Information visualization Project : Mapping Trends and Dynamics in the Video Game Industry: Insights Through Data and Visualizations

Author
Euan Russel Rodger

Author
Jean Edouard Acker

Project type : Application project

Course
CS5346

Professor
Bimlesh Wadhwa

Contents

1 Introduction

In this project, we explore various aspects of the video game industry, with a particular focus on **game reviews** and their impact, especially to shed lights on **review bombing** and its consequences. We also conduct a deep analysis of the trending genres of video games and which factors explain their popularity.

1.1 Review Bombing

One key area of interest is **review bombing**, a phenomenon where users intentionally flood a game with negative reviews, often for reasons unrelated to its actual quality. By analyzing game reviews, we aim to highlight this trend and assess which types of games (e.g., FPS, MOBA, indie games) tend to receive the most **well-structured and positive** reviews.

This could help identify emerging trends in the gaming industry—are FPS games gaining popularity, or are indie titles on the rise?

1.2 Popular genre

We examine the **most popular genres** of video games and their evolution over time. This analysis could help identify which genres are currently trending and which ones are declining in popularity.

1.3 Potential Insights for Stakeholders

These findings could be beneficial for various stakeholders:

- **Game developers**: Better targeting of players and adjustments to game design strategies.
- **Gamers**: Understanding the gaming landscape and making informed choices about which games to play.

2 Datasets and Pre-processing

This section details the datasets used in the project and describes the pre-processing steps applied to them.

2.1 Steam Games Dataset [1]

The primary dataset used for this project comes from a publicly available dataset on Kaggle entitled *Steam Games*, which contains information about nearly 100,000 games distributed through the Steam platform. This dataset includes 39 features, covering a wide range of metadata related to the games, their developers, and user behavior.

For the purposes of this project, only a subset of relevant features was retained, depending on the needs of each visualization. The selected attributes are described below, along with their type and any remarks on data quality:

- **released_date**: The release date of the game (temporal feature). This was generally converted to a lower temporal resolution, either `YYYY/MM` or simply `YYYY`, to allow for time-based aggregation in visualizations.

- **estimated_owners**: An interval indicating the estimated number of owners of the game (interval variable). To make it usable for quantitative visualization, the midpoint of the interval was computed and used as a representative numeric value.
- **publishers**: The name(s) of the studio(s) that developed or published the game (categorical variable). This was retained as is.
- **name**: The title of the game (categorical variable), also kept without modification.
- **genre**: A comma-separated list of genres associated with the game (categorical variable). This field was either parsed into an array of genres or simplified by retaining only the first genre listed, depending on the visualization.
- **average_playtime_forever**: The average number of hours players have spent on the game (ordinal variable). While useful in theory, this feature sometimes contains suspiciously high values (e.g., 1000 or 2000+ hours), suggesting potential inaccuracies or outliers.
- **median_playtime_forever**: The median number of hours played (ordinal variable). Unfortunately, this field also presents similar inconsistencies, casting doubt on its reliability.
- **price**: The sale price of the game in US dollars (ordinal variable). This value was used without transformation.
- **recommendations**: The number of recommendations left by users on Steam (ordinal variable). This feature was directly used for analysis and ranking.

Preprocessing and Derived Data

This dataset was used to generate three derived datasets, each supporting a specific visualization:

HairballDataset.json

Used in the **Hairball of Steam games released during the selected period**. The preprocessing steps were:

- Retained only the features: `release_date`, `estimated_owners`, `publishers`, `name`, and `genre`.
- The dataset was filtered to keep only:
  - The 100 most productive publishers.
  - The 20 most popular genres.
- All games published by the selected publishers were retained, provided their genres were among the 20 most popular.
- The `release_date` was converted to a monthly granularity.
- In the resulting graph structure, a link (or edge) was created between games published by the same publisher.
- The preprocessing steps were performed in `processingDataset2.ipynb`.
- The final dataset was exported in JSON format.

barChart.json

Used in the **Number of games of each genre released over the selected time period**. The processing pipeline included:

- Retained features: `release_date`, `genre`, and `median_playtime_forever`.
- A new `year` column was extracted from the release date.
- For each year, the number of games released per genre was computed. Since a game may belong to multiple genres, it was counted once for each genre it appeared in.
- The preprocessing was performed in `processingDatasetBarChart.ipynb`.
- The final dataset was exported in JSON format.

bubblePlot.csv

Used in the **Bubble plot of the price vs. number of recommendations**. The steps were as follows:

- A random sample of 4000 games was drawn, from which the following features were retained:
  - `name`
  - `estimated_owners`
  - `price`
  - `recommendations`
  - `median_playtime_forever` (preferred over average playtime due to outliers, e.g., 5000+ hours)
  - `genres` (only the first listed genre was retained to avoid redundancy)
- Games with both `estimated_owners` and `recommendations` equal to zero were discarded.
- The `estimated_owners` field, originally in the form of intervals (e.g., "50,000 – 100,000"), was approximated using the midpoint of the interval.
- After filtering, the final dataset contained approximately 600 games.
- All preprocessing was conducted in `processingDatasetScatter.ipynb`.
- The final dataset was exported in CSV format.

2.2 playTracker Scraped Dataset

The second dataset was obtained by scraping the website *PlayTracker* (`https://playtracker.net`), a platform that aggregates information about video games from various sources and estimates player statistics. The data collection was performed using a custom Python program named `playTrackerScraper.ipynb`, which is included with this report.

This scraper collects information for approximately 15,000 games. Compared to the Kaggle dataset, the PlayTracker dataset offers more reliable and refined player-related metrics. The extracted features are the following:

- **estimated_players**: The estimated number of total players for each game (ordinal variable). This metric is generally more accurate than the ownership estimates provided by the Kaggle dataset.
- **estimated_active_players**: An estimation of currently active players (ordinal variable).
- **average_total_playtime**: The average total time spent in the game by players (ordinal variable).
- **average_recent_playtime**: The average time spent recently (e.g., in the last 2 weeks) by active players (ordinal variable).
- **median_recent_playtime**: The median recent playtime, offering a complementary central tendency measure (ordinal variable).
- **release_date**: Both the original release date and the Steam release date are provided (temporal variables). These were parsed into a lower-resolution format (e.g., `YYYY/MM`) for time-based analysis, similarly to the Kaggle dataset.
- **genre**: The primary genre of the game (categorical variable). This field was parsed into a list for multi-genre games.
- **developer**: The studio or team that developed the game (categorical variable).
- **publisher**: The company responsible for publishing the game (categorical variable).
- **engine**: The game engine used (e.g., Unity, Unreal Engine) (categorical variable).
- **mode**: The game modes available (e.g., single-player, multiplayer) (categorical variable).
- **perspective**: The visual perspective used in the game (e.g., first-person, third-person) (categorical variable).
- **theme**: Thematic elements (e.g., Action, Stealth, Adventure) (categorical variable).
- **additional_tags**: Extra tags related to game features or mechanics (e.g., AI, Minigames) (categorical variable).

All of this information was compiled into a CSV file for ease of use and further processing. In terms of preprocessing, the operations applied were minimal:

- **estimated_players**, **estimated_active_players**, **average_total_playtime**, **average_recent_playtime**, **median_recent_playtime**: These were left unchanged.
- **release_date**: Converted into lower-resolution timestamps as done for the Kaggle dataset.
- **genre**: Parsed into lists when necessary.
- **all other fields**: Retained as is.

Preprocessing and Derived Data

Two derived datasets were constructed to support visualizations related to game engine usage: `engine_hist.csv`, used in the **Histogram of the most used game engine**, and `StackedEngine.csv`, used in the **Stacked bar chart representing the percentage of players per game engine**.

engine_hist.csv

This dataset was generated from a larger dataset by processing the `engine` column to unify different versions of the same engine under a single label. For instance, entries such as `Unreal Engine 3`, `Unreal Engine 4`, and `Unreal Engine 5` were grouped under a common label `Unreal Engine` using regular expressions. This normalization enables a more accurate comparison between engines.

The final steps were:

- The cleaned dataframe was exported to CSV format.
- All preprocessing was performed in the notebook `processingEngineData.ipynb`.

StackedEngine.csv

This dataset was built to facilitate temporal and genre-based analysis of game engine popularity, especially in relation to estimated player counts. The construction pipeline is as follows:

- Retained columns: `estimated_players`, `release_date`, `genre`, `developer`, and `engine`.
- Rows with missing values in the `genre` or `engine` columns were removed.
- The `engine` column was normalized, similar to the method used for `engine_hist.csv`, using regular expressions to unify versions of the same engine.
- Any remaining rows with missing values were dropped.
- The release date was parsed into a new `month` column, and the dataset was sorted chronologically by this field.
- The `genre` column, containing strings of genres, was parsed into lists.
- A new dataframe was created with the columns: `month`, `engine`, and one column for each genre.
- For each row in the original dataset, a number of rows equal to the number of genres associated with the game were added to the new dataframe. In each row, the genre-specific column was assigned the estimated number of players for the game. This allows genre-based estimation of game popularity.
- The resulting dataset was grouped by `month` and `engine`, and aggregated to compute the total estimated number of players for each genre, per month and engine.
- The final dataframe was exported as a CSV file.
- All preprocessing was performed in the notebook `processingEngineStacked.ipynb`.

2.3 SteamDB Dataset

The third dataset used in this project was obtained from *SteamDB* (`https://steamdb.info/app/753/charts/#max`), a platform that tracks real-time statistics and historical trends on the Steam platform. This dataset provides a time series of the total number of concurrent users on Steam at specific time intervals.

Preprocessing and Derived Data

This dataset was only used to create the `SteamUserLineChart.csv` dataset for the **Line Chart representing the number of steam user over time**, missing values (particularly in the early stages of Steam) are filled in with the first available values

2.4 Steam Reviews Dataset (2021)

The fourth dataset used in this project is the *Steam Reviews Dataset 2021*, available on Kaggle[1]. This dataset consists of approximately 21 million user reviews covering around 300 games. The data was originally collected using the official Steam Web API.

The dataset includes 23 features, of which the most relevant for our study are:

- **App name**: Name of the game or application (categorical variable).
- **review**: The textual content of the user review (textual variable).
- **timestamp_created**: Timestamp indicating when the review was submitted (temporal variable).
- **recommended**: Boolean flag indicating whether the review is positive or not (categorical variable).

Preprocessing and Derived Data

Several steps were taken to clean and preprocess this dataset in order to extract temporal and sentiment-based signals:

- **Data Cleaning**: Missing and duplicate entries were removed, the index was reset, and data formats were standardized.
- **Date Conversion**: The `timestamp_created` field was converted into weekly periods to allow for temporal aggregation and trend analysis.
- **Review Aggregation**: Reviews were grouped by both `app_id` and `date_created` to compute weekly totals. The number of positive reviews was also recorded. A new column, `negative_prop`, was calculated as:

$$\texttt{negative\_prop} = 1 - \left(\frac{\texttt{positive\_reviews}}{\texttt{total\_reviews} + 1} \right)$$

  The "+1" in the denominator prevents division by zero and mitigates extreme values due to low counts.

- **Review Bomb Detection**: To identify potential review bombing events, rolling averages and standard deviations were computed over a specified time window (e.g., 7 weeks). Z-scores for both `total_reviews` and `negative_prop` were calculated to detect anomalies:
  - A spike in `total_reviews` (high Z-score) suggests a surge in attention.
  - A spike in `negative_prop` (high Z-score) suggests a surge in dissatisfaction.

[1] `https://www.kaggle.com/datasets/najzeko/steam-reviews-2021`

Boolean flags were set to identify weeks where these metrics exceeded predetermined thresholds.

- **Genre Analysis**: The review data was merged with genre information, and genre lists were exploded into individual rows. This enabled the computation of review bomb statistics per genre.

Generated Dataset

The preprocessed data was exported to a CSV file named `steam_bomb_flags.csv`, which contains the following features:

- **app_id**: Unique identifier for each game on Steam.
- **date_created**: Start date of the corresponding weekly period.
- **total_reviews**: Total number of reviews during the week.
- **positive_reviews**: Number of positive reviews during the week.
- **negative_prop**: Proportion of negative reviews.
- **rolling_mean**, **rolling_std**: Rolling average and standard deviation of total reviews.
- **z_score**: Z-score for review volume, indicating statistical outliers.
- **neg_rolling_mean**, **neg_rolling_std**: Rolling average and standard deviation of `negative_prop`.
- **neg_z_score**: Z-score for negative sentiment trends.
- **review_bomb_flag_volume**: Boolean flag indicating a potential review bomb based on review volume.
- **review_bomb_flag_neg**: Boolean flag indicating a potential review bomb based on negativity.

Derived Summary Datasets

Several additional summary datasets were created to analyze review bombing patterns across different dimensions, such as genre, price, language, and special conditions like early access or free distribution.

genre_review_bombs.csv

This dataset aggregates review bomb statistics across different genres. Each row corresponds to one genre (e.g., Action, RPG), with the following attributes:

- **Genre**: Genre label of the game.
- **total_reviews**: Total number of reviews for games in this genre.
- **total_positive_reviews**: Total number of positive reviews.
- **total_negative_reviews**: Computed as `total_reviews` − `total_positive_reviews`.

- **total_bombs_volume**: Number of review bombs flagged due to volume spikes.
- **total_bombs_neg**: Number of review bombs flagged due to negative review spikes.
- **avg_negative_bombs_per_game**: Average number of negativity-based bombs per game in the genre.
- **bomb_rate_volume**: Proportion of volume-based bombs, computed as:

$$\texttt{bomb\_rate\_volume} = \frac{\texttt{total\_bombs\_volume}}{\texttt{total\_reviews}}$$

- **bomb_rate_neg**: Proportion of negativity-based bombs:

$$\texttt{bomb\_rate\_neg} = \frac{\texttt{total\_bombs\_neg}}{\texttt{total\_reviews}}$$

price_review_bombs.csv

This dataset analyzes the distribution of review bombs across price bins:

- **Price_Bin**: Interval representing the price range (e.g., 0–5, 5–10).
- **total_reviews**, **total_positive_reviews**, **total_negative_reviews**: Review counts per bin.
- **total_bombs_volume**, **total_bombs_neg**: Number of volume-based and negativity-based bombs respectively.
- **bomb_rate_volume**, **bomb_rate_neg**: Same as above, computed as proportions over `total_reviews`.

language_posneg.csv

This dataset tracks the sentiment distribution of reviews based on their language:

- **language**: Language of the review.
- **Positive**, **Negative**: Total number of positive and negative reviews.
- **Positive%**, **Negative%**: Computed as:

$$\texttt{Positive\%} = \frac{\texttt{Positive}}{\texttt{Positive} + \texttt{Negative}}, \quad \texttt{Negative\%} = \frac{\texttt{Negative}}{\texttt{Positive} + \texttt{Negative}}$$

received_for_free_posneg.csv

This dataset distinguishes sentiment distributions based on whether the game was received for free:

- **received_for_free**: Boolean variable (True or False).
- **Positive**, **Negative**: Review counts.
- **Positive%**, **Negative%**: Sentiment proportions, computed similarly as above.

written_during_early_access_posneg.csv

This dataset analyzes sentiment depending on whether the review was written during the early access phase:

- **written_during_early_access**: Boolean flag.
- **Positive**, **Negative**: Review counts.
- **Positive%**, **Negative%**: Sentiment proportions.

3 Visualization analysis

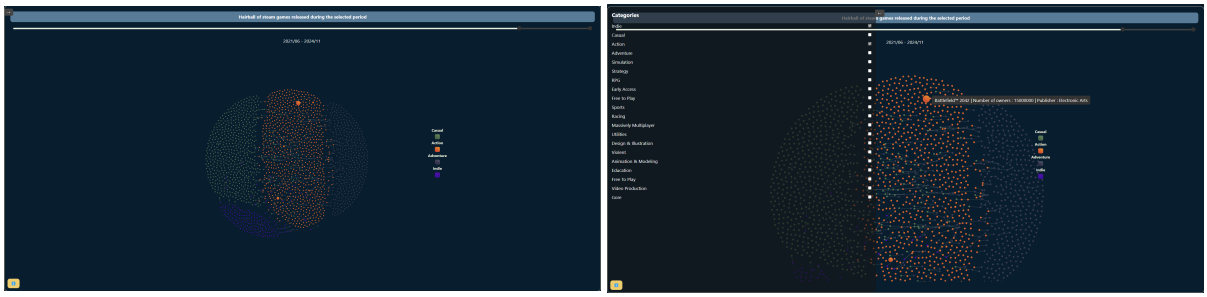

Figure 1: Hairball of steam games released during the selected period

Hairball

This interactive network graph offers an engaging way to explore Steam's game catalogue. Each **node** represents a game, and the **color** indicates its genre. Links are drawn between games developed by the **same studio**, helping to highlight clusters of productions.

The visualization is useful for both **gamers** and **developers**:

- **Gamers** can discover other popular games and track games published by their favorite studios.
- **Developers** can identify currently popular genres or follow the career path of successful studios. By observing the games they released over time (via the release date in the tooltip), developers can evaluate what type of game and level of ambition is realistic at each stage.

Controls

- A **double slider** allows selecting a date range to filter the games displayed.
- A **side panel** filters which genres to display.
- Two sliders let you control the **intra-link** and **inter-link** density to avoid visual clutter or performance lag.
- The visualization can be **zoomed and panned** for better exploration.
- Each node is **hoverable** to display additional details (game title, genre, release date, studio, etc.).
- Nodes can be **grabbed and moved** manually for custom layouting.

Design Choices

- The hairball resembles a **game map**, reinforcing the ludic aspect of this storyboard.
- The **simulation-based layout** of this force-directed graph echoes the physical simulations in game engines, making the choice of this design coherent with the theme.

Number of game of each genre released over the time period selected

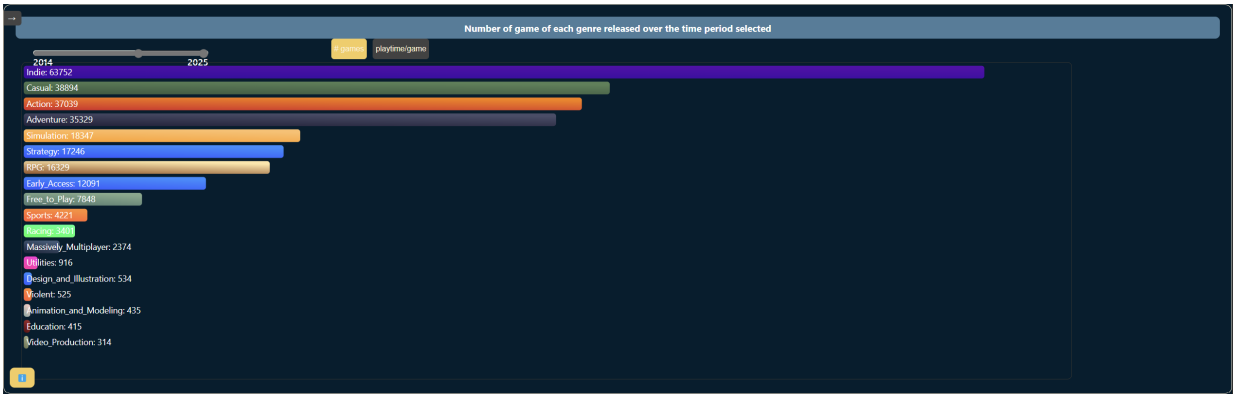

Figure 2: Number of game of each genre released over the time period selected

This plot is a horizontal bar chart representing the number of games per genre released during the selected period. It allows developers and analysts to assess the popularity of each genre at a given time. By analyzing this trend, developers may choose to focus on a genre that has been gaining momentum in recent years.

While the Kaggle dataset includes suspiciously high values for average playtime—sometimes over 1000 hours, which is quite unrealistic—the dataset scraped from PlayTracker provides more coherent values. However, since PlayTracker only includes around 15,000 records, this visualization sticks with the Kaggle dataset. It is therefore recommended to only toggle the number-of-games variant.

The visualization shows a clear dominance of indie games in recent years. This can be explained by the rise of free and accessible game development tools such as Unity and Unreal Engine, empowering individuals to create and publish their own games. This phenomenon is also explored in the **Engine Usage Histogram** and the **Stacked Histogram** below.

Controls

- A **slider** lets you select a custom date range for filtering releases.
- A **side panel** enables you to filter by specific genres of interest.
- A **toggle button** switches between visualizing the raw number of games and the ratio: *median playtime / number of games*, which serves as a proxy for genre popularity.

Bubble plot of the price vs number of recommendations

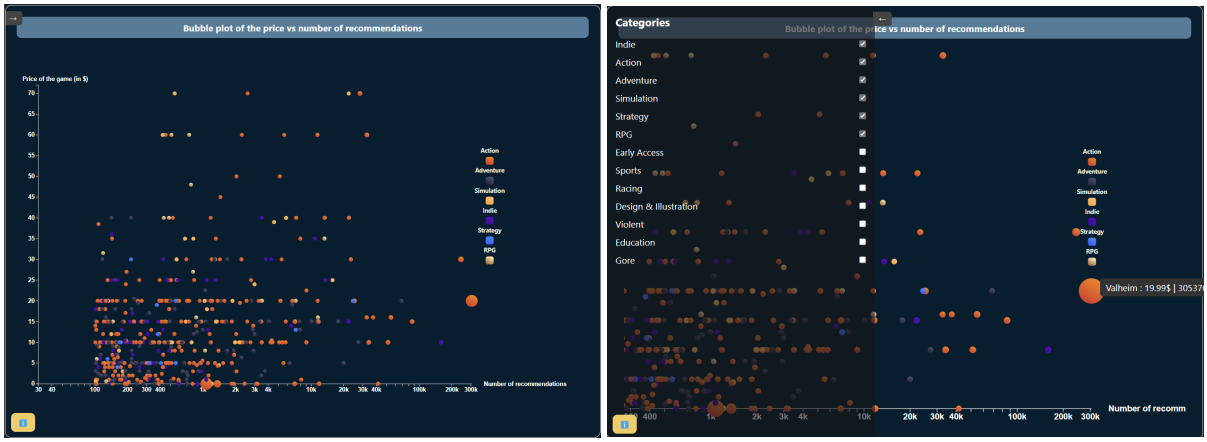

Figure 3: Bubble plot of the price vs number of recommendations

This bubble plot highlights the distribution of games according to their price and the number of recommendations. We can observe that most games are priced below $20, and only a few receive more than 3000 recommendations. The most played and recommended games were action games. However, this observation should be taken with caution, as the dataset used is a small sample of the full Steam library, and we assumed the predominant genre of a game is the first listed in the dataset.

Controls

- **X axis:** Number of recommendations (logarithmic scale, to avoid point squeezing).
- **Y axis:** Game price.
- **Color:** Represents the game genre.
- **Size:** Proportional to the number of game owners.

Design choices

- Bubbles appear progressively to enhance visual engagement.
- Side panels allow filtering by genre of interest.
- Bubbles react on hover, showing a tooltip with detailed information.
- The visualization supports pan and zoom interactions for better exploration.

Line Chart representing the number of steam user over time

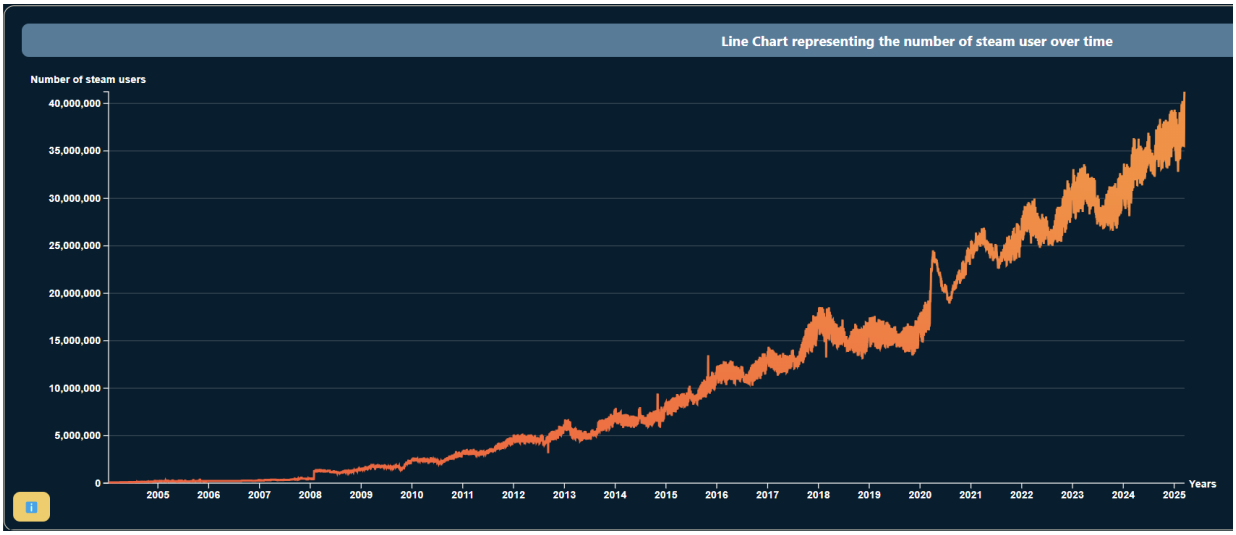

Figure 4: Line Chart representing the number of steam user over time

This plot highlights the **popularity of Steam** as a game marketplace. Thanks to **SteamDB**, user data is easily retrievable — unlike for other platforms like Epic Games, Origin, or Instant Gaming.

Design Choices

- The **Y-axis** represents the **number of users**.
- The **X-axis** represents **time**.
- A **tooltip** reveals more detailed information when hovering over the chart.
- **Ticks** make it easier to read exact user counts.
- The chart supports **zooming and panning** for in-depth exploration.
- The graph is **animated on creation**, and the animation can be replayed via a **Replay** button.

Histogram of the most used game engine

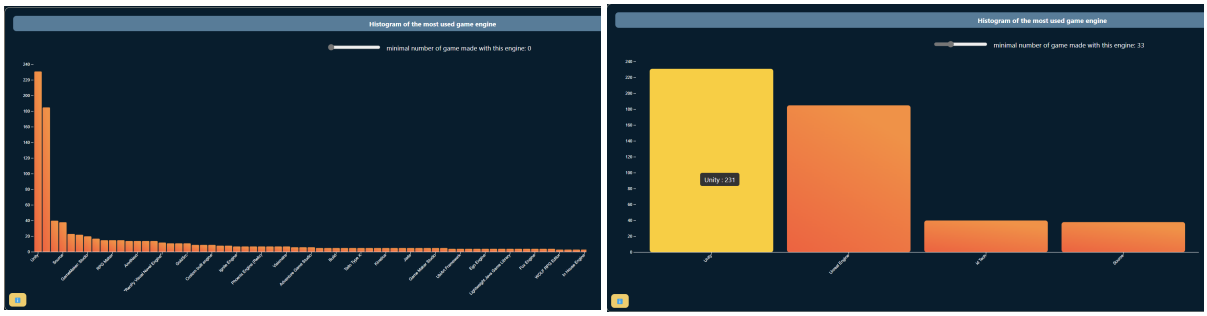

Figure 5: Histogram of the most used game engine

This histogram represents either the **number of games** or the **percentage of games** developed with each game engine.

This visualization allows developers to identify **popular engines** for their game development.

Controls

- A slider allows selecting the **minimum number** of games (or the **minimum percentage** over the dataset) developed with each engine.
- You can switch between displaying the **number of games** and the **percentage of games** using a toggle button.

Design Choices

- Bars are **sorted** to make comparisons easier.
- The histogram is fully **animated** for a more engaging experience, and the animation can be **replayed** using the replay button.
- Zooming is enabled on the **x-axis** for better exploration.
- The number of **x-axis labels** adapts depending on the number of displayed engines, preventing overlap.
- A **tooltip** provides detailed information about each engine, and bars **change color** when hovered for better interactivity.

Stacked bar chart representing the percentage of players per game engine, specifically for the selected genre and over the selected time range

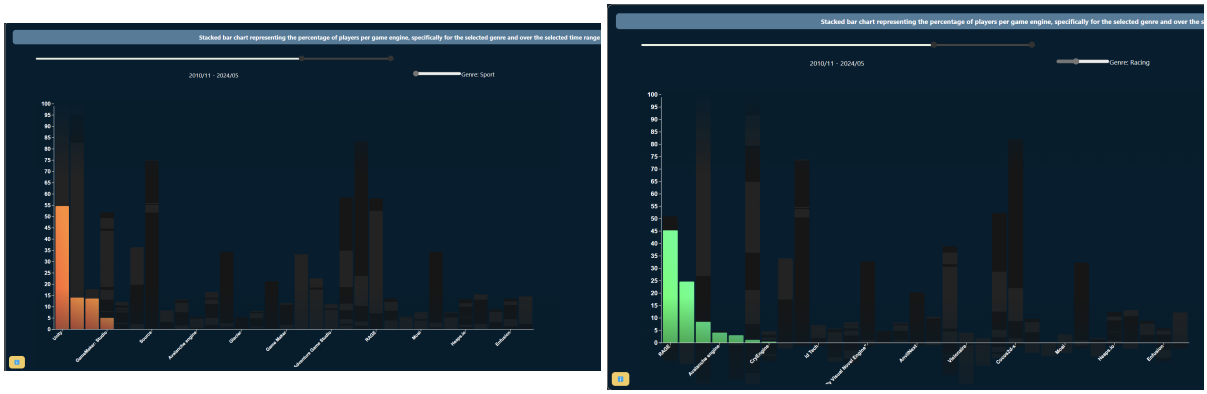

Figure 6: Stacked bar chart representing the percentage of players per game engine

This stacked bar chart represents the **percentage of players per game engine**, specifically for the selected genre. The percentage is averaged over the selected time range.

This visualization is a powerful tool for:

- **Game developers**, who can assess which engine is most suitable for the type of game they want to build.
- **Game engine creators**, who gain insights into how their tools are being used in relation to game genres.

Controls

- A **double slider** allows you to select a range of dates to analyze.
- The number of estimated players is averaged across this date range.
- Since the scraped dataset is only a sample of all available Steam games, we use the **percentage of estimated players** as a metric of popularity. This allows us to measure which engine was the most popular for developing a specific genre at a given time.
- Another slider lets you **scroll through 20 different genres**.
- When you change the selected genre, the corresponding layer in every bar moves up or down to align on the X-axis. This makes it easier to compare the popularity of engines for a specific genre.

Design Choices

- Bars are sorted by the percentage of players for the selected genre to ease comparison.
- The entire histogram is **animated** to make the visualization more dynamic and engaging.
- Initially, the different layers appear progressively. This animation can be **replayed** using the replay button or by adjusting the date range.
- The layer corresponding to the selected genre is **the only one colored**. Other layers are rendered in various shades of gray to emphasize the genre of interest.

- The grayscale tones and moving bars **evoke the aesthetic of scrolling urban landscapes**, a visual motif that fits well within this storyboard's theme.
- The chart supports **zooming on the X-axis** for more precise comparisons.
- The number of X-axis labels is dynamically adjusted based on the number of engines shown, ensuring that labels don't overlap.
- A **tooltip** provides more detailed information when hovering over each engine.
- Since bars may extend beyond the chart bounds, a **fade-out mask** is used to gradually hide the tops and bottoms of overflowing bars, creating a smooth visual effect.

Review bombing analysis

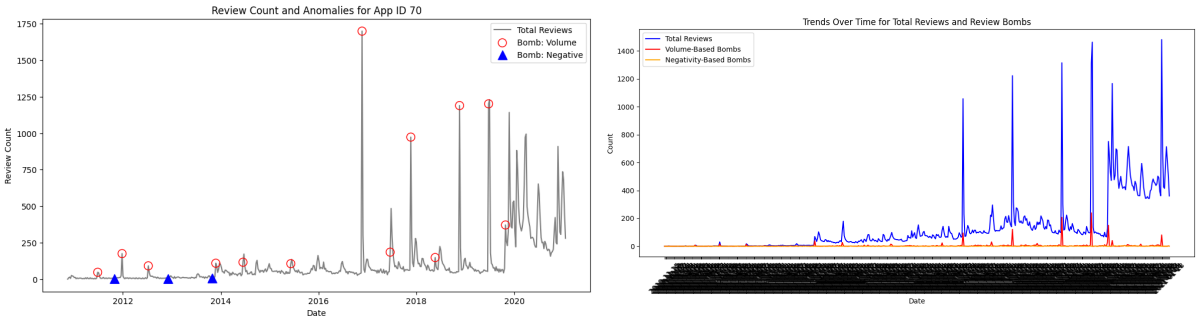

Figure 7: Review bombing detection

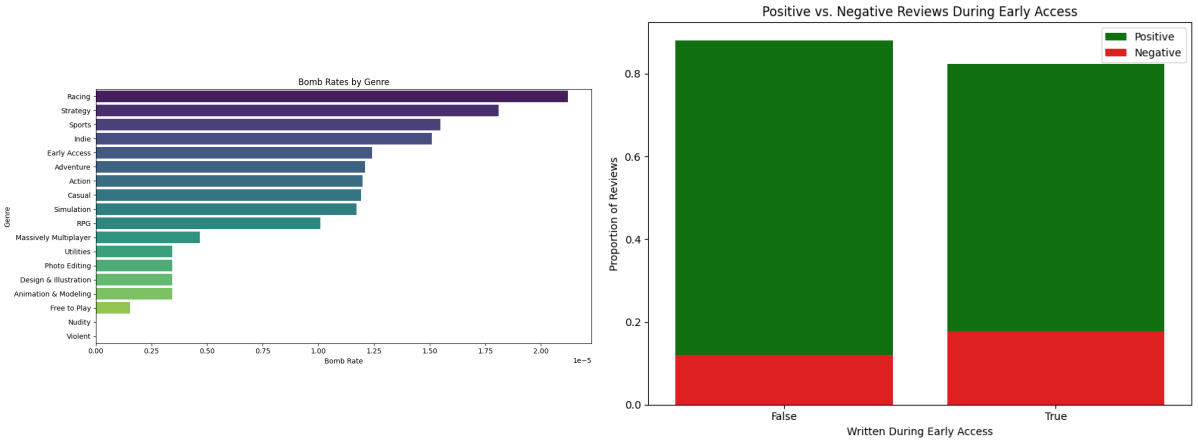

Figure 8: Bomb rate by genre (left) and influence of negative ratings on review bombs (right)

In this section, we address the following research questions:

- What are the trends over time for total review bombs?
- Do negatively rated games get review bombed more or less?
- Are some genres impacted more or less by negative reviews and review bombs?
- How does the price of a game influence how it is reviewed? Are cheaper games more frequently review bombed due to their lower price point?
- How does the distribution of positive and negative reviews vary across languages, gifted games, and early access games?

Details of the Visualizations

1. Stacked Bar Charts

Encoding:

- **X-axis:** Categories (e.g., genres, languages, early access status).

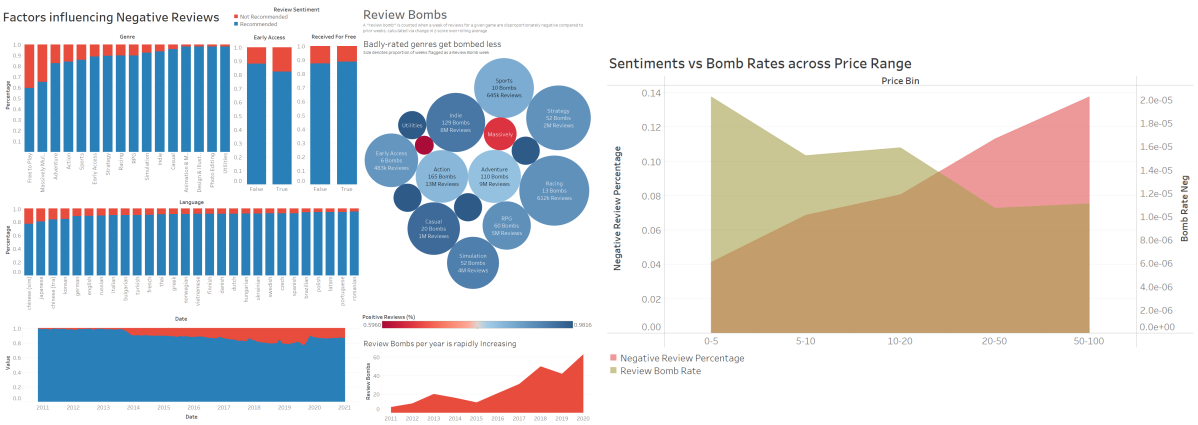

Figure 9: Dashboard with factor influencing review bombs rate (left) and price vs review bomb rate (right)

- **Y-axis:** Percentage of reviews (stacked to 100%).
- **Color:** Blue for positive reviews, red for negative reviews.

Design Choices:

- Bars are sorted by the percentage of negative reviews to facilitate easier comparison.
- Consistent bar lengths highlight proportions rather than absolute values.

2. Stacked Area Chart

Encoding:

- **X-axis:** Time (weekly intervals).
- **Y-axis:** Percentage of reviews (stacked to 100%).
- **Color:** Blue for positive reviews, red for negative reviews.

Design Choices:

- 100% stacking is used to emphasize relative changes over time.
- Smooth transitions between time intervals enhance readability.

3. Dual-Axis Area Chart

Encoding:

- **X-axis:** Game price (binned).
- **Left Y-axis:** Percentage of negative reviews (red area).
- **Right Y-axis:** Review bomb rate (yellow area).
- **Color:** Red for negative reviews, yellow for review bombs.

Design Choices:

- Dual axes allow for the simultaneous representation of different magnitudes.
- Colors are chosen to represent "negative" attributes while avoiding redundancy: red for negative reviews and yellow for review bombs (blue is already used to denote positive aspects in other charts).

4. Bubble Chart

Encoding:

- **Bubble Size:** Review bomb rate (proportion of weeks flagged).
- **Bubble Color:** Positive review rate (red-to-blue gradient).

Design Choices:

- Positional encoding is omitted, as all key variables are represented through size and color.
- Larger bubbles include labels with summary statistics, such as total number of reviews and review bombs for each genre.

5. Area Chart for Review Bombs Over Time

Encoding:

- **X-axis:** Time (weekly intervals).
- **Y-axis:** Total number of review bombs.

Design Choices:

- A minimalist design emphasizes the global trend without distraction.

Conclusion

Through a series of interactive visualizations, we examined the evolution of game genres on the Steam platform and the tools used to create them. The hairball highlighted relationships between studios and their catalogs.

The bar charts and histograms provided quantitative insights into genre popularity, confirming the rise of indie games over the last decade. This surge can be attributed to the democratization of game development, thanks to accessible engines like Unity and Unreal. Complementary visualizations, such as the engine usage histogram and the stacked bar charts, revealed how certain engines became closely associated with specific genres, particularly within the indie scene.

Altogether, these visualizations not only depict the diversity and evolution of the Steam game ecosystem but also offer strategic insights for both developers and analysts. They can identify viable genre-engine combinations, follow market shifts, and align their development choices with evolving player preferences.

We also explore the Steam reviews dataset to analyze trends in user sentiment and review bombing. We first examined factors influencing negative reviews, visualized through stacked bar charts. Microtransaction-heavy genres, such as Free to Play and Massively Multiplayer, had a significantly higher share of negative reviews. In contrast, genres for applications rather than games were reviewed much more positively. For languages, Chinese, Japanese, and Korean reviews showed the highest negative sentiment compared to others, potentially reflecting cultural or regional differences in expectations.

A stacked area chart complements these findings by showing the proportion of negative reviews over time. While negative sentiment has gradually increased, a slight positive shift occurred at the end of 2019, likely due to Steam prompting players with substantial playtime to update their reviews and implementing measures to counter review bombing.

A review bomb refers to a sudden, abnormal spike in negative reviews within a short timeframe, often unrelated to the game's quality. Since no publicly available datasets exist for review bombs, we created one by tagging these spikes. The bubble chart reveals that negatively rated genres are not necessarily subjected to more review bombs, as they exhibit a much lower relative number of review bombs despite their low review scores.

Finally, we analyzed the relationship between price and reviews. A dual-axis area chart shows that while more expensive games tend to receive more negative reviews, cheaper games are more vulnerable to review bombing. This highlights how price influences both sentiment and susceptibility to review manipulation.

References

[1] Martin Bustos Roman. Steam games dataset, 2022.