TwitchViz: A Visualization Tool for Twitch Chatrooms

Rui Pan
Simon Fraser University
250 – 13450 102nd Avenue
Surrey, BC, Canada, V3T 0A3
ruip@sfu.ca

Lyn Bartram
Simon Fraser University
250 – 13450 102nd Avenue
Surrey, BC, Canada, V3T 0A3
lyn@sfu.ca

Carman Neustaedter
Simon Fraser University
250 – 13450 102nd Avenue
Surrey, BC, Canada, V3T 0A3
carman@sfu.ca

CHI'16 Extended Abstracts, May 07-12, 2016, San Jose, CA, USA
ACM 978-1-4503-4082-3/16/05.

http://dx.doi.org/10.1145/2851581.2892427

Abstract
Twitch.tv is a flagship platform for live game streaming between players and viewers. It allows players to broadcast their gameplay to a public audience where viewers chat with each other and discuss gameplay. Current tools for analyzing live game streaming and chat rooms are limited. In this paper, we describe the design of TwitchViz: a new visualization tool with the goal of helping both players and game designers to better understand the relationship between gameplay and Twitch viewers' chatting behaviors. An initial feasibility study showed that TwitchViz supports novel ways to get an insight of gameplay issues from the patterns of chatting behaviors of viewers and highlighted design issues to address in subsequent versions of the tool.

Author Keywords
Twitch.tv; live streaming; chat room; video game; visualization tool

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
Live game streaming provides a new social medium that combines high-fidelity game videos with a low-

fidelity text-based chatting channel [2]. It allows game players to broadcast their gameplay to a public audience and help viewers exchange thoughts in Internet Rely Chat (IRC). Twitch.tv, or simply Twitch, has more than 100 million unique viewers and more than 1.7 million broadcasters per month according to Twitch's official stats [10].

Viewers of Twitch often share their thoughts while they are watching live game streams. This has turned Twitch's chat rooms into a participatory virtual community [2]. However, given the rise in the number of Twitch viewers and chat posts, users are now often overloaded with information; this makes it challenging for streamers to maintain an understanding of their own communities [2]. Twitch has developed features to restrict posters to subscribers-only, but, anecdotally, this does not seem to have reduced the volume of data that appears in a Twitch chat channel. In addition, game designers have no easy way to mine these data for insights about gameplay issues: there are few visualization tools to help game designers 'think in the shoes' of the viewers of the live game streaming website.

In order to address these gaps, we designed an interactive visualization and analysis tool called TwitchViz that visualizes data from Twitch chat channels. In the following sections of the paper, we present related research, describe the design of TwitchViz, and then discuss visualization principles behind the design. We conclude with a discussion of our findings from a small-scale feasibility study that we conducted to inform future redesign of the tool.

Related Work

Live game streaming is a new concept that has emerged in recent years. Kaytoue et al. [3] published the first study of video game live streaming in 2012. They crawled online data for 100 days and characterized a new web community of game streamers and viewers. Hamilton et al. [2] presented an ethnographic study on the virtual communities on Twitch with a discussion of 'hot video medium + cool text medium'. They came up with design suggestions that inspired our design of TwitchViz.

We also see there are system designs focused on the visualization of chat rooms. Chat Circles [11] is a graphical interface for online chatrooms that uses color and form to filter and break conversation groups. Coterie [8] is a dynamic visualization of social interaction in IRC channels which helps users 'see' social patterns. The focus of both visualizations is more general chatting online without a specific context such as playing video games.

There are also many visualization tools focused on game analysis. For example, Data Cracker [5], a visual game analytic tool for analyzing Dead Space 2 (a survival horror game by Electronic Arts), provides a means for game teams to increase game data literacy. VIZMO [4] is a visual game browser that allows users to categorize video games by mood and visual styles. However, none of these tools focus on live game streaming and viewers' reaction to gameplay.

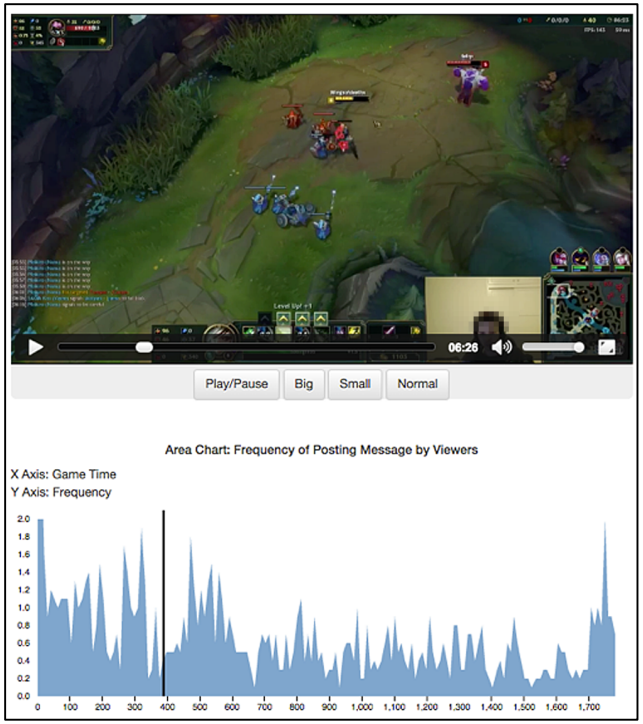

Figure 1. The synchronous navigation bar and video time.

The Design of TwitchViz

The goal of TwitchViz is twofold. First, we want to provide game players with a means to review the chat history that occurred while they played a game. This could help them better understand the community's reaction and discussion around their gameplay. Second, we want to provide game designers with a means to analyze chat history in relation to game play such that they can make informed decisions about future game design as well as understand their current design.

TwitchViz runs on web browsers, built with D3.js and the Twitch API. The layout of TwitchViz consists of three parts: a pre-recorded Twitch gameplay video is placed at the top left of the tool, with a Play/Pause button and three video size change buttons underneath. Below the video, there is an area chart showing frequency of viewer message posting in the chatroom. At the top right of the tool, there is a stacked bar chart comparing the main chat topics of viewers. We crawled the corresponding chat room data when we recorded the gameplay video. The data dimensions that we considered in this tool are timestamp, frequency of posting message by viewers (in 10 second increments), and the content of message.

There are three main functions in TwitchViz: synchronous video navigation, customizable game section codes, and brushing comparison. We discuss each in turn.

Synchronous Video Navigation

We added a vertical black navigation bar into the area chart. When users are watching the recorded gameplay video on TwitchViz, the black navigation bar will synchronously slide along with the playing of the video. In Figure 1, the bar is shown near the 400 second mark on the X-axis. Users can click on the area chart to move the navigation bar and the video will automatically jump to the time that users click. We imagine that game streamers and game designers can use this function as a 'cross reference' to get a sense of what excites viewers at different points in the video. Designers and streamers can also play back interesting parts of the video to explore how they impacted viewers' chat.

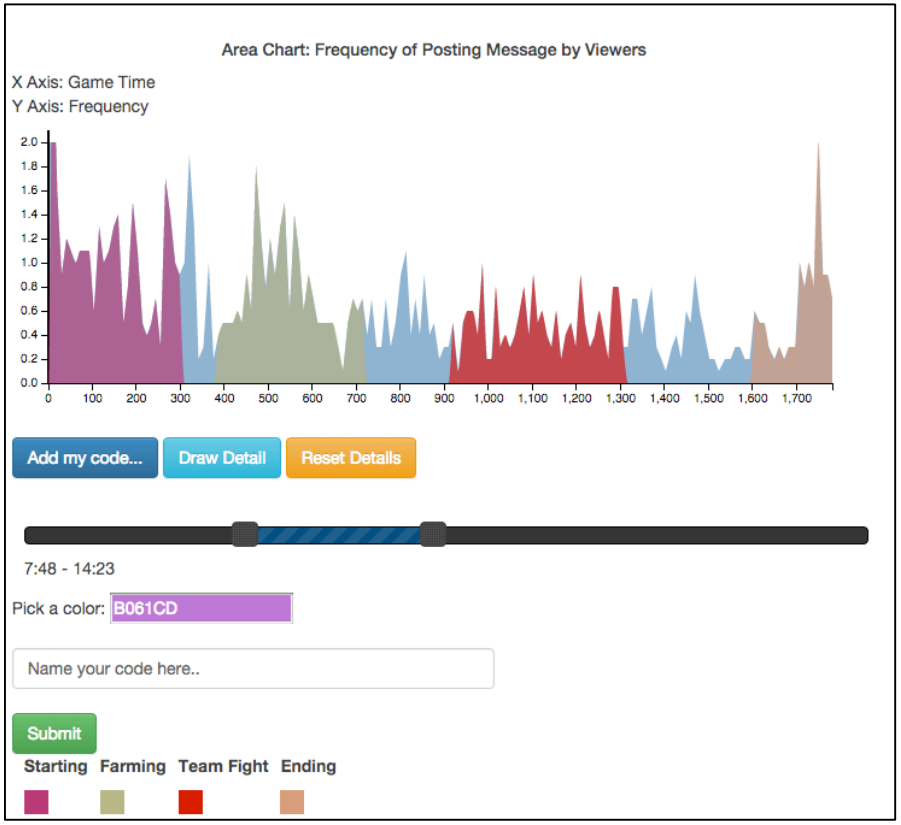

Figure 2. The function of adding user-defined game section codes.

Customizable Game Section Codes

We designed a function that helps users who are game designers assign codes to game video sections. These can be used to describe player behaviors during different time intervals. Figure 2 shows the interface for this function and an example of the area chart after being coded. When a user clicks the 'Add my code…' button, a time slider, a color selection field, and a text input field will appear. Users can slide the bars on the time slider to specify a time interval. The corresponding time of the left and right boundary will be changed while users are moving the slider. Users can choose a color that they like using the color palette. They can also input an exact RGB value. The background of the color text box will automatically change to the RGB value as a preview. Under the color selection field, there is a text field where users can type in their code. For example, a game designer might define the first 5 minutes of a game video as 'Starting' (purple in Figure 2), 10 minutes after it as 'Farming' (green in Figure 2), and some small pieces of time scattered as 'Team Fight' (red). After clicking the 'Submit' button, the section of the area chart will change to the color selected by user. The mapping of colors and codes is placed under the area chart as a reference for the convenience of users.

Brushing & Comparison

Another important function of TwitchViz is brushing and comparison. When users drag their cursor over the area chart, a semi-transparent black brushing area indicating a selected interval will appear (Figure 3). After clicking the 'Draw Detail' button, TwitchViz will draw a stacked bar chart on the top right of the page. Users can compare the different topics that occurred in the chat room during the interval. Multiple interval comparison is also available on TwitchViz. Figure 4 shows the case of using the stacked bar chart to compare chatting topics.

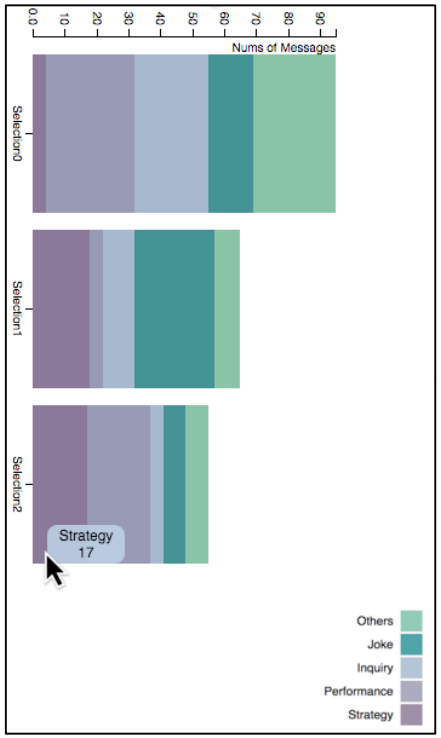

Figure 4. Comparison of multiple intervals' topics of chatting contents in a stacked bar chart. When users hover their cursor over rectangles, a tooltip indicating the accurate number of messages in the attribute will pop out.

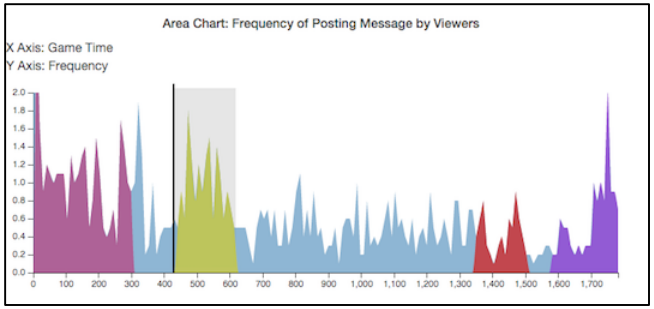

Figure 3. A brushing selection of an interval on the area chart.

Visualization Principles

The design of TwitchViz followed visualization principles such as careful selection of visualization idioms and colors, consideration of space constraints etc. Our initial design and sketching efforts compared line charts with area charts and stacked bar charts for overall usability. Stacked bar charts provide a sense of part-to-whole relationships and they also have good scalability when it comes to expanding stacked attributes [6]. In our case, using a stacked bar chart can help users compare attributes inside a selected interval. They can also compare multiple intervals using the same representation. However, one weakness is that a single representation of a stacked bar chart makes it different to compare across intervals in terms of a specific attribute.

User can define their own game section code with a color that they picked from the color palette because colors are good as categorical attributes for identity channels [6]. We expect that users could distinguish different game time sections that they defined. In terms of the color we chose for the stacked bar chart, we avoided using red and green together in case some of our users might have red-green color blindness. Furthermore, the amount of hues is limited to two because an excessive amount of hues may confuse people [1,7,9].

The brushing provides a pervasive and powerful means that allows users to focus a certain subset data [12]. After users select a subset of the data, a tooltip indicating the accurate number of messages in the attribute will pop out on the stacked bar chart when users hover their cursor over the rectangles. We imagine that this small interaction can help users get accurate data without losing high level overviews.

Feasibility Study

We had four people participate in a feasibility study of TwitchViz. The participants in this study were all graduate students with knowledge about visualizations. We showed each participant the visualization and had them use it to perform several tasks. We asked questions about their general feelings when using the tool. We also asked them to gave us feedback on the visualization techniques used.

All participants expressed their appreciation of TwitchViz. They told us that TwitchViz helped them get insights of gameplay and understand what excited people. One participant spoke highly of the interaction of the tool and easiness of finding patterns with the area chart.

The participants also pointed out the disadvantages of TwitchViz. One participant said that he could not see everything in a single screen and had to scroll a lot. He also mentioned the confusion of the time slider and the brush. Another participant thought that TwitchViz

lacked capabilities for individual viewer analysis. He said it would be interesting to break the area chart down into individual chats.

Conclusion and Discussion

This paper outlines a system design of TwitchViz to help game streamers and game designers better understand viewers' chatting behaviors in live game streaming platforms. We also present a feasibility test of the system where we find that users acknowledge TwitchViz is a novel and useful tool for exploring viewers' interest when they are watching game play streams.

While beneficial, TwitchViz still contains some limitations. It is currently difficult for users to switch switch from one game video to another. Because of the nature of live game streaming, it would be better to have a 'real-time' analysis representation than our current method of recording games for post-play viewing. TwitchViz is also unable to compare multiple game streams. In this case, it is difficult for people to use TwitchViz to compare different genres of gameplay video. Natural language processing and text analysis are not the main focus in this paper, so we did not spend efforts on processing message contents, which could have provided a means to automatically categorize conversations.

For future work, we will iterate the design of TwitchViz based on the feedback from users. We are planning to recruit senior Twitch streamers and professional game designers to conduct a deeper evaluation of the system.

References

1. David Borland and Russell M. Taylor II. 2007. Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 2: 14–17.
2. William A. Hamilton, Oliver Garretson, and Andruid Kerne. 2014. Streaming on twitch: fostering participatory communities of play within live mixed media. ACM Press, 1315–1324. http://doi.org/10.1145/2556288.2557048
3. Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira Jr, and Chedy Raïssi. 2012. Watch me playing, i am a professional: a first study on video game live streaming. *Proceedings of the 21st international conference companion on World Wide Web*, ACM, 1181–1188. http://doi.org/10.1145/2187980.2188259
4. Jin Ha Lee, Sungsoo (Ray) Hong, Hyerim Cho, and Yea-Seul Kim. 2015. VIZMO Game Browser: Accessing Video Games by Visual Style and Mood. ACM Press, 149–152. http://doi.org/10.1145/2702123.2702264
5. Ben Medler, Michael John, and Jeff Lane. 2011. Data cracker: developing a visual game analytic tool for analyzing online gameplay. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2365–2374. http://doi.org/10.1145/1978942.1979288
6. Tamara Munzner. 2014. *Visualization Analysis & Design*. CRC Press.
7. Bernice Rogowitz and Lloyd Treinish. Why Should Engineers and Scientists Be Worried About Color? Retrieved December 17, 2015 from https://www.research.ibm.com/people/l/lloydt/color/color.HTM
8. Dana Spiegel and Judith Donath. Coterie: Live, Dynamic Visualization of Social Interaction Online.

Retrieved December 17, 2015 from http://web.media.mit.edu/~spiegel/research/Coterie/Coterie_UIST_2000.pdf
9. Maureen Stone. Choosing Colors for Data Visualization by Maureen Stone - BeyeNETWORK. Retrieved December 17, 2015 from http://www.b-eye-network.com/newsletters/ben/2235
10. Twitch Inc. About Twitch.tv. Retrieved December 16, 2015 from http://www.twitch.tv/p/about
11. Fernanda B. Viégas and Judith S. Donath. 1999. Chat circles. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, 9–16. http://doi.org/10.1145/302979.302981
12. Graham Wills. 1997. *Visual Exploration of Large Structured Datasets*. IOS Press.