Design Challenges for Livestreamed Audience Participation Games

Seth Glickman
Carnegie Mellon University
Pittsburgh, PA, USA
sglickma@andrew.cmu.edu

Nathan McKenzie
Carnegie Mellon University
Pittsburgh, PA, USA
nmckenzi@andrew.cmu.edu

Joseph Seering
Carnegie Mellon University
Pittsburgh, PA, USA
jseering@andrew.cmu.edu

Rachel Moeller
New York University
New York, NY, USA
rachelpmoeller@gmail.com

Jessica Hammer
Carnegie Mellon University
Pittsburgh, PA, USA
hammerj@andrew.cmu.edu

ABSTRACT
Livestreamed APGs (audience participation games) allow stream viewers to participate meaningfully in a streamer's gameplay. However, streaming interfaces are not designed to meet the needs of audience participants. In order to explore the game design space of APGs, we provided three game development teams with an audience participation interface development toolkit. Teams iteratively developed and tested APGs over the course of ten months, and then reflected on common design challenges across the three games. Six challenges were identified: latency, screen sharing, attention management, player agency, audience-streamer relationships, and shifting schedules. The impact of these challenges on players were then explored through external playtests. We conclude with implications for the future of APG design.

CCS Concepts
•Human-centered computing → Interaction paradigms; User interface toolkits; Interface design prototyping; •Applied computing → Computer games;

Author Keywords
Game design; livestreaming; online communities

INTRODUCTION
Livestreaming platforms such as Twitch.tv allow audiences to watch streamers playing games in real-time, while using text chat among themselves to form community [12, 24, 34]. Streamers and audiences also interact with one another socially, such as when streamers acknowledge audience members who have donated money or subscribed to the stream [5]. However, audience members typically cannot directly participate in the streamer's gameplay with streaming platforms offering only limited supporting features [2]. Livestreamed audience participation games (APGs) challenge this assumption. APGs allow viewers to directly participate in gameplay along with the streamer, typically in a secondary but influential role [54]. For example, in *Choice Chamber*, the streamer controls the main character in a simple platforming game, but audience participants can decide what enemies they face and what abilities the character possesses [3].

Streamers are beginning to embrace new means of audience engagement [24], since engaged audiences are seen as a key step to growing and maintaining a successful channel [4, 22, 50]. Developers see streamers and stream viewers as a critical game audience, and are experimenting with new livestreaming mechanics in their games, including APGs (e.g. [3, 32, 44, 46, 52]). Finally, audience participants have different motivations from either streamers or passive viewers, and seek varying levels of interaction and control [54]. APGs can address the needs of all three of these stakeholder groups.

The technical and social context of streaming platforms shapes the design opportunities available to APGs. For example, latency between the stream and the chat [1, 66] can interfere with the provision of feedback to audience players. Based on a prior study of the design space [54], we developed a toolkit for creating lightweight custom audience participation interfaces that could communicate with Unity games. The toolkit addressed some of the existing challenges with APG development, such as difficulty providing hidden information to players, by using a secret IRC channel to send game-relevant messages and creating a persistent HTML5 visual interface.

To discover additional challenges and opportunities in this space, we provided our toolkit to three game development teams and asked them to create new APGs across a range of game genres. Over the course of ten months, these games were developed using an iterative design process [8, 13, 18]. Design teams were asked to reflect on their process and identify common design challenges across the teams. We then conducted external playtests, with a research confederate playing the role of the streamer(s), and collected feedback from audience participants. In this paper, we share the six challenges identified

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
CHI PLAY '18, October 28-31, 2018, Melbourne, VIC, Australia
©2018 Association for Computing Machinery.
ACM 978-1-4503-5624-4/18/10…$15.00
DOI: http://dx.doi.org/10.1145/3242671.3242708

by design teams: latency, screen sharing, attention management, agency, audience-streamer relationships, and shifting schedules. We then illustrate how the APGs developed by participating design teams address those challenges, and explore how playtesters reacted in practice. We conclude with broader thoughts about design strategies for APGs.

RELATED WORK

Twitch.tv, YouTube Gaming, and other game streaming platforms allow players of video games to stream their gameplay in real time to live audiences [24, 35]. This democratization of livestream broadcasting has converted millions of gamers to broadcasters, and created crowdsourced game event content that audiences can virtually attend [12, 21, 24, 40, 45, 55]. While most streaming platforms are functionally similar, Twitch.tv is both the largest and has the highest proportion of game content [45, 55]. In the current state of game streaming, viewers are heavily skewed toward a small number of games [12] and toward the most popular streamers [25, 39]. Viewers also have high persistence in the stream once acquired [67]. Taking these factors together, game streaming audiences vary wildly in size, and can grow very large at the top end.

Socially engaged viewers seek to connect with the streamer and other audience members, and to acquire social status within the audience community [15, 42]. Viewers also value the quality of liveness. Even when video is archived and available at a later date, viewers rarely return to earlier videos [31]. Part of the pleasure of viewing game streaming is being present when game events occur or when the streamer has strong reactions [14, 24, 49].

Audience Participation Games

We define audience participation games (APGs) as games that empower audience members, understood as mutually-aware viewers, to affect gameplay [54]. Livestreamed APGs are APGs that take place using the medium of a game streaming platform. Formative efforts in the digital space include *Beach Ball Cursor Game*, which used the shadow of a beach ball maneuvered by the audience as a live element in the game, while an adapted version of the racing game *Pole Position* let the audience control the movement of a car based on which way they leaned their bodies [37]. At a larger scale, the popular television show *Who Wants to Be a Millionaire?* allowed players to poll the studio audience for help once per game. Audience members could vote for one of four answers to a trivia question using a keypad attached to their seat [11].

In the livestreaming context, some existing games have been adapted to the streamer/audience format. For example, streaming-adapted *Quiplash* allows up to eight players to answer a series of humorous questions, while audience participants vote on which answer is the best [19]. *Twitch Plays Pokémon* parsed audience commands in a Twitch chat stream to control games from the Nintendo series [33, 47]. Other games have been specifically created for livestreamed audience participation. For example, *Choice Chamber* and *Legend of Dungeon: Masters* allow audience participants to influence mechanics and challenges [3, 32], while *I'll Hide You* lets audience members influence the behavior of players filming in city streets [48]. Major streaming services are beginning to support and develop APGs such as *Superfight* and *Breakaway* [44, 46, 52]. However, relatively few APGs exist, and many design spaces for APGs remain to be explored [1, 54].

Prior work on APGs [54] proposed a set of three design challenges which emerged from a development and playtesting process: facilitating meaningful audience impact on gameplay; creating performable gameplay; and conveying asymmetric information. We explore additional design opportunities and challenges through development and testing of novel APGs.

Participation Interfaces in Game Streams

Streamers use third-party broadcasting tools to distribute the audio/video feeds of their games on these streaming platforms. To facilitate a performative environment [43], they commonly superimpose a camera view of themselves playing and talking to the audience [55]. Streamers also build community using external tools to communicate [1, 28, 43], moderate [36, 56], provide viewer feedback [30], and personalize the experience of their audience [38].

Viewers participate in streams by visiting a webpage on their computer or mobile device, which provides an audio-visual feed of the streamer. A streamer's page also includes a text chat interface through which they can communicate both with the streamer and other online members of the audience in the channel [24, 34]. Regardless of audience size, all viewers are provided with a single chat channel, which creates a single audience [17]. Chat communications fall into a variety of categories from game commentary to socializing to trolling and other maladaptive behavior [53].

When audiences can affect gameplay directly, the most common pattern is for audience participants to type keywords in the game chat, which are then aggregated by voting [1, 62]. For example, *Twitch Plays Pokémon* uses audience chat to crowdsource gameplay decisions [33], while *Choice Chamber* gives audiences control over the enemies, obstacles, and power-ups of a platforming game [3]. Some external tools support other forms of audience participation [34]. *StrawPoll* helps streamers tally viewer keywords [27]; *Helpstone* provides contextual information about the game *Hearthstone* for stream viewers, and lets audience participants suggest hints for the streamer [35]; *Streamote* allows audience members to form groups, vote outside the main chat stream, and place bets with virtual currency [29]. Most recently, Twitch has developed Extensions, interactive overlays that a streamer can add to their video stream [59].

METHODS

We used two complementary methods to explore the design challenges associated with APGs. First, we explored challenges in the *process* of design, by working with game design and development teams. Second, we used external playtesting to better understand how players experienced those challenges.

Game Design and Development

We recruited three game design teams to develop APGs. All three teams were explicitly informed that they would be designing livestreamed APGs before joining the project, and agreed that they would be willing to create games for this context. The teams ranged in size from a solo developer to a team of seven. All teams' backgrounds included at least one individual with game design training, either in the classroom or as an industry professional. Design team members ranged from undergraduates to experienced industry veterans (15 years in industry). Recruitment was conducted on the university campus.

Each team iteratively designed and developed a game over the course of ten months, using playtesting and other feedback processes to guide design [8, 18]. Design teams were required to create livestreamed APGs, but chose their own themes, mechanics, and game genres. During this period, teams periodically archived working builds of their game. They also produced supporting materials such as design diaries, sketches, paper prototypes, and presentations. All game artifacts were collected for analysis.

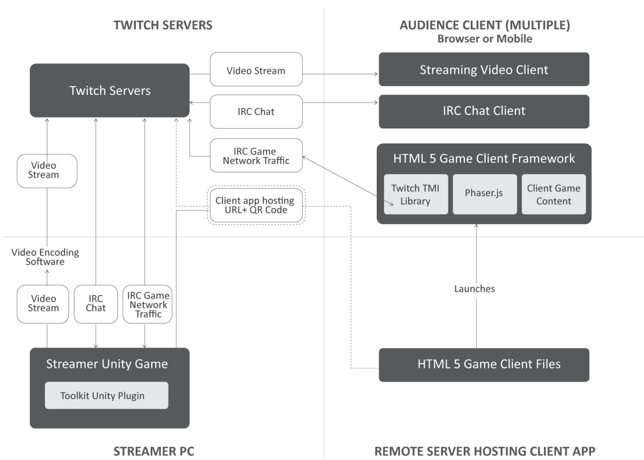

Figure 1. Toolkit technology overview

During the game design process, representatives from all teams attended regular design meetings. Meetings were used to discuss research papers related to APGs, play exemplar games, and provide peer feedback on design problems faced by each team. Design meetings were also used for internal playtesting. Meeting notes were kept for these meetings, and were used as reflective artifacts for teams (see below). Finally, the teams were able to communicate in a shared Slack channel.

All games were developed in Unity and deployed on Twitch for audience participation. Design teams were provided with an audience participation interface design toolkit, which allowed them to develop and deploy HTML5 interfaces for audience participants that could affect the streamer's game. The toolkit contained three components to handle the developers' pipeline. A Unity plugin provided the networking infrastructure and initiated the client launch. An HTML5 framework handled client login and interfacing with Twitch, client game life cycle issues, and client networking. Finally, an IRC-based protocol enabled game control over Twitch chat servers.

Designers also used the Twitch API to embed Twitch streams and chat channels in their audience participation interfaces.

External Playtesting

We conducted four external playtest sessions; all three games developed in this study were played in each session in a random order, for a total of twelve game experiences. During each playtest session, research team members took the role of the streamer(s), and playtesters were audience participants. Group sizes ranged from two to five playtesters per session. Participants had a wide range of experience with the Twitch platform, ranging from nearly none to extensive expertise.

To simulate the livestream experience while still allowing for direct observation, playtests included both users participating in person in a lab environment, and users logging in remotely. Research team members observed playtester actions and communication during the sessions, and recorded audio and video. Chat and logic channels on Twitch for the various games were logged and time-coded to match the lab media recordings. After each game, a researcher facilitated a focus group discussion about the user experience of that game. Questions included open-ended inquiries about their experiences, as well as directed probes using screenshots of the game and/or questions about specific actions they had taken. For example, participants were asked to explain the function of specific elements in both the viewer and the streamer interface. After all three games, users filled out a survey including demographic information and comparisons between all three games.

To understand differences between users, one researcher created profiles for each playtester by combining survey data and game behavior metrics. These profiles were shared with the rest of the research team. The qualitative data was analyzed using thematic analysis [6]. All data was reviewed collaboratively by the research team and discussed until consensus about themes and interpretations was reached.

GAMES

Gods of Socks and Spoons

Gods of Socks and Spoons (GoSaS) is a two-streamer action-arcade game. Gameplay takes place in two alternating 45-second phases. In the streamer phase, the streamer's character can fly around the screen and blow gusts of air. They use these abilities to pop item-filled balloons and to protect their audience players from whimsical monsters that hurl socks, spoons, and other odds and ends at them. Streamers are invulnerable to the monster attacks, but can ultimately be harmed by the other team's audience members. When a streamer takes enough damage from audience participants, they are defeated and the opposing team wins.

Audience participants join the streamers' teams, five to each side. They appear on the ground below their streamer, and may choose one of six buildings to stand in front of. During the streamer phase, participants use an audience participation interface to make selections about their individual actions. If they and an audience participant on the other team have chosen the same building, they can choose to attack that audience player. They can also collect resources from their current building, use resources previously collected to attack the opposing streamer, and select their location for the next round. Following each phase of streamer play, an audience phase

Figure 2. ***GoSaS*** **(Streamer interface) ©Nathan McKenzie**

Figure 3. ***GoSaS*** **(Audience interface) ©Nathan McKenzie**

plays out in a turn-based battle mode, similar to the *Final Fantasy* series. All participants can see the outcome of the actions they selected during the streamer round including damage done, resources collected, and their own health status.

Audience members are able to join at launch or in mid-stream. Open participant character slots are automatically filled by simple AI-driven non-player characters (NPCs) when an audience member is absent, and are replaced when a new player joins the game. Play strategy includes coordinating with teammates to collect the resources that will allow for special attacks, while predicting and avoiding the tactics used by opposing audience team members to disrupt the resource collection process.

Pitcrew

Pitcrew is an audience participation racing game. To win, a streamer must drive seven laps around an obstacle-filled track without running out of gas. In single-player mode, streamers compete against their previous best times. In multi-player mode, they compete against other live streamers, who have their own audience participant pit crews. Obstacles and hazards obstructing the streamer's path along with variations in car handling increase the streamer's game difficulty level.

To replenish the car's fuel, the streamer must occasionally visit one of the pit stops located at various points on the track. Each pit stop is controlled by a team of audience participants. In addition to replenishing the streamer's fuel, the audience participants at that pit stop can replace parts on the streamer's car, adjust the car's turning radius, driving stability, track visibility and more, just as pit crews do for real-world racers. Streamers communicate with their pit crews to request specific car builds for their upcoming laps (e.g., optimized to overcome upcoming obstacles, for time or handling, etc.).

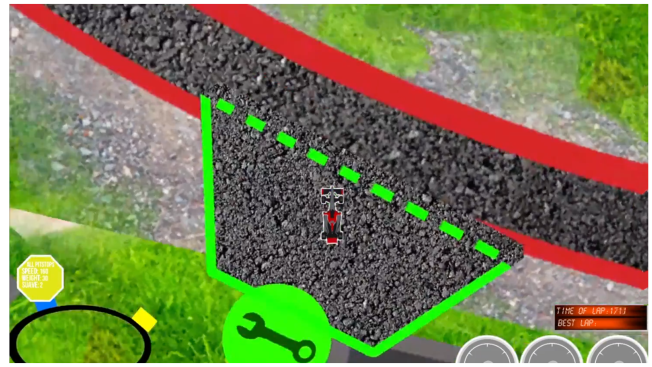

Figure 4. ***Pitcrew*** **(Streamer interface) ©Rachel Moeller**

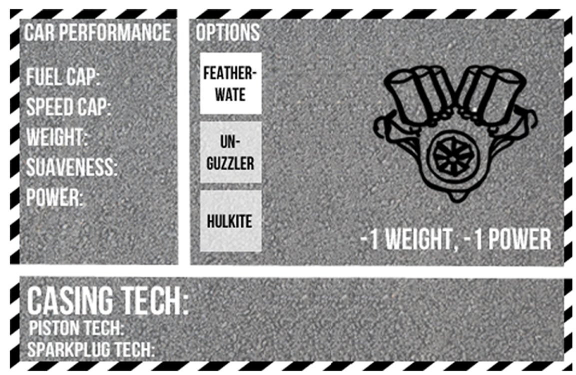

Figure 5. ***Pitcrew*** **(Audience interface) ©Rachel Moeller**

Each audience participant controls one aspect of the pit crew, such as replacing tires or tuning up the engine. Although the total effect on the streamer's car is based on the aggregate of the pit crew's choices, each pit crew member has unique items available to them, that only they can use. Crews must therefore negotiate who will deploy which items, and what their collective goal will be. For example, they choose whether to improve the streamer's experience by fulfilling the streamer's requests, or to make it more difficult for them to succeed.

What Lurks in the Dark

What Lurks in the Dark (What Lurks) is a camera-based horror game for a single streamer and any number of audience participants. The streamer explores a haunted house, trying to accomplish all the tasks necessary to lift its curse. The house's ghost pursues them, and can kill them if it catches them. However, the streamer cannot see the ghost.

Only the audience participants can see the ghost using a special audience interface, in which they can select among multiple cameras throughout the twelve rooms of the house. Audience members can communicate information about the ghost to the player, discuss with each other via chat, and vote to activate jump scares for the streamer in different areas of the house.

They also have access to a live video feed of the streamer's webcam and their gameplay exploring room-to-room.

Audience members may choose to warn streamers of ghost-sightings they see in the camera feeds of the audience interface. However, they can also root against the streamer and provide misleading information. A compelling feature of the audience experience is the ability to watch the streamer's reactions to unexpected moments. Latency in the live streamer video feed also becomes a game factor as the information viewed in the cameras of the audience interface may be out of date, leading some to provide the streamer with inaccurate information.

Figure 6. ***What Lurks in the Dark*** **(Streamer interface) ©Jessica Hammer**

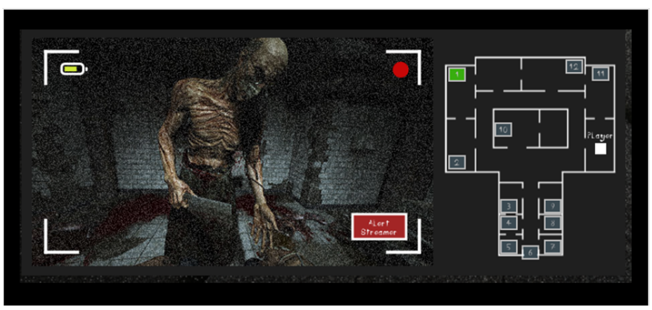

Figure 7. ***What Lurks in the Dark*** **(Audience interface) ©Jessica Hammer**

To win the game, the streamer must complete all exorcism-related tasks before the ghost catches them. If the ghost catches them, they lose.

Design Challenges

In this section, we present six key challenges that were identified by both novice and expert designers, and that came up across all three games despite their genre differences. As noted above, design teams reflected on their work in a range of ways during the iterative game design process. Teams kept design diaries, held internal playtest sessions, and brought artifacts to research meetings for discussion. Using this material, as well as notes from group meetings, each team was asked to reflect on their design process and describe key challenges they faced when working on APGs. Expert designers within the teams had better language to articulate the problems they identified and a wider range of exemplars from which to draw inspiration; however, both expert and novice designers described the six challenges we present here as issues for APG design. Similarly, the design challenges manifested differently across the three games, in part because of genre differences, but all three teams described addressing each of these challenges through design decisions. A sample of those decisions are presented here.

Latency

Designers were concerned with *latency*, or the lag between when the streamer took an action and when it was shown to viewers on the stream. On Twitch, stream latency has been measured at 12 seconds for browsers and 21 seconds for the mobile client, while chat channels (and our chat-based audience participation interfaces) have minimal latency [66]. While this degree of latency is acceptable to stream viewers, it could create problems when audiences become players [23, 26]. Designers framed this challenge as one of feedforward and feedback to audience participants. For example, how could audience participants choose meaningful actions when their information about game state was out-of-date? Would audience participants understand how their actions were affecting the game, if they might not see the impact until more than 20 seconds later? Design teams took advantage of the audience participation interface's reduced latency, compared to video, to give audience participants a better sense of their interactions with the streamer's game. However, they felt that this did not entirely solve their latency problems.

For example, *GoSaS* addresses latency by asking audience participants to commit to actions that will play out in a future round of the game. Successful play means correctly reading the current situation and identifying the best future action to take under conditions of uncertainty. Because rounds are 90 seconds long, which is considerably longer than typical stream latency, audience participants' experience of the stream delay is subsumed in their experience of the round delay. The round delay is in turn core to gameplay, since successful prediction means successful play.

Shared screen

Designers described grappling with the game stream's role as a *shared screen*, which needed to provide information to streamers, to audience participants, and to non-participating viewers. Designers used the audience participation interface to provide some degree of personalized information to audience participants, and to separate information directed at them from information directed purely at spectators or at viewers. However, designers wanted to maintain audience participants' connection to the game stream, which was understood as a key space for communicating and connecting with audience participants, as well as for allowing them to shift back and forth between participation and spectatorship over the course of a game. Stream real estate that was *specifically* designated for audience participants was therefore treated as a valuable and limited resource by all design teams, particularly when thinking about how games might scale for arbitrarily sized audiences.

For example, *What Lurks* reduces the pressure on shared screen space by focusing audience attention on the "ghost viewing" cameras scattered throughout the haunted house, which are only visible in the audience participation interface.

The stream view follows the streamer's character as they explore the haunted house, but on the stream, the ghost is invisible. Only when watching the cameras in the audience participation interface can the ghost be spotted. The most efficient strategy for audience players is to coordinate so that they are distributed across the many different views of the haunted house, including the stream. The audience players receive feedback in the stream only when the streamer is attacked by the ghost, an experience which does not need to scale based on how many audience participants were present.

Managing attention

Designers were keenly aware of the competing demands on stakeholders' *attention*. For example, game streamers are typically expected to split their attention between their in-game performance, the verbal commentary they provide, and engaging with audience chat [5]. Designers thought that audience participants would have to split their time between watching the stream, chatting with other audience members, strategizing about gameplay, and interacting with the audience participation interface. Additionally, they believed that streamers would not want audience participants to be too distracted from the streamer's performance and the game stream. In response, they described aiming for audience participation interfaces that were functional, but that they believed would not be too complex or distracting.

For example, *What Lurks* asks players to *watch* the cameras in the audience interface, so that they can find the ghost. However, the game minimizes the cognitive burden of this task. Camera views are low-resolution and grainy; the player cannot interact with the camera, even to pan or zoom it. Collectively, these decisions suggest that players can easily *listen* to the streamer, even if their eyes are elsewhere. In fact, a smart streamer will use narration to their advantage, directing viewers to observe different parts of the house and keeping the audience aware of their current location and trajectory. The game includes an audio cue when the ghost is near the player, in part so that audience participants can return their visual attention to the main screen in time to see a ghost-streamer encounter, which could result in a loss for the streamer or a harrowing escape. Either result is something that audience participants would not want to miss, and audio cues mean they do not have to.

Player agency

Designers were aware of research findings showing that many (but not all) audience participants value the ability to affect the outcome of the game [54]. They identified player *agency* as a key design concern. Latency and shared screen space created pragmatic issues around communicating feedback to players, but their central concern was rather how to give agency to multiple audience participants without compromising the experience for any of them. Existing solutions such as voting were criticized for aggregating player inputs, reducing the sense of agency for each individual audience participant. Designers explored using the audience participation interface to differentiate audience player experiences, for example by giving audience participants different abilities or information, so that individual audience participants would feel a greater sense of contribution.

For example, *Pitcrew* creates a hierarchy of participation using a "fractal" design. Each audience participant joins a pit crew, a team of three audience players. Each pit crew member can independently and individually decide what piece to add to the streamer's car; however, if they collaborate with their teammates, they can amplify the effect that their changes have by making selections that affect the car in similar ways. Similarly, the effect a given crew has on the streamer lasts only until the streamer can change out the relevant parts on their car at another pit stop. However, if players coordinate across pit stops, the audience's influence is longer lasting. Individual players can feel empowered at the level of their team, but with effective group coordination they can also see the influence of those decisions echo across the entire game system, even if they are not personally responsible for doing so.

Relationship design

Designers were highly conscious of the *relationship* between the streamer and the audience participant. In particular, the typical streamer-viewer relationship was understood as hierarchical. Viewers competed for the limited attention of the streamer, which served as validation of their social status and made them feel connected to the streamer. Meanwhile, streamers benefited from viewers' attention and presence in their channel, including financially through Twitch's partnership programs. While the literature certainly supports this model [4, 22, 50], designers saw APGs as an opportunity to playfully reconfigure this relationship and give viewers more power in the stream.

For example, in *GoSaS*, each streamer takes on the roles of protector and resource dispenser to their team of audience players. The audience players are weak and vulnerable for most of the time they are playing; however, if they collectively accumulate enough of the right resources, they can damage or destroy the opposing streamer. In contrast, the streamers are impervious to harm most of the time. They can direct resources to their team, interfere with resources being directed to the other team, or harm the opposing player's team directly. However, they cannot harm the other streamer directly, nor can they be harmed except under exceptional circumstances. The streamer who is best able to protect their team, and let their team do the work of destruction, is the one who wins.

Shifting schedules

Designers were highly conscious that streamers and audience participants might be playing on different *schedules*; for example, audience participants might drop in and drop out during a single streaming game session. At-will participation was treated as a key difference between streamers, who are often committed to a regular streaming schedule, and audience participants, who value their flexibility, and the focus was on supporting the latter. Some game elements that might otherwise have been shared in the stream, such as game tutorials, were moved to the audience participation interface so that audience participants could engage with them without depending on the streamer. Additionally, designers considered what might happen if audience participants left the stream mid-game; would the remaining participants and the streamer still be able to play?

For example, in *Pitcrew*, audience participants can join or leave a pit stop at any time. Since part selection is a one-click activity and can be changed anytime before the streamer's car arrives, a new audience player can get started in just a few seconds. If an audience participant leaves during the game, their choices are held constant until the beginning of the next game. For the remainder of the game, their teammates must work around the selected part as a fixed choice; it becomes a gameplay obstacle for them to accomplish their goals. Similarly, if all audience players leave the game, pit stops will randomly generate car builds. Reacting to random builds is a different type of challenge, and a different type of fun, for both the streamer and the community.

INSIGHTS FROM PLAYTESTING

These six design challenges (latency, shared screens, managing attention, player agency, relationship design, and shifting schedules) emerged from the game design process, including internal playtesting. During external playtests, we explored how these challenges were experienced by players who were not familiar with APGs. In this section, we report key insights that emerged across all three games.

Co-presence as design material

Prior research suggests that *liveness* is a key draw for viewers of streaming games [14, 24, 49]. In our games, we observed that the quality of liveness could be converted into game design opportunities by treating mutual awareness and co-presence as design materials.

In the Twitch context, viewers have a strong sense of the streamer's presence, but the streamer may not be equally aware of audience members. Mutual awareness between audience members becomes difficult in larger streams, especially as viewers may join and leave the stream at any time. Across all three games, playtesters were aware of the streamer's presence, but wanted to be more aware of the streamer's action, location, and choices. They also wanted to use audience participation mechanics to control their awareness of the streamer's activities, such as by "zooming in" to follow them more closely. Distributed multi-player games already use a range of techniques to make players mutually aware [64], including non-verbal communication mechanics such as mini-maps that show the location of other teammates or the ability to annotate the environment with messages for other participants [58]. In this situation, we interpret players' reactions not as a desire for *more* awareness of the streamer, but rather as wanting their awareness of the streamers' activity to shift as *they* shift between observing, planning, and participatory roles.

Although players described wanting to manipulate their level of awareness of the streamer, shifting awareness systems do not have to be player-controlled. For example, *Pitcrew* uses both camera controls and audience participation mechanics to affect player awareness of streamer actions. As the streamer circles the racetrack, the camera follows their car. Only the audience players who are near the streamer can be seen on screen; similarly, only the section of track immediately ahead of the streamer's car is visible for planning purposes. This camera choice helps audience participants be aware when their choices are likely to be relevant. At the same time, the camera's distance from the streamer's car can be manipulated by the audience players through their car builds. As the car's "suaveness" increases, the camera moves away from the streamer, showing more of the track around the streamer. When more track is visible, both audience participants and the streamer can plan more effectively. However, moving the camera also forces the streamer to adjust their driving style to address the changed scale of controls and turning ratios.

When it came to awareness of other participants, playtesters were torn between *agency* and *community*. Some playtesters thought that having more participants in the game would create a sense of shared purpose and activity, of "doing something together" (P10). However, others thought that more players would mean their voice would be lost in the crowd (P9), or that their contributions to the game would not have an impact (P4). This difference reflects, but does not completely map to, the difference between *individual* and *social* agency [54]. For example, some playtesters were worried about their voice being lost (P11) (reduction in social agency), while others were concerned about difficulty affecting game activities (P4) (reduction in individual agency). Interestingly, playtesters were also sensitive to the presence of NPCs (computer-controlled non-player characters) (P3,P4,P5,P8), whose presence might also dilute their social or agentic contribution.

We observed that playtesters were sensitive to how different games coordinated and aggregated audience input. For example, *What Lurks* disaggregates audience contributions by distributing their attention over multiple ghost-sensitive camera views. When a player spots the ghost, they can individually raise the alarm. Players described their participation as affecting the streamer's in-game behavior (e.g. stopping them from entering a particular room). They also had a clear sense of how their participation formed a social relationship between themselves and the streamer. As one playtester described it, "we were kind of like the [streamer's] eyes" while the streamer "did the legwork" (P4), framing the streamer and audience participants as complementary parts of the same body. In *GoSaS*, however, where a fixed number of non-player and audience-controlled characters contribute to the game on each turn, players were torn between developing community and losing their unique ability to make a contribution. This was in part due to their expectations of "how much I should have been doing" (P1); game design decisions can help set these expectations appropriately, but streamers and community members can also help develop participation norms.

Finally, we noted that while playtesters interacted with the audience participation interfaces provided, they did not participate in table talk [65]. Table talk, or game-inspired conversation between players at the game table, creates ways for all players to participate, even when it is not their turn. Audiences on Twitch already respond to game events in chat [17], but typically in ad-hoc ways. Treating conversations between players as game elements is another approach that can help translate live co-presence into shared playful experiences [57].

Orchestration of play activities

In our design process, we identified audience-centric activities, such as watching the streamer, and participation-centric activities, such as interacting with game interface elements. As designers, we developed techniques to transfer attention between audience interactions and the streamer's activities. For example, in *GoSaS*, streamers harvest resources for audience players, audience players spend those resources on abilities, and those abilities then impact the streamers. Resource flow between audience participants and streamers shifts player attention between the different parts of the game ecosystem; the phase-based gameplay, in which streamers and audience participants take turns at center stage, orchestrate what might otherwise be individual decisions into a larger whole.

In playtesting, we observed that when players were frustrated or did not understand the game, they typically expressed it by shifting their attention. In an early playtest of *GoSaS*, an error in the audience participation interface code meant that the Twitch chat overlapped with the audience interface; both were difficult to use. During this playtest, playtesters were observed looking away from the screen an average of once per minute during the gameplay. This behavior was only occasionally observed during other playtests of the same game.

Additionally, we observed that simply *shifting* audience participant attention was insufficient. Rather, playtesters wanted to be able to *predict* where their attention should be. For example, in *GoSaS*, participants initially felt uncomfortable shifting their attention from the screen to the audience interface. While this may have partly been due to the fast-paced motion of the main stream, we did not see this behavior in *Pitcrew*, where there are also fast-moving game elements. Rather, players quickly understood that in *Pitcrew*, they could predict where their attention would be needed based on the streamer's car's position along the track. In *GoSaS*, it took players some time to understand that streamers and audience participants would alternate turns being at the focus of play. Until players were confident about how future attentional handoffs between audience and streamer would be handled, they were reluctant to devote too much attention to the audience participation interface and reported finding it confusing. However, when playtesters watched several rounds of gameplay before encountering the audience interface, they did not report frustration. We believe that they had developed a mental model of when it was safe to move their attention away from the stream, and were therefore able to pay attention to the audience interaction components of the game.

This insight allows us to expand existing ways of thinking about game design, such as the notion of the interest curve [51]. Interest curves represent the sequencing of game activities to stimulate and retain user engagement over time. However, our design explorations suggest that for APGs, a single interest curve may not be enough. To represent engagement with different types of game activities, such as watching the streamer and participating in gameplay, we need to consider the *orchestration* of *multiple* interest curves. For example, game designers need to understand moments when an audience participant is primarily engaged with the stream or with the audience participation interface, as well as designing hand-off strategies between the two. Additionally, designers may need to be cautious about creating moments of high engagement with both the stream and the audience interface; while expert players may find those moments stimulating, new players may simply avoid engaging with the audience participation elements of the game in this situation. Finally, the current understanding of interest curves may need to be expanded to incorporate notions of participant role and social obligation.

Communicating relationships

APGs explore *relationships* between the streamer and the audience. In our designs, audience participants contribute to relationship development through game mechanics, such as giving audience players in *GoSaS* the ability to damage the opposing streamer and win the game for their own streamer, and through game-related chat discussions, such as conversations within a *Pitcrew* team about how to affect the streamer's car. We can conceptualize these game elements as prompts that imply particular types of relationships, and that support audience participants in producing them.

During playtests, we observed that audience participants needed strong scripts to help them understand the relationship between the audience and the streamer. For example, playtesters understood their role in *What Lurks* as being the "intel guy sitting in the back helping out" (P4). While participants used a range of metaphors to describe this relationship, it clearly informed their understanding of the game. Playtesters were able to describe relevant goals, such as giving the streamer useful information, and predict relevant outcomes, such as the streamer avoiding being caught by the ghost. They also expressed a sense of connection to the streamer, such as being nervous for them.

The role of relationship scripts differed based on how much prior experience users had with Twitch. Novice Twitch users did not have an existing model of what a social relationship with a streamer might be like; when given a strong model, they used it to understand their role in the game and hypothesize about possible actions.

More experienced Twitch users had a strong mental model of what streams should be like, with active streamer-performers and passive viewers [63]. Although *What Lurks* cast the streamer in a vulnerable role, the *activities* of the game itself aligned with these expectations, and audience participants emphasized their role as watchers, observers, and scouts. When games did not align with expert Twitch users' expectations, however, audience participants struggled to understand their relationship to the streamer. When asked to describe *GoSaS*, players who were familiar with Twitch described the streamers as "leaders" and explained that they "protect us" (P5). Although true, this description omits significant aspects of gameplay; audience participants on one team can directly affect the other team's streamer, and in fact doing so is the only way to win the game.

Surprisingly, all participants understood that audience participants could collaborate on audience-centric tasks such as collecting resources, although this behavior is not a common pattern in streams. Relationship scripts that tell audience participants how to relate to *one another* may be a fruitful area for exploration in the future.

Finally, we note that interpersonal interactions in our playtests were neutral or positive, despite the common appearance of trolling behavior on Twitch. We suspect that the social script of "participating in a research study" reduced the chance of antisocial behavior. Understanding the social and relational cues that promote or reduce trolling is also an area of interest when designing for participation on Twitch.

DISCUSSION

In many cases, Twitch can be understood as a *digital stadium*, in which crowds coalesce to watch a live competition or performance while communicating as a mass. There is substantial evidence that Twitch successfully provides a stadium experience. For example, esports events dominate the most popular Twitch streams [12]; esports fans are similar in their motivations and pleasures to sports fans, except perhaps even more dedicated [7]; and massive chat streams operate like crowds rather than conversations [17]. Perhaps unsurprisingly, games that are popular in the digital stadium (e.g. *League of Legends* [20]) demand the audience's close attention, maintain a high level of challenge and excitement, and require the audience to have significant skills to 'read' the game.

Twitch streamers often express the desire for a different type of stream experience, one that affords conversation and relationship-building [24]. Hamilton et. al. describe this as the desire for a *third place*, a shared space in which people form and maintain communities [24, 41]. However, streams may challenge some of the assumptions of digital third places. For example, digital third places erase distinctions between visitors, fostering connections between people who might otherwise never connect [16]. Many Twitch streams, however, perpetuate such distinctions by giving different communicative affordances to viewers, subscribers, moderators, and, of course, to the streamer themselves [24].

While understanding Twitch as a third space can be useful, our design insights suggest an alternate model: the Twitch stream as *distributed digital hearth* [61]. Around the hearth, being together is important for its own sake; attention shifts dynamically between different participants; and relationships are developed over time through shared activities. Similarly, by gathering around the distributed hearth of a Twitch stream, players can connect socially, appreciate one another's performance, shift attention among different stakeholders, and quickly scaffold relationships. Gameplay is the *ostensible* purpose of the gathering, but in practice the game serves to facilitate connection among participants [9].

Executing this strategy in a livestreaming context, however, can be difficult. Players want both the experience of being part of something larger that the entire audience experiences together, while also wanting their individual actions to carry weight and meaning. We identify a central challenge as information-theoretic: how much of the total information in the game can audience participants see while watching, and how much new information can each participant contribute? As the number of players increases, and the frequency of meaningful input per player increases, the ability for players to make sense of the game as a system decreases. This model suggests design tradeoffs between number of players, meaningful input, and sense-making ability.

Conventional large scale multi-player games, such as MMOs, competitive FPS games, or MOBAs, expect extremely high frequency input from all of their players. To keep play comprehensible, they structure play in such a way that there is a constrained number of individual players interacting with each other at any moment, with clusters of interacting players acting like separate nodes in a loosely coupled network (particularly as player counts go up). In a sense, each individual player mostly has a unique, non-shared experience, because the design of the games intentionally keeps the vast majority of other player input and interactions out of sight, and out of scope of any individual player. While many players may be playing these games, they are not necessarily creating a group experience.

Conversely, experiences like board games and tabletop role-play experiences have small group sizes, *and* they ask for relatively low-frequency input compared to a typical MOBA or RTS. These choices leave players much more time to watch other players making choices and to build relationships through table talk [65]. Creating the shared narrative of the experience is a cognitively and socially demanding activity, one that takes time and attention and effort, and that in turn requires watching and communication between players. These activities are hard to do when a player is busy with demanding high-stakes real-time interactions. In a human-administered context such as board games, watching time is also needed so that players make sure that game rules are executed correctly (which in turn grants the activity fairness and legitimacy). Once a game goes digital, however, watching loses this purpose, as the game rules are programmatically administered. Similarly, while watching other live humans sequentially make their choices is a possible context for social tension, and excitement, and ribbing, and drama, and shared meaning making - to draw an analogy from film, functioning like reaction shots that create the social meaning of a game action - watching a slew of computer opponents work through their choices is typically tedious.

How, then, can we allow substantial player input, even at scale, and still retain the possibility of a participatory shared group experience?

One design strategy to address this challenge is what we might call the *Counterstrike* [10] strategy. In *Counterstrike*, the beginning of each match is chaotic; a large number of players are providing high-frequency input, and managing an extremely difficult sense-making process is the core of the game. However, as characters die and players are eliminated from the match, they enter spectator mode, seeing only from the views of remaining players. As the number of players dwindle, the number of views dwindle too, and more and more players become spectators. Likewise, as the number of players dwindle, so does the amount of input in the system, making the entire game state more and more comprehensible to the attention of a single spectator. As the amount of watching increases, and the amount of system input decreases, the sense of a shared group experience for the endgame ramps up, and when something exciting or novel happens, it is often the case that many (dead) players were watching it happen in real time, while chatting with each other about what's going on.

A second strategy would be to look to sports, not only as models for esports but also to create more participatory experiences. Real-time team sports often have multiple parties taking real time actions in parallel that, by default, produce experiences that might have too high of an input density to produce meaningful shared experiences. However, sports typically include breaks during play, where action is broken up for viewers to engage in meaning-making and for tension to build. Additionally, viewers are supported in chunking and other sense-making activities by post-game activities such as dissemination of play diagrams and coverage in sports publications. These activities can be considered part of the larger play cycle, even though they take place outside the game itself.

A third approach would be to use visualization and summarization techniques to help with player sense-making. For example, AI agents could identify moments where an audience participant's input would have the largest impact on the game, and direct players' attention to those. Rather than tying gameplay to a streamer's perspective, audience participants might get visualizations of game data in real-time, for example in an overlay on the game stream, that would help them understand the high-level patterns of the game.

In short, we see the promise of APGs as the ability to harness the possibilities for scaling up that come with having rules and play structure coordinated by programmatic rules, while fostering meaning-making and a shared group experience. While there are some existing models for doing this, we will also need to discover new strategies for sense-making, activity coordination, and social connection in games. We recognize that these designs push against the financial and social dynamics of Twitch as currently constructed, which reward large streams and stadium-style play. However, streamers want better ways to connect with audiences, reducing the impact of celebrity provides more opportunities to members of marginalized groups, and Twitch itself has recently moved to financially and socially support streamers at smaller scales [60]. We therefore see the digital distributed hearth as advantageous to a range of stakeholders.

CONCLUSION AND FUTURE WORK

In this paper, we report design and playtest insights from a ten-month game development process in the genre of livestreamed APGs. We identified six key design challenges for APG designers: latency, shared screens, managing attention, player agency, relationship design, and shifting schedules. However, these areas of challenge are also areas of opportunity. The limitations of a game development form often define its strengths. For example, Twine games, despite the constraints of working primarily in text, have created a flowering of innovation around text-based games. We consider that the constraints of APGs may also produce "freeing limitations," limitations which are in fact generative for new designs. We identify this design space as the *distributed digital hearth*, or streaming experiences in which being together becomes important for its own sake. We therefore look forward to developing additional games in this space, as well as facilitating game design with our audience participation interface toolkit.

In addition to the challenges around sense-making that we have discussed in this paper, a significant design space remains to be explored around how audience participation in gameplay relates to existing social structures on Twitch. How do audience participants get selected for play? Can anyone become a participant? If there are gatekeepers, who are they, and what power do they have? How long does a given audience participant remain active? What happens when they leave, are removed, or engage in trolling behavior? We expect that these issues will affect both the gameplay experience of the APG, and the larger social structures of the channel. To facilitate game designers exploring these interactions, we can create toolkit elements that are oriented toward supporting the larger social ecosystem of an APG. For example, we could provide a persistent database to allow streamers to track audience participation across game sessions, or a moderator interface that allows quick assignment of audience players to different roles.

Finally, our designs suggest opportunities to improve Twitch in ways that have implications not only for APGs, but that also provide new experiences in other types of Twitch streams. For example, our games surface screen sharing between streamers and audience participants as a design issue. However, if viewers could customize their stream experience, for example by highlighting their own character, shared-screen game designs would become more feasible. As of this writing, only the streamer can customize a given stream, and their choices affect all viewers. Per-viewer stream customization could facilitate joint participation in gameplay, offer developers new ways to shift attention between audience interfaces and streaming video, and potentially allow in-stream audience participant interface designs. Per-viewer stream customization also has implications for meeting user needs that go far beyond APGs. For example, novice viewers could add tutorials to their streaming view, viewers with accessibility needs might enlarge their view of the game, and viewers unfamiliar with the game's language could add on-screen translations for game text.

Taken together, we see APGs as valuable to explore in their own right, both for how they extend the pleasures of livestreamed play and for how they challenge some of livestreaming's tacit assumptions. We look forward to research- and design-based explorations of the social relationships developed through livestreamed APGs, as well as to the new technologies and tools inspired by this domain.

ACKNOWLEDGMENTS

Our thanks to Amazon for their generous support of this work. We are grateful to Safinah Ali and Cody Fraser for their contributions, and to the game designers who participated in the project, including Justin Fanzo, Rachel Gu, Jinchao Han, Ketul Majmudar, Tony Deng, Flora Cheng, and Chaoya Li. Whittney Chu, Shih-Yu Lin, Siheon Lee, Anurati Sodani, and Tian Luo assisted with playtesting. Finally, we would like to thank all our players and playtesters.

REFERENCES

1. Safinah Ali, Rachel Moeller, Judeth Oden Choi, and Jessica Hammer. 2016. Analytic Frameworks for Audience Participation Games and Tools. (2016).
2. Sky LaRell Anderson. 2017. Watching People Is Not a Game: Interactive Online Corporeality, Twitch. tv and Videogame Streams. *Game Studies* 17, 1 (2017).
3. Studio Bean. 2014. Choice Chamber. (2014). `http://choicechamber.com`
4. Hugh Beyer and Karen Holtzblatt. 1997. *Contextual design: defining customer-centered systems*. Elsevier.
5. Christopher Bingham. 2017. *An Ethnography of Twitch Streamers: Negotiating Professionalism in New Media Content Creation*. Ph.D. Dissertation.
6. Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
7. Kenon A Brown, Andrew C Billings, Breann Murphy, and Luis Puesan. 2017. Intersections of Fandom in the Age of Interactive Media: eSports Fandom as a Predictor of Traditional Sport Fandom. *Communication & Sport* (2017).
8. Judeth Oden Choi, Jodi Forlizzi, Michael Christel, Rachel Moeller, MacKenzie Bates, and Jessica Hammer. 2016. Playtesting with a Purpose. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. ACM, 254–265.
9. Mia Consalvo. 2017. When paratexts become texts: de-centering the game-as-text. *Critical Studies in Media Communication* 34, 2 (2017), 177–183.
10. Valve Corporation. 1999. Counter Strike. (1999). `http://blog.counter-strike.net`
11. Michael Davies. 2002. Who Wants to be a Millionaire? (2002).
12. Jie Deng, Felix Cuadrado, Gareth Tyson, and Steve Uhlig. 2015. Behind the game: Exploring the twitch streaming platform. In *Proceedings of the 2015 International Workshop on Network and Systems Support for Games*. IEEE Press, 8.
13. Alena Denisova, A Imran Nordin, and Paul Cairns. 2016. The convergence of player experience questionnaires. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. ACM, 33–37.
14. Judith S Donath. 2002. Identity and deception in the virtual community. In *Communities in cyberspace*. Routledge, 37–68.
15. Steven Drucker, Li-wei He, Michael Cohen, Curtis Wong, and Anoop Gupta. 2002. *Spectator games: A new entertainment modality of networked multiplayer games*. Technical Report.
16. Nicolas Ducheneaut, Robert J Moore, and Eric Nickell. 2007. Virtual "third places": A case study of sociability in massively multiplayer games. *Computer Supported Cooperative Work (CSCW)* 16, 1-2 (2007), 129–166.
17. Colin Ford, Dan Gardner, Leah Elaine Horgan, Calvin Liu, Bonnie Nardi, Jordan Rickman, and others. 2017. Chat Speed OP PogChamp: Practices of Coherence in Massive Twitch Chat. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 858–871.
18. Tracy Fullerton. 2014. *Game design workshop: a playcentric approach to creating innovative games*. CRC press.
19. Jackbox Games. 2015. Quiplash. (2015). `http://jackboxgames.com/project/quiplash/`
20. Riot Games. 2009. League of Legends. (2009). `https://na.leagueoflegends.com/en/`
21. John Gaudiosi. 2016. Twitch Ups Its Game to Compete with YouTube Gaming. *Fast Company* (2016). `http://fortune.com/2016/02/26/twitchupsitsgametocompetewithyoutubegaming`
22. Barney G Glaser and Anselm Strauss. 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, Chicago, IL, USA.
23. Tor-Morten Grønli. 2014. Latency Thresholds for Usability in Games: A Survey Kjetil Raaen. (2014).
24. 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 32nd annual ACM conference on Human factors in computing systems*. ACM, 1315–1324.
25. Erik Harpstead, Thomas Zimmermann, Nachiappan Nagapan, Jose J Guajardo, Ryan Cooper, Tyson Solberg, and Dan Greenawalt. 2015. What drives people: Creating engagement profiles of players from game log data. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*. ACM, 369–379.
26. Tristan Henderson and Saleem Bhatti. 2003. Networked games: a QoS-sensitive application for QoS-insensitive users?. In *Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?* ACM, 141–147.
27. Curse Inc. 2017. StrawPoll. (2017). `https://strawpoll.me`
28. Discord Inc. 2015. Discord. (2015). `https://discordapp.com`
29. Proletariat Inc. 2016a. Streamote. (2016). `https://streamote.tv`
30. StreamElements Inc. 2016b. StreamElements. (2016). `https://streamelements.com`
31. Adele Lu Jia, Siqi Shen, Dick HJ Epema, and Alexandru Iosup. 2016. When game becomes life: The creators and spectators of online game replays and live streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 12, 4 (2016), 47.

32. Robot Loves Kitty. 2015. Legend of Dungeon: Masters. [PC Computer]. (2015). http://robotloveskitty.com/LoD/

33. A Leavitt. 2014. TwitchPlayedPokemon: An Analysis of the Experimental Interactive Phenomenon. In *Game Developers Conference*.

34. Pascal Lessel, Michael Mauderer, Christian Wolff, and Antonio Krüger. 2017a. Let's Play My Way: Investigating Audience Influence in User-Generated Gaming Live-Streams. In *Proceedings of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video*. ACM, 51–63.

35. Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. 2017b. 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*. ACM, 1571–1576.

36. NightDev LLC. 2017. NightBot. (2017). https://beta.nightbot.tv

37. Dan Maynes-Aminzade, Randy Pausch, and Steve Seitz. 2002. Techniques for interactive audience participation. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*. IEEE Computer Society, 15.

38. Muxy. 2015. Muxy. (2015). https://muxy.io

39. Supun Nakandala, Giovanni Luca Ciampaglia, Norman Makoto Su, and Yong-Yeol Ahn. 2016. Gendered conversation in a social game-streaming platform. *arXiv preprint arXiv:1611.06459* (2016).

40. Kevin Ohannessian. 2014. How twitch hooked 45 million viewers to watch 13 billion minutes of gaming a month. *Fast Company* (2014). https://www.fastcompany.com/3027452/how-twitch-hooked-45-million-viewers-to-watch-13-billion-minutes-of-gaming-a-month .

41. Ray Oldenburg. 1999. *The great good place: Cafes, coffee shops, bookstores, bars, hair salons, and other hangouts at the heart of a community*. Da Capo Press.

42. Jedrzej Olejniczak. 2015. A linguistic study of language variety used on twitch. tv: desriptive and corpus-based approaches. *Redefining Community in Intercultural Context* (2015), 329–344.

43. Anthony J Pellicone and June Ahn. 2017. The Game of Performing Play: Understanding streaming as cultural production. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4863–4874.

44. Pipeworks. 2015. Superfight. (2015). https://www.superfightgame.com/

45. Karine Pires and Gwendal Simon. 2015. YouTube live and Twitch: a tour of user-generated live streaming systems. In *Proceedings of the 6th ACM Multimedia Systems Conference*. ACM, 225–230.

46. Luke Plunkett. 2016. Twitch Is Helping Build Games An Audience Can Play Too. *kotaku* (2016). https://kotaku.com/twitch-is-helping-build-games-an-audience-can-play-too-1764877430 .

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

48. Stuart Reeves, Christian Greiffenhagen, Martin Flintham, Steve Benford, Matt Adams, Ju Row Farr, and Nicholas Tandavantij. 2015. I'd Hide You: Performing live broadcasting in public. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2573–2582.

49. Raquel Robinson, Katherine Isbister, and Zachary Rubin. 2016. All the Feels: Introducing Biometric Data to Online Gameplay Streams. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*. ACM, 261–267.

50. Saiph Savage and Andrés Monroy-Hernández. 2015. Participatory Militias: An Analysis of an Armed Movement's Online Audience. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 724–733.

51. Jesse Schell. 2014. *The Art of Game Design: A book of lenses*. AK Peters/CRC Press.

52. Dennis Scimeca. 2016. Streamline could revolutionize the way we play games on Twitch. *The Daily Dot* (2016). http://www.kotaku.com.au/2016/03/twitchishelpingbuildgamesanaudiencecanplaytoo/

53. Joseph Seering, Robert Kraut, and Laura Dabbish. 2017a. Shaping pro and anti-social behavior on twitch through moderation and example-setting. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, 111–125.

54. Joseph Seering, Saiph Savage, Michael Eagle, Joshua Churchin, Rachel Moeller, Jeffrey P Bigham, and Jessica Hammer. 2017b. Audience Participation Games: Blurring the Line Between Player and Spectator. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. ACM, 429–440.

55. Thomas Smith, Marianna Obrist, and Peter Wright. 2013. Live-streaming changes the (video) game. In *Proceedings of the 11th european conference on Interactive TV and video*. ACM, 131–138.

56. Marcin Swierzowski. 2017. Ankhbot. (2017). https://www.ankhbot.com

57. Alexandra To, Ali Safinah, Geoff F Kaufman, and Jessica Hammer. 2016. Integrating Curiosity and Uncertainty in Game Design.. In *DiGRA/FDG*.

58. Zachary O Toups, Jessica Hammer, William A Hamilton, Ahmad Jarrah, William Graves, and Oliver Garretson. 2014. A framework for cooperative communication game mechanics from grounded theory. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*. ACM, 257–266.

59. Twitch. 2017a. Extensions: A Revolution in Live Streaming. (2017). https://www.twitch.tv/p/extensions/

60. Twitch. 2017b. Twitch Affiliate Program. (2017). https://affiliate.twitch.tv/

61. Amy Voida and Saul Greenberg. 2009. Wii all play: the console game as a computational meeting place. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1559–1568.

62. Mirjam Vosmeer, Gabriele Ferri, BAM Schouten, and Stefan Rank. 2016. Changing roles in gaming: twitch and new gaming audiences. In *DiGRA/FDG 2016-Abstract proceedings of the first international joint conference of DiGRA and FDG, Dundee, Scotland Google Scholar*.

63. Donghee Yvette Wohn, Guo Freeman, and Caitlin McLaughlin. 2018. Explaining Viewers' Emotional, Instrumental, and Financial Support Provision for Live Streamers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 474.

64. Jason Wuertz, Sultan A Alharthi, William A Hamilton, Scott Bateman, Carl Gutwin, Anthony Tang, Zachary Toups, and Jessica Hammer. 2018. A Design Framework for Awareness Cues in Distributed Multiplayer Games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 243.

65. Yan Xu, Evan Barba, Iulian Radu, Maribeth Gandy, and Blair MacIntyre. 2011. Chores Are Fun: Understanding Social Play in Board Games for Digital Tabletop Game Design. In *DiGRA Conference*.

66. Cong Zhang and Jiangchuan Liu. 2015. On crowdsourced interactive live streaming: a twitch. tv-based measurement study. In *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 55–60.

67. Cong Zhang, Jiangchuan Liu, and Haiyang Wang. 2016. Towards hybrid cloud-assisted crowdsourced live streaming: measurement and analysis. In *Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 1.