Tracking Dream SMP's Twitch Viewership Movements

Sabina Chen
sabinach@mit.edu

Abstract

We present Tracking Dream SMP's Twitch Viewership Movements, an interactive visualization to explore underlying viewership movement information between Twitch streamers within the Dream SMP community. Using real-time data gathered from the Twitch API, viewership movement can be mapped between streamers across different points in time. Our visualization uses animations to encode movement direction and viewership values. It also features a Minecraft-themed pixelated game design to encourage users to engage with the visualization. We show that our visualization enables users to take an exploratory approach to viewing real-world viewership data and be able to find basic trends in streamer interactions within the community.

1 Introduction

Twitch is a live-streaming platform for gamers and other lifestyle creators to "broadcast" their gameplay and life online, so that fans can hear, watch, and interact with the creators in real-time through an interactive messaging platform [1]. Twitch streamers can choose to stream a wide variety of games. Some of the more popular games include GTA, League of Legends, Call of Duty, Fortnight, Among Us, etc.

Depending on the popularity of the game, communities of viewers are usually formed around the games themselves, or around the specific content creators and their friend groups. One such community that has recently grown in popularity is the Dream SMP community [2], which is a subset of streamers under the Minecraft community [3], who have become known for streaming role-play under the Dream SMP Minecraft server. Top streamers within the Dream SMP community often average 100k+ viewers for every stream. For example, TommyInnit's recent lore stream on April 29, 2021 averaged over 600k+ concurrent viewers! These viewership values are huge, compared to normal Twitch streamers who tend to average anywhere between 0-1000 viewers, which is what makes this community of streamers interesting to explore.

As avid Twitch viewers, one thing we noticed is that when a popular streamer goes online, viewership numbers for other streamers within the same community will usually go down. This is not just the case for streamers within the Dream SMP community, but across all communities i.e. OfflineTV. This is expected, because viewers who watch one streamer will usually also watch another streamer within the same community. This begs the following questions though:

- What are the underlying viewership movements between streamers within the same community?
- How does Twitch viewership evolve through time?
- Which streamers consistently hold the most number of active viewers at any given time?

Enabling the ability to explore these questions is the basis of our visualization. Our visualization focuses on mapping viewership movements between streamers between separate points in time. Using the D3 framework, we developed an interactive visualization that enables users to explore viewership dynamics between members of the Dream SMP Minecraft community.

2 Related Work

Our visualization was largely inspired by *Visualizing Twitch Communities* by Kiran Gershenfeld [4]. *Visualizing Twitch Communities* provides a high-level overview of the overall Twitch streaming landscape across separate snapshots in time. The static graphs produced using Gershenfeld's visualization algorithm represent data from the top 100 top streamers across one week of each of the specified months. Streamers are grouped together into color-coded "communities", which are defined to be collections of streamers watched by the same viewers. Independent static visualizations have been generated using Gephi for the months of December 2020, February 2021, and March/April 2021. After seeing how the separate static graphs evolved from month-to-month (ie. streamers and communities shrinking/growing depending on popularity, shared viewership changing between streamers, new communities being formed), we were curious to understand how exactly viewers were moving between these individual snapshots of time.

For our visualization, we chose to focus on only the streamers within the Minecraft community, specifically streamers who are members of the Dream SMP Minecraft server. By focusing on a smaller community, viewership movements can be more effectively and thematically mapped. Because the Dream SMP Minecraft server focuses mainly on role-playing, previous data visualizations developed tended to focus more on character relationships [5], story timelines [6], or YouTube subscriber counts [7]. Thus, for our visualization, our goal was to visualize real-world Twitch viewership relationships between streamers within the Dream SMP Minecraft community.

3 Methods

In this section, we describe our dataset and the dataset retrieval process, as well as the tools and techniques used to build our visualization.

3.1 Dataset

3.1.1 Retrieving Data

Data is retrieved in real-time using the Twitch API [8]. API requests are scripted in NodeJS via Axios. Using a background cron scheduler, we ping the API to retrieve unique viewership IDs of all the verified viewers in chat for each live streamer hourly, at 15 minutes past the hour. The raw data is stored in a .json file that is later used to determine viewership movement statistics.

Data is retrieved at 15 minutes past the hour, as opposed to on the hour, because (1) there is a known delay in updating viewership info innately within the Twitch API, and (2) streamers often start streams on the hour. Therefore, to allow time for both viewers to come into the stream and for the API to update, we chose to retrieve data at the 15-minute mark. The raw data retrieved includes id information on the broadcaster, moderators, vips, admins, and viewers.

To determine which streamers to retrieve data from, we keep a list of all known Dream SMP members in a separate .csv file, which is updated as new members are added to the server. This list includes information on each member's preferred character name, main Twitch account id, and alternative Twitch account id (since some streamers stream on multiple accounts). We only retrieve data for streamers on the list who are currently live at the time of data retrieval. The Twitch API does not store old data, therefore, in order to save a history of all of our desired datetimes, all data must be retrieved in real-time and stored locally.

3.1.2 Processing Data

After the raw data has been retrieved, a separate cron scheduler used to calculate viewership statistics is executed immediately

afterwards. Active viewership numbers for each streamer are automatically included in the raw data.

To calculate the number of viewers who have moved from one streamer to another, viewer ids are mapped between the raw data of each live streamer of the previous hour, to the raw data of each live streamer of the current hour. Viewer ids that existed in the previous hour, but are no longer available in any of the live streamers of the current hour, are deemed to have gone "Offline". Our definition of "Offline" means that either the viewer has logged off of Twitch or that they have moved on to watch another streamer that lies outside the Dream SMP community. Because our visualization focuses only on viewership relationships within the Dream SMP community, any viewers outside the community are discounted in this way and considered "Offline".

Finally, the overall minimum and maximum viewership statistics across all datetimes are then calculated and stored. These min/max values are used later to normalize the sizes of the nodes and links of our visualization for visual consistency purposes (ie. 250k viewers should always create nodes of 110x110px regardless of datetime).

3.2 Development

The majority of the visualization is scripted using the D3 (version 3) visualization library [9].

3.2.1 Interactive Graph

Our primary visualization is the node graph located at the center of the web page. This visualization shows how viewership numbers change between streamers within the Dream SMP community at specific points in time. Each streamer is represented by their associated Minecraft skin. The size of each node visually encodes the number of viewers currently watching the selected member's stream. The link thickness encodes how many viewers have moved from one stream to another compared to the previous hour. Hovering over the nodes and links provide information on both active viewership counts and movement numbers, respectively. Legends are placed symmetrically on either side of the node graph to specify the meaning of the nodes and links.

Interactive elements such as pressing the "back", "forward", and "play" buttons enable users to have greater control over the desired datetime to view. Pressing the "play" button will automatically step through each datetime available. All datetimes are in EST. The overall visualization is Minecraft themed, since the majority of streamers within this community are known for their Minecraft streams along with their role-playing activities on the Dream SMP Minecraft server.

3.2.2 Nodes

The nodes are drawn using D3's built-in rects, images, and text attributes. Nodes representing each Dream SMP member are positioned in a circle in the node graph. Hovering over the nodes provides the exact active viewership values for the specific node, along with the total active viewership number within the Dream SMP community at that specific datetime. Minecraft health hearts also pop up on hover to more effectively encode this ratio (comparing how many hearts are filled/not filled is more intuitive than trying to calculate the ratio of numeric values).

To create these nodes, cartesian coordinate information for each node must first be calculated. To ensure that the nodes are evenly spaced around the circle, the step angle between each node is calculated by dividing 2π by the number of Dream SMP members. The radius of the node graph is then hardcoded. Using the polar coordinates found for each node (ie. angle and node graph radius), the cartesian coordinates can finally be calculated using geometry.

The "Offline" node is the only node that is placed slightly outside the specified node graph diameter. This is to help differentiate movement going in and out of the circle, and to ensure that users do not confuse the "Offline" node with a regular Dream SMP member node.

Node sizes are determined using D3's built-in scaleLinear function by mapping the minimum and maximum viewership values calculated during (Section 3.1.2 Data Processing) to the minimum and maximum desired dimensions of the input images. The node size of each individual Dream SMP member is mapped within this range, and visually encodes the number of viewers currently watching the selected member's stream.

3.2.3 Links

The links are drawn using D3's built-in SVG path vectors. Links represent viewership movement from one stream to another. Animated webflow lines move from source to target nodes on the respective links to encode the movement direction.

On hovering over a node, the links connected to the selected node will turn either red or green, while other links not directly relevant to the hovered node are greyed out. Red links represent viewership moving outbound from the selected node, while green links represent viewership moving inbound into the selected node.

Hovering over the links themselves will provide exact viewer movement numbers between nodes. The associated full body minecraft skins of the source node also pop up on hover to represent how the specified viewership movement values compare to the total viewership movement value at that specific datetime. Link thickness is determined similarly to that of calculating node sizes, described in (Section 3.2.2 Nodes), by using scaleLinear to map the current viewership movement value to the min/max viewership movement range and its associated link thickness size.

3.2.4 Slider

The slider located at the top of the web page enables users to control which datetime data to load, which consequently updates the node graph. The available datetimes range from May 02, 2021 to May 19, 2021, as this was the timeframe we were able to retrieve data for via the Twitch API. The slider handle is represented as a grass block, as per our visualization's Minecraft theme. Dragging the grass block enables users to quickly view the streaming landscape at different desired times.

On pressing "play", the slider will step through all the available datetimes at one second intervals, to simulate the passage of time. Each step jumps forward one hour in time. On each slider step update, the node graph is cleared and redrawn to ensure that all the data and visualization elements (ie. links, nodes, text) from the previous datetimes are not unintentionally propagated. On pressing the "left" and "right" buttons, users can step through each datetime individually, giving them greater control when interacting with the visualization.

3.2.5 Minecraft Theme

The visualization is Minecraft themed because the Dream SMP community is known for their Minecraft role-playing lore (ie. story) streams. This theme is propagated across all elements of the visualization. For example, all text uses either the Minecraftia font or MinecrafterAltRegular font, which mimics the font used in the actual game.

Visualization elements are replaced with Minecraft-themed assets, such as the grass block for the slider handle, health hearts for the active viewership ratio, full body skins for the viewership movement ratio, Dream SMP character faces for the nodes, and even the background image of the web page, which is an image texture of the white wool found by shearing sheep in the game. Additionally, non-black font colors, such as the "play" button and text links, were also chosen to match the colors found in typical Minecraft grass blocks, i.e. dark green and dark brown. Overall, the majority of visual assets in this visualization were taken from either the Minecraft Wiki or Dream SMP wiki.

4 Results

Visualizing Dream SMP's Twitch Viewership Movements enables users to compare viewership statistics in an interactive manner. It achieves this by including summary overview, interaction tips, statistics, and tool tips. To encourage users to engage with the visualization, our visualization is themed around a pixelated game design inspired by the Minecraft game itself.

4.1.1 Interactive Graph

The interactive node graph shows the active viewership numbers for each member of Dream SMP, along with viewership movement information between streamers within the same community. To avoid overwhelming the user, nodes and links within the node graph can be individually selected on hover by greying out all irrelevant links. This enables users to be able to focus on only a few nodes and link interactions at a time.

The node graph can be updated by datetime via dragging a slider located on the top of the web page. This slider can also be automatically executed by pressing the "play" button. To ensure that this functionality is not overlooked, the "play" button is intentionally chosen in another font and color, to differentiate the text from the rest of the visualization.

Legends mapping link thickness and node sizes to viewership movement numbers and active viewership counts, respectively, are located symmetrically on either side of the nodegraph. Additionally, usage instructions and helpful hints are displayed at the top of the web page to guide users in learning how to interact with the visualization.

4.1.2 Statistics

Hovering over each node and link displays text at the center of the node graph, so that users who are interested in seeing the exact viewership values can do so without obstruction. Minecraft-themed assets in the form of health hearts and full body character skins are also displayed below this numeric text. These playful images enable users to more easily compare the ratio of values between the selected and total viewership numbers.

5 Discussion

5.1.1 What are the underlying viewership movements between streamers within the same community?

One of the driving forces of this visualization was to understand where viewership was coming from, especially when large streamers go live. Do the majority of viewers come from other streamers from within the same community? Or do they come from offline? Also, what happens after the stream when these large streamers go offline? Where do the viewers disperse? Our visualization allows us to answer these questions.

Let's look at the May 03, 2021, 2pm EST case study. Looking at the visualization, we can see that GeorgeNotFound goes online during this time. Based on the size of the node and by comparing it to that of the legend, we can see that his active viewership count is around 190k. On hovering over the node, we can see that he actually has exactly 191,404 active viewers at this specific point in time. Additionally, looking at the size of the link moving from Offline to GeorgeNotFound compared to all the other links in the graph, although a large portion of viewers do still come from other streamers, most viewers moving into GeorgeNotFound's stream are coming from offline. An hour afterwards, on May 03, 2021, 3pm EST, we can see that GeorgeNotFound goes offline, and that the majority of his viewers also follow him offline, as opposed to going to other streamers in the community.

While for this specific case study, we can answer that, yes, for large streamers, the large majority of their viewers come from offline as opposed to other streamers within the community, this answer can vary depending on the exact streamer and viewership scenario, and can be determined on a case-by-case basis. However, as shown in this example, using our visualization we can answer the questions posed above.

5.1.2 How does Twitch viewership evolve through time?

Using the visualization, we can also determine the approximate streaming schedules of each streamer through observation. While the most number of concurrent streamers active are usually around the evenings (ie. between 5pm to 11pm), many streamers within the community also often hold streams past midnight (ie. between 1am to 4am), with 5am to 8am being the time with consistently the least amount of activity within the community. This is interesting to observe, as it seems that most streamers are most active during the night, which makes sense as this is usually the time when viewers most likely have free time, since they are out of work and school.

5.1.3 Which streamers consistently hold the most number of active viewers at any given time?

Between the dates of May 02, 2021 and May 19, 2021, we can compare the approximate active viewership values across datetimes by comparing the node sizes of each member of all the graphs. Stepping through each datetime individually, we can see that the top streamers in this community (ie. the streamers with consistently the largest node size in the graphs), are GeorgeNotFound, Quackity, KarlJacobs, Ranboo, Sapnap, WilburSoot, and TommyInnit.

6 Future Work

While this visualization enabled us to make interesting insights into general viewership movement trends, there is still room for improvement. Proposed topics for future work are discussed below.

One limitation to our current visualization is that much of the backend data retrieval process and data processing is done locally on a personal computer, while the visualization itself is hosted on Github pages. This causes two issues: (1) if the local computer is powered off, no data gets retrieved or processed, and (2) the visualization does not get automatically updated since propagating updated data to GitHub pages is a manual process. Thus, one topic for future work would be to transition the website to an external server that can retrieve, process, and update data all on the same server continuously and automatically, so that the visualization itself can be updated live at all times.

Additionally, currently the datetime range explorable is static and limited. One area of improvement would be to allow the user to dynamically select the range of datetimes that they want to explore. This can be done by using adjustable sliders, or by adding pop-up calendars on each end of the slider so that users can manually select the dates they wish to visualize. The addition of adjustable datetime ranges will provide users with greater control over the timeframes they want to explore.

Another area of future work is to enable greater time resolution in the data collected. Currently, data is retrieved every hour. However, some streamers often only stream for less than one hour at a time. This means that our data collection process may be missing useful viewership interactions that happen within these small timeframes. Thus, rather than requesting data every one hour, one improvement would be to increase the rate of requesting data, ie. retrieve data every 15 minutes or 30 minutes, instead of only once every hour. However, this proposed solution will come with challenges in itself. The amount of memory required to store the retrieved data will linearly increase with each resolution increase. Thus, finding a valid server to store all the collected data is important.

References

[1] Twitch. (2011). Twitch. https://www.twitch.tv/

[2] Dream SMP. (2020). Dream Team Wiki. https://dreamteam.fandom.com/wiki/Dream_SMP

[3] Minecraft Wiki. (2011). Minecraft Wiki. https://minecraft.fandom.com/wiki/Minecraft_Wiki

[4] Gershenfeld, K. (2020). Visualizing Twitch Communities. GitHub.https://github.com/KiranGershenfeld/VisualizingTwitchCommunities/tree/CloudCompute

[5] The Dream SMP Relationship Chart. (2021). Reddit. https://www.reddit.com/r/dreamsmp/comments/k6mng7/the_dream_smp_relationship_chart/

[6] Timeline of the Dream SMP. (2021). Dream Team Wiki. https://dreamteam.fandom.com/wiki/Timeline_of_the_Dream_SMP

[7] DREAM SMP Subscriber Count History (+Future) [2012-2022]. (2021, March 1). [Video]. YouTube. https://www.youtube.com/watch?v=7S6-mdnC9vA

[8] Twitch API. (2011). Twitch Developers. https://dev.twitch.tv/docs/api/

[9] D3.js. (2011). Data-Driven Documents. https://d3js.org/