

Channeling End-User Creativity: Leveraging Live Streaming for Distributed Collaboration in Indie Game Development

LINGYUAN LI, Clemson University, USA

GUO FREEMAN, Clemson University, USA

NATHAN J. MCNEESE, Clemson University, USA

282

This paper explores the role of live streaming in distributed collaborative software development using indie game development, an end-user driven creative community, as an example. We conducted 27 in-depth interviews with indie game developers from various cultures and countries, who had engaged in live streaming for collaborative software development either as a streamer or a viewer. Our findings show how live streaming can be used by indie game developers to support their endeavors to innovate the traditional game development model, which goes beyond just learning and teaching technical skills. We also highlight the potential challenges indie developers face in this process. We thus make unique contributions to CSCW by bridging the previously often disconnected research agendas on collaborative software development and live streaming. We also provide potential directions for designing future live streaming platforms to better support distributed collaboration in emerging end-user driven creative activities.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing; Empirical studies in collaborative and social computing;**

Additional Key Words and Phrases: indie game development, distributed collaboration, computer-mediated collaboration, live streaming

ACM Reference Format:

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, Article 282 (November 2022), 28 pages. <https://doi.org/10.1145/3555173>

1 INTRODUCTION

As an emergent form of interactive media, live streaming has become a growing research agenda in CSCW and HCI [51, 53, 74, 112, 118]. Live streaming started as a niche, non-mainstream media platform for video game players to share their gameplay in real time and to communicate with other players. More recently, it has evolved into an interactive online space for learning, teaching, and mentoring various creative, artistic, and technological skills [19, 25, 26, 33, 34, 49, 114]. It has also been widely used by communities that focus on creativity and innovation to live broadcast their projects and technological processes, seek feedback and support, and interact with the audience in real time [4, 18, 25, 49].

One such community is the independent [indie] game development community. Indie game developers are those who do not affiliate with large game companies or publishers but make and

Authors' addresses: Lingyuan Li, lingyu2@g.clemson.edu, Clemson University, Clemson, South Carolina, USA; Guo Freeman, guof@clemson.edu, Clemson University, Clemson, South Carolina, USA; Nathan J. McNeese, mcneese@clemson.edu, Clemson University, Clemson, South Carolina, USA.

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.

© 2022 Association for Computing Machinery.

2573-0142/2022/11-ART282 \$15.00

<https://doi.org/10.1145/3555173>

publish games in alternative ways such as self-funding/publishing, small teams/studios, and free labor. They are playing an increasingly important role in shaping the global gaming community [32, 35, 88]. For them, live streaming seems to demonstrate the potential to support new and promising forms of distributed collaboration, especially in the context of global software development (GSD) – software development practices where geographically distributed collaborators are mutually engaged and thus interdependent in development activities in order to design, program, and implement a technical system or product [6, 11, 76].

However, despite growing research interests in collaborative software development (e.g., [6, 11, 76, 80]) and live streaming (e.g., [18, 25, 49, 61, 103]) in CSCW and HCI, these areas have been studied separately without research integration. For example, studies on live streaming tend to focus on game streaming [61, 103], esports commentating [62, 66] or social interaction [16, 51, 64, 112]. A small body of work has also highlighted the nuances of using live streaming for learning and teaching coding and programming [4, 18, 25, 49]. How live streaming can be further leveraged to support creative and innovative practices beyond teaching and learning skills is understudied. In contrast, studies on collaborative software development tend to focus on workplace settings (e.g., [6, 11, 76]) or open-source software (OSS) communities (e.g., [22, 78, 97]) with professional and/or expert developers. How emerging technologies may support new distributed collaborative dynamics in software development by developers of various levels and experiences requires more research.

This disconnect thus motivates this study to explore collaborative practices occurring during live streaming as an emerging example of collaborative software development. We believe that such an investigation is critical for the CSCW and HCI community. There are two reasons for that: one is that live streaming has grown beyond gaming and plays an increasingly important role in people's online lives. How it can be designed to support and facilitate the ways people share and create content and work together in various contexts, including creativity and innovation, needs to be examined. The other is that while distributed collaboration has been a long-standing research focus in CSCW, global events such as COVID-19 have placed even more focus on understanding and re-imagining remote experiences. For example, the viewership of Twitch.tv increased by 31% in March 2020 [104]. We may expect live streaming to become even more important for computer-mediated collaboration, such as collaborative software development, in the near future.

In particular, we chose indie game development, rather than other development communities that have been studied before (e.g., online programming mentoring as in [19, 25, 26] or OSS communities [22, 78, 80, 97]), as our research context to bridge the two research agendas mentioned above. We understand indie game development as a modern and novel exemplar of end-user driven creative activities – activities that everyday users can engage with technology in novel and more participative ways through which their labor, innovation, and power can be fostered, acknowledged, and distributed [40, 41, 113]. These activities, such as DIY making, hacking, crafting, open design/manufacturing [7, 50, 69, 70], are outcomes of the new bottom-up technological innovation model and the democratization of technology (i.e., making technology more broadly distributed and accessible to more people) [69, 70, 102]. Indie game development attracts geographically distributed amateurs, non-traditional technology users, and people who have little to no prior experience in game development [36, 37]. This differs from traditional workplace settings or OSS communities that mainly focus on tech savvy or expert developers. As the indie community is making tremendous contributions toward the booming creative industry, how developers of all levels work together to shape modern software/game development must be investigated to explicate new forms of distributed collaborative practices.

In addition, the indie community demonstrates unique labor setups, production methods, and expectations for collaboration and participation with a broader aspiration to innovate the traditional

game development model [32, 35, 77, 88]. Their collaborative practices thus go beyond just online programming mentoring, teaching, and learning [19, 25, 26, 49]. Studying this community, therefore, can inform our understanding of nuanced live streaming experiences in collaborative software development beyond the educational aspect. Such knowledge can also help us better understand how live streaming can be designed to support other similar end-user driven collaborative activities that focus on innovation and creativity (e.g., DIY making, hacking, crafting, and open design and manufacturing).

Drawing on 27 in-depth interviews with indie game developers from various cultures and countries, who had engaged in live streaming for collaborative software development either as a streamer or a viewer, we focus on the following research questions:

RQ1: How do indie game developers use live streaming to support their collaborative game development beyond learning and teaching programming?

RQ2: What challenges do indie game developers often encounter when collaborating through live streaming?

We make unique contributions to CSCW by bridging the previously often disconnected research agendas on collaborative software development and live streaming. We do so in three ways. Firstly, using indie game development as an example, we draw on collaboration theories to explain emerging practices of end-user driven collaborative software development through live streaming. We thus extend existing CSCW research on distributed collaboration by focusing on how collaborative software development can be conducted in new ways and by new populations beyond the traditional workplace settings or OSS communities that often focus on professional or expert developers. Secondly, we contribute towards prior literature on live streaming for coding, programming, and online mentoring (e.g., [19, 25, 26]) by shedding light on how live streaming can be used beyond teaching and learning skills, such as for community building and/or advocating aspirational agendas. Thirdly, we identify challenges and risks involved in using live streaming for distributed collaboration in software development. We also provide potential directions for designing future live streaming platforms to better support distributed collaboration in emerging end-user driven creative activities.

2 RELATED WORK

Our work is grounded in three interlinked strands of research in CSCW: distributed collaboration and software development; indie game development as an emerging form of collaborative software development; and live streaming for interacting, learning, and programming. We believe that our focus will uniquely bridge prior literature on collaborative software development and live streaming in a creative and innovative context.

2.1 Distributed Collaboration and Software Development

Collaboration is essential to gather and organize crowds and to optimize their power. D'Amour et al. summarize five basic elements of collaboration: sharing, membership, interdependency, power, and process [24]. *Sharing* includes a series of shared practices, such as shared responsibilities, shared decision-making, and shared values. *Membership* requires open communication, mutual trust, and respect. *Interdependence* means mutual dependence (e.g., see [27]). *Power* implies relationships and interactions between team members, and *process* points out that collaboration is not static or predetermined but changeable over time. Based on these elements, collaboration is considered as a powerful dynamic, a major social activity, and an important interpersonal cohesion.

As collaborative technologies continue to help connect people in different times and spaces [12, 43], distributed collaboration in various contexts has become a central research agenda in the CSCW community. While theories of distributed collaboration work also continue to evolve (Table 1),

Table 1. Three Main Theories on Collaboration. Adapted from D'Amour et al. [24], Biuk-Aghai et al. [10], and Olson et al. [12, 84, 85]

Theory	Elements	Description	Examples
D'Amour et al.	Sharing	A series of shared practices Shared decision-making Shared values	Shared responsibilities
	Membership	A collegial-like relationship Mutual trust and respect	Open communication
	Interdependency	Mutual dependency Maximized individual contributions	Synergy among members
	Power	The simultaneous empowerment of each participant whose respective power is recognized by all Knowledge and experience	A product of the relationship and interactions between team members
	Process	Collaboration is an evolving process A transforming process An interpersonal process A structuring of collective action	A dynamic and interactive process
Theory	Patterns	Description	Examples
Biuk-Aghai et al.	Artefact-Exchange	Two or more users exchanging artefacts	Uploading, replacing, and opening an artefact
	Artefact-Management	Two or more users managing the set of artefacts	Deleting, assigning, and de-assigning artefacts
	Group-Discussion	Two or more users exchanging discussion statements	Posting and opening discussion statements
	Message-Exchange	Two users exchanging directed messages	Sending messages
	Workspace-Setup	Two or more users configuring the setup of a workspace	Adding users to a workspace, creating a role, creating a discussion forum
Theory	Aspects	Description	Examples
Olson et al.	Common Ground	The knowledge that people share and that they know the other shares	Describing the same event or idea differently when talking to different people
	Coupling of Work	The extent and kind of communication required by the work	Coauthoring an article
	Collaboration Readiness	The motivation for coworkers to collaborate	Learning a technique if being paid overtime
	Collaboration Technology Readiness	The difficulties faced in adapting, adopting, and bringing collaboration technologies into use	Spending time to develop technological frames
	Organizational Management	The practices by which management activities are part of shaping the fundamental premises for collaboration without proximity	Having respective travel policies in each local organizations from the same large global organization

special attention has been paid to distributed collaboration in work environments where creativity and innovation is considered a hallmark of successful work (e.g., software development) [9, 63]. Therefore, a large body of work has explored collaborative practices in global software development (GSD) – software development practices where geographically distributed collaborators are mutually engaged and thus interdependent in development activities in order to design, program, and implement a technical system or product [6, 11, 76].

For example, focusing on organizational dynamics and the workplace, Biuk-Aghai and Simoff identify five patterns of virtual collaboration in online collaboration systems, namely, artifact-exchange, artifact-management, group-discussion, message-exchange, and workspace-setup [10]. Olson and Olson highlight five aspects of distributed work in GSD – common ground, collaboration readiness, collaboration technology readiness, coupling of work, and organizational management [84]. In addition, *awareness*, including both a general awareness of the entire team and more detailed knowledge of individual collaborators on the team, is generally considered necessary to overcome

time and distance challenges in distributed software development teams [46]. Yet, challenges to foster awareness have been identified, including the tendency to ignore unpopular ideas and lose track of novel ideas as well as the lack of critical evaluation of opinions and reflections [29]. Building *trust* is also found to be challenging in distributed software engineering teams [3, 105]. However, maintaining socially correct behavior, exhibiting technical competency, and demonstrating concern for others as well as informal, non-work related communication can promote trust [3, 105]. Matthiesen et al. further highlight several cultural blind spots in conducting distributed collaboration in GSD as well, including an increased number of interruptions, lack of translucence of remote colleagues' work, and the re-definition of boundaries between work and articulation work [76].

Another strand of important research focuses on collaborative software development beyond the organizational and workplace settings, such as the open source software (OSS) development community [22, 78, 80, 97]. OSS development often requires advanced technical skills to enter and extensive efforts to support [116]. Unlike traditional software development, OSS development can be viewed as a participative system where developers are bonded together through collaboration [81]. In particular, members of the OSS community are highly motivated to make their own contributions, work closely with other developers on various tasks, and use communication tools extensively [116]. Even if OSS development lacks many of the traditional coordination mechanisms for software development [80], its focus on critical peer review and idea sharing may make it do better than a proprietary organizational form [59].

In summary, existing CSCW research has demonstrated the continuous need and challenges for supporting distributed collaboration and varied socio-technical relationships in different contexts of software development practices (e.g., organizational, workplace, and open source) [6, 11, 58, 76, 91]. However, prior research seems to mainly focus on tech savvy or expert developers' distributed collaborative practices in software development. Examples include workplace settings like an international IT firm, or knowledge-based online communities like OSS development where professional developers can write and share code and collectively contribute to developing and upgrading sophisticated software (e.g., operating systems) [79, 97]. Advanced technical skills and extensive experience are often required to engage in these practices. In contrast, while new forms of end-user driven collaborative activities that focus on innovation and creativity (e.g., DIY making, hacking, crafting, open design and manufacturing, and indie game development) are actively restructuring today's tech industry [69, 70, 102], these activities often are conducted by end users with varying degrees of technical skills and through new and novel computer-mediated methods. How these new phenomena and populations may shape the landscape of distributed collaboration in software development will need more research. We thus introduce indie game development as a growing form of collaborative software development, which goes beyond a traditional workplace setting and is open to developers at all levels.

2.2 Indie Game Development as an Emerging Form of Collaborative Software Development

Indie games have been broadly defined as games that are consciously created within alternative production and distribution structures outside of the mainstream game companies [71]. Drawn upon this understanding and grounded in our engagement in the indie game development community since early 2017, in this paper we define indie game developers as *people who do not affiliate with large game companies or publishers but make and publish games in alternative ways such as self-funding/publishing, small teams/studios, and free labor*. Recent industry reports have highlighted the success of indie games in economic growth and technological innovation: indie gaming has become one of the fastest-growing and most influential segments of the video game industry, which

produces 80 billion dollars in annual revenue [8]; indie games have also dominated the mobile gaming market, accounting for 68% of all mobile game sessions [28].

As highlighted at the beginning of this paper, we chose indie game development as our research context because it is a modern and novel exemplar of end-user driven distributed software development. First, indie game development attracts geographically distributed amateurs and non-traditional technology users [32, 36, 37, 86, 88], which differs from OSS communities that often consist of expert developers and focus on sophisticated software products. Therefore, indie game development shows potential to establish an equitable and inclusive space for traditionally excluded groups in game development: financially independent developers self-fund their production; the creativity independence originates from autonomous decisions of the developer; and indie games can be self-published as a common practice in the shareware era [42]. In this sense, indie game development is highly collaborative and often depends on collaborators with geographical, time, and even cultural differences, which serves as a foundation to engage a broad community of developers of all levels [32, 88, 95, 108].

Second, compared to the mainstream gaming industry, the indie game development community demonstrates different labor setups, development life cycles, production models, and expectations for collaboration and participation in innovating game development. The indie community often considers itself anti-establishment and anti-authoritarian [71]. O'Donnell points out that indie games focus on a small number of clear design and aesthetic goals and can be developed at a much smaller scale in terms of workload and manpower [86]. Freeman et al. add that indie games are also developed by small teams of distributed developers who emphasize developer-player interaction [36]. "Being indie" thus seems to represent a new business and innovation model of game development, which has the potential to reconfigure and revitalize how modern digital games are developed, distributed, and perceived [77]. Therefore, at least a subset of the indie game development community self-identifies as a new form of "rebellious entertainment" [99] and a new counter-culture that advocates a more participative and democratic form of game development through collaborative efforts [32, 35, 88].

Over the past several years, live streaming has been effectively incorporated by the indie community in these efforts and aspirations. Game Development has become a popular content tag on Twitch, one of the biggest and most popular live streaming platforms in the world. The subcommittee of "Science & Technology" in the Twitch Creative community hosts most game development live streams, which has over 447,000 followers [1]. Twitch Developers was also created as a main hub for game developers and designers on Twitch to share and deliver their development practices through live streaming [2]. In addition, Twitch has published a "playbook" to encourage indie developers to integrate Twitch into any part of their game development life cycle, including acquiring new players, creating new engaging experiences, retaining players by creating a watch/play loop, and monetizing [4]. Prior research has also shown that many indie developers use streaming platforms such as Twitch to socialize, promote, and share what they are working on [35].

In this sense, live streaming seems to play an increasingly important role in advocating the indie values and aspirations as a counter culture and supporting indie developers' various development practices, such as helping novice indie developers jumpstart their game design and development [36]. While similar practices have been reported in previous live streaming studies on learning and teaching game-related coding and programming [19, 25, 26, 49], in this paper, we explore how indie game developers may use live streaming beyond just as an online learning space to collaboratively pursue their aspiration for innovating the traditional game development model. In the next section, we review existing live streaming scholarship and how our study may extend this body of research by focusing on indie game developers' collaborative practices through live streaming.

2.3 Live Streaming for Interacting, Learning, and Programming

In recent years, popular live streaming platforms and services such as Twitch.tv, YouTube Live, and Facebook Live have become part of pop culture and attracted millions of viewers and streamers. For example, Twitch.tv, one of the primary live streaming platforms, has approximately 15 million daily active users and 7.7 million active channels [56]. Different from traditionally televised broadcasting and pre-recorded videos that can be cut and edited, live streaming is simultaneously recorded and broadcast in real-time on media platforms. It combines high-fidelity video presentation with a low-fidelity text-based communication system [51]. This increases user engagement by enabling viewers to interact with each other and with the streamer. Other interactive features are also embedded in live streaming platforms to strengthen social presence and interaction, such as gifting [74, 118] and donating [111].

Focusing on these unique interaction dynamics, existing literature has explored live streaming as a nuanced online social space [16, 17, 53, 115], a form of cultural production [44, 89], part of digital economy [60, 111], an engagement with cultural heritage [73], a demonstration of affective labor and online performance [66, 112], and an online community involving content moderation and privacy concerns [68, 92, 93, 109]. A growing body of work has also explored how live streaming can be used for learning and teaching various skills (including creative, artistic, and technological skills) in an interactive way. Such research has highlighted the contexts and main features of learning and teaching through live streaming:

Contexts of Learning/Teaching. Live streaming can facilitate teaching and learning in diverse contexts and topics. For example, computer science practitioners can learn and teach programming through live streaming [49]: developers can learn writing code either in streams with a small online audience [25] or at scale with a large number of viewers [19]. Artists can gather around certain stream channels and learn drawing using particular design software [33, 114] and other related creative skills such as video editing, crafting, and music [34]. Live streaming can also be used to facilitate language learning – e.g., streaming online language lessons to offer more real-time interactive experiences between language teachers and learners [17].

Features of Learning/Teaching. Previous studies have identified three main features of learning/teaching practices through live streaming. First, many online users' learning/teaching practices mainly benefit from the interactivity of live streaming regardless of their various learning contexts/topics. For example, multiple mentoring types could emerge during real-time interaction through live streaming: streamers and viewers could mentor each other; and viewers and viewers could also have the chance to mentor each other [26]. Second, live streaming can facilitate effective learning/teaching because participants engage in these practices through a combination of live audio, video, and text interaction. This is especially useful for teaching and learning creative practices: streaming authentic projects from start to finish at real time makes the learning and teaching process more understandable and approachable [34]. Third, streaming communities are often formed around a shared identity drawn from the stream's content and the shared experiences [51]. Therefore, live streaming helps cultivate communities of practice (i.e., groups of people who share a concern or a passion for a domain they do and do it better in a process of collaborative learning and regular interaction to share knowledge, stories, and skills [26, 107]). This particularly benefits community-based learning and teaching by focusing on collaboratively learning skills through doing.

In summary, prior work has revealed the nuances of live streaming as a collaborative and interactive environment for learning and teaching skills (e.g., coding and programming). In this paper, we endeavor to explore the potential of leveraging live streaming to support end-user driven creative practices, such as collaborative software development in the indie context. A core value

in end-user driven creative communities, such as the indie game development community, is to make technological practices more participatory, democratized, and accessible to everyday users through collaborative efforts [5, 69, 102]. While teaching and learning can be important to indie game developers' collaborative practices (e.g., sharing new skills and tools), their collaboration goes beyond just teaching and learning to advocate the indie culture.

3 METHODS

Data Collection. This study is part of a broader, multi-year research project on studying indie game development as public engagement in technological innovation. To recruit participants, we posted messages on Reddit, Facebook groups, and multiple Discord channels for game developers to recruit interviewees who self-identify as indie game developers and have experienced live streaming (either as a streamer or a viewer or both) for game development in the past 12 months. We also directly contacted game developers in the USA who we already knew and asked their willingness to participate using a snowball sampling. All developers who responded to our requests and agreed to participate were interviewed. In total, 27 semi-structured in-depth interviews were conducted. Interviews were conducted via voice or text chat through Discord, Google Hangouts, or Skype based on participants' preferences from October to November in 2019. No video calls were conducted in order to protect participants' privacy. The average length of interviews was 60 minutes, and participants were given a \$20 gift card after they completed the interviews.

Participants. Out of the 27 participants, 4 self-identify as woman, 1 as gender non-binary, and 22 as man. Participants aged from 18 to 51 years with an average age of 28.5 years. Eight of them (29.6%) self-reported their engagement in indie game development as a full-time level and 19 (70.4%) self-reported as a part-time or hobbyist level. Specifically, participants considered themselves "full time" if they were developing indie games as a full time job. They considered themselves "part time" or "hobbyist" if they had other full time job(s). We acknowledge that these self-reported categories do not always reflect participants' engagement level in the indie community - some participants may engage with the indie game community to an extreme extent but they are not developing indie games full-time. Participants have also engaged in indie game development in various roles, such as organizers of related events, developers, sound engineers, artists, studio owners, producers, and educators. Participants were located in different countries, including the USA, Canada, India, Malaysia, South Korea, Australia, Nigeria, Brazil, Russia, Japan, the UK, Belgium, and Austria. In the study, they engaged in live streaming either as a streamer or a viewer or both; and most participants streamed or watched streams on Twitch Creative. Table 2 summarizes participants' demographic information.

Interviews. Interviews started with basic demographic questions and moved to participants' general experiences of indie game development. Then participants were asked about their experiences and attitudes towards using live streaming in game development, including any collaborative experiences or activities through live streaming. Examples of interview questions included: *"What do you often use live streaming for?"*, *"Have you engaged in any form of collaborative activities through live streaming? What are they? How did you conduct such activities?"*, and *"Have you interacted with streamers or other viewers? Have you conducted any form of collaborative activities with streamers or viewers? And how?"* They were also asked about the impacts of policies or politics on their collaboration and the role of live streaming in broadening participation and collaboration in game development. A set of interview questions was about participants' experiences of virtual and/or face-to-face game jams, which was reported in our previous works [38, 65].

Data Analysis. Using a Grounded Theory Approach [15, 98], we conducted an empirical, in-depth qualitative analysis of the collected data. Based on Charmaz's guidelines for qualitative analysis [15], we first closely read through the transcribed interview data line by line to acquire

Table 2. Demographic information of interviewees

ID	Gender	Age	Country	Occupation	Full/Part time Indie	Experience (Years)
P1	Man	30	USA	Unemployed	Hobbyist	2
P2	Man	26	Austria	Unemployed	Hobbyist	4
P3	Woman	24	N/A	Software Engineer	Hobbyist	2.5
P4	Man	35	USA	Wood Worker	Hobbyist	9
P5	Man	28	Malaysia	UX Designer	Hobbyist	4
P6	Non-binary	27	South Korea	Teacher	Hobbyist	N/A
P7	Man	18	Canada	Student	Hobbyist	4
P8	Man	19	USA	Student	Hobbyist	6
P9	Man	31	USA	Architect	Hobbyist	19
P10	Man	18	USA	Student	Hobbyist	4
P11	Man	50	USA	Indie Developer	Full Time	28
P12	Man	20	USA	Student	Hobbyist	2
P13	Man	18	India	Student	Hobbyist	7
P14	Man	23	Nigeria	Indie Developer	Full Time	4
P15	Man	24	Brazil	Indie Developer	Full Time	7
P16	Man	22	N/A	IT Contractor	Hobbyist	10
P17	Man	40	USA	IT	Hobbyist	2
P18	Woman	30	USA	Indie Developer	Full Time	3
P19	Man	25	Belgium	Indie Developer	Full Time	7
P20	Woman	18	USA	Artist	Hobbyist	1
P21	Man	51	Russia	Indie Developer	Full Time	15
P22	Man	46	Japan	Indie Developer	Full Time	23
P23	Man	28	Australia	Unemployed	Hobbyist	15
P24	Man	37	UK	Unemployed	Hobbyist	27
P25	Man	22	USA	Student	Hobbyist	2
P26	Man	24	UK	Software Developer	Hobbyist	12
P27	Woman	36	USA	Indie Developer	Full Time	9

Note: Country – country of origin; N/A – participants preferred not to answer.

a general understanding of the whole picture regarding indie game developers' collaborative experiences via either watching or streaming game development practices. We then conducted open coding of the data inductively to generate initial categories and wrote conceptual memos (e.g., emotional support during streaming and viewers' participation). Next, we identified a preliminary set of narrative categories (e.g., social and technological support during streaming, challenges due to the platform design itself) through axial coding, which highlighted the ways of live streaming can be used beyond teaching and learning skills and the potential challenges. For example, we combined categories such as "information overload", "much social interaction with audiences", and "social stress to manage live streaming technology" together as a broader category "too much distraction during streaming." We carefully examined and reviewed the categories regarding each research question for further analysis by focused coding. We also worked together in an iterative process to extract quotes and discuss, combine, and refine categories. Lastly, we used the extracted quotes and refined categories to generate a synthesized description to answer our research questions.

4 FINDINGS

In this section, we first describe how indie game developers use live streaming in various aspects to support their collaborative game development beyond learning and teaching skills (RQ1). We then identify major challenges for indie developers to use live streaming as a collaborative space for game development (RQ2).

4.1 Live Streaming for Collaboration in Indie Game Development beyond Learning & Teaching

In addition to learning and teaching technical skills, our participants leverage live streaming to support their collaborative efforts for innovating game development in three ways, including facilitating a democratic and participatory model for game development, building a collective

presence to increase the public visibility of indie game development, and fostering an open and supportive indie culture for collaboration.

4.1.1 Leveraging Live Streaming to Facilitate a Democratic and Participatory Model for Game Development. Similar to prior literature on live streaming, our participants also acknowledge that the uniqueness of live streaming is that streamers and viewers can “bounce ideas off people” (P8, man, 19, USA, Hobbyist) in real time. This feature allows indie developers to collaboratively and synchronously design, develop, solve problems, and socialize with streamers and viewers in the moment. In this sense, live streaming platforms can be used to facilitate a democratic and participatory game development model where potential game players (e.g., audiences) can actively engage in the game design and developmental process as partners.

For instance, participants highlight how they can use live streaming to transform potential players from passive recipients of their games to active partners in their game development through two-way communication, which seems to foster a more democratic model of game development. P10 (Man, 18, USA, Hobbyist) mentions that indie developers who stream their ongoing development processes also effectively collect various feedback from the audiences, which can be incorporated into their final products:

“If somebody is making a game, and I’m interested in following it, I like seeing the development process and maybe they’ll be drawing up some art and they’ll ask the chat: ‘Hey, how does my cool enemy look?’ ‘Oh, we should do it more this way.’”

According to P10, live streaming indie game development is similar to online participatory design [90]: developers who are streaming often invite viewers (who are potential players) to participate in the development process by giving comments, feedback, or suggestions through live chat. In this way, potential players actively engage in various aspect of game development (e.g., ranging from game background music production and avatar’s visual design to game storyline) early in the development cycle. This thus changes viewers’ role from potential users to “co-designer” [90] (e.g., people who are not trained in design working together with designers in the design and development process) of a digital game and makes the game development process highly collaborative.

Similarly, P1 (man, 30, USA, Hobbyist) adds,

“People might make suggestions to the way a game goes. And so it’s a little bit like they’re voting and giving feedback. And I think that can be a really cool thing for a programmer developer because they’re getting feedback on the way people wanted it to.”

P1’s account further highlights how live streaming can foster a democratic space for indie game developers and potential online players to work together – it is almost as if players, as audiences, collectively contribute to the developers’ decision-making (e.g., by voting on the game design feature that they liked most). In a stream, every viewer is able to type their own thoughts and communicate with other viewers through the embedded live chat. All chats and comments are also publicly visible to all viewers in the stream and the streamer. In this process, both the streamer and the viewers engage in collaboratively evaluating and identifying the most popular, reasonable, and best-supported feedback or suggestion in real time. The developer/streamer can then incorporate these feedback into their game development in turn.

For some other developers, the benefits of using live streaming for democratic and participatory game development with players may even go beyond the design stage. Many perceive live streaming

as a useful hub for evaluating how their games work by watching how players interact with their games after releasing:

“It’s incredible that you can learn so much from watching people play or do things that you would never expect people to do. I think that it’s much more rapid feedback than ever before.” (P11, man, 50, USA, Full Time Indie)

According to P11, the democratic nature of live streaming facilities the playtest phase. This provides “*much more rapid feedback than ever before*” and helps him further improve the design of his game. Especially, unlike lab-based playtests, such spontaneous “live streaming user tests” may produce more authentic feedback and reveal how players actually play the game in real-world (e.g., “*do things that you would never expect people to do*.”)

Therefore, for indie game developers, live streaming is not merely about “watching” or “interacting” but “participating,” which promotes a sense of interdependency and democracy. On the one hand, indie developers can be streamers who stream their game development process to gather feedback from viewers and then incorporate those into their products. They can also be viewers who watch streams of gameplay to collect spontaneous playtest data. On the other hand, potential game players actively participate in the game design process as viewers (audiences) and help the streamer/developer succeed, so the final product can better meet their own interests and play needs.

4.1.2 Leveraging Live Streaming to Collaboratively Build the Presence of the Indie Community. People are able to locate streams they want to watch or promote by searching keywords or adding hashtags to a live stream. Many streaming platforms also allow users to broadcast their events to a massive audience. Our participants also mention that they often leverage these benefits of live streaming to collaboratively build the presence of the indie community. This is critical to many indie developers as visibility has been recognized as a significant challenge for promoting and facilitating indie game development [35]. As an end-user driven creative community, indie game developers share the aspiration for standing together to have a stronger voice.

Therefore, P11 (man, 50, USA, Full Time Indie) views live streaming as an important means of “collaborative marketing” for indie game developers to collectively build their stronger presence together online:

“I enjoy sharing my process with others and explaining what I’m doing, [...] and letting people play it, see what’s coming up or see how I make things or give a little preview of whatever new features that are coming to the game. Those are things I do. So it’s marketing. The number of indie developers who stream is also increasing so we can market the concept of indie games all together.”

According to P11, live streaming provides indie game developers with a free public platform to present their talents. It also provides viewers with a channel to learn about indie developers and even reach out to them. In addition, live streaming attracts and gathers indie game developers all over the globe and helps them present indie games as a growing industry trend and indie game development as a novel tech community. This collective presence goes beyond individual developers’ personal efforts, which further reinforces the visibility of indie game development to the general public. It also motivates developers’ collaborative efforts to gain acknowledgment, understandings, and recognition. P23 (man, 28, Australia, Hobbyist) and P17 (man, 40, USA, Hobbyist) both explain how such collective presence may emerge:

“Rather than just watching something passively, you can actively communicate with the person on the screen and you can ask them questions about the game, you can talk about your game if they’re open to it. [...] And if they can do this with

many developers, they kinda get a good idea about what indie game development is in a collective way." (P23)

"It can be a good way to start to build an identity of indie game developers as a community. With a more visible community, people are going to be interested in your end result so it can basically be PR for a game or your team." (P17)

According to P23, a unique benefit of using live streaming in indie game development is that indie developers can interact with viewers in a more identifiable and personable way – because they can present themselves as real human beings, rather than faceless names attached to their products. This makes both developers and viewers more willing to participate in online interactions and communications. It also helps developers grow closer to their audiences and helps the general public better understand the indie culture and practices. Similarly, in P17's account, indie game developers can cultivate a sense of their own community and identity by having a collective presence on live streaming. This helps spread knowledge and information about indie game development through word of mouth, which increases public awareness of this particular tech community.

4.1.3 Leveraging Live Streaming to Foster an Open and Supportive Indie Culture for Collaboration. It is also important for our participants to leverage live streaming to foster an open and supportive indie culture for collaboration. This is not only "valuable to help create a community" (P7, man, 18, Canada, Hobbyist) but also to maintain the tenacity and closeness within the community. They especially highlight two main ways through which live streaming can be used to foster such an collaboration-friendly atmosphere: maintaining collaborative relationships in a more natural way, and encouraging various types of community support.

Maintaining Collaborative Relationships in a More Natural Way. Participants point out that live streaming can be used as a casual, natural, and less awkward way to maintain relationships with potential collaborators. P16 (man, 22, N/A, Hobbyist) reveals,

"Compared to just DMing somebody out of the blue or Tweeting at somebody, a very nice, casual, no pressure way to keep up with people is to just start live streaming or go on to somebody's live stream and have a chat with them."

With the feature of casualness and spontaneity, idle chatting via live streaming becomes an ideal way for indie game developers to get closer with each other and build friendships for potential future collaboration. P26 (man, 24, UK, Hobbyist) and P22 (man, 46, Japan, Full Time Indie) share:

"I tend to chat with them and make fun of the others as well." (P26)

"Sometimes just like socializing, like 'How are your kids?' [...] Just whatever the natural kind of emerging social culture that seems to come from that sidebar with everyone talking in it." (P22)

For both participants, the importance of engaging in live streaming is not merely about learning skills and discussing game development. Rather, they often engage in topics beyond gaming and are willing to joke and chat about personal stories and their daily lives. For them, live streaming helps them "bond" with other indie developers and seek potential collaborations based on the developing relationships.

Maintaining a certain frequency of live streaming also helps strengthen the "bond" between streamers and viewers, as P4 (man, 35, USA, Hobbyist) points out,

"It's sort of being on a schedule that forces you to stick to, even if there's only one person that ever watches if they're expecting you to be on a certain time."

For P4, keeping a commitment to the stream schedule adds a level of accountability to the communication between streamers and viewers. This may help build public credibility and trust

for collaborative activities in turn. Accordingly, audiences may naturally give “*live support and encouragement*” (P24, man, 37, UK, Hobbyist) to developers as streamers, especially when they are participating in designing the game together.

Encouraging Various Types of Community Support. Many participants mention that they have used live streaming to either share or receive various social and emotional support. Such support makes them even more open and willing to engage in potential collaborative activities through live streaming. For example, P16 (man, 22, N/A, Hobbyist) highlights the donation feature of live streaming:

“A lot of live streaming like Twitch have some kind of donation systems, some way to tie that into your crowdfunding...I think it'd be cool.”

Donating to indie developers who are streamers not only expresses viewers’ financial support to the developers but also practically “*takes some of the economic anxiety off*” (P6, non-binary, 27, South Korea, Hobbyist) to help indie developers devote themselves to their game development practices. Such support further solidifies a sense of the indie community on live streaming, making people more open to exchange mutual help. P18 (woman, 30, USA, Full Time Indie) locates in a place where she is unable to connect with other indie developers. By using live streaming, she is able to overcome the isolation and engage in the indie community. She explains,

“I would say that it just kind of adds on to the sense of community especially for people like me. I don’t even interact with people, so I’m kind of a weird case. I live in the middle of nowhere so I definitely can’t participate in any game jams. But it’s just a way to interact with the other developers and interact with the community around the game jam but without having to be there.”

For developers like P18, live streaming allows them to still have a sense of community despite lacking offline social networks and in-person interaction with other indie game developers due to geographical limitations. Some even are able to find future collaborators, as P1 (man, 30, USA, Hobbyist) shares:

“Because instead of having to download each game and play it yourself, you can watch it (live streaming) from your TV or your phone. And then you can be like, ‘Hey, Dave, that was really cool. And I want to follow you on Twitter.’ and ‘Maybe I could do music for your next projects or something like that!’”

In this quote, live streaming seems to become a new means of socializing and networking for many indie developers. It is convenient and efficient, as people can just watch without the efforts to download and play any game. It can also be watched from anywhere - TV, phone, or tablets rather than sitting in front of a computer. This convenience helps people to stay connected and engage in relationship building whenever and wherever. This constant connection, therefore, makes asking for collaboration, help, or building friendship easier and more feasible.

4.2 Challenges for Collaboration through Live Streaming in Indie Game Development

Yet, our participants also highlight several challenges emerging in this progress of leverage live streaming platforms to innovate game development collaboratively: difficulties to identify streams relevant to game development; increased distractions from collaborating on game development itself due to information overload, the presence of audience, and streaming tools; copyright concerns; and privacy issues.

4.2.1 Difficulties to Identify Streams Relevant to Game Development. Live streaming has grown beyond its original use for streaming gameplay and has been increasingly used by the creative communities (e.g., Twitch Creative). However, many participants still feel that existing live streaming

platforms are not designed for supporting collaborative software development such as indie game development. For example, our participants explain the frustration to identify streams relevant to game development. P1 (man, 30, USA, Hobbyist) points out that Twitch took “indie game development” category away, making it challenging for both streamers and viewers to locate “game development” streams on different platforms:

“Minecraft or Fortnite, those are all categories and art is a category. But this person who is streaming game development has the stream under science and technology. And so like, if somebody sees ‘Oh, these people are streaming under science and technology that doesn’t sound as exciting or descriptive as game development to me.’”

In P1’s opinion, even Twitch, the largest live streaming platform, does not provide a clear and straightforward way to label and categorize “game development” relevant streams. Rather, such streams are mixed up with other gameplay streams or science and technology streams. This not only weakens the public visibility of the indie community but also creates additional barriers for indie game developers to identify potential collaborators and interact with one another.

Therefore, some participants propose that it is necessary and important to devote a live streaming platform or channel specifically to indie game development. Such a platform/channel would better serve the nature of indie game development as collaborative software development. P24 (man, 37, UK, Hobbyist) highlights,

“I think on the whole, live streaming for game devs would be a much more enjoyable thing if there were a specific place for them, as it stands on Twitch, there’s no specific game dev category, all the fields are separated [...] right now on Twitch you have all your artists doing actual art, you got 3D modellers just doing random 3D (not always for game) and it would be nice to specifically target these areas for game dev as they are often so different.”

According to P24, indie game development typically involves visual design, storytelling, arts, programming, music, sound effects creation, and many other aspects. This is why it is challenging to list it under a single tag on live streaming platforms. However, existing live streaming platforms are often not designed to co-list multi-disciplinary content for a specific technological practice such as indie game development. In this sense, how to make live streaming platforms meet the collaborative and multi-disciplinary needs of indie game development will be an important question for designing and developing future live streaming platforms to further support creative technological practices.

4.2.2 Easy Distraction due to Information Overload, the Presence of Audience, and Streaming Tools. Another major challenge that many participants mention is the easy distraction from collaborating on game development itself due to the overloaded information produced in live streaming, the presence of audience, and the use of various streaming tools.

Distraction due to Information Overload. Compared to watching regular videos, developers who engage in live streaming either as streamers or viewers have to process the increasing amount of information both from the streamed content and the two-way communication system (e.g., live chat). P17 (man, 40, USA, Hobbyist) notes that this information overload can easily distract and interrupt indie developers, which may hinder a potential collaborative process:

“Somebody just randomly made a comment that gave me an idea or a good idea, but it didn’t necessarily help my creativity. I think it may actually hurt because it was pulling part of my attention.”

While feedback from live chat can be highly helpful for indie game development, this interactive nature of live streaming may also become a distraction that slows down the actual development progress. Indie developers like P17 often have to spend time and attention to filter, screen, and evaluate these information to identify potential collaborative opportunities. Likewise, P22 (man, 46, Japan, Full Time Indie) considers being forced to pay attention to excessive information in live streaming as "*certainly the cost and time that can detract from your actual development.*" For him, the interactive feature of live streaming harms, rather than facilitates, his potential collaborative practices in game development – because he is overwhelmed by the information from interacting with audiences.

In addition, participants consider viewers' feedback both "*good and bad aspects for indie developers' innovation*" (P2, man, 26, Austria, Hobbyist). P2 expresses his concerns:

"There was a streamer B [a streamer's name] just let the chat have input on the game or make decisions. So maybe during the time he designed the character, he would ask questions like what features should the character have? Is the face okay? Should the headphones be changed? And it was a lot of engagement with the community in the chat. But it's difficult because making a change can require a lot of time."

In P2's account, this streamer is an excellent example of engaging potential players in collaborative game development. While P2 in general does not oppose this practice, he considers seeking and incorporating viewers' feedback in every step of game development unnecessary and risky. This not only produces too much information that requires further filtering and evaluation but also delays the development process indefinitely. For him, these information and input become "noises" rather than catalysts for collaborative software development.

In this sense, though information overload has been reported as a challenge in various live streaming practices such as gaming [51, 66, 110], indie game developers who stream or watch a stream face even more difficulties to deal with their information load. Streamers in gaming or other contexts can simply ignore the overwhelming information from the live chat or depend on dedicated moderators to filter such information [51, 66, 82, 110]. In contrast, indie game developers situate in a unique situation. On the one hand, they endeavor to facilitate a democratic and participatory model for game development. This means that as streamers, they have to continue to attend to and process viewers' comments and feedback themselves (e.g., to incorporate them into their game development). On the other hand, they are streaming or viewing highly creative and technical content (e.g., game development) in real time. This requires them to concentrate on the details. Therefore, the constant updates from the chat can become noises. For them, it seems to be challenging to achieve a balance between effectively monitoring useful suggestions from all parties and concentrating on their own creative practices without being distracted from monitoring the ongoing information flow. In addition, indie developers may also need to be careful about the fine line between collecting ideas and viewpoints for their game development from the audience and their own independent thinking in this collaborative process. Over-depending on such information from audiences may become *information overload* that dilutes their creativity and independence.

Distraction due to the Presence of Audience. Live streaming can also be a distraction for developers simply because someone is "watching." P27 (woman, 36, USA, Full Time Indie) shares that developers who stream have to maintain a healthy social atmosphere to ensure a steady streaming process:

"We don't have much negativity on our channel because I try to deflect people that come in. Like they'll have a favorite coding language. And they'll be like

saying, ‘Oh, it’s better than all the others’ and I’ll try to be like, ‘Well, we don’t really encourage that here because there are reasons to use different ones.’”

In this example, P27 has to be sensitive to contentious comments from her audience so that she can block harmful or toxic information that is destructive to a potentially collaborative atmosphere. This thus somehow distracts her from engaging in communicating/discussing with her audience about her game development.

P16 (man, 22, N/A, Hobbyist) also adds that he naturally feels “*uncomfortable*” because of the social pressure and anxiety involved in live streaming:

“When you’re starting out, it can be a little bit less casual when you have zero viewers. And someone just pops in and starts talking to you. That can be a lot of pressure. It’s a very weird relationship that you’re having with this person where they can see you, they can hear you and you can only see what they type. I’ve definitely been uncomfortable while I’m streaming before because there’s never a way to shake people off.” (P16)

The fact that P16 is being “watched” itself distracts him enough from fully engaging in conducting game development or seeking collaborative effort from the audience in front of the camera. As he mentions, there is “*never a way to shake people off*” when streaming. Likewise, P17 (man, 40, USA, Hobbyist) points out the difference between indie developers who focus on collaborative game development and streamers who focus on pleasing audiences. For him, live streaming is not about streaming his own game development practices but about putting on a performance (e.g., “*your personality and your stage presence*”). In this sense, he may need to pay more attention to how to please his audience rather than concentrate on streaming his game development practices.

Distraction due to Navigating Streaming Tools. Another distraction may come from how to appropriately navigate and set up live streaming technologies. P27 (woman, 36, USA, Full Time Indie) expresses her concerns:

“How are they probably hitting barriers early and never trying it beyond that, like people who might be like dancers or musicians or artists or like this who’s out there making cool stuff? They’re so overwhelmed by having a good webcam, having a good microphone [...] then suddenly there’s a bandwidth issue and they don’t know what to do about it.”

As P27 mentions, many indie game developers are amateurs and non-professional game developers. In addition to navigating game development as an advanced technological practice, they also have to learn how to appropriately use non-intuitive live streaming tools such as Stream Labs OBS and solve emerging technical issues (e.g., bandwidth). For them, live streaming creates additional technical barriers to engage in the already challenging collaborative software development practices.

4.2.3 Copyright Concerns regarding Collaboration through Live Streaming. Our participants are aware of the open and sharing nature of live streaming, which well aligns with the indie culture. For example, many indie-focused game jams (e.g., Global Game Jams) use the open intellectual property ownership model. In this model, indie developers own the original content but are expected to share the created games, assets, and source code so that other people can download, modify and distribute for non-commercial purposes. However, some participants are concerned that they may accidentally disclose some sensitive information about their technological practices in live streaming. Many of them find themselves in a dilemma: they hope to openly present and promote their game development to the public so as to seek potential collaboration and input from the audience; yet, they are also worried about their copyright and the possibility that others may steal their ideas and take the credit. P3 (woman, 24, N/A, Hobbyist) reflects,

"I have not used live streaming for indie game development much, because I kind of want it to be private and mine, confidential. Not that somebody's going to watch my stream, copy my game. But you know, it's always the thing that freaks me out. Maybe they'll steal my ideas if I'm programming it right in front of them."

For P3, streaming her projects on live streaming poses a risk to her intellectual property rights. Similar to many indie-focused game jams mentioned above, some live streaming platforms implement open intellectual property ownership. Some other platforms do take actions to protect streamers' copyright. For example, they would censor uploaded streams to identify copyrighted content if it matches any third-party content. However, either mechanism cannot protect developers' copyrights outside the live streaming platform. Therefore, there is always a risk that indie developers' ideas, designs, and codes shown in streams can be used in others' products without informing or crediting them.

Moreover, some mainstream game companies are able to exclusively stream certain content where they own intellectual property rights. For indie game developers who significantly rely on open-source resources (e.g., game engines, platforms, patches, and art assets) released from these companies, what they can stream and share is restricted. P4 (man, 35, USA, Hobbyist) explains the issue:

"Because most of the projects I work on, I have to sign NDAs and things like that. So, I can't actually just show the world what I'm working on. But back then, when I was working on smaller things and things that could be shown." (P4)

P4 needs to pay special attention to the scope of his streamed content so he does not cross over the line accidentally. Likewise, P15 (man, 24, Brazil, Full Time Indie) gives an example of YouTube removing everyone's Let's Play (a Switch game) videos from their platform. For him, many live streaming platforms provide indie game developers with an easy way to cultivate their communities and careers. However, many aspects of indie game development, such as the community connection, sharing knowledge, and marketing, are heavily restricted by mainstream gaming companies. This thus leads to copyright concerns. These concerns may also cast doubt on the independence of indie game developers as they claim to be – as we have seen, mainstream game companies are still able to control and manipulate what and to which degree indie game developers can share online from the intellectual property perspective.

4.2.4 Privacy Issues when Sharing Technological Practices through Live Streaming. The open and sharing nature of live streaming may not only risk indie game developers' intellectual rights but also their privacy. For many, how to approach and manage privacy has become a central concern. At the personal level, many developers are worried that they disclose personal information unintentionally when sharing game development practices through live streaming. For example, P27 (woman, 36, USA, Full Time Indie) shares how someone stalked her and violated her privacy:

"It was really uncomfortable because he followed me on all my social media after coming into the chat. And I didn't know if this person was going to seriously harass me more. We blocked him. But I didn't know if it would keep going, because for some people it does. And I think that's really scary. And you also have to be really smart about what you are sharing. And are you sharing things that are too much? Like could people figure out where you live, would be a really scary thought."

In P27's case, streaming her game development practices allows someone to invade her privacy and track her online beyond the live streaming platforms. For her, streaming may inevitably reveal her personal information and put her personal safety at risk. In addition, no effective tool

seems to exist for protecting streamers because even blocking may not stop people from harassing or following them. For indie game developers, they have to restrict the range of interaction, communication, and the specific streamed content to protect their personal privacy, as what P27 said "*be smart about what you are sharing*". Yet, in doing so, they may miss the opportunity to fully leverage the open and interactive nature of live streaming to facilitate their collaborative practices and productivity.

5 DISCUSSION

Our findings have shown that in addition to teaching and learning technical skills, live streaming can be used to support indie game developers' collaborative efforts for innovating game development in three ways: facilitating a democratic and participatory model for game development, building a collective presence to increase the public visibility of indie game development, and fostering an open and supportive indie culture for collaboration (RQ1). However, our findings also reveal several challenges for indie developers to engage in collaborative game development through live streaming. They often have to face difficulties to identify streams relevant to game development, increased distractions from collaborating on game development itself due to information overload, the presence of audience and streaming tools, copyright concerns, and privacy issues (RQ2). We now use our findings to discuss the implications of this work for bridging existing HCI/CSCW research agendas on collaborative software and live streaming. We also provide potential design recommendations for addressing challenges emerging in distributed collaborative practices through live streaming.

5.1 Bridging Collaborative Software Development and Live Streaming in the Context of Indie Game Development

An important insight from this study is how our findings uniquely bridge previous research on collaborative software development and live streaming. On the one hand, we focus on indie game development, which leverages live streaming to support distributed collaboration between developers and between developers and players to collectively advocate the indie culture. This community represents an emerging form of end-user driven collaborative software development, which differs from distributed development teams in workplace settings (e.g., [6, 11, 76]) or OSS communities (e.g., [22, 78, 97]) that mainly focus on expert or professional developers. This thus sheds light on how collaborative software development can be conducted and approached in new ways and by new populations.

On the other hand, indie game developers indeed share similar collaborative learning, teaching, and online programming mentoring practices through live streaming as reported in previous research [19, 25, 26]. Yet, their aspirations to innovate the traditional game development model also motivate them to leverage live streaming to support their collaborative efforts beyond just learning and teaching skills. As our findings have shown, they have leveraged live streaming to seek an alternative game development model that focuses on democratic participation; they have also used live streaming to increase the public visibility of their community. In this sense, the indie community seems to acknowledge the key role of streaming in the contemporary gaming ecosystem. For them, streaming has become essential for the "discoverability" and the ultimate sales of their games [87, 101], as well as for fostering an open and supportive indie culture.

Therefore, these insights can inform the design and development of future live streaming platforms to better support similar end-user driven creative practices (e.g., making, hacking, crafting, and marketing), which may not only focus on teaching, learning, and sharing but also emphasize community building and/or advocating aspirational agendas. Specifically, our study bridges the above-mentioned two research agendas in three ways.

5.1.1 Explaining Collaborative Software Development through Live Streaming Using Collaboration Theories. Using indie game development as an example, we are able to explain collaborative software development through live streaming using collaboration theories (Table 3).

Table 3. Mapping Findings with the Three Main Theories on Collaboration

Theory	Elements	Description	Examples
D'Amour et al.	Sharing	Shared experiences in indie game development	P4: "We are all in this together so I'm happy to help in any way"
	Membership	Mutual technical help	P13: "If there's some bug out, usually with Neil, there's this back and forth between us."
	Interdependency	Spontaneous co-designing based on two-way real-time communication	P21: "I had some feedback given by players and that I finally used because they were better than mine."
	Power	Collectively increase the visibility of indie game development	P1: "When you see there are many other indie developers streaming, you kinda build a collective identity to promote indie games."
Process		Establish "bond" between streamers and viewers	P26: "I tend to chat with them and make fun of the others as well."
Theory	Patterns	Description	Examples
Biuk-Aghai et al.	Artefact-Exchange		
	Artefact-Management		
	Group-Discussion	Ask viewers for feedback	P10: "They'll ask the chat: 'Hey, how does my cool enemy look?' 'Oh, we should do it more this way."
	Message-Exchange	Direct messaging between streamers and viewers	P22: "Sometimes just like socializing, like 'How are your kids?'"
	Workspace-Setup	Use live streaming as a co-working space	P4: "Leave it in the background while I'm working [...] It's sort of like a fill in for co-working spaces."
Theory	Aspects	Description	Examples
Olson et al.	Common Ground	Collaborate	P13: "Trying to figure something out together."
	Coupling of Work	Co-design via live streaming	P21: "If this project is commercial, we'll obviously have early feedback."
	Collaboration Readiness	Promote products	P11: "We can market the concept of indie games all together"
	Collaboration Technology Readiness	Dealing with live streaming tools	P27: "Suddenly there's a bandwidth issue and they don't know what to do about it."
	Organizational Management		

First, based on the five elements of collaboration [24], the democratic and participatory model of game development fostered by live streaming emphasizes the notion of sharing and allows for collective mutual efforts oriented toward a common goal [106]. This highlights *sharing* and *interdependency* of collaboration. Specifically, for indie developers, the synergy across streamers and viewers helps deepen aspects of sharing such as decision-making, skills, and information/data. Developers also note the need for different people's skills and abilities at different times throughout collaboration, highlighting the importance of *interdependency*.

Second, as our findings have shown, participants endeavor to collectively increase the visibility of indie game development by making live streaming a marketing tool. In this process, everyone involved in a stream enjoys the *power* associated with helping increase awareness and spread potential impact [54]. Such power is not based on respective roles or titles in the stream but their experiences, knowledge, and motivations within a collaborative and collective community.

Third, various forms of social and emotional support provided in live streaming further strengthen the aspects of *membership* and *interdependency* of collaboration. In our study, indie developers are aware of the mutual support and the resulting benefits, which helps community building based on the mutual dependency and synergy. These efforts maximize individual contributions to make the output of the whole much larger than the sum of inputs from each part [24] to benefit indie game development. Lastly, all practices through live streaming are shaping indie game developers' distributed collaboration in a dynamic, interactive, and evolving *process* by enhancing interpersonal communication, accumulating collective efforts, and encouraging mutual engagement through the live nature of streaming.

Our findings also shed light on how collaborative software development through live streaming reflects certain aspects of Biuk-Aghai et al.'s collaboration theory [10] and Olson et al.'s theory [84, 85]. For example, indie developers' efforts to ask for feedback from viewers in live streaming initiate *group discussion* in collaboration [10]; and the motivation to build a collective identity to promote indie games demonstrates the *collaboration readiness* in the indie game community [84, 85]. The fact that indie developers, whether as streamers or viewers, have to deal with live streaming technological issues such as microphone setup and bandwidth, also represents the collaboration technology readiness in a collaborative practice [84, 85].

5.1.2 Implications of Live Streaming for Collaborative Software Development beyond Learning and Teaching. Our findings extend prior literature on live streaming for coding, programming, and online mentoring (e.g., [19, 25, 26]) by shedding light on how live streaming can be used beyond teaching and learning skills, for example, for helping address issues emerging in collaborative software development as identified in previous literature.

Above all, the difficulty of awareness including staying aware of others [46, 47], workspace [23], or emotion [48] has been pointed out many times in distributed software development. Live streaming with two-way real-time input allows indie developers both as streamers and viewers to have a consistent awareness of the workflow, the created content, and the feedback from both sides in real time.

In addition, distributed software development often suffers from the difficulty to build trust among team members in teams and collaboration [52, 57, 96]. Previous studies have shown that live streaming can promote online community formation and build supportive relationships between viewers and streamers [94, 111]. In our study, indeed, live streaming seems to create a natural and casual atmosphere for indie developers to establish and maintain collaborative relationships. It also encourages various types of community support to enhance such relationships and mutual dependency with different affordances, such as the live chat system and the feature of donation. In the past, the indie community often had to depend on specific social networks such as Kickstarter or Indiegogo for crowdfunding [71]. Now, live streaming seems to introduce new and more effective ways for collaboration in the indie culture by engendering trust and offering more support. Additional focused work is still needed to specifically investigate how the overall environment and processes associated with live streaming impact trust. Yet, that being said, our study points to the possibility that the environment may lead to fruitful socio-emotional behaviors, such as casual chat for networking and relationship building, donation, and community support.

Live streaming also further helps to limit cultural blind spots through a diversity of perspectives and cultures that global software development usually sees as pain-points [55, 75, 83]. In some sense, live streaming is growing into a suitable platform for indie developers across the world to share, learn, and collaborate together by increasing more authenticity, transparency, spontaneity, accountability, and respect. For them, these characteristics integrate inquiry or discussion within the democratic and participatory model of game development that they advocate into part of the streaming to weaken interruptions [76] and increase real-time communication to reduce uncertainty and misunderstanding [13].

5.1.3 A Critical View of Using Live Streaming for Distributed Collaboration in Software Development. Despite these nuances and benefits, we also offer a critical view of using live streaming platforms for distributed collaboration in software development. As our findings have shown, being live and providing real-time two-way communication presents both pros and cons for such collaboration. As far as being “live” is concerned, on the one hand, the spontaneous nature of live streams sets an authentic, transparent, and personable tone to help establish an environment of trust for online collaboration [20]. On the other hand, being live not only results in viewers not having flexible control over the information received but also puts more pressure (e.g., being watched by viewers) on streamers in the collaborative process.

In addition, the real-time two-way connection between streamers and viewers as well as among viewers makes it easier to produce both more relevant and irrelevant information, leading to “noise” for developers’ collaborative practices. Although previous research has shown that these features can benefit general live streaming practices by forming new ties and connections [94, 111], they may in fact hinder indie game developers’ productivity. As our data have shown, indie developers often find it challenging to balance between effectively gathering useful suggestions and feedback from the live chat and concentrating on actually conducting their creative practices. In this sense, the behavior of streaming or watching a stream itself seems to be a distraction from the practice of creativity and innovation *per se* that requires abundant concentration.

Further, as a creativity-centric community, how to better protect indie game developers’ intellectual property rights and copyright seems to conflict with the open and sharing nature of live streaming, which has been a long standing issue in online creative communities [30, 31]. This concern, along with the privacy risk of disclosing too much personal information [39, 68], makes indie game developers reluctant to collaborate through live streaming as they are worried about potential online stalking and harassment.

In general, better understanding these new perspectives and challenges in parallel leads to an increasing need for a broader examination of the role of live streaming in distributed collaboration in various contexts, including software development. This study shows our efforts to bridge the previously often disconnected research agendas on collaborative software development and live streaming, so we believe that the issues and new research questions surrounding how to expand collaborative environments for new technological practices (e.g., indie game development) will continue to grow in their importance.

5.2 Designing Live Streaming for Distributed Collaboration

Our findings highlight that in many ways, indie game developers are using the live streaming platforms in a manner that they were not initially designed for. This in turn brings forth both positive and negative effects. Therefore, a reconsideration of how we can (and should) improve the design of live streaming platforms to address the many challenges emerging in distributed collaboration for end-user driven creative activities (e.g., DIY making, hacking, crafting, and indie game development) seems to be necessary. In this section, we outline three main design implications.

1. Make End-User Driven Creative Streams/Communities More Identifiable and Traceable. In our study, participants often complain about the troubles associated with identifying live streams focusing on indie game development. This may partly come from the fact that indie game development, a segment of the game industry, has been under-emphasized or little discussed when compared to mainstream fields in many aspects [45, 72, 100]. As live streaming has become an important collaborative space, the challenge to identify a subsection or sub-community of certain end-user driven creative streams (e.g., those focusing on DIY making, hacking, open design/manufacturing, crafting, and indie game development) can be the first barrier to hinder these users from seeking potential collaborators and tracking down related streams and communities. Therefore, there is a clear and urgent need to make specific end-user driven creative streams and communities more identifiable and traceable on live streaming.

To address this issue, our first recommendation is to make these specific end-user driven creative streams and communities stand-alone categories within the overall live streaming community and across different live streaming platforms. For example, instead of moving streams regarding indie game development to science or technology category as a secondary category, adding a category specifically for indie game development is straightforward for indie game developers to locate what they need. We also suggest that streamers always use hashtag keywords like “indie game development” in their live streams to increase the identifiability of their streams, which can also be helpful to collectively strengthen the public awareness of their practices and community. Our third recommendation focuses on allowing those streams whose content may be closely related to more than one category to fall under different categories simultaneously. It may also be helpful to design secondary tags to provide viewers with more specific information about the stream (e.g., a stream about how to make music effects for an indie game can be given a secondary tag “Music”).

2. Mitigate Distractions during Live Streaming. The interactive and multimedia nature of live streaming can produce much more communication, participation, and management [51, 114, 117]. However, some of these are entirely unrelated to the game development process itself and may distract it from a perspective of collaborative practice. Therefore, design implications relating to these issues should put forth efforts to control information input and lower the threshold of streaming tool use to help mitigate distractions.

We recommend that direct messages to streamers should be highlighted to keep them from being missed or lost in the chat system so that developers can get the important information easily. To avoid getting flooded with messages, we also recommend live streaming platforms to offer streamers more options to filter messages in chat system based on keywords, which may help them find the necessary information and reduce irrelevant information. Another important recommendation is to make streaming tools simple and easier to use. For example, we recommend providing general users with a detailed plain-language guidebook to help solve various issues such as webcam settings, microphone settings, and bandwidth problems. For users who focus on collaborative software development and coding (e.g., indie developers and hackers), the chat system can also be designed to detect codes to offer professional-looking and readable formatting or keep the original formatting.

3. Offer More Protection for Intellectual Property, Copyright, and Privacy. Copyright concerns are not new in online environments [67]. Indeed, live streaming, as an open online space, requires much more direct engagement and interaction and reveals more bioinformation (e.g., appearance, gender, and voice). This makes sharing practices the challenging facet of collaboration from perspectives of copyright infringement [14, 68]. In our study, participants share concerns that the open and sharing nature of live streaming may harm indie game developers’ intellectual property rights. Therefore, for copyright concerns, we suggest that live streaming platforms and all users should supervise copyright infringements jointly. We recommend that live streaming platforms can collaborate

with other venues such as main game jam contests to expand indie games database as third-party content for matches; the live streaming system can also offer users an one-step report function to report streams that include copyright infringements easily. In addition, platforms can offer a method to record and timestamp sharing activities privately or store such records within the main database in case a dispute over copyright arises. In this way, a creator could point to a moment when they shared certain content or activity in their stream and have proof they did it first if needed. These mechanisms should also be used collectively with effective strategies to protect indie game developers' personal privacy, in addition to protecting the products and content created by them. These strategies may include a better way to block unwanted messages and an more effective way to prevent cross-platform stalking.

5.3 Limitations

This study has a few limitations. Our findings do not differentiate how different live streaming platforms may affect indie developers' distributed collaboration experiences in various ways. To address this limitation, more work will be conducted to compare diverse collaborative practices mediated by different live streaming platforms. In addition, we acknowledge that our sample leans heavily towards male indie game developers. This seems to be consistent with the general gender distribution of game developers: a worldwide game developer survey shows that more than 75 percent of responding game developers are men, and about 20 percent are women from 2014 to 2019 [21]. As the gaming/tech field continues to promote gender equality and diversity, we will recruit a more diverse sample to explore the gendered perspective of distributed collaboration through live streaming in the traditionally male-dominated tech space.

6 CONCLUSION

How does emerging novel technology continue to shape people's collaborative practices over distance in more nuanced ways? In this paper, we have focused on the role of live streaming in distributed collaborative software development as shown in indie game development, an end-user driven creative community. We have especially highlighted how live streaming can be used by indie game developers beyond learning and teaching technical skills to support their endeavors to innovate the traditional game development model and the potential challenges they face in this process. In doing so, we uniquely bridge previous research on collaborative software development and live streaming. As global events (e.g., COVID-19) have placed more focus on remote work, we hope that these insights not only extend existing theories of collaboration but also point to future research directions to better support emerging forms of distributed collaboration in various types of end-user driven creative activities.

ACKNOWLEDGMENTS

We thank our participants and the anonymous reviewers. We also thank Jake Armstrong for data collection. This work was supported by the National Science Foundation under award 1849718.

REFERENCES

- [1] 2020. Science & Technology. <https://www.twitch.tv/directory/game/Science%20%26%20Technology>
- [2] 2020. Twitch Developers. <https://dev.twitch.tv/>
- [3] Ban Al-Ani, Matthew J Bietz, Yi Wang, Erik Trainer, Benjamin Koehne, Sabrina Marcza, David Redmiles, and Rafael Prikladnicki. 2013. Globally distributed system developers: their trust expectations and processes. In *Proc. of the 2013 Conf. on Computer supported cooperative work*. 563–574.
- [4] Timothy Aldridge. 2019. Twitch for Game Developers. <https://blog.twitch.tv/en/2019/03/29/twitch-for-game-developers-b664163bed65/> (2019).

- [5] Morgan G Ames, Jeffrey Bardzell, Shaowen Bardzell, Silvia Lindtner, David A Mellis, and Daniela K Rosner. 2014. Making cultures: empowerment, participation, and democracy-or not? In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 1087–1092.
- [6] Gabriela Avram, Liam Bannon, John Bowers, Anne Sheehan, and Daniel K Sullivan. 2009. Bridging, patching and keeping the work flowing: Defect resolution in distributed software development. *Computer Supported Cooperative Work (CSCW)* 18, 5-6 (2009), 477.
- [7] Shaowen Bardzell, Jeffrey Bardzell, and Sarah Ng. 2017. Supporting cultures of making: Technology, policy, visions, and myths. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 6523–6535.
- [8] Stephanie Barish. 2018. The Independent Game Development Boom: Interview with Stephanie Barish, CEO of IndieCade. New York Film Academy.
- [9] Mattias Bergström, Peter Törlind, et al. 2007. Examining creative collaboration in distributed and co-located design teams. In *DS 42: Proc. of ICED 2007, the 16th Int'l Conf. on Engineering Design, Paris, France, 28.-31.07. 2007*. 327–328.
- [10] Robert P Biuk-Aghai and Simeon J Simoff. 2004. Patterns of virtual collaboration in online collaboration systems. In *Proceedings of the IASTED International Conference on Knowledge Sharing and Collaborative Engineering, St. Thomas, USVI, November*. Citeseer, 22–24.
- [11] Pernille Bjorn, Jakob Bardram, Gabriela Avram, Liam Bannon, Alexander Boden, David Redmiles, Cleidson de Souza, and Volker Wulf. 2014. Global software development in a CSCW perspective. In *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing*. 301–304.
- [12] Pernille Bjørn, Morten Esbensen, Rasmus Eskild Jensen, and Stina Matthiesen. 2014. Does distance still matter? Revisiting the CSCW fundamentals on distributed collaboration. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 5 (2014), 1–26.
- [13] Greg Borchers. 2003. The software engineering impacts of cultural factors on multi-cultural software development teams. In *25th International Conference on Software Engineering, 2003. Proceedings*. IEEE, 540–545.
- [14] Maurizio Borghi. 2011. Chasing copyright infringement in the streaming landscape. *International Review of Intellectual Property and Competition Law* 42, 3 (2011).
- [15] Kathy Charmaz. 2006. *Constructing grounded theory: A practical guide through qualitative analysis*. sage.
- [16] Chia-Chen Chen and Yi-Chen Lin. 2018. What drives live-stream usage intention? The perspectives of flow, entertainment, social interaction, and endorsement. *Telematics and Informatics* 35, 1 (2018), 293–303.
- [17] Di Chen, Dustin Freeman, and Ravin Balakrishnan. 2019. Integrating Multimedia Tools to Enrich Interactions in Live Streaming for Language Learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [18] Yan Chen, Walter S Lasecki, and Tao Dong. 2020. Towards Supporting Programming Education at Scale via Live Streaming. *arXiv preprint arXiv:2010.15015* (2020).
- [19] 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.
- [20] Xusen Cheng, Terry Nolan, and Linda Macaulay. 2013. Don't give up the community: a viewpoint of trust development in online collaboration. *Information Technology & People* (2013).
- [21] J. Clement. 2021. Distribution of game developers worldwide from 2014 to 2021, by gender. <https://www.statista.com/statistics/453634/game-developer-gender-distribution-worldwide/> (2021).
- [22] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*. 1277–1286.
- [23] Daniela Damian, James Chisan, Polly Allen, and Brian Corrie. 2003. Awareness meets requirements management: awareness needs in global software development. In *Proc. of the Int'l Workshop on Global Software Development, International Conference on Software Engineering (ICSE 2003)*.
- [24] Danielle D'Amour, Marcela Ferrada-Videla, Leticia San Martin Rodriguez, and Marie-Dominique Beaulieu. 2005. The conceptual basis for interprofessional collaboration: core concepts and theoretical frameworks. *Journal of interprofessional care* 19, sup1 (2005), 116–131.
- [25] Travis Faas, Lynn Dombrowski, Erin Brady, and Andrew Miller. 2019. Looking for group: Live streaming programming for small audiences. In *International Conference on Information*. Springer, 117–123.
- [26] Travis Faas, Lynn Dombrowski, Alyson Young, and Andrew D Miller. 2018. Watch me code: Programming mentorship communities on twitch. tv. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–18.
- [27] CLAIRE M Fagin. 1992. Collaboration between nurses and physicians: No longer a choice. *Academic medicine: journal of the Association of American Medical Colleges* 67, 5 (1992), 295–303.
- [28] Peter Farago. 2012. Indie Game Makers Dominate iOS and Android. *Flurry. com [online]. [cit. 2013-01-31]. Dostupné z: http://blog.flurry.com/bid/82758/Indie-Game-Makers-Dominate-iOS-and-Android* (2012).

[29] Umer Farooq, John M Carroll, and Craig H Ganoe. 2007. Supporting creativity with awareness in distributed collaboration. In *Proc. of the 2007 Int'l ACM Conf. on Supporting group work*. 31–40.

[30] Casey Fiesler and Amy Bruckman. 2014. Copyright terms in online creative communities. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 2551–2556.

[31] Casey Fiesler, Jessica L Feuston, and Amy S Bruckman. 2015. Understanding copyright law in online creative communities. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. 116–129.

[32] Stephanie J Fisher and Alison Harvey. 2013. Intervention for inclusivity: Gender politics and indie game development. *Loading...* 7, 11 (2013).

[33] C Ailie Fraser, Mira Dontcheva, Joy O Kim, and Scott Klemmer. 2019. How live streaming does (and doesn't) change creative practices. *interactions* 27, 1 (2019), 46–51.

[34] 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.

[35] Guo Freeman, Jeffrey Bardzell, Shaowen Bardzell, and Nathan McNeese. 2020. Mitigating Exploitation: Indie Game Developers' Reconfigurations of Labor in Technology. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–23.

[36] Guo Freeman, Jeffrey Bardzell, Shaowen Bardzell, and Nathan J McNeese. 2019. The Innovation Ecology: Collaborative Information, Community Support, and Policy in a Creative Technology Community. In *International Conference on Information*. Springer, 614–624.

[37] Guo Freeman and Nathan J McNeese. 2019. Exploring Indie Game Development: Team Practices and Social Experiences in A Creativity-Centric Technology Community. *Computer Supported Cooperative Work (CSCW)* 28, 3-4 (2019), 723–748.

[38] Guo Freeman and Nathan J McNeese. 2021. A Tale of Creativity and Struggles: Team Practices for Bottom-Up Innovation in Virtual Game Jams. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–27.

[39] Guo Freeman and Donghee Yvette Wohm. 2020. Streaming your Identity: Navigating the Presentation of Gender and Sexuality through Live Streaming. *Computer Supported Cooperative Work (CSCW)* 29, 6 (2020), 795–825.

[40] Verena Fuchsberger, Silvia Lindtner, Martin Murer, and Manfred Tscheligi. 2015. Rethinking Technology Innovation: Factories, Fabrication & Design Research.(2015).

[41] Verena Fuchsberger, Martin Murer, Manfred Tscheligi, Silvia Lindtner, Andreas Reiter, Shaowen Bardzell, Jeffrey Bardzell, and Pernille Björn. 2015. The future of making: Where industrial and personal fabrication meet. *Aarhus Series on Human Centered Computing* 1, 1 (2015), 4.

[42] Maria B Garda and Paweł Grabarczyk. 2016. Is every indie game independent? Towards the concept of independent game. *Game Studies* 16, 1 (2016).

[43] Lucy L Gilson, M Travis Maynard, Nicole C Jones Young, Matti Virtiainen, and Marko Hakonen. 2015. Virtual teams research: 10 years, 10 themes, and 10 opportunities. *Journal of management* 41, 5 (2015), 1313–1337.

[44] Kishonna L Gray. 2017. They're just too urban": Black gamers streaming on Twitch. *Digital sociologies* 1 (2017), 355–368.

[45] Orlando Guevara-Villalobos. 2011. Cultures of independent game production: Examining the relationship between community and labour. In *DiGRA Conference*.

[46] Carl Gutwin, Reagan Penner, and Kevin Schneider. 2004. Group awareness in distributed software development. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. 72–81.

[47] Carl Gutwin, Kevin Schneider, David Paquette, and Reagan Penner. 2004. Supporting group awareness in distributed software development. In *IFIP International Conference on Engineering for Human-Computer Interaction*. Springer, 383–397.

[48] Emilia Guzman and Bernd Bruegge. 2013. Towards emotional awareness in software development teams. In *Proceedings of the 2013 9th joint meeting on foundations of software engineering*. 671–674.

[49] Lassi Haaranen. 2017. Programming as a performance: Live-streaming and its implications for computer science education. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*. 353–358.

[50] Julia Katherine Haines. 2014. Emerging innovation: the global expansion of seed accelerators. In *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing*. 57–60.

[51] 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.

[52] Adrián Hernández-López, Ricardo Colomo-Palacios, Ángel García-Crespo, and Pedro Soto-Acosta. 2010. Trust building process for global software development teams: a review from the literature. *International Journal of Knowledge*

Society Research (IJKSR) 1, 1 (2010), 65–82.

[53] Zorah Hilvert-Bruce, James T Neill, Max Sjöblom, and Juho Hamari. 2018. Social motivations of live-streaming viewer engagement on Twitch. *Computers in Human Behavior* 84 (2018), 58–67.

[54] Mingyao Hu and Sohail S Chaudhry. 2020. Enhancing consumer engagement in e-commerce live streaming via relational bonds. *Internet Research* (2020).

[55] Halyan Huang and Eileen M Trauth. 2008. Cultural influences on temporal separation and coordination in globally distributed software development. *ICIS 2008 Proceedings* (2008), 134.

[56] Influencer Marketing Hub. 2020. 40 Useful Twitch Stats for Influencer Marketing Managers. <https://influencermarketinghub.com/twitch-stats/> (2020).

[57] Syed Roohullah Jan, Faheem Dad, Nouman Amin, Abdul Hameed, and Syed Saad Ali Shah. 2016. Issues in global software development (communication, coordination and trust) a critical review. *training* 6, 7 (2016), 8.

[58] Chris Jensen and Walt Scacchi. 2005. Collaboration, leadership, control, and conflict negotiation and the netbeans.org open source software development community. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. IEEE, 196b–196b.

[59] Justin P Johnson. 2006. Collaboration, peer review and open source software. *Information Economics and Policy* 18, 4 (2006), 477–497.

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

[61] Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira Jr, and Chedy Raïssi. 2012. Watch me playing, i am a professional: a first study on video game live streaming. In *Proceedings of the 21st international conference on world wide web*. 1181–1188.

[62] Lucas Kempe-Cook, Stephen Tsung-Han Sher, and Norman Makoto Su. 2019. Behind the Voices: The Practice and Challenges of Esports Casters. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[63] Jan Kratzer, Roger Th AJ Leenders, and Jo ML van Engelen. 2005. Keeping virtual R&D teams creative. *Research-Technology Management* 48, 2 (2005), 13–16.

[64] 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*. 1571–1576.

[65] Lingyuan Li, Divine Maloney, and Guo Freeman. 2021. Collaboration, Dedication, and Social Pressure: A Comparative Analysis of Virtual and Face-to-Face Game Jams. In *Proceedings of the Annual Hawaii International Conference on System Sciences*.

[66] Lingyuan Li, Jirassaya Uttarapong, Guo Freeman, and Donghee Yvette Wohn. 2020. Spontaneous, Yet Studioes: Esports Commentators' Live Performance and Self-Presentation Practices. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–25.

[67] Xigen Li and Nico Nergadze. 2009. Deterrence effect of four legal and extralegal factors on online copyright infringement. *Journal of Computer-Mediated Communication* 14, 2 (2009), 307–327.

[68] Yao Li, Yubo Kou, Je Seok Lee, and Alfred Kobsa. 2018. Tell me before you stream me: Managing information disclosure in video game live streaming. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–18.

[69] Silvia Lindtner, Shaowen Bardzell, and Jeffrey Bardzell. 2016. Reconstituting the utopian vision of making: HCI after technosolutionism. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1390–1402.

[70] Silvia Lindtner, Garnet D Hertz, and Paul Dourish. 2014. Emerging sites of HCI innovation: hackerspaces, hardware startups & incubators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 439–448.

[71] Nadav Lipkin. 2013. Examining Indie's Independence: The meaning of "Indie" Games, the politics of production, and mainstream cooptation. *Loading...* 7, 11 (2013).

[72] Nadav D Lipkin. 2019. The Indiedocalypse: The Political-Economy of Independent Game Development Labor in Contemporary Indie Markets. *Game Studies* 19, 2 (2019).

[73] 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.

[74] Zhicong Lu, Haijun Xia, Seongkook Heo, and Daniel Wigdor. 2018. You watch, you give, and you engage: a study of live streaming practices in China. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.

[75] Marcelo Marinho, Alexandre Luna, and Sarah Beecham. 2018. Global software development: practices for cultural differences. In *International Conference on Product-Focused Software Process Improvement*. Springer, 299–317.

[76] Stina Matthiesen, Pernille Bjørn, and Lise Møller Petersen. 2014. "Figure out how to code with the hands of others" recognizing cultural blind spots in global software development. In *Proceedings of the 17th ACM conference on Computer*

supported cooperative work & social computing, 1107–1119.

- [77] Christian McCrea. 2013. Australian Video Games: The Collapse and Reconstruction of an Industry. In *Gaming Globally*. Springer, 203–207.
- [78] Nora McDonald and Sean Goggins. 2013. Performance and participation in open source software on github. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 139–144.
- [79] Ines Mergel. 2015. Open collaboration in the public sector: The case of social coding on GitHub. *Government Information Quarterly* 32, 4 (2015), 464–472.
- [80] Audris Mockus, Roy T Fielding, and James D Herbsleb. 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 11, 3 (2002), 309–346.
- [81] Kumiyo Nakakoji, Kazuaki Yamada, and Elisa Giaccardi. 2005. Understanding the nature of collaboration in open-source software development. In *12th Asia-Pacific Software Engineering Conference (APSEC'05)*. IEEE, 8–pp.
- [82] Azadeh Nematzadeh, Giovanni Luca Ciampaglia, Yong-Yeol Ahn, and Alessandro Flammini. 2019. Information overload in group communication: From conversation to cacophony in the twitch chat. *Royal Society open science* 6, 10 (2019), 191412.
- [83] Brian Nicholson and Sundeep Sahay. 2001. Some political and cultural issues in the globalisation of software development: case experience from Britain and India. *Information and organization* 11, 1 (2001), 25–43.
- [84] Gary M Olson and Judith S Olson. 2000. Distance matters. *Human-computer interaction* 15, 2-3 (2000), 139–178.
- [85] Judith S Olson, Erik C Hofer, Nathan Bos, Ann Zimmerman, Gary M Olson, Daniel Cooney, and Ixchel Faniel. 2008. A theory of remote scientific collaboration. *Scientific collaboration on the internet* (2008), 73–97.
- [86] Casey O'Donnell. 2012. The North American game industry. *The video game industry: Formation, present state, and future* (2012), 99–115.
- [87] Felan Parker and Matthew E Perks. 2021. Streaming ambivalence: Livestreaming and indie game development. *Convergence* 27, 6 (2021), 1735–1752.
- [88] Felan Parker, Jennifer R Whitson, and Bart Simon. 2018. Megabooth: The cultural intermediation of indie games. *New Media & Society* 20, 5 (2018), 1953–1972.
- [89] 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*. 4863–4874.
- [90] Elizabeth B-N Sanders and Pieter Jan Stappers. 2008. Co-creation and the new landscapes of design. *Co-design* 4, 1 (2008), 5–18.
- [91] Walt Scacchi. 2010. Collaboration practices and affordances in free/open source software development. In *Collaborative software engineering*. Springer, 307–327.
- [92] Joseph Seering, Robert Kraut, and Laura Dabbish. 2017. 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*. 111–125.
- [93] Joseph Seering, Tony Wang, Jina Yoon, and Geoff Kaufman. 2019. Moderator engagement and community development in the age of algorithms. *New Media & Society* 21, 7 (2019), 1417–1443.
- [94] Jeff T Sheng and Sanjay R Kairam. 2020. From Virtual Strangers to IRL Friends: Relationship Development in Livestreaming Communities on Twitch. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–34.
- [95] Bart Simon. 2013. Indie Eh? Some Kind of Game Studies. *Loading...* 7, 11 (2013).
- [96] Anne-Marie Søderberg, Sundar Krishna, and Pernille Bjørn. 2013. Global software development: commitment, trust and cultural sensitivity in strategic partnerships. *Journal of International Management* 19, 4 (2013), 347–361.
- [97] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*. 1379–1392.
- [98] Anselm L Strauss. 1987. *Qualitative analysis for social scientists*. Cambridge university press.
- [99] Keith Stuart. 2012. Us and the game industry—How indie games are the new counter-culture. *The Guardian* (2012).
- [100] Yanhui Su, Per Backlund, and Henrik Engström. 2020. Business intelligence challenges for independent game publishing. *International Journal of Computer Games Technology* 2020 (2020).
- [101] Dean Takahashi. 2016. Twitch's research shows that live streaming leads to better game sales. <https://venturebeat.com/2016/07/13/twitchs-research-shows-that-livestreaming-leads-to-better-game-sales/>
- [102] Joshua G Tanenbaum, Amanda M Williams, Audrey Desjardins, and Karen Tanenbaum. 2013. Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2603–2612.
- [103] TL Taylor. 2018. *Watch me play: Twitch and the rise of game live streaming*. Princeton University Press.
- [104] Grand view research. 2021. Video Streaming Market Size Share Report, 2021-2028. <https://www.grandviewresearch.com/industry-analysis/video-streaming-market> (2021).

- [105] Yi Wang and David Redmiles. 2016. Cheap talk, cooperation, and trust in global software engineering. *Empirical Software Engineering* 21, 6 (2016), 2233–2267.
- [106] Doris Wells-Papanek and Luigi A Pecoraro. 2017. Co-Work, Co-CREATE, Co-Innovate: It's the Future. *Design Management Review* 28, 3 (2017), 42–48.
- [107] Etienne Wenger. 1999. *Communities of practice: Learning, meaning, and identity*. Cambridge university press.
- [108] Emma Westcott. 2013. Independent game development as craft. *Loading... The Journal of the Canadian Game Studies Association* 7, 11 (2013), 78–91.
- [109] Donghee Yvette Wohn. 2019. Volunteer moderators in twitch micro communities: How they get involved, the roles they play, and the emotional labor they experience. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [110] Donghee Yvette Wohn and Guo Freeman. 2020. Audience management practices of live streamers on Twitch. In *ACM International Conference on Interactive Media Experiences*. 106–116.
- [111] 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*. 1–13.
- [112] Jamie Woodcock and Mark R Johnson. 2019. The affective labor and performance of live streaming on Twitch. *tv. Television & New Media* 20, 8 (2019), 813–823.
- [113] Anna Xambó, Robin Laney, and Chris Dobbyn. 2010. TOUCHtr4ck: democratic collaborative music. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. 309–312.
- [114] 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*. 1–12.
- [115] 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*. 55–60.
- [116] Luyin Zhao and Fadi P Deek. 2004. User collaboration in open source software development. *Electronic Markets* 14, 2 (2004), 89–103.
- [117] Jilei Zhou, Jing Zhou, Ying Ding, and Hansheng Wang. 2019. The magic of danmaku: A social interaction perspective of gift sending on live streaming platforms. *Electronic Commerce Research and Applications* 34 (2019), 100815.
- [118] Zhenhui Zhu, Zhi Yang, and Yafei Dai. 2017. Understanding the gift-sending interaction on live-streaming video websites. In *International Conference on social computing and social media*. Springer, 274–285.

Received April 2021; revised November 2021; accepted March 2022