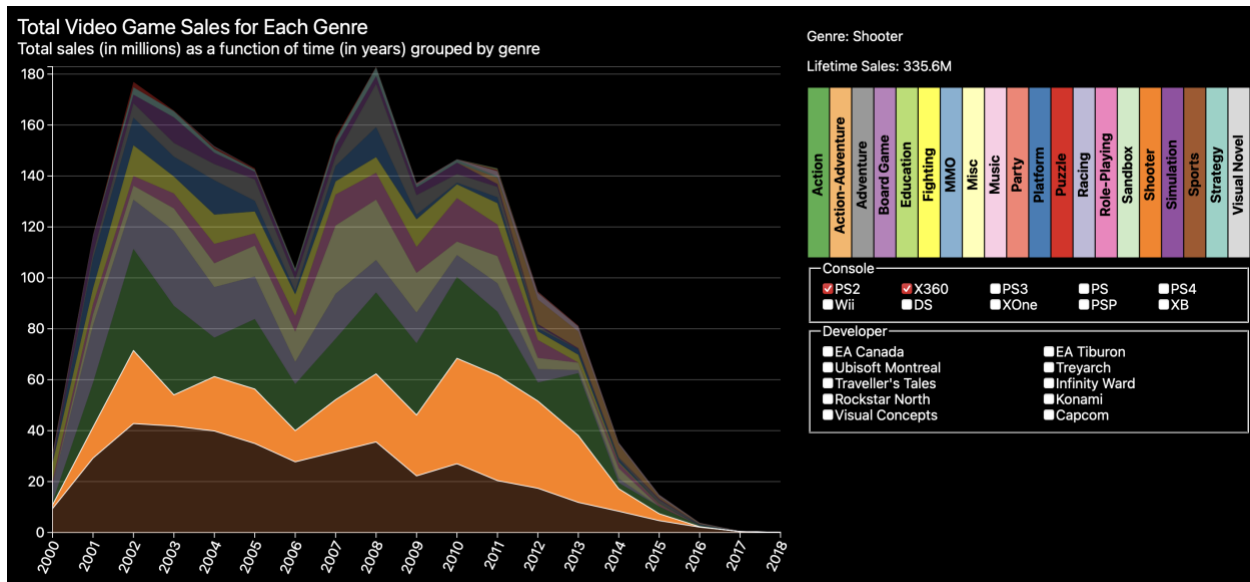
Final Interactive Visualization



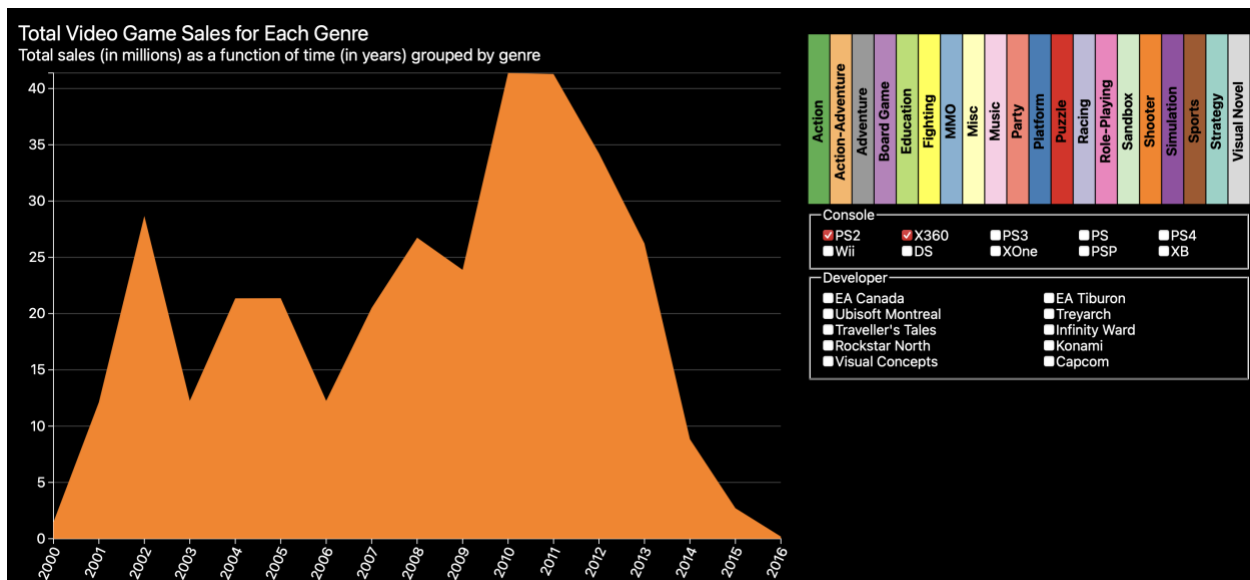
This is a screenshot of what users would see first prior to any interactions. It depicts the total sales (in millions) over time grouped by each genre. There are the top 10 consoles and developers on the right-hand side accompanied by a legend that maps each color to its respective genre in the visualization.



Here we see the hover functionality selecting a given genre and displaying some additional information on the right side above the color legend.



Upon selecting a few filters on the right side, we see that the lifetime sales value changes as well as the visualization, displaying games that are on the PS2 and Xbox 360.



Upon clicking on one of the genres, you can zoom into that genre to see it. This is useful for genres that have very few copies sold compared to the other genres.

Issues and Tradeoffs with the Visualization

A few issues and tradeoffs that come with this visualization is that the smaller stacks/sections are difficult to interact with. These sections can be difficult to interact with, and by introducing a minimum stack height, it would allow easier visibility (but potentially skew the data). Users could also use the legend to interact with the genre sections, circumventing the data skewing caused by the previous solution. Additionally, it can be a bit confusing to understand what a

user is looking at when they click on a specific genre, so it could be useful to incorporate a breadcrumb navigation system to allow users the ability to navigate back to all genres.

The filtration system could result in an empty set when selecting certain filtration criteria. This was a large issue during development, and it was mitigated by reducing the number of filters that a user could select, only limiting it to the top 10 developers/consoles with the highest total sales figures. However, this could still be an issue if the user selects a developer that is exclusive to certain consoles.

Development Process + Changes Between Storyboard and Final Result

I began with doing a bit of research on how stacked area charts worked as well as some research on d3 documentation and functions. Afterwards, I began with processing the data to make it the correct shape. This was possibly the longest and most difficult part as I was unsure of how to adequately sum up the total sales for each genre within d3. Additionally, the majority of the functions I tried ended up making the data much more precise and I was unsure of why it was not looking like my storyboard visualizations. Eventually, I realized that I was incorrectly, summing the data and not grouping by year, but rather by the exact date of occurrence. After this, the hover and click functionality were relatively easy to implement as well as the filtration checkboxes. Formatting the legend and attempting to address an issue of the colors for each genre changing on filtration/click took a decent amount of time as well, but upon spending some time reading the existing legend code, as well as some linear algebra (used for placement of text labels after rotation), it became quite easy. As for the colors not syncing across interactions (clicking + filtering), I had to pass the color scale into every function such that the correct color was shown for each genre across interactions.

The biggest theme within the tradeoffs that I had to make in this visualization was the intuitiveness of the interactions. While checkboxes on the right side does tell users that they are able to filter by consoles and developers, the hover feature could have led users to believe that there was no way to zoom into the individual genre stacks. I solved this issue by using a few CSS styles that changed the cursor into a pointer whenever the user hovered. But this came with another issue in that users might not know how to navigate back to the unzoomed view, so by using the same common design practice of a pointer instead of a cursor, users can return to the original view. While it may have been easier to implement a back button or “unzoom” button, this would provide visual clutter, resulting in more distractions for users. In short, clicking on a genre stack zooms you in and clicking again returns you to the original view with all the stacks displayed.

There was also the issue of small genre stacks which made it difficult or near impossible to see specific data. To address this, I implemented the hover feature that dimmed the other stacks, focusing users on the one they hovered over. Additionally, the zoom feature also made it so that each genre could be inspected.

Time Spent on Project

Most of the time on this project was spent on cleaning up the data and formatting it in a way that d3 could create stacks from. Additionally, the actual implementation of the stacked area chart was a bit confusing as the incorrect data formatting made it seem as if it was an issue with my implementation. Other than those two, I also spent a lot of time on formatting the legend and unanimous color scales, as was previously mentioned. Overall, this project took me around 16 hours including research/planning, pre-processing/formatting, and implementation.