Twitch.tv a Streaming Odyssey

Emmitt Johnson, Cody Malick, and Melissa Sheridan

Fig. 1. Twitch.tv Logo.

Abstract—This paper will attempt to explain the streaming statistics of Twitch.tv. This will serve as a useful starting point for streamers who are looking for the best time to stream, as well as who is the chattiest viewer. Twitch.tv is a popular platform for gamers to stream their game play for others to watch and comment. The chat system on Twitch.tv is a great way for viewers to interact with the player, and have allowed for some streamers to gain immense popularity get better from feedback and interact with others. While Twitch.tv is robust for what it is, it does not show statistics that would be useful for the viewers and streamers. There are plenty of statistics that would be interesting, but we will be focusing on the chat statistics as this is the way users interact with the streamer directly. This project should allow aspiring streamers and viewers to make the most of their streams.

Index Terms—Radiosity, global illumination, constant time

1 Introduction (Motivation)

Twitch.tv is a powerful platform for gamers to stream their game play and interact with others who are interested in the game. Twitch.tv is also a platform for people to make money if they hit a certain number of viewers and this makes this platform invaluable to some talented people. This project is useful for aspiring streamers who want to show off to the world, and make the most of the platform. The information presented will show when the best time to stream is, which is useful for deciding when to stream to maximize viewers, thus creating a fan base. We will also cover who had the most active chat, so aspiring streamers can model off successful streamers who have amassed a large viewership. Lastly spammers are a part of life, and Twitch.tv allows people to block people who spam, and take away from the stream itself. This information will allow the user to watch out for potential spammers who make the chat room not enjoyable and distracting.

Twitch.tv has is a good platform for viewing and streaming, but it does not easily provide statistics for potential streamers who want to jump into this new way of playing. This data will provide an easy way to:

- See when to stream, using a simple visual technique. Different circles will represent the value of the messages. A simple hover over a circle will reveal the information about the circle, be it user name, time, number of messages.
- Who to model off to have a successful stream. This information will give information of the top 20 streamers on Twitch.tv.
- What users might be distracting. From the top 20 streams we

- *Emmitt Johnson. E-mail:johnemmi@oregonstate.edu.*
- *Cody Malick. E-mail:malickc@oregonstate.edu.*
- *Melissa Sheridan. E-mail: sheridme@oregonstate.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx/

have information about who sent the most messages, as well as how many messages were sent in a one hour period over the course of a day.

- What information we as researchers can get from this information, given a time of day, and a stream.

- The product will be a web page using D3. This allows for easy access to the information from whatever device is available to the user.

To achieve these goals, we worked off a working Twitch irc bot implementation Cody wrote a few months ago. We had to re-write the data collection implementation to get useful data out. Next we had to parse the data and analyze it to get the information we wanted. Finally, we had to implement a way to easily visualize and see the data and make it meaningful for the user. We did this with D3js.

2 Related Works

Finding Twitch.tv chat analysis is hard aside from the random image posted to some forums every now-and-again. We did end up finding three solid examples.

2.1 popcorncolonel

Popcorncolonel has done a word cloud and message per minute analysis on Twitch chat. This study was done on chat scraped on August 6, 2014, which was a Sunday, and they found that some of the most active time was around 6:30pm EDT. This does vary slightly from our time, but this was roughly 3 years ago. Events and other things could easily change the data as Twitch is a service which thrives on events.

There was some slight differences between how they chose to represent the data. Popcorncolonel opted to go for a word cloud rather than the bubbles that we used. Word clouds tend to be too busy, and they don't convey much useful information, except for the top few words. [5]

2.2 Elliot Star - Twitch Science team

This study was done by Elliot Star, who is a part of the Twitch Science team. The Twitch Science team does analysis on large data sets from Twitch in general. This specific study was done on how users overlapped between channels. We had thought about doing this, but we opted to go from a more chat based analysis.

The analysis that was done is on how some of the bigger games on Twitch are related. The goal was to see how many users overlapped between channels. These graphs show that people are more likely to watch streamers within a certain game, but do branch out to other games. We think the visualization here is cool, and it would have been a very interesting project to take on if we had access to the relevant data. [7]

3 Method

This section will go over the methods we used to visualize this data, as well as the justification for why we used specific techniques for representing the data we pulled. We will start with an overview of the two techniques we used. Then we will go over justification for why we chose each technique.

3.1 Method Overview

We used two methods to portray our data in a way that is easy for the human eye to parse and we tried to be as intuitive as possible. For the data that was focused on frequency or quantity we chose to represent this with circles. As show in Figure 2. For data that dealt with time we chose a simple bar graph, as this is the easiest way to show change over time without implementing animation. It is also the most familiar to users, and therefor easy for the user to parse and understand.

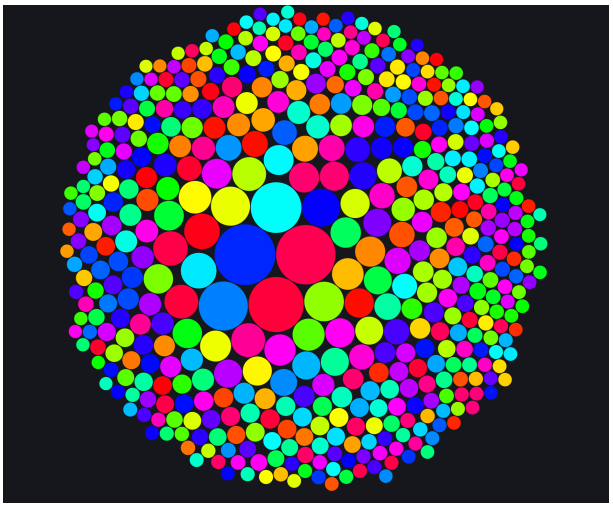

Fig. 2. Representation of our data about a quantity or frequency

3.2 Circles for Representation Purposes

Most of the data we chose to represent is trying to convey a quantity or frequency. This data is not necessarily dependent on time, nor is that the focus of this data. To portray this in the easiest sense we had multiple choices. Circles, pie charts, even word clouds. We ruled pie charts because it is good for describing a portion of whole, and not necessarily a frequency. If we had a breakdown of what each user sent, then we would use a pie chart. Word clouds, while they look cool, seemed to be taboo in the Visualization community. They can be misleading, not accurately display the data, and take away from the message with their distracting formatting.[4] A series of circles with different radii is easy for users to see that each circle represents an entity, and the size of the circle represents the quantity or frequency the entity represents. For example, if we took popularity of pets, dogs and cats would most like have a bigger radius than a snake. We use this technique for the data describing, what messages were sent the most in an hour, which stream had the most active chat, and who sent the most messages in a chat room. We choose these data sets to be circles because they are clearly representing frequency and quantity.

To make this come to light, the entity the circle represents was the name of the user, stream or specific message. From there we took the values in the set, and scaled them down to a scale that will fit on the page. This was done automatically for us with D3js. This new value was then applied to create the circle of different sizes. This gives us the nice variation as seen in Figure 2. We then assigned a random color to each circle to allow for differentiation between each entity or circle. We thought about having a specific color assigned to each circle but that took away from the individual circle when looking at the image from a distance and not on a zoomed screen, especially with circles of smaller radii.

3.3 Bar Graph for Representation Purposes

For our second technique, we used a simple bar graph. We had some data that shows a change over time and circles were not appropriate. So, we had three basic choices, a line graph, bar graph, and a scatter plot. We did not have enough points for a scatter plot to be interesting. A line graph was not appropriate either because it was not the change of one entity but many, so the graph would have been messy and hard to read. We decided on a bar graph as this allowed us to show time, and the number of messages sent in a particular time frame.

The bar graph is simple, as we boiled it down to its essence to get the point of the data across, which is when is Twitch the most active. In this case, we assigned the y-axis to be the number of messages that were sent and the x-axis to be the time of day broken out into hour slots. While this is the least interesting visualization, this is the simplest way to represent a change over time.

Table 1. Schema of 'user' Table

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	Primary	NULL	auto_increment
username	text	YES		NULL	

Table 2. Schema of 'channel' Table

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	Primary	NULL	auto_increment
name	text	YES		NULL	

4 IMPLEMENTATION

We had the fortune to work with the full stack on this project. From data collection, analysis, and visualization, our team built it all. This section will be divided into the corresponding components, and how each was designed and executed.

4.1 Data Collection

Step one of any project is acquisition of data. For this project that meant tapping into Twitch.tv's channel chat. In the past, Cody wrote a basic Twitch bot that was used as the foundation of the project. A few optimizations and alterations were made to provide data collection functionality.

The bot, written in the Go programming language (commonly referenced as Golang) is simple in functionality. It logs into a Twitch channel designated by user input, and captures all messages as they pass by. Each user that submits a message is then recorded in a MariaDB database. The bot makes extensive use of the go-ircevent library to hook into the chat itself.[8] It also makes extensive use of the Twitch Developer API to successfully send and receive messages.[9]

4.2 Data Storage and Schema

For data storage, we used MariaDB, an open source implementation of MySQL.[2] It made sense to use this for our project as it is provided for free, and uses the SQL syntax we're all familiar with. We had a total of three tables: 'user' (table 1), 'message' (table 2), and 'channel' (table 3). The format of the data was important because we needed to make sure we can extract the information. The 'channel' and 'user' tables are fairly straightforward. The 'message' table was the keystone of the project. It contains the information we're interested in: what messages were sent, when the messages were sent, and what channel the message was sent on.

4.3 Data Processing

To extract the information that we wanted, we wrote a simple analysis program using the Go language.[3] The program pulls the set of data we need, in this case all messages, related users, and channels from April 15th, 2017. To pull the data we need, we translate April 15th, 2017 into Unix time stamp format. With that, we can pull all info from Unix time stamp 1492214400 to 1492300800.

After pulling the relevant data, we sort out the information we need to answer the questions we want to answer. For review, we want to answer the following:

- The most active hours for users based off of messages sent per hour
- Which streams are the most successful (based off a prepossessed selection of channels)
- Over the course of an hour, who is sending the most messages
- Most common messages sent in the most active hour of the day

Table 3. Schema of 'message' Table

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	Primary	NULL	auto_increment
channelid	int(11)	YES	Foreign	NULL	
userid	int(11)	YES	Foreign	NULL	
message	text	YES		NULL	
timestamp	int(11)	YES		NULL	

For the most active hours for users, we simply sorted messages by time stamp, and associated each of the twenty-four hours in a day with the number of messages sent. From that, we accomplished the third goal, most common message sent in the most active hour. Next, we got a metric which streams were most popular by checking messages sent per stream over the whole day. Lastly, we checked which users were sending the most messages across all the channels we monitored by associating each message with the user, and counting.

4.4 Visualization

We used D3js to bring the data to life. D3js is a powerful visualization library for JavaScript that allows the user to take data in any form and create a web page with visuals based on the data. We used this tool because it requires little computing power on the client end, which allows for anyone to access the data visualizations. For us, we took the data from the databases, and created comma separated variable (csv) files, and fed this into the D3 library. This was the easiest way for us to handle this data as well, and morph this project from a static data store to a dynamic one in the future. It also allowed for easier development because no special hardware was required to run it.[1]

Users could answer the questions using three different avenues. First the size of the circles will tell the number of messages sent or frequency it was sent from. From the first glance, the user can easily see and distinguish between the most popular message, and streamer, as well as the most active person in the chat rooms themselves. The second method is a click on the circle to get the name of the person or the message itself. Last, a hover over the circles will give detailed information about what the circle represents. The circles are easy to parse for a human as it is easy to distinguish between a large circle and a small one, the titles above each graphic should help hone in on what the information is portraying.[6]

The bar graph is natural to parse because everyone has been exposed to to bar graphs at least once in their lives. This graph simply shows progression of a day on Twitch and will show what time(s) are the most popular for streamers.

5 RESULTS

We had some interesting results from this exercise in visualization. The first is that we had a far greater volume of messages than we expected. Although there was no initial guess of what the results would be, we expected to get roughly fifty-thousand messages. We ended up with about one-hundred twenty-thousand! The first observation we made was that only a few of the channels get a lion's share of the messages sent. About three of the channels received the majority, and the others all received similar numbers, with a few small counts who did not stream during our recording time.

From that data, we gleaned that the average user does not comment too often, but when users do comment, then comment quite a bit. Of all users, however, we see that three of the top five users are bots. This could make sense if the bot is a common community bot used for displaying status messages for a given channel, or if it is used to respond to user input on a channel. Another insight we discovered was that the Twitch channel emoji are used incredibly often. Out of the top phrases we extracted, most of them were emojis. Twitch uses a phrase, for example "WutFace", to display a small emoji icon. The only thing more popular than a few common emoji was the phrase "lul." As for how useful that information is for someone looking to extract information for their channel, we're not sure.

Lastly, we determined that the time that users are most active, or most likely to be watching on Twitch, is around 5:00 PM PST. This result could be skewed from the fact that we picked popular streams, or a coincidence in that the streams we picked are mostly active in that time frame. Regardless, it was an interesting observation.

6 Conclusion

Overall, this was a really fun project that we all enjoyed working on. We were able to get the answers to the questions we asked, and provide more insight into the audience of Twitch in general. We're looking forward to the next assignment.

Acknowledgments

The authors wish to thank Twitch, for creating an awesome streaming service, as well as our patient TA for reviewing our document. This work was supported in part by my parents who paid for my education to OSU.

References

[1] M. Bostock. D3js, 2015.

[2] M. Foundation. Maria database, may 2017.

[3] Google. The go programming language, may 2017.

[4] J. Harris. Word clouds considered harmful, Oct. 13 2011.

[5] popcorncolonel. Chat stats, Oct. 10 2014.

[6] E. Sinar. 7 data visualization types you should be using more (and how to start), Feb 14 2016.

[7] E. Star. Visual mapping of twitch and our communities, cause science!, Feb. 4 2015.

[8] thoj. Event based irc client library in go (golang), Dec 2016.

[9] Twitch.tv. Chat and irc, may 2017.