Towards a Design Framework for Data-Driven Game Streaming: A Multi-Stakeholder Approach

NOOR HAMMAD, Carnegie Mellon University, USA
ERIK HARPSTEAD, Carnegie Mellon University, USA
JESSICA HAMMER, Carnegie Mellon University, USA

Research on live streaming systems that incorporate real-time data, such as game or viewer data, have been a topic of HCI research for some time. Despite the potential of data-driven game streaming interfaces, translating this research into practice faces two key challenges. First, the design space afforded by data-driven game streaming systems is not yet well understood, making it difficult to identify how designs might meet users' existing and potential needs. Second, adoption of these systems relies on engagement with the entire streaming ecosystem, which includes developers, streamers, moderators, and viewers, rather than with just one group. Through a two-phase design study, we investigate the expectations, desires, and experiences of streaming stakeholders, shedding light on how data-driven game streaming systems can meet their needs. Building upon these insights and drawing upon previous research, we propose a design framework aimed at analyzing and generating data-driven game streaming designs, thereby moving toward formalizing the design and development of such systems.

CCS Concepts: • **Human-centered computing** → **Participatory design**.

Additional Key Words and Phrases: Live Streaming, Twitch, Game Development, Game Design, Participatory Design

ACM Reference Format:
Noor Hammad, Erik Harpstead, and Jessica Hammer. 2024. Towards a Design Framework for Data-Driven Game Streaming: A Multi-Stakeholder Approach. *Proc. ACM Hum.-Comput. Interact.* 8, CHI PLAY, Article 342 (October 2024), 28 pages. https://doi.org/10.1145/3677107

1 INTRODUCTION

Game live stream platforms such as Twitch [1] can be understood as data-driven ecosystems. Games, produced by game developers, use player interaction data to change the game state. Streamers share live video of themselves and their gameplay. Viewers and moderators engage the streamer by typing or sharing emojis in a simultaneous text chat channel. However, by default these stakeholders have very limited capacity to interact with one another, and participants' capabilities are primarily determined by their role in the ecosystem. Viewers, for example, are often restricted from modifying game state data.

Systems that leverage *data-driven game streaming* can offer novel interaction and communication mechanisms through the collection, propagation, and rendering of *live* data. Data-driven game streaming systems use data to power novel live streaming features, which can be collected from multiple sources, namely the game being streamed (game data), the streamer playing the game

Authors' addresses: Noor Hammad, nhammad@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA; Erik Harpstead, eharpste@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA; Jessica Hammer, hammerj@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

ACM 2573-0142/2024/10-ART342
https://doi.org/10.1145/3677107

(streamer data), or the viewers and moderators watching the live stream (viewer data). Using these diverse sources of data, data-driven game streaming systems can enable features such as participatory mechanics for viewers, rendered overlays for streamers, or communication mechanisms for streamers, viewers, and moderators.

Some existing systems explore approaches to augmenting game streams with live data. Audience Participation Games, or APGs, offer the viewers the ability to affect the game being played by the streamer, often through the use of voting or aggregation mechanisms [15, 35]. Lessel et al, through their HelpStone system for the game *Hearthstone* [4], offer additional communication and interaction avenues for viewers via an overlay rendered onto the video stream of a game by utilizing game log data generated by the streamer's gameplay [22]. Similarly, Hammad et al's MARS (Metadata Augmented Real-Time Streaming) enables real-time rendering of metadata from Unity games [18] to support personalized viewing interfaces across many genres of games. Even outside of the HCI research community, systems such as Twitch's own extension tools are attempts to provide additional avenues for viewers and streamers to interact with each other and the game being played [16]. These systems vary in their approaches to data, how they define stakeholder capacities, and what types of features they support. However, they collectively demonstrate a hunger on the part of developers, streamers, viewers, researchers, and other stakeholders for richer game streaming experiences.

Although this hunger continues to drive system development, we do not yet have a framework to talk about the relationship between different data-driven game streaming systems, nor do we have a design language to talk about what kinds of experiences they make possible. To continue the exploration of data-driven game streaming experience established by these previous works, and to legitimize the data-driven game streaming paradigm, it is important to formalize the shared design space that unites all data-driven game streaming experiences.

Stakeholders who produce live stream data are also stakeholders in *how* that data is used to augment the streaming experience, and their buy-in is needed to make systems that work in practice. Game developers need to implement data-driven features in their games; streamers need to enable the data-driven functionalities on their channel, and leverage them to build relationships with their audience; streaming interface developers need to render incoming data; viewers need to engage with the new functionalities using those interfaces; and moderators need to maintain community relationships and guidelines in the presence of these new tools. Because these stakeholders are tightly coupled, we must work with all stakeholders *together* to understand what challenges they face, how their needs might be met, and what design opportunities exist.

Towards the goal of formulating the design space of data-driven game streaming, this paper shares the results of a two-phase design study conducted with target stakeholders in the distributed data-driven game streaming ecosystem, using an existing data-driven game streaming system to support participant ideation [18]. Stakeholders including streamers, developers, designers, and viewers served as our participants, with the goals of 1.) understanding their needs from a data-driven game streaming ecosystem, and 2.) utilizing their unique perspectives as our target stakeholders to understand how they think about the design space. The results from this study offered insight into how to build data-driven game streaming systems with the target users in mind. We propose a design framework that integrates the findings of our study with previous insights from other design space explorations of data-driven game streaming instances [15, 35, 37]. This framework provides a structured approach for designers to create, analyze, and improve data-driven game streaming designs; our framework asks designers to identify design objectives and target user (Experience), define abstract actions that users can take (Abstract Actions), describe how users can take those actions across available interaction affordances (User Interaction), and to determine the stakeholders' relationship to the data needed to power the design (Data Behavior). In summary, our work here contributes the following:

(1) A set of design goals that stakeholders desire from data-driven game streaming systems
(2) A series of general design considerations to maintain the technical, ethical, and social integrity of these systems
(3) A framework that incorporates the above findings to help generate, evaluate and improve data-driven game streaming designs

2 RELATED WORK

Prior work shows that players already use their own game data to improve performance [32, 41]. However, game data has potential to play a greater role in the play experience if it is used as a "creative asset" [40] that serves both players and other stakeholders in a streaming context. In framing live streaming as a data-driven experience, we build on the concept of data as a creative asset that shapes the experience of all streaming stakeholders. To this end, we situate our study in the areas of data-driven streaming research and in research on streaming stakeholders.

2.1 Data-Driven Game Streaming Systems

Existing work on data-driven game streaming systems uses different data channels for different stakeholders and purposes. One common system design leverages or manipulates chat data. In particular, a series of systems have used chat data to enable sensemaking of conversations at scale. Conversation Chat Circles [28] is a tool to limit the number of chat messages the viewers sees, thus helping viewers communicate and make sense of the chat in streams with many viewers. The system dynamically adjusts the chat volume by using "neighborhoods" that group viewer chat data based on a number of criteria. Chung et al's VisPoll system serves a similar sensemaking purpose for a different audience, helping streamers make sense of visual viewer input data at scale [10]. This visual viewer interaction data is categorized using a visual input management framework (VIMF), which allows streamers to specify, aggregate, visualize, and make sense of this data. Another system called TwitchViz [29] uses chat data for visualization to help improve the streamer's understanding of the relationship between chat behavior and moments of gameplay on stream. Another use for real-time chat data is to mitigate toxicity, as is the case with Yen et al's StoryChat system [47]. This is a system built to encourage prosocial behavior on stream by generating graphical narratives from viewer chat messages, an approach which particularly supports moderators who would otherwise have to intervene.

Stream viewer chat data can also be used as a means of viewer participation in game streams. *Twitch Plays Pokemon*, for instance, aggregated chat data to drive the game mechanics in *Pokemon Red* [31]. *Choice Chamber*, another early audience participation game, allows viewers to choose what obstacles the streamer faces through chat [3]. Lessel et al's Crowdchess system used aggregated live stream chat data from viewers to allow them to play Chess against an AI on stream [21]. Seering et al also leveraged chat data in their first formal exploration of live streamed APGs [35]. This was followed by Glickman et al's toolkit that enabled the creation of arbitrary HTML interfaces for APGs, this time using the chat channel as a hidden means of encoding the viewers' interaction data for gameplay [15]. Both explorations of APGs revealed critical insights about the design of such experiences, paving the way for the exploration of the design space and its associated challenges. A new form of chat-based audience participation game known as Danmaku Participation Games (DPG) emerged on the live streaming platform BiliBili which shows promise in innovating aggregation-based APG mechanics [43]. Wang et al shed light on this new form of stream-based gameplay, finding key design attributes that differentiate the independent viewer control of gameplay in DPGs from previous iterations of APGs. APGs such as the ones discussed here are a well-established form of data-driven game streaming system, however APGs are merely one type of of data-driven game streaming system which focus on offering viewers the ability to impact gameplay. We therefore argue that APGs are a subset of data-driven game streaming, and that the overall design space includes other types of experiences.

A second form of data-driven system involves incorporating multiple data sources, including game data, and rendering that data onto the video feed in an overlay format. The HelpStone system made use of game log data from the deck building game *Hearthstone* [4], combined with viewer data from an interactive overlay, to enhance streamer-viewer communication in real-time. Building upon this work, the MARS system (Metadata Augmented Real-time Streaming) formally introduced the concept of game-aware interfaces, or interfaces that allowed audiences to interact with game objects from a streamed game. MARS is a generalizable architecture which can be integrated into any Unity game, and makes use of Twitch's Extension tools to enable personalized data-driven interfaces for viewers. Twitch extensions are a recent addition to Twitch which enable overlays for viewers of streams [16]. Extensions can be developed using any modern web framework, and as of now there are hundreds of extensions available on the platform. In a data-driven context, extensions can offer many capacities for viewers and streamers. This includes game-specific overlays for popular games such as *Destiny 2*, which let viewers visualize game data in real-time. Extensions can also enable broader data-driven capabilities using an "Extension Backend Service", which allows an extension to communicate with an external data source. Nowadays, extensions power many functionalities for viewers, including iterations on the "Twitch Plays" experiences that kickstarted the APG phenomenon, such as the "Crowd Control" extension which can be used to activate audience participation in over one hundred games.

Finally, some systems integrate data sources outside the streaming platform itself. For example, Robinson et al incorporate real-time biometric data from the streamer to convey their emotional state to their viewers using a Heads Up Display [33].

We note that many interfaces are designed specifically to enhance the eSports spectatorship experience, given the popularity of the genre. For example, tools like Charleer et al.'s real-time dashboards visualize real-time data from the popular eSports titles *Counter Strike: Global Offensive* and *League of Legends* [6]. Stahlke et al also explored spectator interfaces for eSports, particularly focusing on the design of the spectator-player experience and identifying mechanics to support spectators [36]. Relatedly, there are a number of data-driven streaming systems which, while relevant, do not apply strictly to games [24, 26, 45, 46]. While eSports remain an integral part of the modern live streaming landscape, the unique qualities of eSports, such as the nature of viewer engagement being more akin to a "digital stadium" as opposed to a "digital hearth" [35] and the use of eSports casters to provide commentary, do not apply to the broader game streaming ecosystem. As such, we chose to focus our efforts on game streaming experiences where a streamer is playing a game on their channel for viewers to engage with. While we are focused on games *outside* the context of eSports, we found that the data-driven game streaming systems we are interested in rely on similar data sources as those eSports: chat data, game log data, overlay data, and data sources from outside the core stream.

Many of the above papers identify design implications of their respective systems in the form of design goals and considerations [15, 28, 35]. Taken separately, these design insights lay the foundation for the design spaces for different types of data-driven game streaming systems. Taken together however, these systems lack a design language that unites the overall space of data-driven game streaming. Our goal then, is to move towards establishing that design language to pave the way for best practices in the field.

2.2 Streaming Stakeholders

Streaming stakeholders include streamers, viewers, moderators, game developers, and more [18]. These stakeholder groups have different needs and priorities, even as they work together to create engaging live-streamed game experiences.

A substantial body of work explores the needs of individual stakeholder groups, including how those needs might inform design. For example, Wohn et al conducted interviews with streamers to understand their audience management practices. Their findings yielded design recommendations to create streamer tools for audience management to support knowledge discovery and growth management [44]. Mallari et al interviewed game streamers to understand their data analytics needs [27]. They found that streamers have a range of information needs ranging from content, marketing, and community data which are not easily met by existing tools, and have demonstrated the importance of designing effective analytics interfaces for game streamers. In terms of viewers, Wang et al's exploration of the recent Danmaku Participation Game (DPG) phenomena involved semi-structured interviews with viewers on the BiliBili live streaming platform to understand their practices and experiences [43]. Also working with viewers, Lessel et al surveyed over 400 viewers to understand the elements they value in a live stream setting [20]. Their findings ranked 58 elements of a live streaming experience, with interactive components being the most valued by viewers.

While need-finding with streamers and viewers is relatively common, investigations of other stakeholders' needs are rare. For example, Li et al's study of indie game development communities on Twitch interviews game developers in a streaming context, revealing how members of the indie game developer community leveraged live streaming for a distributed software development setting that encouraged a democratic model of game development [23]. However, the paper effectively treats the developers as streamers and viewers, using live streaming as a means to build community, rather than focusing on their underlying game development needs. Similarly, the diversity of streaming moderator motivations and experiences is only now beginning to be understood in ways that can inform design [5, 34].

Although our focus is on game streaming, we note that studies of live streaming outside games often engage multiple stakeholder groups at the same time, and illuminate how their needs can be complementary. Interviews with streamers and viewers of intangible cultural heritage streaming found that streamers had altruistic intentions, and viewers were motivated to learn and engage with the arts [24]. Similar work was conducted with streamers, viewers, and moderators of educational streams to understand their current practices for knowledge sharing, with the findings being used to build a knowledge sharing system for viewers [25]. In a similar educational context, interviews were conducted with streamers and viewers in a classroom setting, where instructors took on the role of streamers to continue teaching during the COVID-19 pandemic [8]. In the context of creative streams, streamers and viewers were interviewed and surveyed to identify how to best support the goals of streamers in their creative process [13].

While many of the papers discussed in the previous section worked with viewers and streamers for formative or evaluative purposes [22, 28, 29], streaming stakeholders are typically not directly involved in the design process. A key exception is Glickman et al's work with APGs [15], where researchers worked with game developers for ten months to create audience participation games. This process moved the design space of APGs forward by yielding six design challenges for APGs, such as latency and sharing the screen, along with worked examples by game professionals. Striner et al extended this work by engaging six student game developer teams, which resulted in three sensitizing concepts for APG design: agency, pacing, and community [37]. In working with developers, the above moved the field of APG design forward, underlying the importance of incorporating developer feedback in data-driven game streaming design.

Taken together, these studies involving stakeholders highlight the importance of understanding the data practices of individual stakeholder groups, such as streamers' needs around data analytics [27]; learning about synergies between groups so that designs can address multiple stakeholder needs simultaneously [24]; and engaging stakeholders in design, particularly game developers and others who are already designing systems for streaming, to leverage their insights and expertise [15]. We designed our study to directly engage stakeholders *together* to explore the potentials of data-driven game streaming systems, arguing that their perspectives and interdisciplinary collaboration through our study can help contribute to our understanding of their varying, and often conflicting needs. We thus work with streaming stakeholders to understand what features they want to see in these systems, what concerns they have with this technology, and how we can collectively move towards defining a shared language for the design of data-driven game streaming systems.

3 IDEATION AND DESIGN STUDY

In order to better understand the design space for data-driven game streaming, we executed a multi-stakeholder study structured around the following research questions:

(1) What features do streaming stakeholders want in a data-driven game stream system?
(2) What challenges and concerns do stakeholders have when it comes to novel data-driven game streaming features?
(3) How can stakeholders contribute to the design of data-driven game streaming systems?

To address these research questions, we designed a two-phase study where game developers, live streamers, viewers, moderators, designers, and researchers were invited to collaboratively envision the future of data-driven game stream experiences. In the first phase of our study (Phase One), participants brainstormed desired features (Design Goals) and ways to mitigate challenges (Design Considerations) for data-driven game streaming, generating *affinity diagrams* from their ideas [30]. The research team then reviewed Phase One data to synthesize themes and insights. We used our analysis to create prompts for the second phase of the study (Phase Two), a participatory design workshop where participants envisioned sample data-driven game interfaces. Our analysis of the Phase Two data allowed us to see how stakeholders understand data being created, maintained, and rendered.

Below, we provide a detailed breakdown of our study, including the methodology and results of each of the study phases.

3.1 Participants

To ensure our study included the diverse perspectives within the data-driven game streaming ecosystem, we targeted a range of participants: game developers, front-end developers, live streamers, viewers, moderators, designers, and live stream researchers. More specifically, our inclusion criteria specified that participants be residents of the United States, be aged 18 or older, have a familiarity with the Twitch live stream platform, and have experience in one of the roles we defined.

We utilized multiple channels to reach potential participants, including social media platforms (X, Facebook, Reddit) and informal networks within the Twitch streaming community. We also recruited heavily through specialized Discord servers, namely servers for game developers (IGDA, Global Game Jam, Unity Developer Community), user researchers (IGDA Games User Research and UX SIG), Games Related Academic conferences (CHI PLAY, Foundations of Digital Games), and Twitch extension developers (TwitchDev). We did not utilize eSports community channels in our recruitment, as our scope was on other forms of game streaming. Interested parties first filled out an online screener survey, and selected participants were emailed the consent form and scheduling information.

In Phase One of our study, we recruited a total of 26 participants, who were distributed across six study sessions. For Phase 2, we retained 21 out of the initial 26 participants, distributed across four study sessions. While all of our participants were representative of the stakeholders in the distributed developer ecosystem, many of them identified as having multiple roles; for example, a participant who was a professional game developer also live streamed on Twitch as a hobby, while another participant was a professional streamer who also had experience as a moderator. Fourteen of our participants identified as male, eleven identified as female, and one participant identified as non-binary. In total, ten participants identified themselves as *streamers*, ranging from full-time professional streamers who stream at least three times a week with 50-200 viewers per stream, hobbyists who stream once or twice a month with 10-20 viewers per stream, to individuals who did professionally stream in the past but no longer do so. Nine participants identified themselves as *game developers*, four of whom are current graduate students in a game design program. A single participant identified themselves as a professional *UX designer*. Three participants identified themselves as *moderators*, all of whom had experience moderating multiple channels. Two participants were academic *researchers* interested in live streaming. Finally, six participants identified themselves as *viewers*, all of whom regularly watch Twitch streams on a weekly basis.

3.2 Phase One: Collaborative Ideation

Phase One of our study was primarily focused on addressing the first two research questions; What features streaming stakeholders want in a data-driven game stream? What challenges and concerns do stakeholders have when it comes to novel data-driven game streaming features? To accomplish this, we used affinity diagramming [30], a collaborative ideation and organization method which can be conducted with target stakeholders to generate information and categorize that information into related groups.

To help support participants' ideation, and to establish a shared meaning of what data-driven game streaming is capable of, Hammad et al's MARS system was used as an example model of data-driven game streaming. MARS was selected for the study for three reasons; it is a recent system that demonstrates the current cutting edge capabilities of data-driven game streaming; its architecture offers generalizeability to many genres of games; and it is designed specifically with a "distributed developer ecosystem" in mind [18].

Each affinity diagramming session consisted of 3-6 participants, and was conducted on Zoom and Figma. The placement of participants into sessions was largely decided based on the availability of the participants, however the research team ensured no session consisted of all participants who identified as the same stakeholder type. Sessions lasted approximately two hours, and were facilitated by a member of the research team. Two additional research team members supported the session lead by addressing technical issues as they occurred. To maintain participant privacy in the group setting, each participant was given a pseudonym as soon as they entered the Zoom waiting room. Participants introduced themselves using their pseudonym, and all study data was recorded using this identifier.

Next, the facilitator set the context of the session by describing the MARS system. The overview of MARS included a discussion of the system architecture as well as a video demo to show examples of the system's capabilities. We provided time for questions to ensure that all participants understood the concept of data-driven game streaming and how it might be instantiated in a system.

After context setting, we moved to a brainstorming phase. Participant brainstorming was broken into *individual brainstorming* followed by *collaborative brainstorming* [11]. This strategy maximizes the diversity of participant ideas, while providing shared context during the group discussion.

As individuals, participants were given fifteen minutes to generate ideas on sticky notes on Figma, in response to three guiding questions:

(1) How do you envision MARS being used by streaming stakeholders such as viewers, streamers, moderators, and developers?
(2) How can people benefit from this technology?
(3) What features would you like to see in MARS supporting?

Participants were then instructed to look at and annotate all sticky notes on the board. If they liked an idea, they could drop a “kudos” sticker on it. If the note was not clear, participants could leave a question mark sticker. Facilitating natural discussions in a workshop among strangers was difficult when done remotely [12], and the kudos and question mark approach provided a segue for participants to start discussing with one another. Once participants were done annotating, the facilitator read the notes marked with the question marks out loud, which gave the participants an opportunity to clarify and discuss their ideas with the whole group. At this point, participants were given five minutes to collectively answer one final guiding question:

(1) What ethical, social, personal, or technical challenges can you foresee occurring when using this technology?

The final hour of the session focused on sorting and grouping the generated notes through affinity diagramming. Following standard affinity diagramming protocol, participants took turns placing one single note into an existing group; if the note did not fit within that group, participants created a new note grouping. Sorting and grouping of the notes was done based on the similarities or connections among the underlying ideas, as participants were regularly reminded not to group notes based on keywords. Throughout this grouping process, participants were engaging in discussions to negotiate the placement and refine the clusters.

When all notes were grouped, and the participants were satisfied with their groupings, the session concluded by generating theme sentences for each note cluster. These had to be short, descriptive sentences that captured the underlying relationship between a group of notes. To conclude the grouping activity, participants were instructed to take a final pass of the groups of notes and their associated themes to discuss any final changes.

3.3 Inter-Phase Analysis and Phase Two Preparation

In total, we conducted six affinity diagramming sessions in Phase One involving 26 participants, which generated approximately 370 notes in 58 groupings. Our next task was to consolidate knowledge across the six different affinity diagrams (one from each session). Following our established affinity diagramming protocol, we categorized the themes and their corresponding groups into higher-level groupings. This process mirrored the affinity diagramming procedure employed by participants, with each researcher taking a note group and systematically placing these groups into larger categories. This approach took multiple iterations before all research team members were satisfied with the final resulting affinity diagram.

In generating the finalized affinity diagram, two distinct higher level categories emerged; feature-driven categories that focus on an experience the design should be be built around, and categories which should apply to all data-driven game streaming experiences. We named these two categories *Design Goals* and *Design Considerations*, respectively. Design Goals encompassed specific functionalities. Categories labeled as Design Goals used the "As a [role], I want to [action]" format. For example, "As a viewer, I want to meaningfully contribute to the game being played by the streamer" denotes a Design Goal category. A related, albeit distinct Design Goal category that emerged was "As a viewer, I want to be able to play a game with/against the streamer." Such Design Goals could potentially be realized through a data-driven game stream experience found in pre-existing audience participation games [35]. However, not all data-driven game stream experiences need to support these gameplay-oriented features; an interface for visually impaired viewers, for example, need not enable game contributions from the viewer for it to be successful. On the other hand, Design Considerations, unlike Design Goals, emerged as a category that describes guidelines applicable to *all* data-driven game streaming experiences. Categories labeled as Design Considerations were largely focused on mitigating challenges and concerns that participants had about data-driven game streaming, and are framed as questions. For example, "How does the system enable voluntary interaction/opt-in mechanisms for viewers?," is a Design Consideration that emphasizes that no viewer should be forced to participate if they do not wish to, reflecting the stakeholders' emphasis on respecting viewer autonomy.

In total, we identified eight Design Goal categories and seven Design Consideration categories, listed in Table 1 and 2 respectively. It is worth noting that of the eight Design Goals, only one is focused on streamers and the rest are viewer-facing goals. This is to be expected, as viewers are often the main recipients of novel features to help bolster their engagement, especially since viewers often have the least amount of agency to influence live gameplay [18, 35]. This in turn could have influenced our participants, who may already have familiarity with existing data-driven game streaming systems that focus on viewer-facing features. Besides viewers, the needs of the other stakeholders in our study largely manifested as Design Considerations, most of which related to ethical challenges and managing the workloads of streamers, moderators, and developers.

We then used these findings to prepare for the second phase of the study, a design workshop in which returning participants would create sample data-driven interfaces. Due to the use-case-specific nature of the Design Goal category, we matched each Design Goal with specific games (for our final choices, see Table 1). To ensure we were selecting games our participants could use as design material, we sent a survey to all participants before Phase Two commenced. In the survey, participants were asked to rate their familiarity with each of 29 possible games on a scale of 1-5, with 5 meaning they are confident they can comfortably design for that game. We then used this data to create groups of 2-3 participants who were all familiar with the same game, but who had diverse perspectives based on their Twitch stakeholder type.

With this comprehensive understanding, we transitioned to the next phase, the design workshop, leveraging these insights to explore the design space of data-driven game streaming interfaces with stakeholders.

3.4 Phase Two: Design Workshop

After the affinity diagramming from Phase One, we were able to address RQ1 and 2 through a list of Design Goals and Design Considerations essential for data-driven game streaming interfaces. With these Design Goals and Considerations in hand, Phase Two shifted focus to address the last research question: How can stakeholders contribute to the design of data-driven game streaming systems? Thus, the overarching goal of Phase Two was to address this research question by exploring the insights from Phase One through a design workshop with stakeholders.

We conducted four Phase Two sessions with a total of 21 returning participants. While Phase Two had availability constraints which affected the session groupings, such as needing to ensure all participants were familiar with a game if they were assigned a design for it, more attention was paid in Phase Two to place participants into groups for brainstorming purposes. Each group needed to have multiple perspectives of stakeholders to ensure fruitful discussion, meaning no group was made up entirely of viewers, for example. Instead, participants were placed into groups with other participants who identified as a different stakeholder type. Each session started with a recap of Phase One, followed by an overview and discussion of our results, the derived Design Goals, and the Design Considerations. We then divided participants into their assigned groups.

Design Goals	Games
As a viewer, I want to personalize and control the information I see during the stream.	*The Witcher 3: Wild Hunt*
As a viewer, I want the system to give me contextual information that is needed to understand the game stream	*Baldur's Gate 3*
As a viewer, I want to meaningfully contribute to the game being played by the streamer	*Portal 2*
As a viewer, I want to be able to play a game with/against the streamer.	*Among Us*
As a viewer, I want to use the system to engage with other viewers	*Animal Crossing: New Horizons*
As a streamer I want to use data from the system to better understand my audience and/or my gameplay during or after the stream	*Sid Meier's Civilization VI*
As a viewer with visual impairments, I want to use the system to make the game stream accessible to me	*The Elder Scrolls V Skyrim*
As a viewer who is not fluent in English, I want the system to translate the in-game text to a language of my choice	*Cyberpunk 2077*

Table 1. The Design Goals derived from Phase One and the games they were matched with in Phase Two

No.	Design Considerations
1	How can the system enable features which excite streamers without adding too much to their workload?
2	How can the system incorporate features to help moderators do their job?
3	How does the system mitigate trolling/cheating behavior?
4	How does the system implement industry-standard practices such as privacy and security measures?
5	How does the system streamline the development process to minimize the added work developers need to do to get the system working for them?
6	How does the system foster communication between all stakeholders such as viewers, streamers, and developers?
7	How does the system enable voluntary interaction/opt-in mechanisms for viewers?

Table 2. Summary of Design Considerations found in Phase One

Each participant group was assigned a specific game and Design Goal for which they were to design data-driven game streaming prototypes. Figma boards were set up for each group before the session, containing their assigned Design Goal, relevant game information, and designated areas for designing. Unlike Phase One, where participants used Figma largely for writing on sticky notes and dragging them across the board, Phase Two emphasized Figma's design features for remote collaboration and low-fidelity prototyping.

Before starting group-based design work, participants were first given 15 minutes for individual brainstorming of possible prototypes to create. This was followed by participants being placed in breakout room sessions with their group members for collaborative brainstorming and design refinement. Each group was asked to create two separate designs, with 15 minutes dedicated to each, ensuring equal time allocation for both ideas.

Similar to Phase One, guiding questions were once again provided to participants to aid their design process. These questions were asked when group design activities commenced; What kind of data would you need in order to make the overlays for the game? Would the data need to be bi-directional? Can the viewers send data back to the game or to the viewer? How would the front-end display/utilize this data? How would the user/viewer interact with the overlay?

After the collaborative prototyping period, participants were instructed to refine both of their designs by reflecting on the Design Considerations from Phase One. The full list of Design Considerations from Table 2 was provided to participants to encourage them to think about if their designs were not meeting concerns such as enabling voluntary interaction mechanisms for viewers. They were given ten minutes to collaboratively iterate their low-fidelity prototypes with these Design Considerations in mind.

Upon returning from breakout rooms, participants presented their designs to the whole group, facilitating discussions on design intent and allowing researchers and other participants to ask clarifying questions.

In total, participants generated sixteen design prototypes for eight different games and their corresponding Design Goals. To analyze these designs, we employed an iterative coding approach inspired by qualitative coding. Initially, the lead researcher annotated each prototype using sticky notes, capturing key features as described by participants. The goal of this initial annotation was to get a feel for the characteristics of each prototype. Examples of annotations include descriptions of audience participation features in the *Portal 2* designs. Following this initial annotation, the lead researcher engaged in discussions with the research team to review the observed patterns and themes.

Subsequently, a more comprehensive analysis was conducted. The lead researcher categorized each design, utilizing a spreadsheet to delineate more specific features. These attributes, including factors such as chat implementation, data usage, and target user (e.g. streamer or viewer) were continuously refined with each iterative pass of the data. As patterns emerged, additional columns were added to the spreadsheet to accommodate the evolving understanding of the designs. The lead researcher iteratively added columns until every prototype was sufficiently described, leading to a total of 19 columns.

The goal of the previous prototype analysis was to be able to thoroughly capture the facets of each prototype. The next step in our analysis was to explore the underlying relationships between the designs.In addition, we also studied the *process* through which the participants generated their ideas, collected in our observational notes. We collated behaviors across participants to define common steps that participants initiated in their ideation and analysis, such as their approaches for thinking about what data is needed for their designs and how to render that data. We then integrated the process data with our spreadsheet analysis of the designs, as well as with our data from Phase One of the study, to understand the underlying patterns and relationships between prototypes. This relationship exploration was an iterative process of the lead researcher reviewing the data, sketching possible models, and discussing the meaning of these models with the group.

This process facilitated more robust models that organized the shared attributes we found in the data, ultimately revealing insights which contributed to the formulation of an early-stage design framework for data-driven game streaming experiences. This framework provided a structured approach to understanding and categorizing the diverse array of design elements observed across the study, offering valuable insights into the potential implementation and implications of data-driven game streaming interfaces.

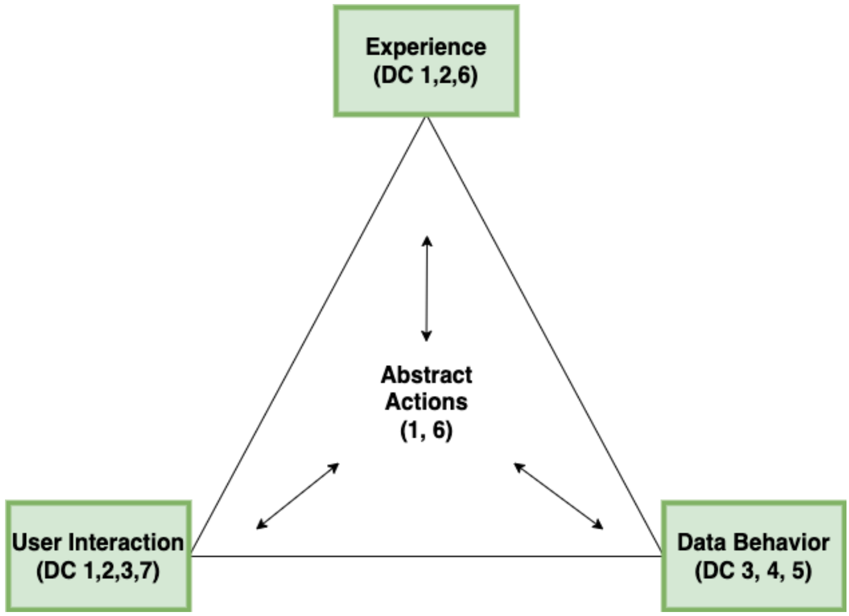

Fig. 1. The design framework for data-driven game streaming

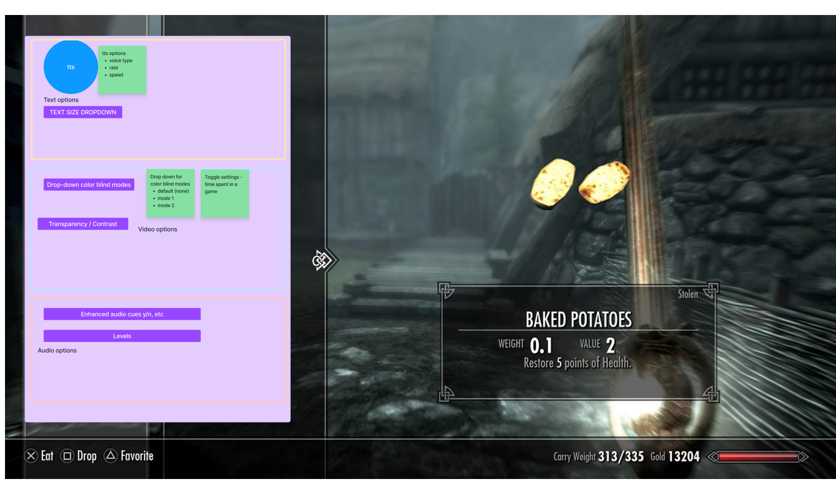

Fig. 2. A design for visually impaired users using the game *The Elder Scrolls V: Skyrim*

4 DESIGN FRAMEWORK FOR DATA-DRIVEN GAME STREAMING

As part of the sense-making process for our multi-phase study, we created a design framework that incorporates Design Goals and Design Considerations from Phase One, the analysis of the design process from Phase Two, and theories from existing literature on APG design [35, 37]. This framework is illustrated in Figure 1, which shows the relationship between the core *components* of our framework. Each of the outer components of the triangle represent a necessary facet of the design: establishing the desired experience (*Experience*), creating an effective interface (*User Interaction*), and powering the user needs and the interface with data (*Data Behavior*). In Phase

Two, participants approached their design in a sequential order (Experience, Abstract Actions, User Interaction, and Data Behavior) and were not able to revisit previous steps in their process due to time constraints and the nature of the study procedure. Part of our decision making in structuring the framework in this way is because we noticed early on in our analysis that there was an interconnectedness between the different decisions that a designer would have to make. Decisions related to abstract actions in particular clearly had greater implications for the other aspects of the design. We adapted the framework to reflect that interconnectedness such that design teams, which are comprised of streaming stakeholders, can iterate within each of these components, or move back and forth between them, as part of their iterative design process.

Fig. 3. A design for streamers interested in visualizing data generated from their stream of the game *Sid Meier's Civilization VI*

Central to the framework and linking the other components we place *Abstract Actions*, which serve as an *alignment mechanism* between these three perspectives in the design process [7]. These abstract actions help creators consider each of the components within relation to one another, helping them instantiate the experience in interfaces that respect the technical capabilities of a data-driven game streaming environment. The bi-directional arrows to and from Abstract Actions represent this, and highlight that movement between the outer components requires alignment with the abstract actions first. As an alignment mechanism, abstract actions also serve to help interdisciplinary teams stay on the same page, even when they are working on different components of the project.

While Abstract Actions are the core of the framework, design teams can begin ideation from any of the outer components as well. For example, a design team might start from a set of Design Goals based on user research (*Experience*) and use these goals to then consider what interaction mechanisms (*User Interaction*) would be necessary to create them and what in-game metadata (*Data Behavior*) is required. Alternatively, the design team can take a data-first approach, starting from a sense of what metadata can be sent from the game (*Data Behavior*) and from there align with Abstract Actions to ideate on possible interfaces (*User Interaction*) and what novel experiences (*Experience*) they can create. As designers move through these outer components, they can make use of the idea of Abstract Actions as a lens to help coordinate relationships between them.

To illustrate each of the elements of our framework, we make use of three designs created by participants from Phase Two. These include a design for the game *The Elder Scrolls V: Skyrim* [38], which offers accessibility features to visually impaired viewers (figure 2); a design for streamers to make sense of data generated while playing the game *Sid Meier's Civilization VI* (figure 3) [14]; and a design that enables viewers watching a stream of *Portal 2* [39] to contribute to gameplay (figure 4). We then demonstrate how the framework can be used to better understand existing systems by using it to analyze *Helpstone* and how the interaction opportunities afforded by the system relate to its target experience and available data [22].

4.1 Abstract Actions

The central component of the framework is about understanding abstract user actions by asking the fundamental questions: What will the user do? What actions are they expected to take? These abstract actions are *interface-agnostic*, meaning they are not concerned with the visual appearance or specific implementation of the design. The focus of this component is to identify *what* actions need to be performed rather than dwell on *how* these actions can be taken.

In our Phase Two study, we extracted abstract actions from the process we observed from our participants in Phase Two, where they only brainstormed the user actions after discussing the Experience and never iterated upon these actions. However, in a more realistic situation, design teams would repeatedly return to the Abstract Actions component to ensure that their design vision, the interactions they create, and their technical and data behavior decisions are aligned. For example, a team might revisit their abstract actions after modifying their target user persona, after testing a particular interaction, and/or after identifying and implementing a new data pipeline they can integrate. Any resulting changes to the abstract actions then inform the iteration at the outer components of our diagram.

As part of Abstract Actions, design teams are invited to consider how their work enables exciting features without overwhelming streamers and how the actions can foster communication among stakeholders (Design Considerations 1 and 6) The abstract actions for the *Skyrim* design are:

(1) Manipulate game text properties
(2) Manipulate visual properties of the game such as contrast
(3) Enhance in-game audio volume
(4) Filter out audio channels

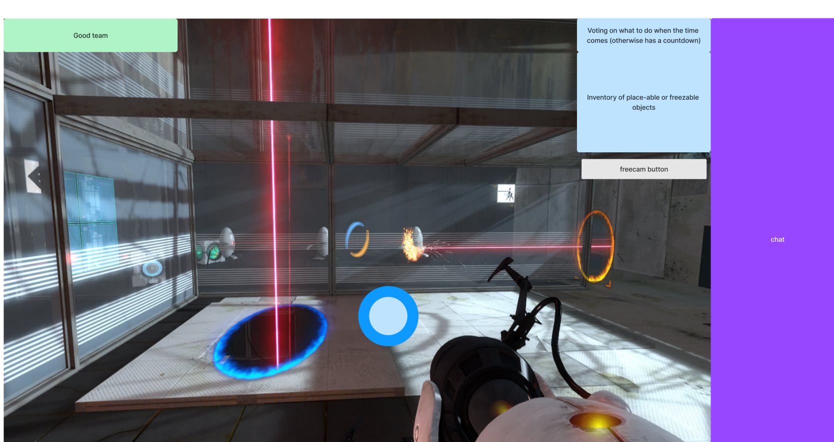

Fig. 4. A design for viewers to contribute to gameplay in *Portal 2*

The abstract actions for the *Civilization VI* design are:

(1) View notable chat messages and keywords sent by viewers
(2) View region-based statistcs about viewers
(3) View metrics across the timeline of the stream such as viewer retention data and key moments. This timeline feature allows streamers to see trends across the whole streaming session and can be revisted for post-hoc analysis after the stream ends

Finally, the abstract actions viewers can take in *Portal 2* are:

(1) Select team (with streamer, against streamer) to join
(2) Contribute to the decision of what objects to add to the environment to help/hinder the streamer
(3) Contribute to the decision of what objects to freeze in the environment to help/hinder the streamer
(4) View the game level or environment

In particular, note that the *Civilization VI* actions are focused on viewing and sense-making, instead of being limited to activity verbs as one might expect in a game environment [2]. Similarly, note that the actions in *Portal 2* are interdependent with those taken by other players, which we address in the User Interaction component of the framework.

4.2 Experience

The primary focus of the Experience component is on establishing a clear design goal and target user. In our study, participants began their prototyping in Phase Two at this component since they were assigned a Design Goal at the start of the study session. The Experience component is representative of the Design Goals derived from Phase One, and the key questions addressed in this component are: What is the overall goal of the design? Who are the intended users? While our sample primarily targeted viewer needs, design teams can extend beyond this stakeholder group; additionally, even within these groups they can identify specific user subtypes as well, such as visually impaired viewers. This component is also where design teams consider how they can accommodate streamers without overloading them with additional tasks that hinder their experience, how their goals impact moderators, and how their goals can foster communication across stakeholder groups (Design Considerations 1,2, and 6).

Our examples demonstrate some of these possibilities. For the *Skyrim* design, the design objective was to create a system to make a game stream accessible to a viewer with visual impairments. For *Civilization VI*, the objective was to enable a streamer to visualize real-time data from a stream to better understand their audience or gameplay. For *Portal 2*, the objective was enable gameplay motivated viewers [35] to meaningfully participate in the streamer's game.

4.3 User Interaction

While abstract actions asks design teams to identify *what* abstract actions the user can take, the User Interaction component focuses on *how* the user completes those actions. Our approach to this component is designed to help teams navigate the specific challenges that come with prototyping interfaces for streaming (e.g. [15]). In our study, the participants began sketching low-fidelity prototypes after discussing the goals and actions they wanted to explore (Experience and Abstract Actions components), and we derived the elements of this component when analyzing their resulting designs. Design teams could start at this component if they have particular features or user interface features they wanted to prototype first. They can then return to this component after re-aligning with the abstract actions of the design.

Along with prototyping, this component comprises of two stages where design teams must first establish the *interaction dimensions* followed by the *responsiveness* of the target user's actions.

Mode	Target	Overlay	Game Environment	Game Mechanics	Communication
Individual		View notable chat messages/keywords, view region-based metrics, view timeline-based metrics	None	None	None
Collective		None	None	None	None

Table 3. Example of the interaction dimensions with the *Sid Meier's Civilization VI* design

Mode	Target	Overlay	Game Environment	Game Mechanics	Communication
Individual		Select team to play with or against streamer using overlay buttons, select "Freecam" button to explore game environment	"Freecam" feature to view game environment independently from other viewers and streamer	None	None
Collective		Select what objects to place or freeze from available pop-up	None	Click to vote on objects to freeze or place	None

Table 4. Example of interaction dimensions with the *Portal 2* design

4.3.1 Interaction Dimensions. Interaction dimensions can be broken down into the *modes* and *targets* where they take place. Here, the emphasis is on describing and classifying interaction opportunities available to the user. Our use of modes and targets builds on Striner et al's design space of *agency* [37], and on individual and social agency as defined by Seering et al [35]. We expand this prior work to generalize the desire for "personalized control" beyond just play and APG-based mechanics, and to capture a broader array of multi-user interactions in the data-driven game streaming design space that do not necessarily have to be "social".

The two modes of interaction are *individual* and *collective*, each having the possible targets of *overlay*, *game environment*, *game mechanics*, and *communication.*

In an individual interaction mode, the result of an interaction is, as the name suggests, personalized to a single user. The interaction does not impact nor rely on the experience of other users of the system. For example, an interface that allows viewers to independently hover over game elements to view their attributes is considered an individual interaction, as viewers are free to hover on any element they want without the hover interaction affecting other viewers. Interactions classified under the collective mode, on the other hand, offer capacities that impact and/or rely on other users' actions, resulting in a shared interaction experience among users. For example, an interface that allows viewers to click on a point in space and add a sticker to the game's environment is a collective interaction, as that sticker is now visible by other viewers. In our study, we observed individual and collective interactions manifesting in four main ways.

We capture these as four types of interaction, which offer different types of *agency* for the user to impact the experience. These targets are: overlay, game environment (which includes visuals and audio), game mechanics, and communication. The overlay target pertains to interactions that directly manipulate the rendered overlay. The game environment target grants users control over what they see and hear within the game, while the game mechanics target allows users to engage with game mechanics to progress gameplay, such as helping or hindering mechanics in APGs [15, 35]. Lastly, the communication target encompasses the channels dedicated specifically to facilitating communication among stakeholders like viewers, streamers, and developers.

Taken together, the interaction modes and targets allow designers to break down the user experience into interactable components, which can help design teams evaluate the patterns and tradeoffs in their design. Tables 3 and 4 show example task breakdowns of the *Civilization VI* and *Portal 2* designs respectively. The *Civilization VI* example shows how multiple types of interaction may target the same mode (individual) and target (overlay). While there are differences between the actions, they all involve the individual viewer interacting with overlay data.

The *Portal 2* example, on the other hand, demonstrates how a single design can include actions in multiple modes (individual, collective) and with multiple targets (overlay, environment, mechanics). More specifically, this demonstrates how a single abstract action such as "Contribute to the decision of where to place added object" can be broken down into multiple target interactions. Here, the overlay target describes the UI components that users interact with, such as using a pop-up which appears when voting for object placement is available. Along with this overlay target, the underlying mechanism powering the decisionmaking of where to place the added object is a game mechanics target, which is controlled using a voting system.

4.3.2 Responsiveness. In addition to interaction modes and targets, the next aspect to consider is the *responsiveness* of the design. This is the frequency of interaction and updates, which can be inferred by the interaction dimensions. For example, actions in the collective mode often involve coordinating with other users, which can slow the pace of updates. and result in lower responsivness.

Given that accurate response times are impossible to generate at the low fidelity prototyping stage, designers can estimate how long the user needs to wait to see the results of their actions as "low", "medium" or "high." For example, a low responsiveness action in the *Portal 2* design is when a viewer places a vote for their decision to freeze an object to help or hinder the streamer using the voting pop-up. After a viewer places their vote, they have to wait until the voting system aggregates all votes to see the outcome of that action. Conversely, in the *Skyrim* example, the interaction to manipulate the text properties of the game manifests as highly responsive in this design. This is due to the design's use of individual modes of interaction that do not depend on other stakeholders.

Table 5 shows the responsiveness of the *Portal 2* design. The table lists the interactions from the interaction dimensions table for that design to illustrate the impact those decisions have on responsiveness. Here, the individual interactions of the design result in high responsiveness, whereas the collective interactions yielded low responsiveness.

In terms of Design Considerations, the design team is again asked to consider how their evolving design is accommodating the streamers' workloads, impacting moderators' workflow, mitigate cheating, and enable opt-in participatory mechanisms for viewers (Design Considerations 1,2, 3,7).

4.4 Data Behavior

The Data Behavior component is concerned with one of the critical pillars of a data-driven game streaming system; the data itself. In our study, we invited our participants to consider the data needed to power their designs towards the end of the study sessions after they had completed two iterations of prototyping. However, design teams can start here if they already have existing

Interactions	Modes	Targets	Responsiveness
Select team to play with/against the streamer using overlay buttons	Individual	Overlay	High
Select what objects to place/freeze using available pop-up window	Collective	Overlay	Low
Click to vote on objects to place or freeze	Collective	Game Mechanics	Low
Use freecam feature to view game environment	Individual	Overlay, Game Environment	High

Table 5. Example of responsiveness with the *Portal 2* Design

data pipelines they want to leverage or if they want to design around the technical constraints of their systems. Design teams can return to the Data Behavior component if further iterations of the design require a re-evaluation of the required data behavior.

To capture the essential data operations needed from each stakeholder, we borrow from file system terminology to define the *data privileges* of each stakeholder. In considering data operations, four primary data privileges emerge: reading, writing, deleting, and manipulating data. While reading, writing, and deleting data are fairly self-explanatory, the concept of data manipulation warrants further exploration. Data manipulation refers to taking existing data and altering its properties. Filtering menu data or altering the color of video data are examples of manipulations. The framework uses these four data operations to prompt design teams to consider what data is needed to power their design and what the stakeholders' relationship to that data is.

Along with data privileges, we define the concepts of data *synchrony* and *throughput*, which capture how often data needs to be sent and the volume of data that needs to processed. Similar to responsiveness, throughput can be estimated by design teams using "high", "medium", and "low" to denote volume.

By comprehensively analyzing data behavior, the framework empowers design teams to tailor data operations to suit the diverse needs of stakeholders and, to understand the constraints and opportunities the available data place on the design, and to optimize the overall user experience. The data behavior of the *Skyrim* and *Portal 2* designs can be found in Tables 6 and 7 respectively. The *Skyrim* design enables viewers to personalize their game viewing experience for accessibility purposes, which means viewers *read* incoming data and can *manipulate* its properties. The nature of this design requires synchronous data because viewers expect to see their personalizations happen in real-time; if the data pipeline is disrupted, the user experience is disrupted. The data required is classified as high throughput, since audio, visual, and text data can often be large in size.

Conversely, the voting features of the *Portal 2* designs means that viewers need to *write* that data for developers to *read* in the game. Developers then need to *write* data in response to those voting actions. Additionally The voting data needed for the *Portal 2* design is asynchronous in nature, since delays are necessary to wait for all viewers to finish voting. This data in this case is also classified as low throughput, since vote tallying does not require large volumes of data to be written by each viewer. Though it is theoretically possible that a stream may have hundreds of thousands or millions of viewers attempting to vote, we still classified this as a low throughput data operation since it is unlikely that a stream will have enough viewers to throttle the data pipeline with viewer votes.

With respect to the Design Considerations at this component, design teams are asked to consider the aspects of the data that can mitigate trolling/cheating, implement privacy and security measures, and reduce the workload on developers (Design Considerations 3,4, and 5).

Data	Viewer Privileges	Streamer Privileges	Developer Privileges	Synchrony	Throughput
Audio channel properties	Read, Manipulate	None	Write	Synchronous	High
Game visual properties	Read, Manipulate	None	Write	Synchronous	High
Game text	Read, Manipulate	None	Write	Synchronous	High

Table 6. An example of the Data Behavior component with *Skyrim*

Data	Viewer Privileges	Streamer Privileges	Developer Privileges	Synchrony	Throughput
Camera location	Read, Manipulate	None	Write	Synchronous	High
Freezeable and placeable objects list	Read	None	Write	Asynchronous	Low
Voting data	Read, Write	None	Read, Write	Asynchronous	Low

Table 7. An example of the Data Behavior component with *Skyrim*

5 ANALYSIS OF AN EXISTING DATA-DRIVEN GAME STREAMING SYSTEM

Thus far, we have illustrated our framework using three design examples taken from Phase Two of our study. These three designs were created by our participants through a procedure that started with the Experience component. To demonstrate the framework's flexibility and its applicability to existing data-driven game streaming systems [22], we use it to analyze *HelpStone*. The *HelpStone* system improves communication between the streamer and their viewers, and it does so by offering various interactions for viewers to engage in the gameplay. For this reason, we begin our analysis from the User Interaction component; for reasons of scope, we demonstrate our analysis using the viewer interface rather than also including the corresponding streamer interfaces. Table 8 classifies the interaction dimensions of *Helpstone* into their respective modes and targets. It offers communication channels that viewers could use to quickly convey information via the thumb up/down feature, or to provide lengthier feedback in the form of text comments. It also allows viewers to interact with the overlay both individually and collectively.

While *HelpStone* excels at providing a variety of interaction opportunities for viewers, most interactions happen at the overlay target, and there are no interactions that occur at the game environment or mechanics targets. This is due to the fact that the available interactions for the viewer provide suggestions for the streamer that the streamer can take up or ignore, rather than taking actions in the game space directly. Taken together, these insights allow us to see possible future design opportunities for *Helpstone* by allowing viewers to manipulate the game environment properties, or to directly affect gameplay mechanics. For example, a future iteration of *HelpStone* could allow users to customize the game's sound effects (game environment) or add cards to the streamer's deck (gameplay mechanics).

With regards to the responsiveness of the interactions, Table 9 summarizes the responsiveness of the available interactions. One notable aspect about the design of *HelpStone* is that its interaction design of features such as drawing zones of interest allow for high responsiveness even in a collective mode. Enabling collective interactions that have high responsiveness was something

Mode	**Target**	**Overlay**	**Game Environment**	**Game Mechanics**	**Communication**
Individual		Use clock widget to synchronize with the streamer, hover over cards to browse historical info, use widget to browse through previous turns and see what cards were played	None	None	None
Collective		Draw lines between zones of interest, use archetype widget to vote on play style	None	None	Use thumbs up or thumbs down to convey feedback to the streamer, write comments to the streamer

Table 8. A breakdown of the interaction dimensions in *Helpstone*

Interactions	**Modes**	**Targets**	**Responsiveness**
Use widget to synchronize time with streamer	Individual	Overlay	High
Draw lines between zones of interest	Collective	Overlay	High
Use widget to browse cards played in a previous turn	Individual	Overlay	High
Use archetype widget to vote on playstyle	Collective	Overlay	Low

Table 9. *Helpstone's* interaction responsiveness

our participants struggled with, yet *HelpStone* demonstrates that it is possible to create collective interactions that are highly responsive for viewers. In parallel, *HelpStone* effectively uses voting mechanisms to achieve its goals, particularly in the play style interaction where viewers vote on the play style they would like the streamer to take on. This is an example of how low- and high-responsiveness interactions can co-exist in the same experience.

We can identify the abstract actions of *HelpStone* from the User Interaction component analysis. The abstract actions for viewers in *HelpStone* are:

(1) Synchronize to streamer view
(2) Provide gameplay suggestions to the streamer
(3) View historical information about previously played cards
(4) Provide real-time feedback to the streamer's actions
(5) Comment on gameplay to communicate with the streamer or other viewers

Using Abstract Actions, we are able to see the data needed to power these interactions. Table 10 summarizes a subset of the data needed to power the interface. In this case, our table shows that game developers are not highly involved in reading data, which is due to all of the data needed to power these interactions being *retrospective* as it is obtained from game logs [22]. In other words, the data is generated by the game, but then is read-only by all other stakeholders. This constraint provides technical reasons why *Helpstone* as originally designed does not interact with the game environment or the mechanics.

Data	Viewer Privileges	Streamer Privileges	Developer Privileges	Synchrony	Throughput
Streamer Clock Data	Read	None	Write	Synchronous	Low
Viewer Clock Synchronization Input	Write	None	Write	Asynchronous	Low
Viewer line drawing data	Write	Read	None	Synchronous	Low
Viewer voting data	Write	Read	None	Synchronous	Low

Table 10. The Data Behavior breakdown of *HelpStone*

In the case of *Helpstone*, the designers had clear Experience goals and shared them in their research paper [22] (Fostering viewer-viewer engagement, contextual information, and meaningful game contribution for viewers). It is perhaps disingenuous of us to pretend that we are not aware of these goals during this analysis. However, we can see that the goals as discussed in the paper are linked directly to the abstract actions that we describe. Because these actions are then connected to both the user interface and to the data layer, we can see how abstract actions provide a way to connect these three design approaches regardless of where one begins the process.

6 DISCUSSION

In conducting this study, our aim was to move the design space of data-driven game streaming forward by working with target stakeholders. We approached this by asking what features streaming stakeholders wanted to see in data-driven game streaming systems (RQ1), what challenges and concerns stakeholders have when it comes to this technology (RQ2), and how stakeholders can contribute to the design of such systems (RQ3). Our work yielded Design Goals and Design Considerations that stakeholders believed were important, and we synthesized these findings into our design framework to contribute the stakeholders' perspectives in the forming of this design space.

Our pursuit of these research questions revealed further lessons and broader implications for future work in this area. We elaborate on these lessons by first discussing in-depth the insights derived from analyzing the Phase Two designs using the framework. We then discuss the methodological insights we derived from working with many types of stakeholders together with regards to managing incentives. Finally, we discuss the implications and limitations of our work in developing the design space to include other forms of data-driven streaming.

6.1 Insights From Designs

While analyzing our data and creating the framework, we noted how interconnected the components of our framework were. A notable pattern that demonstrated this interconnection within the framework was the *tradeoffs* introduced when iterating between or within components . Most notably, choosing between individual or collective modes has implications for which targets, responsiveness levels, or data behaviors are easiest or most salient to envision interactions for.

Across the designs our participants envisioned, concepts that used an individual mode never used a game mechanics target. This means there were no capabilities for users to influence gameplay individually, only as a collective. Further, when game mechanics targets were included in collective mode concepts, they exclusively made use of voting mechanisms to deal with aggregation of inputs.

This is demonstrated in the *Portal 2* design, and all other designs in our data used similar voting mechanisms. This is a known trade off between enabling individual capacities to affect gameplay and accounting for scale [35]. However, there are demonstrated ways to afford individual capacities [43] as well as different strategies of aggregation beyond the standard majority rule [21]. This highlights the potential for a broader set of interactions with gameplay mechanic targets which afford individual capacities or deeper collective capacities, as seen in our *Helpstone* analysis.

Similarly, we saw a tradeoff at the responsiveness level. Individual interactive modes, such as the ones found in the *Skyrim* design, tended to be highly responsive while collective mode interactions, such as *Portal 2*'s, were lower in their responsiveness. This is partly a consequence of technical constraints imposed by collective modes' need to coordinate data between multiple parties such as the game, steamers, and other views. Individual model interactions, on the other hand, are not reliant on all parties coming to collective agreement.

From applying the framework to the concepts our participants created, we observed that data behaviors were often emergent from the prior choices rather than being explicit choices in themselves. For example, the collective designs that relied on voting/input for game mechanics tended to be asynchronous and low throughput, as demonstrated by *Portal 2*'s data behavior table (Table 7). Conversely, individual designs that have lots of interactions to enable a personalized experience tend to rely on synchronous, high throughput data, as seen in *Skyrim*'s design and Table 6. Moreover, because individual modes did not tend to offer direct impacts on the game, users were more often consuming or manipulating data rather than acting as data producers.

An additional notable pattern that we observed was specific to the Design Goal "as a streamer, I want to use data from the stream to better understand my audience and/or my gameplay during or after the stream", which was the design objective for *Civilization VI*. In Phase One, participants commonly expressed a desire for analytics solutions for streamers, which is corroborated in the literature as a common need [27]. However, when asked to envision new concepts to address this need, it was challenging for participants to come up with novel concepts, instead focusing only on using only an overlay target for interaction as opposed to a broader mix of targets.

We hypothesize this might be because of a few different reasons. First, analytics systems are complex and require effective integration and visualization of data, which might be hard to imagine for our participants. Second, participants might be anchored to their prior experience with analytics interfaces, particularly Twitch's built-in data dashboards. Third, there may be a limited data literacy in this population which makes it difficult to understand how different sources of data can serve them and their needs.

6.2 Managing Stakeholder Incentives

As a research community, we endeavor to design and develop solutions to support streaming stakeholders, thereby advancing the design space of data-driven game streaming. Creating data-driven streaming ecosystems that are successful in the long run will need the buy-in from all stakeholders involved. However, it is crucial to acknowledge that stakeholders across the streaming ecosystem may have conflicting needs and incentives, and trying to meet a particular stakeholder's needs may impact those of another stakeholder. For example, much has been written about the community and entertainment motivations of viewers and streamers [9, 17], however streamers, particularly those whose primary employment is streaming, have financial incentives that their audience does not [42]. Similarly, game developers and publishers have their own financial concerns [19].

In our study, these conflicting motivations and needs manifested in the Design Considerations from Phase One, particularly relating to maintaining autonomy and reducing workload. All stakeholders, be they viewers, streamers, moderators, or developers, valued autonomy in their data-driven game stream engagement. Where they differed, however, was how that autonomy is conceptualized. Viewers in our study were eager to design interactions so that they can personalize and impact the stream, while streamers and moderators sought to maintain autonomy in how they manage their communities. As a result, streamers and moderators at times expressed concern that by giving viewers individualized interfaces, it would make it easier for bad actors to disrupt the stream. Maintaining viewer autonomy while allowing streamers and moderators to maintain control of their communities is one possible conflict that can emerge in this context.

Concerns about workload were particularly pronounced among streamers, moderators, and developers, who expressed a desire to innovate without significantly increasing their workload. Streamers are tasked with playing, socializing, entertaining, and managing while live [44], and game developers are asked to enable features and data collection in their games. In our study, both parties were eager to design novel data-driven game streaming features that viewers wanted, however a major concern was how enabling those novel features could add to their workloads. Our findings underscore the importance of designing systems and interventions that alleviate these concerns, ensuring that the development of data-driven game streaming experiences does not unduly burden stakeholders.

Another source of conflict that poses additional challenges relates to the financial incentives of streamers and developers [42]. In the standard Twitch ecosystem, developers would need to ask streamers permission for advertisements. In our study, a professional streamer expressed apprehensions about how enabling data-driven game streaming features at the game-level may result in game developers bypassing them in monetization efforts. Developers on the other hand, may be more inclined to contribute features to a game being streamed if there are financial incentives at play. Balancing these incentives is crucial for fostering a healthy and sustainable ecosystem, necessitating nuanced approaches that account for the diverse motivations of stakeholders.

Our study highlights the significance of understanding the roles and concerns of each stakeholder in data-driven game streaming ecosystems. As researchers, we must integrate these considerations into our methodologies, creating spaces for stakeholders to voice their concerns and designing interventions that address their needs. Moving forward, future research should continue to explore the nuances of these systems, treating stakeholders as shared members of an ecosystem rather than individual groups with isolated needs. By doing so, we can better capture the complexities of data-driven game streaming experiences and foster collaborative innovation within the ecosystem.

6.3 Toward a Unified Design Language

In our framework, we capture the desired features of stakeholders in the form of Design Goals along with the Design Considerations to mitigate their concerns. We also provide a structured approach to explore design across various dimensions, including abstract actions, interactions, responsiveness, and data behavior. However, this framework represents just the beginning of a broader exploration within the research community. To advance the design space further, we propose two key areas of focus: incorporating a broader set of data-driven streaming use cases and considering additional data behaviors.

One use case that warrants further attention is eSports. While our framework applies in some ways to eSports spectatorship, there are nuances specific to eSports that we have yet to account for. For instance, eSports has a competitive motivation akin to traditional sports, with viewers often seeking to analyze gameplay strategies rather than directly impacting gameplay. Moreover, eSports may involve higher production values and additional stakeholders such as color casters, professional eSports teams, and post-match analysts. Financial concerns in eSports, including revenue generation channels, also differ from other forms of game streaming and merit consideration. Seen through the lens of data-driven streaming, the attributes of eSports introduces additional stakeholders while also placing different responsibilities on others. Analysts and commentators, for instance, are eSports-specific stakeholders whose role in the data pipeline remains unclear.

Additionally, it is worth considering how a how data-driven streaming may differ in contexts that do not involve traditional video games, such as coding streams or educational content. These streams have their own norms and constraints, and the absence of a "game" as a digital data source raises similar questions about the nature of the stakeholders' relationship to the data. In a coding stream context, for example, data may come from official technical documentation in addition to the direct content of the stream, suggesting the existence of additional interaction targets that would be context or use case dependent. Expanding the framework to capture additional data channels presents challenges, as it introduces complexity to the data landscape and may affect stakeholder relationships with the data. For example, in an educational context there may be additional privacy concerns related to the use data from students. Stakeholders' perceptions of data may not align with their perceptions of desired features, highlighting the need for a clearer understanding of their existing relationships with the data. Furthermore, the inclusion of new data sources and channels may impact stakeholders in unforeseen ways, necessitating careful consideration and exploration within the design space. By providing a common language for framing these concerns our framework creates a foothold to explore these issues.

7 LIMITATIONS AND FUTURE WORK

Our study encountered several limitations that point to avenues for future research and development in the domain of data-driven streaming interfaces. One notable limitation pertained to the low-fidelity nature of the designs generated during our study, coupled with incomplete details regarding data flow processes. This aspect suggests a need for refining the designs and elucidating the intricacies of data flow to enhance the practicality and comprehensiveness of proposed solutions. Additionally, while our framework provides a structured approach to analyzing data-driven game streaming interfaces, its efficacy in guiding design decisions warrants examination in later stages of the development process. Longitudinal studies could offer insights into the framework's performance throughout the design lifecycle, offering opportunities for refinement and optimization.

Another limitation we encountered is that there may blind spots in our findings as a consequence of how we structured the brainstorming and affinity diagramming activities. For instance, while the Design Considerations in our study offered a more balanced view of the concerns of all stakeholders involved, the Design Goals notably focused predominantly on viewers. Future work is necessary to develop design goals that address the experience needs of the broader ecosystem beyond viewers.

Another limitation related to the structuring of the study is that observed that participants primarily defaulted to the existing text chat rather than propose new communication channels, which is why the communication target was often not accounted for in interactions. This may be because participants were not explicitly prompted to think of new ways to communicate with stakeholders.

While our study included a diverse set of stakeholders across the data-driven game streaming ecosystem, our study was constrained by the limited number of moderators involved. Like streamers and developers, the desires of the moderators in our participant pool only manifested in the Design Considerations. Future research endeavors should prioritize recruiting a wider pool of stakeholders to ensure comprehensive coverage of Design Goals that reflect the needs of the diverse stakeholder ecosystem. More specifically, further collaboration with moderators is needed to establish effective Design Goals that moderation practices which are tailored to the unique dynamics of data-driven game streaming ecosystems.

Additionally, the exclusion of eSports and non-gaming streaming contexts from our study's scope limits the generalizability of our findings. An immediate next step for this work is to expand the design space to encompass these domains, exploring how data-driven streaming interfaces can adapt to suit their specific requirements and user behaviors.

Finally, while we identified key Design Considerations, integrating them into the framework remains a challenge. As a result, the Design Considerations mainly come into play as a final reflection in each component of the framework that encourages designers to incorporate the considerations in their design. Future iterations of the framework should prioritize seamless integration, ensuring that Design Considerations are not treated as an afterthought but rather as integral components at every stage of the design process. Addressing these limitations and pursuing future research directions will contribute to the advancement of data-driven game streaming interfaces, facilitating the development of more inclusive, engaging, and ethically grounded streaming experiences for all stakeholders involved.

8 CONCLUSION

In conclusion, our research on data-driven game streaming systems unveils critical insights into the needs, expectations, and experiences of stakeholders within the streaming ecosystem. By conducting a comprehensive two-phase design study involving streamers, developers, designers, moderators, and viewers, we have identified key Design Goals desired from data-driven streaming systems and outlined essential Design Considerations to ensure the technical, ethical, and social integrity of these systems. Building upon these insights, we propose a novel design framework aimed at analyzing and generating data-driven game streaming interfaces, thus paving the way for a more formalized approach to the design and development of such systems. Our framework provides a structured approach for designers to create, analyze, or improve data-driven game streaming experiences, emphasizing the importance of Experience, Abstract Actions, User Interaction, and Data. By providing a framework for future research, our work contributes to advancing the field of HCI and facilitating the development of more inclusive, engaging, and ethically sound data-driven streaming experiences. In doing so, we push the streaming experience to move past its humble video-and-text origins and into a new generation of live interaction and shared engagement.

ACKNOWLEDGMENTS

We would like to thank Jessie Chen and Seyoung Lee for their contributions to this project. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1942087.

REFERENCES

[1] 2021. Twitch.tv. https://www.twitch.tv/
[2] Anna Anthropy and Naomi Clark. 2014. A Game Design Vocabulary: Exploring the Foundational Principles Behind Good Game Design. (2014).
[3] Studio Bean. 2015. *Choice Chamber.* Game [Steam]. Studio Bean. Last played October 2016..
[4] Blizzard. 2014. *Hearthstone.* Game [Multiple Platforms].
[5] Jie Cai, Donghee Yvette Wohn, and Mashael Almoqbel. 2021. Moderation Visibility: Mapping the Strategies of Volunteer Moderators in Live Streaming Micro Communities. In *Proceedings of the 2021 ACM International Conference on Interactive Media Experiences* (Virtual Event, USA) *(IMX '21)*. Association for Computing Machinery, New York, NY, USA, 61–72.
[6] Sven Charleer, Kathrin Gerling, Francisco Gutiérrez, Hans Cauwenbergh, Bram Luycx, and Katrien Verbert. 2018. Real-Time Dashboards to Support eSports Spectating. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play* (Melbourne, VIC, Australia) *(CHI PLAY '18)*. Association for Computing Machinery, New York, NY, USA, 59–71.

[7] Tianying Chen, Jessica Hammer, and Laura Dabbish. 2019. Self-Efficacy-Based Game Design to Encourage Security Behavior Online. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3312935

[8] Yan Chen, Walter S Lasecki, and Tao Dong. 2021. Towards Supporting Programming Education at Scale via Live Streaming. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (2021), 1–19.

[9] Gifford Cheung and Jeff Huang. 2011. Starcraft from the stands: understanding the game spectator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 763–772.

[10] John Joon Young Chung, Hijung Valentina Shin, Haijun Xia, Li-Yi Wei, and Rubaiat Habib Kazi. 2021. Beyond Show of Hands: Engaging Viewers via Expressive and Scalable Visual Communication in Live Streaming. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21, Article 109)*. Association for Computing Machinery, New York, NY, USA, 1–14.

[11] Steven Dow, Julie Fortuna, Dan Schwartz, Beth Altringer, Daniel Schwartz, and Scott Klemmer. 2011. Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC,Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2807–2816.

[12] Therese Fessenden. 2020. When Remote Workshops Fail. https://www.nngroup.com/articles/remote-workshop-fail/. Accessed: 2024-2-22.

[13] C Ailie Fraser, Joy O Kim, Alison Thornsberry, Scott Klemmer, and Mira Dontcheva. 2019. Sharing the studio: How creative livestreaming can inspire, educate, and engage. In *Proceedings of the 2019 on Creativity and Cognition*. 144–155.

[14] Firaxis Games. 2016. *Sid Meier's Civilization VI*. Game [Multiple Platforms].

[15] Seth Glickman, Nathan McKenzie, Joseph Seering, Rachel Moeller, and Jessica Hammer. 2018. Design Challenges for Livestreamed Audience Participation Games. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play* (Melbourne, VIC, Australia) *(CHI PLAY '18)*. Association for Computing Machinery, New York, NY, USA, 187–199.

[16] Emily Halpin. [n. d.]. Introducing Extensions: A Streaming Revolution. https://blog.twitch.tv/en/2017/08/31/introducing-extensions-a-streaming-revolution-c31762addcd9/

[17] William A Hamilton, Oliver Garretson, and Andruid Kerne. 2014. Streaming on twitch: fostering participatory communities of play within live mixed media. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1315–1324.

[18] Noor Hammad, Erik Harpstead, and Jessica Hammer. 2023. The View from MARS: Empowering Game Stream Viewers with Metadata Augmented Real-time Streaming. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA,USA) *(UIST '23, Article 62)*. Association for Computing Machinery, New York, NY, USA, 1–13.

[19] Mark R Johnson and Jamie Woodcock. 2019. The impacts of live streaming and Twitch.tv on the video game industry. *Media Cult. Soc.* 41, 5 (July 2019), 670–688.

[20] Pascal Lessel, Maximilian Altmeyer, and Antonio Krüger. 2018. Viewers' Perception of Elements Used in Game Live-Streams. In *Proceedings of the 22nd International Academic Mindtrek Conference* (Tampere, Finland) *(Mindtrek '18)*. Association for Computing Machinery, New York, NY, USA, 59–68.

[21] Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. 2017. CrowdChess: A System to Investigate Shared Game Control in Live-Streams. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (Amsterdam, The Netherlands) *(CHI PLAY '17)*. Association for Computing Machinery, New York, NY, USA, 389–400.

[22] Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. 2017. Expanding Video Game Live-Streams with Enhanced Communication Channels: A Case Study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1571–1576.

[23] Lingyuan Li, Guo Freeman, and Nathan J McNeese. 2022. Channeling End-User Creativity: Leveraging Live Streaming for Distributed Collaboration in Indie Game Development. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2 (Nov. 2022), 1–28.

[24] Zhicong Lu, Michelle Annett, Mingming Fan, and Daniel Wigdor. 2019. " I feel it is my responsibility to stream" Streaming and Engaging with Intangible Cultural Heritage through Livestreaming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[25] Zhicong Lu, Seongkook Heo, and Daniel J Wigdor. 2018. StreamWiki: Enabling Viewers of Knowledge Sharing Live Streams to Collaboratively Generate Archival Documentation for Effective In-Stream and Post Hoc Learning. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW (Nov. 2018), 1–26.

[26] Chance Lytle, Parker Ramsey, Joey Yeo, Trace Dressen, Dong Hyun Kang, Brenda Bakker Harger, and Jessica Hammer. 2020. Toward Live Streamed Improvisational Game Experiences. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (Virtual Event, Canada) *(CHI PLAY '20)*. Association for Computing Machinery, New York, NY, USA, 148–159.

[27] Keri Mallari, Spencer Williams, and Gary Hsieh. 2021. Understanding Analytics Needs of Video Game Streamers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama,Japan) *(CHI '21, Article 337)*. Association for Computing Machinery, New York, NY, USA, 1–12.

[28] Matthew K Miller, John C Tang, Gina Venolia, Gerard Wilkinson, and Kori Inkpen. 2017. Conversational Chat Circles: Being All Here Without Having to Hear It All. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2394–2404.

[29] Rui Pan, Lyn Bartram, and Carman Neustaedter. 2016. TwitchViz: A Visualization Tool for Twitch Chatrooms. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 1959–1965.

[30] Kara Pernice. 2018. Affinity Diagramming for Collaboratively Sorting UX Findings and Design Ideas. https://www.nngroup.com/articles/affinity-diagram/. Accessed: 2024-2-21.

[31] Dennis Ramirez, Jenny Saucerman, and Jeremy Dietmeier. 2014. Twitch plays pokemon: a case study in big g games. In *Proceedings of DiGRA*. 3–6.

[32] Frans Rijnders, Günter Wallner, and Regina Bernhaupt. 2022. Live Feedback for Training Through Real-Time Data Visualizations: A Study with League of Legends. *Proc. ACM Hum.-Comput. Interact.* 6, CHI PLAY (Oct. 2022), 1–23.

[33] Raquel Robinson, Zachary Rubin, Elena Márquez Segura, and Katherine Isbister. 2017. All the feels: designing a tool that reveals streamers' biometrics to spectators. In *Proceedings of the 12th International Conference on the Foundations of Digital Games* (Hyannis, Massachusetts) *(FDG '17, Article 36)*. Association for Computing Machinery, New York, NY, USA, 1–6.

[34] Joseph Seering and Sanjay R Kairam. 2022. Who Moderates on Twitch and What Do They Do? Quantifying Practices in Community Moderation on Twitch. *Proc. ACM Hum.-Comput. Interact.* 7, GROUP (Dec. 2022), 1–18.

[35] Joseph Seering, Saiph Savage, Michael Eagle, Joshua Churchin, Rachel Moeller, Jeffrey P Bigham, and Jessica Hammer. 2017. Audience participation games: Blurring the line between player and spectator. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 429–440.

[36] Samantha Stahlke, James Robb, and Pejman Mirza-Babaei. 2018. The Fall of the Fourth Wall: Designing and Evaluating Interactive Spectator Experiences. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)* 10, 1 (2018), 42–62.

[37] Alina Striner, Andrew M Webb, Jessica Hammer, and Amy Cook. 2021. Mapping Design Spaces for Audience Participation in Game Live Streaming. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21, Article 329)*. Association for Computing Machinery, New York, NY, USA, 1–15.

[38] Bethesta Game Studios. 2011. *The Elder Scrolls V: Skyrim.* Game [Multiple Platforms]. Bethesda. Last played September 2023.

[39] Valve. 2011. *Portal 2.* Game [Multiple Platforms].

[40] Günter Wallner and Michael Lankes. 2023. The Ludic Potentials of Player Data. In *Companion Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (Stratford<, ON,Canada) *(CHI PLAY Companion '23)*. Association for Computing Machinery, New York, NY, USA, 209–211.

[41] Günter Wallner, Letian Wang, and Claire Dormann. 2023. Visualizing the Spatio-Temporal Evolution of Gameplay using Storyline Visualization: A Study with League of Legends. *Proc. ACM Hum.-Comput. Interact.* 7, CHI PLAY (Oct. 2023), 1002–1024.

[42] Dennis Wang, Yi-Chieh Lee, and Wai-Tat Fu. 2019. "I Love the Feeling of Being on Stage, but I Become Greedy": Exploring the Impact of Monetary Incentives on Live Streamers' Social Interactions and Streaming Content. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW (Nov. 2019), 1–24.

[43] Piaohong Wang and Zhicong Lu. 2023. Let's Play Together through Channels: Understanding the Practices and Experience of Danmaku Participation Game Players in China. *Proc. ACM Hum.-Comput. Interact.* 7, CHI PLAY (Oct. 2023), 1025–1043.

[44] Donghee Yvette Wohn and Guo Freeman. 2020. Audience Management Practices of Live Streamers on Twitch. In *ACM International Conference on Interactive Media Experiences* (Cornella, Barcelona, Spain) *(IMX '20)*. Association for Computing Machinery, New York, NY, USA, 106–116. https://doi.org/10.1145/3391614.3393653

[45] Saelyne Yang, Changyoon Lee, Hijung Valentina Shin, and Juho Kim. 2020. Snapstream: Snapshot-based Interaction in Live Streaming for Visual Art. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI,USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12.

[46] Saelyne Yang, Jisu Yim, Juho Kim, and Hijung Valentina Shin. 2022. CatchLive: Real-time Summarization of Live Streams with Stream Content and Interaction Data. In *Proceedings of the 2022 CHI Conference on Human Factors in*

Computing Systems (New Orleans, LA, USA) *(CHI '22, Article 500)*. Association for Computing Machinery, New York, NY, USA, 1–20.

[47] Ryan Yen, Li Feng, Brinda Mehra, Ching Christie Pang, Siying Hu, and Zhicong Lu. 2023. StoryChat: Designing a Narrative-Based Viewer Participation Tool for Live Streaming Chatrooms. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23, Article 795)*. Association for Computing Machinery, New York, NY, USA, 1–18.

Received February 2024; revised June 2024; accepted July 2024