CrowdChess: A System to Investigate Shared Game Control in Live-Streams

Pascal Lessel
DFKI GmbH
Saarland Informatics Campus
pascal.lessel@dfki.de

Alexander Vielhauer
Saarland University
Saarland Informatics Campus
s9alviel@stud.uni-saarland.de

Antonio Krüger
DFKI GmbH
Saarland Informatics Campus
krueger@dfki.de

ABSTRACT
Recently, gaming live-streams appeared in which the audience interacts without a streamer. These settings sometimes allow the audience to select from a set of input aggregators to mediate how individual contributions are aggregated. We developed *CrowdChess*, a system which is optimized for the live-streaming context: multiple viewers play chess together in a single game instance against an AI; they suggest moves and further select one of six aggregators to mediate their contributions. In contrast to existing approaches, *CrowdChess* allows reasoning about the quality of individual and aggregator contributions, and thus it serves as a test bed for how effectively an audience interacts in such settings. We conducted a study in a live-streaming setup (n=12) and contribute insights on the player perception of audience-driven games and show how aggregators in such a small-scale setting are used, for example, the tendency to use group-based aggregators instead of empowering an individual.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
Wisdom of crowds; input aggregation; co-presence; self-administration; audience control; audience-driven games

INTRODUCTION
User-generated live-streaming of games [6, 27] has become popular in recent years. Twitch, one of the major live-streaming platforms, was amongst the top five internet traffic generators of 2014 in the US [4], and is in the 20 top sites in the US according to Alexa traffic rank [1]. Streamers can reach many viewers in parallel, who not only watch them playing in, for example, competitive e-sport matches [3] or games for entertainment purposes ("Let's Plays") [27], but also make use of the social component of these live-streaming platforms: a real-time chat is offered that allows the viewers to chat with each other and with the streamer. While in the beginning of live-streaming this feature was used only for communication, recently it has provided viewers more influence options: games appeared that enable the audience to alter what happens in the game by entering chat commands (e.g. ChoiceChamber [28]); some live-streaming formats, for example interactive pen&paper role-playing sessions, allow the audience to alter the story by using the chat (as done for example by Rocket Beans TV [12]), and specific streams, such as "Twitch Plays Pokémon" (TPP) [9, 12, 18], allow the audience to share control over the game and play together without a streamer. Especially the latter has sparked much interest and led to a new experience on live-streaming platforms that has not yet received much scientific attention.

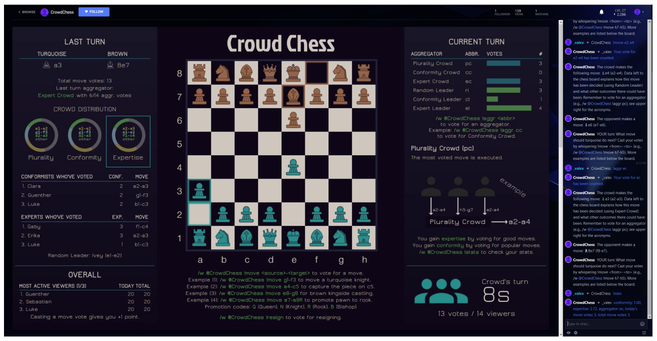


Figure 1. A screenshot of *CrowdChess* running on the live-streaming platform Beam (today called "Mixer"). Left: our video stream. Right: the channel's chat.

How to handle individual contributions is an important question in all the aforementioned examples: either all individual contributions are used and, for example, interpreted as game commands (such as in the beginning of the TPP stream) or the individual contributions are aggregated first. For the latter, polls based on a plurality voting scheme are commonly used in streams today. However, it is questionable whether this kind of aggregation is optimal, especially as investigations exist that show different approaches to input aggregation in the domain of crowd-sourcing that are more effective than a majority/plurality vote [10, 11]. First investigations in the live-streaming domain indicated that an audience which has the option to change input aggregators also selects aggregators other than majority/plurality aggregators [12]. Additionally, an analysis of the TPP phenomenon also indicates that an audience is not uniform in what they expect of such shared game control settings, and sometimes a suboptimal aggregator was selected for entertainment reasons [18].

The games used in these settings are often too complex to objectively reason whether an individual user contribution is beneficial towards the game's goal. Even though specific actions could in theory be rated (e.g., going left is faster), contextual factors and multiple sub-goals in games make a metric for assessing individual actions too complex or even impossible (e.g., in the given example, going right might earn the player an item that might help later). This makes it difficult to reason about interactions in a shared game control setting. With this paper we follow two goals: first, we scientifically investigate such a setting in more detail, and second, we want to be able to reason about the quality of individual and aggregator decisions. Towards these goals, we developed *CrowdChess*, a chess game that is designed for the live-streaming context (see Figure 1) and for audience control. *CrowdChess* should serve as a test bed for shared game control settings. We evaluated *CrowdChess* on the live-streaming platform Beam[1] with twelve participants and found that an expertise voting scheme is favored over a plurality voting scheme, that participants did not enable individual experts to decide alone and that even under time pressure, in small-scale settings, they still communicate over the chat to come to a group decision.

RELATED WORK

Gaming live-streams are relevant for research as they attract many users [6, 27] and findings here relate to several other research areas, for example, Social TV [2], crowd-sourcing [11, 29] and computer-mediated communication [30].

The current body of research mainly focuses on live-streaming usage data on these platforms [8, 21, 22] or considers the social aspects of streamers and their audience. An example of the latter is the work of Hamilton et al. [6]. The authors investigated Twitch streams under the aspect of virtual third places and found that the integrative options offered by the live-chat attract viewers, especially as this is not possible in television. The authors also elaborate on how audience integration looks today, based on interviews with several viewers and streamers. This was also investigated in the first case study of Lessel et al. [12] by analyzing a popular format on Twitch which attracts several thousand viewers regularly, and some overlap with Hamiliton et al.'s findings was reported: besides the fact that showing individual viewer contributions (e.g., viewers' self-made art) in streams is a common theme, plurality polls are also often used to find out what most of the audience wants. In another work, Lessel et al. [13] reported problems when viewer-streamer interactions happen on live-streaming platforms: a) a technological lag leading to asynchrony between chat messages entered and video content shown; b) information overload in channels where more than 150 people are active chatters and c) conveying historical information to newly joining viewers. The authors created a system for the digital card game *Hearthstone* in which they target these issues with the goal to improve the communication between viewers and streamers. They offered a direct interaction option on the video stream, and aggregations of individual viewer opinions were used to reduce the information overload. For the latter, they follow the "ballot box communication" concept [32] but also only use aggregations based on a majority scheme. The direct interaction was perceived well, but is not easy to achieve with the current live-streaming platforms. This hints that input aggregation is necessary in these settings (especially in larger streaming channels) and that simple majority/plurality voting schemes are usually used.

Audiences that interact as a group were already investigated in other domains (e.g. [20]), but investigations have also appeared recently in the live-streaming domain: "Twitch Plays Pokémon" (TPP) had the idea that the audience itself plays a game, instead of simply watching a person playing it. Every Twitch viewer could enter chat messages that were interpreted as game commands. This experiment attracted large numbers of viewers (1.1 million people entered 122 million commands, with 121,000 playing in parallel [23]). The creator recognized that forwarding every viewer command made it hard to finish the game and added an aggregator based on a plurality voting scheme, i.e., only the command entered by the largest number of participants in a time-frame was forwarded. When this aggregator was active was controlled by the audience. This feature of self-administration was frequently used [9]. The first game was finished after 16 days. Ramirez et al. [25] conclude that TPP is not comparable to the "Infinite Monkey Theorem" [31], i.e., it was not the case that the audience finished the game simply because they entered so many commands. Margel also considered TPP and reports explanations for when the audience switched to the plurality voting scheme [18], namely, when situations became too difficult, and not overcoming the obstacle would reduce the entertainment value of the stream. In their second case study, Lessel et al. also investigated TPP with a custom setup, in which several input aggregators were offered [12]. In a lab study they found that there are different reasons when specific aggregators were selected by the audience. They report that viewers still had the feeling of making no progress in the game, but the authors did not further elaborate on whether the audience objectively made progress; therefore, it is currently unclear whether the audience can pick the best input aggregator in a given situation in terms of effectiveness. With *CrowdChess* we want to provide further insights into this question.

Input aggregation is a topic that is investigated in the context of crowd-sourcing: work such as [7, 10, 11, 15] has investigated aggregation schemes for different contexts in which sometimes every input entered is used, without any aggregation, similar to the case of TPP. Other aggregators are used that build on plurality or majority voting schemes (e.g., "Vote" in [11]), weighted plurality/majority voting schemes with weights either based on the expertise of crowd-workers (which is adapted based on task performance [14]) or conformity, i.e., how similar the crowd-worker is to other workers (e.g., based on the idea of "Leader" in [11]). Furthermore, sophisticated algorithms exist that select an input without aggregating. Here, it is not the inputs that are mediated but the selection of one input source. Such approaches have been shown as relevant in the live-streaming context as well [12]. We consider these approaches to inform our selection of input aggregators and will elaborate on them in the next section.

[1]Beam has recently been rebranded to "Mixer": https://mixer.com/, last accessed: 31/07/2017.

CROWD CHESS

For a test bed that allows reasoning about whether an audience of gaming live-streams can effectively self-administrate, we needed a game which allows evaluations on the effectiveness of actions and is still interesting in the context of live-streaming platforms. As games usually have multiple goals, evaluating the available game actions in terms of whether they are helpful towards these goals is not trivial. For example, the game *Pokémon*, which was investigated in [12], allows various actions that cannot easily be judged as the goals the players follow might be different (e.g., catching a new Pokémon, leveling one up, defeating a certain NSC, exploring the optional area X before Y, etc.), and there is no metric that defines exactly how to calculate effectiveness here. In contrast, we decided to use the game chess. Chess software today allows for judging moves and therefore allows comparisons. In addition, the game itself has a complexity that makes playing it interesting enough to attract viewers on live-streaming platforms (e.g., on Twitch several chess channels exist that attract several hundred viewers on average[2]).

Live-streaming challenges

Gaming live-streams that aim for integrating the audience pose different challenges. One challenge is how to enable the audience to enter game inputs. Even though platforms such as Beam allow channel owners to customize their channel page with HTML input elements (such as buttons) and to programmatically listen to viewer interactions, these options are still limited. For example, it is not possible to provide a virtual representation of the chess board to allow viewers to suggest moves via direct interactions, similar to other chess games, while providing them with feedback. Using external web pages (as used in [13]) is potentially not helpful, as there is (to our knowledge) no analysis yet of how open viewers of live-streams are to switch to these during a stream. Another challenge, especially on Twitch and YouTube, is a lag [13, 33] that causes an offset between entered chat messages (which are presented for everyone in nearly real time) and the video shown in the stream (which in relation to the source is already outdated by 15-45 seconds and varies for every viewer). This makes it difficult to relate chat messages to the situation in the stream. In settings in which audience input is interpreted as game commands, this is especially problematic and needs to be considered in the underlying game concepts. Finally, using the chat (as it works in real time, in contrast to the video stream) to provide information to game situations and individual feedback poses the challenge that platforms have limitations on how many messages can be sent in a minute. Although white-listing options might be granted by platform vendors for interesting channel concepts, this cannot be assumed unconditionally. Even a white-listed channel might still be too limited to provide individuals with enough feedback, depending on the channel size. Exceeding the limits can, depending on the platform, mean that the messages are simply not sent to other viewers, or in the worst case that the sending account is banned for multiple hours.

[2]For example https://www.twitch.tv/chess, last accessed: 31/07/2017.

System overview

With respect to these challenges we created *CrowdChess*. It consists of four components: the chess engine, the aggregator system, the live-streaming integration and the user interface shown in the stream. *CrowdChess* is designed for audience vs. AI, but it can easily be adapted to allow for streamer vs. audience or audience vs. audience matches. For this first investigation, we wanted to rule out effects that might be introduced by a streamer that hosts the game (e.g., encouraging aggregator scheme changes), but we see such an investigation as an interesting next step. As chess is already partitioned into turns and does not require fast interactions, it helps to lower the impact of the lag issue mentioned before.

Chess engine and AI

We use Stockfish [26], one of the strongest open-source chess engines available, to evaluate the individual and aggregator-derived move suggestions. Besides using these results as data points for our study, we use them to inform the expertise-based aggregators (see below). This engine is too strong as an opponent to be a motivating challenge in live-streaming settings. Therefore, we use some of the engine-offered options to limit the engine's power (such as the maximal search depth for possible moves). *CrowdChess* offers 16 AI levels; the level increases (decreases) by one when the audience wins (loses). While the upper levels utilize the full capabilities of the engine, the lower levels can be easily beaten even by chess beginners.

Aggregator system

CrowdChess allows the audience to alter the way individual move suggestions are aggregated by voting for an aggregator scheme. Six different schemes are offered (and are partially based on Lessel et al.'s investigation of aggregators in their custom Twitch Plays Pokémon setup [12]) and we distinguish between two classes: those that aggregate the input (named "crowd") and those that select an individual based on certain attributes and use their move suggestion (named "leader").

- **Plurality Crowd (PC)**: A plurality vote, i.e., the move suggested by most viewers is executed. This is a common aggregator in live-streaming polls and was also available in (for example) "Twitch Plays Pokémon". In [12] the authors showed that other aggregators were used more often.
- **Expert Crowd (EC)**: A weighted plurality vote with weights based on individual expertise levels. How the expertise is calculated is not disclosed to the viewers; they can only see the rating, which is updated for every move suggestion given. Every viewer starts with an expertise of 1. A given move suggestion is evaluated by the chess engine in relation to the current board situation. This increases (decreases) the viewer's expertise by up to 2 (-2) points depending on how good (bad) the suggestion was. For this, it is considered whether the suggestion leads to the opponent's mate, to one's own mate (although other moves would have been possible), whether the viewer has missed mate in 1/2/3 turn(s) and finally, how the suggestion score relates to the score of the best possible move in the given board situation[3]. Using expertise is relevant in many

[3]For more details on the expertise and conformity algorithms see the appendix.

crowd-sourcing approaches [24] and is also interesting for our consideration. An audience having the goal to beat the chess AI should empower experts, as they are more likely to provide good move suggestions. The "expertise weighted vote aggregator" had the highest up-time in [12].

- **Conformity Crowd (CC)**: This aggregator is also a weighted plurality vote where the weights are based on the individual conformity levels. Move suggestions given by viewers are continuously compared to suggestions provided by other viewers in the same turn. Suggesting moves that were also suggested by the majority increases a viewer's conformity by up to 2 (starting at 1), depending on the uniformity among move suggestions. The level is decreased (by up to -2) when suggestions are made which only small portions of (or no one else in) the audience also made. This aggregator follows the idea introduced in Lasecki et al.'s work [11]. As soon as the "expertise weighted vote" aggregator became available, the conformity-based "crowd weighted vote" aggregator had nearly no up-time anymore in [12]. Thus, it is interesting to learn whether the CC aggregator in our setting will be used at all.
- **Random Leader (RL)**: The random leader aggregator randomly selects one of the viewers' move suggestions. Such an aggregator relates to the "Anarchy" aggregator in the original TPP, in which all commands were executed and thus all viewers could contribute something. Using the concept of "Anarchy" is possible, but seems unreasonable for a chess setting, as the chance of doing something "worse" and not being able to reverse this decision through other viewer commands, in contrast to Pokémon, is much higher. RL in contrast ensures that in theory all viewers can participate in the game independent of their conformity/expertise level, but also allows the matches to still be playable. A similar aggregator was introduced in Lasecki et al.'s work [11] and was also offered in [12], but had a low up-time there.
- **Expert Leader (EL)**: In contrast to RL, EL uses the move suggestion of the viewer with the highest expertise level. [12] has shown that the audience empowers individuals in difficult situations. In chess settings, it seems reasonable to offer an aggregator that provides the best player with full control. Similar to EC, empowering the best user seems reasonable, when the goal is to beat the AI.
- **Conformity Leader (CL)**: In contrast to EL, here the conformity level is the selection criterion. CL follows the "Legion Leader" approach of Lasecki et al. [11]. In [12] this kind of aggregator was also available (but no counterpart to EL was available in their setting) and this aggregator was used in difficult situations to provide an individual with complete control. It is interesting to see whether this aggregator is also used when EL is offered in parallel.

The selection of the active aggregator is an ongoing plurality vote. Viewers can switch their vote anytime and the one with the most votes becomes active at the end of a turn. The time left in a turn is always shown to the viewers. In the case of a draw, one of the aggregators is randomly selected. If the active aggregator provides the same value for multiple moves, one of these is also randomly selected.

Live-streaming integration

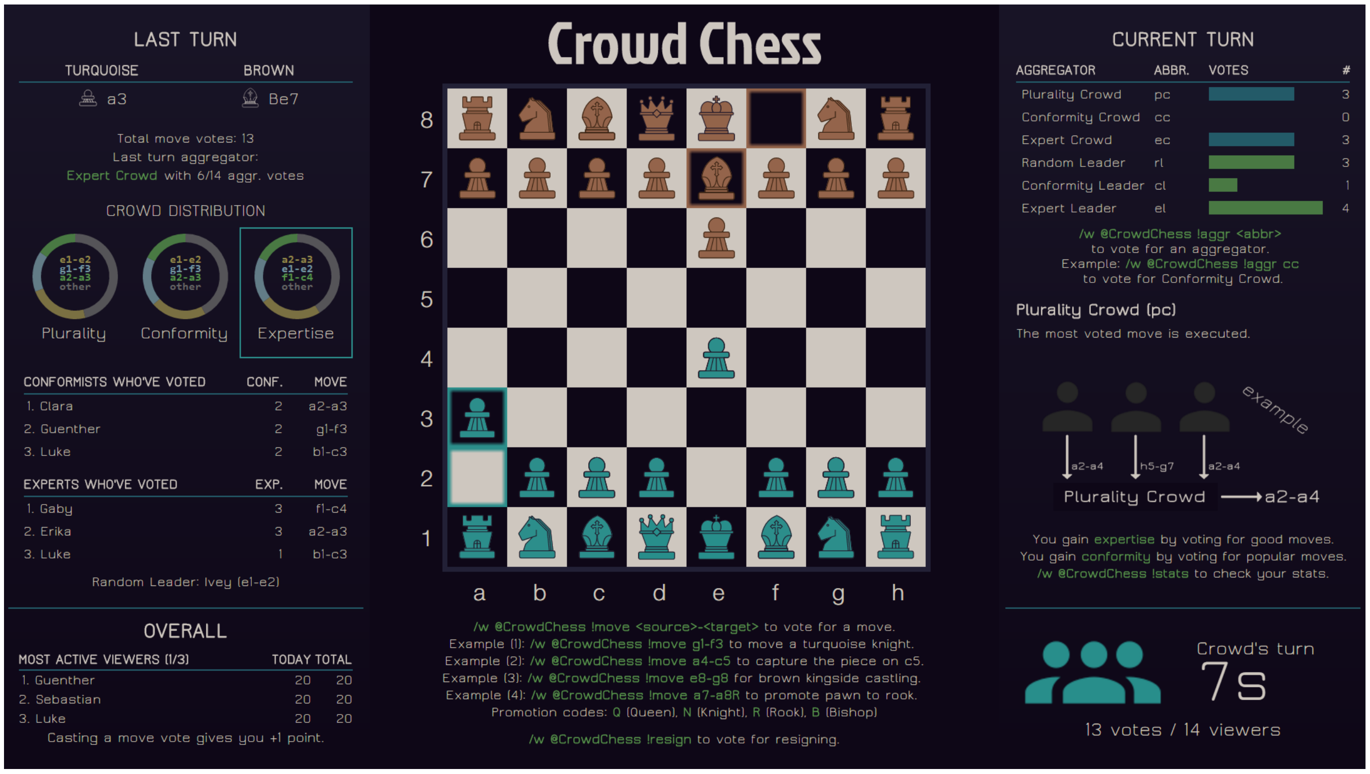
CrowdChess uses a chat bot to inform viewers about their interaction options via the channel's chat. These messages also provide insights into the current game state (e.g., whose turn it is and which piece was moved last). Especially for the aforementioned lag issue, this allows the viewers to think about the next turn, even if the video stream does not yet show the corresponding state visually. Three example messages are:

> *YOUR turn: What move should turquoise do next? Cast your votes by whispering !move <from>-<to> (e.g., /whisper @CrowdChess !move h7-h5).*
>
> *The crowd makes the following move: pawn h4 (h2-h4). Data to the left of the chess board explains how this move has been decided (using Random Leader) and what other outcomes there could have been. Remember to vote for an aggregator (e.g., /whisper @CrowdChess !aggr pc); see upper right for the acronyms.*
>
> *The opponent makes a move: pawn e4 (e2-e4).*

We require viewers to interact with *CrowdChess* via whispers. We did not want viewers—for this first exploration—to influence others merely by disclosing what they want to do, by simply displaying their vote command publicly. Such social effects were already reported by other research [16, 17, 19]; we try to reduce these effects with whispering. These whisper messages are also the primary feedback channel of *CrowdChess*. We use this channel to acknowledge that a viewer has correctly entered a move suggestion, to inform him/her that a move is not possible in the current board situation or that an ill-formed command was entered. If viewers enter commands into the chat instead of whispering, the chat bot removes the entered command from the chat and sends a message informing all users that they need to whisper.

Valid move suggestions follow the form *!move <from field>-<to field>*. This notation is used for all kinds of move suggestions in *CrowdChess* (e.g., capturing other pieces and castling) to simplify the interaction. A user is able to enter multiple move suggestions per turn, but only the last one is considered at the end of a turn. By entering *!aggr <aggregator name>* players can switch their aggregator vote. *!stats* provides the user with his/her current statistics (i.e., his/her conformity/expertise level, which aggregator he/she has currently voted for and the numbers of votes given today and overall). Entering *!resign* is treated as a special move suggestion. If the active aggregator selects this suggestion, the match ends. We implemented a queuing system to cope with the situation of having no white-listing and to avoid exceeding the limits the live-streaming platforms allow.

We use the channel description to give an overview on the command options and the aggregators. Together with the streamed interface of *CrowdChess* (see below) and the chat bot's proactive and reactive messages, viewers have three sources to learn how to use *CrowdChess*. Finally, the game is designed to run unsupervised and continuously (similar to TPP). As soon as a match is over (or no viewer is available anymore), the game restarts with an adapted AI level and thus allows for continuous playing.



LAST TURN

TURQUOISE	BROWN
a3	Be7

Total move votes: 13
Last turn aggregator:
Expert Crowd with 6/14 aggr. votes

CROWD DISTRIBUTION

CONFORMISTS WHO'VE VOTED	CONF.	MOVE
1. Clara	2	a2-a3
2. Guenther	2	g1-f3
3. Luke	2	b1-c3

EXPERTS WHO'VE VOTED	EXP.	MOVE
1. Gaby	3	f1-c4
2. Erika	3	a2-a3
3. Luke	1	b1-c3

Random Leader: Ivey (e1-e2)

OVERALL

MOST ACTIVE VIEWERS (1/3)	TODAY	TOTAL
1. Guenther	20	20
2. Sebastian	20	20
3. Luke	20	20

Casting a move vote gives you +1 point.

/w @CrowdChess !move <source>-<target> to vote for a move.
Example (1): /w @CrowdChess !move g1-f3 to move a turquoise knight.
Example (2): /w @CrowdChess !move a4-c5 to capture the piece on c5.
Example (3): /w @CrowdChess !move e8-g8 for brown kingside castling.
Example (4): /w @CrowdChess !move a7-a8R to promote pawn to rook.
Promotion codes: Q (Queen), N (Knight), R (Rook), B (Bishop)

/w @CrowdChess !resign to vote for resigning.

CURRENT TURN

AGGREGATOR	ABBR.	VOTES	#
Plurality Crowd	pc		3
Conformity Crowd	cc		0
Expert Crowd	ec		3
Random Leader	rl		3
Conformity Leader	cl		1
Expert Leader	el		4

/w @CrowdChess !aggr <abbr>
to vote for an aggregator.
Example: /w @CrowdChess !aggr cc
to vote for Conformity Crowd.

Plurality Crowd (pc)
The most voted move is executed.

You gain expertise by voting for good moves.
You gain conformity by voting for popular moves.
/w @CrowdChess !stats to check your stats.

Crowd's turn
7s
13 votes / 14 viewers

Figure 2. The user interface of *CrowdChess* consists of last turn information (left), the chess board (middle) and the current turn information (right).

User interface

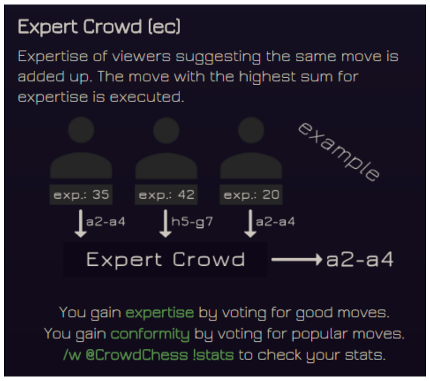
Figure 2 shows the *CrowdChess* user interface consisting of three thematic areas: historical information (left), the chess board (center) and the current information (right).

Historical information: The historical information shows data on the last turn (upper part) and the overall game (lower part). At the top we display the last audience and AI move. Below, we show the aggregator that was active and how many viewers wanted it in relation to all given aggregator votes. Additionally, we provide information on how many move suggestions were entered by the viewers. This area also shows, based on all provided move suggestions, which moves the aggregators' would have selected (the last active aggregator is additionally marked with a cyan box). This should help viewers to make their aggregator vote decisions (e.g., when other aggregators seem to be more suitable for individual goals). We display pie charts for the three crowd aggregators. In each chart we display the top three moves the aggregator outputs in the middle and the relative distribution of each of them in the circle. For EL and CL we display tables showing the top three most conforming/experienced viewers that entered move suggestions for the last turn, together with their current conformity/expertise level and which move they suggested. Finally, we also show which viewer would have been selected randomly by RL together with his/her move suggestion.

The area in the lower part switches every 20 seconds between showing the top nine active viewers (in groups of three), how many matches the audience/AI has won today/overall and the current AI level.

Chess board: In the center the chess board is shown together with the corresponding column letters and row numbers as these need to be entered by the viewers in their move suggestions. On the board we also highlight the last turns made in the color of the audience/the AI. Below the board, examples on how to enter move suggestions are given. It is shown how to do a normal move, how to capture pieces, how to do special moves (castling/promotion) and how to resign the game.

Current information: Information relevant for the current turn is shown on the right side. The upper part displays the aggregators' distribution by showing the different aggregator names (and their abbreviations) and how many viewers have voted for every aggregator. This is followed by a short explanation on how a viewer can switch his/her aggregator vote and a large area providing an example for every aggregator. Every 20 seconds a different aggregator is presented here (see Figure 3). With this element, viewers do not need to read the channel description to understand how aggregators work. We also give information on what expertise and conformity mean in our context, and provide the information that one's own expertise/conformity value can be queried with the *!stats* command. Below this area, we show whether the audience or the AI needs to make a move. When it is the audience's turn, we also show the remaining time. To further mitigate the lag issue, we distinguish between lag and turn time. The lag time denotes a time before the turn timer starts. The lag and turn time can be adjusted, but are fixed at run time (i.e., the audience cannot vote to set up the time by themselves). Below the time we indicate how many viewers are present and how many have entered a move suggestion.


Expert Crowd (ec)

Expertise of viewers suggesting the same move is added up. The move with the highest sum for expertise is executed.

You gain expertise by voting for good moves.
You gain conformity by voting for popular moves.
/w @CrowdChess !stats to check your stats.

Figure 3. The explanation shown for the Expert Crowd aggregator.

STUDY

CrowdChess offers a test bed for various scenarios in the live-streaming context. In an exploratory fashion, we conducted a first study, where we were interested in how it is perceived, how the players use the aggregators and how these perform.

Method

CrowdChess is implemented to work with arbitrary live-streaming platforms. For this first test, we decided to use the platform Beam mainly because the lag is reported to be below one second. Thus, viewers would directly see moves on the streamed video and are not restricted to the chat-bot until the lag time is over. Simply launching a channel on a live-streaming platform was shown to be problematic in several related works that crawled platform data (for example [8, 34]): most of the channels on live-streaming platforms do not attract many viewers in parallel, or any viewers at all. Therefore, we advertised an event in which "you should play chess as part of a group on a live-streaming platform to beat an AI" over Facebook, chess communities and student mailing lists (consisting of computer science, media informatics and psychology students). As prerequisites, we mentioned that players should know the chess rules (but that skill level is irrelevant) and that a Beam account is necessary. During run time it was also possible that Beam users joined who were not explicitly recruited in this way. In general, this kind of advertisement is similar to the usual live-streaming case in which streamers also, for example, announce their streaming times and content via social media platforms. The event ran for 45 minutes. Before the event began, we streamed a slide with information on the event (such as that the channel description provided all necessary commands). Here, we also stated that all communication should be done over the channel's chat and not over tools such as Skype. The turn time was restricted to 60 seconds (and one second lag time), leading to 61 seconds between the opponent move being shown in the chat and the next turn of the opponent. The time restriction was to allow a faster gameplay and ensure that more moves could be carried out in the event. The AI was initially set up with a depth of five (easy) and was then set to ten (rather challenging) after the first match, to let the audience play against an easy and a harder opponent. After the 45 minutes, we provided a link to a questionnaire (which was available in English and German) for the participants via the channel's live-chat that assessed demographics, a self-assessment of one's own skill level, the perception of *CrowdChess* based on statements to be answered on 4-point scales (with the labels *disagree*, *rather disagree*, *rather agree*, *agree*) and some optional questions that allowed for free-text answers. We logged the chat messages, recorded the live-stream and persisted all interactions with our system.

Results

We clustered the results into general usage statistics, qualitative feedback, move suggestion quality and aggregator usage.

General usage statistics

18 registered users visited our channel during the experiment: 13 wrote at least one message (independent of whether it was a chat message or command) and three users remained as spectators for an average of two turns (without doing any interactions before leaving the channel); the other two users visited and left immediately. One user only entered chat messages (this user joined a few seconds before we closed the event). The remaining twelve users all entered at least one move or aggregator change command, i.e., they participated directly in the game (and will be called "players" subsequently). Nine of them used the chat to communicate (and wrote a total of 153 chat messages). The players entered 214 move suggestions (mean M=17.8, standard deviation SD=8.1, median Mdn=21), 24 aggregator commands (M=3.4, SD=2.3, Mdn=2; done by seven players) and 8 stats requests (M=1.6, SD=0.9, Mdn=1; done by five players). Two matches were finished: the first one against AI level five was won by the audience, the match against AI level ten by the AI. Ten players played both matches; two players entered player commands in the second match only. Overall, the audience played 31 chess turns. In match one (13 turns) the average number of move suggestions per turn was 7 (min=4, max=9); in the second match (18 turns) 6.8 (min=3, max=11). Table 1 shows the number of actions per player, their average expertise/conformity value and percentages on how active they were in relation to the number of turns they were on the channel. 75% of the players participated in more than 50% of their witnessed turns.

Twelve participants (7x male, 3x female, 2x no answer; 2x <19, 5x 19-25, 2x 26-32, 2x 33-39 years old, 1x no answer) finished the questionnaire and provided us with their Beam name, showing that all active players did the questionnaire. All but one (Scottish) reported to be German. The Scottish player joined the event by coincidence, while the remaining players heard about it via our advertisement. We let the participants rate their own skill level (see also Table 1) by providing them with statements that indicate different playing skills and they were sorted from lowest to highest. They either selected *"I know the chess rules, but I have not played chess much so far"* (which we denoted with inexperienced in the table) or *"I know the chess rules and I think that I can win against other casual chess players"* (some experience). The next statement *"I know the chess rules and I think that I can win against players that play chess regularly but are not chess club players"* was not selected, indicating that overall our players were not skilled chess players. Furthermore, no one reported playing chess regularly. Only four participants reported consuming live-streams and three had participated in a shared game control setting (here: "Twitch Plays Pokémon").

ID	**Number of actions**				**Avg. expertise value**	**Avg. conformity value**	**Witnessed turns**	**% of witnessed turns in which the user did a player action**	**% of witnessed turns in which the user did any action**	**Skill self-assessment**
	Move	Aggregator	Stats	Chat				*!move* and/or *!aggr*	any command/chatted	
01	27	6	0	8	17	13	31	87%	97%	Some experience
02	21	7	0	2	4	5	31	74%	77%	Inexperienced
03	26	0	3	18	10	11	31	84%	87%	Inexperienced
04	18	1	0	11	13	10	31	61%	65%	Some experience
05	6	2	0	10	2	3	31	26%	39%	Inexperienced
06	15	2	2	13	6	5	31	52%	71%	Some experience
07	22	0	0	0	7	11	31	71%	71%	Inexperienced
08	23	4	1	28	12	9	31	81%	97%	Some experience
09	24	2	1	61	9	7	31	81%	97%	Inexperienced
10	21	0	0	0	6	9	31	68%	68%	Inexperienced
11	8	0	0	2	3	2	23	35%	39%	Inexperienced
12	3	0	1	0	2	2	11	27%	27%	Some experience

Table 1. Player overview. As players were able to join the game later, we added indications on how active they were in relation to their witnessed turns.

Qualitative feedback

Seven participants stated they liked *CrowdChess* (three times agree, four times rather agree), leading to an average of 2.75 on the 4-point scale. Seven participants reported having fun playing it (M=2.6, only asked with a one item question) and four participants stated that it was more fun than playing "normal" chess (M=2.1). The idea of playing chess as a group was liked by half the participants (M=2.5), and five would continue to play *CrowdChess*. Potential aspects that might have impacted the perception of *CrowdChess* were:

- **Time**: No participant reported being familiar with playing chess under time pressure. Eleven participants (rather) disagreed with the statement that the turn time was too long (M=1.3) and ten disagreed with the statement that they had enough time to play and consider the historical information (M=1.3). In the free text questions, six reported that they wanted to participate in every turn and thus had insufficient time to consider all parts of the interface. As this study tried to mimic the live-streaming setting (in which visitors also simply join the channel and are confronted with it directly), we did not explain the user interface beforehand. In this study situation with a clearly communicated time limit of 45 minutes, though, participants might have thought that they needed to provide a move in every turn (which was not necessary) or found it more appealing to enter move suggestions. Concerning time, two participants suggested in the free-text answers to either add an automatic turn time adaptation (by considering how much discussion happens in the chat) or audience-based options to adjust the time. They also reported that they wanted an option to skip the remaining time, if the audience has already decided.
- **User interface**: Seven participants (rather) agreed to the statement that they like the graphical appearance of *CrowdChess* (M=2.4). Five times it was mentioned in the free-text answers that the UI is too overloaded and three times it was reported that the user's monitor (and thus the video stream window) was too small to see all elements properly.
- **Text-based interaction**: Seven participants (rather) agreed to the statement that the interaction via chat messages was acceptable (M=2.4). Four participants explicitly stated that they would rather interact directly with the chess board on the stream. This is in line with the findings of Lessel et al. [13] related to direct interaction. Nine participants (rather) agreed to the statement that they always knew how to enter a move suggestion (M=3.0). 6 syntactical incorrect and 41 invalid (with respect to the chess rules) move commands were entered, but considering who entered those, no relation to the aforementioned question was found. Even though the chat bot provided an error message via whisper in this case, this might have been overlooked.
- **Group feeling**: Two participants reported that they were frustrated because moves were executed that they assessed as worse than their move suggestions. Only five (rather) agreed to the statement that they felt part of the group (M=2.3) and six reported that they thought that their move suggestions mattered (M=2.3). Two participants suggested replacing the historical information with the current move suggestions all players have provided in the current turn to come to a better decision.

We also provided statements on the offered features: Nine participants found the chat bot messages useful (M=2.9) and nine liked to see the current aggregator distribution (M=3). Seeing examples of the aggregators in the stream was liked by seven (M=2.6), the high score list of the most active players by nine (M=2.8) and ten participants liked to see the win/loss ratio against the AI (M=3.3). Seeing which moves the active aggregator has also considered was liked by seven participants (M=2.8); what the other aggregators would have selected by six (M=2.7). Seven (rather) agreed to the statement that they understood the aggregator explanations in the stream (M=2.7), but only four (rather) agreed to the statement that they understood all components of the history (M=2.2). The aspects (time and UI) mentioned above might be explanations for this.

Eight participants were interested in the expertise (M=2.6) but only two in the conformity ratings (M=1.6). None of the participants reported using the stats request command frequently to check his/her expertise (M=1.1) or conformity (M=1.1) (also in line with the actual frequency of the *!stats* command) and only five (one) reported checking the table in the history to see the expertise (conformity) distribution. As an optional

Agg.	% best move suggestion selected	Avg. % of move suggestions better than move selected by aggregator (across all turns)	% worst move suggestion selected
PC	64.5%	13.6%	29%
CC	71%	13.1%	19.4%
EC	71%	11.7%	25.8%
RL	51.6%	24.8%	32.3%
CL	64.5%	17.8%	29%
EL	71%	15%	29%

Table 2. Aggregator performances based on all board positions (31) and all move suggestions (214).

question we asked whether the expertise/conformity values derived by *CrowdChess* seemed plausible; while the three participants that answered this (rather) agreed for conformity (M=3.3), three (of four) participants that answered this for the expertise value rather agreed to that statement (M=2.8).

Move suggestion quality

We first analyzed the 214 move suggestions for how different they were for every turn. We counted how many groups of same move suggestions in a turn were provided and divided this by the number of all suggestions in this turn. Averaged over all turns, a number near 0 would indicate that in most turns the audience would have provided only uniform suggestions; a number near 1 would indicate that the audience only provided different move suggestions in most turns. Both would render most of the aggregators useless, but this was not an issue in our case (M=0.5, SD=0.23, Mdn=0.4). Using the chess engine, we analyzed all 214 move suggestions. Only 40% of the suggested moves would have changed the board situation in favor of the crowd. This is in line with Table 1 and the skill self-assessment. For every board position (31) we aggregated the given move suggestions by using all six aggregators regardless of which one was actually active in the turn. For the random leader aggregator, we used the suggestion of the player that was randomly selected in this turn; for the expert/conformity leaders we utilized the values that the players had at the point of time when the suggestion was made. Based on this, in 25 of 31 turns (81%) at least one aggregator provided the best result amongst the user suggestions (i.e., it selected the move suggestion that received the highest engine score). Table 2 shows the performances of the aggregators. The table represents the case without filtering situations in which the worst was the same as the best move suggestion (e.g., all move suggestions were equivalent). Column 2 of this table shows that the offered aggregators are not perfect. PC, for example, only selects the best suggestion in 64.5% of the turns. This is not surprising as this aggregator was only able to select the move when the majority of the players suggested it. Even the expert aggregators did not clearly excel here, indicating that no player (especially the ones that had a slightly higher expertise value) consistently provided better moves than other players. In column 3 we present an average value: for every turn, we checked how many move suggestions were provided that were better than the selected one. Here, the "crowd" aggregators seem to be better than the "leader" aggregators. This is explainable with the concept of the wisdom of crowds [29], i.e., that a group of people comes to a better decision compared to an individual decision. Especially in our case, in which no proficient chess players were present, this seems reasonable. Column 4 shows how often an aggregator selected the worst suggestion amongst the user votes. Overall, the performance of the random leader is (as expected) worst.

Agg.	#Turns active (# of last active turn)	Draw wins	% best move suggestion selected	Avg. % of move suggestions better than move selected by aggregator (across all active turns)	% worst move suggestion selected
PC	7 (#16)	5	57.1%	13%	14.3%
CC	0 (-)	0	-	-	-
EC	17 (#31)	2	76.5%	14.3%	35.3%
RL	5 (#19)	5	40%	31.6%	60%
CL	2 (#2)	2	100%	0%	0%
EL	0 (-)	0	-	-	-

Table 3. Aggregator performances while active. All values are in relation to the turns in which the aggregator was active.

Aggregator uses

Table 2 shows only a generalized view that does not consider when an aggregator was actually activated by the audience. For example, if an aggregator outputs the best move amongst all available suggestions only 40% of the time, it does not necessarily mean that this aggregator is bad. If the audience selected this aggregator only in situations in which it outputs the best move of the available user move suggestions, this would lead to two conclusions: first, it would show that the audience is able to self-administrate itself (by knowing which aggregator is currently good) and second, that even though the overall performance of the aggregator is suboptimal, in terms of how it is used, it is optimal. We found that in 20 of 31 turns (65%), the audience indeed activated an aggregator that selects the best move amongst the user suggestions. Considering that in only 25 turns the aggregators could have provided the best outcome, this is an encouraging result. Table 3 shows how often an aggregator was active and its performance. These numbers also represent the case without filtering situations in which the worst and best move suggestion were the same.

The following results could be derived from Table 3: first, the most active aggregator was EC. This is in line with our expectation (cf. aggregator section). Interestingly, EL was never activated. It seems that the audience still wanted to empower their members to contribute something instead of simply providing one expert with the option to play alone. Another explanation might be that they were aware that no single player alone excels. Second, activating EC was a reasonable decision, as it selected the best suggestion in 13 of the 17 turns it was active. We analyzed what happened in the other four turns: two times at least one other aggregator would have provided the best result: PC for one turn and RL for the other. As choosing RL could not be accounted for as a deliberate "better" choice, this led to only one turn in which another aggregator would have provided the best result. By additionally considering how often any aggregator would have provided a better result (and not necessarily the best), one instance of RL and one instance of PC/CC/CL would have. Second, compared to Table 2, EC and CL performed (slightly) better, i.e., they seemed to be activated at the right time. Third, we had many turns (14/31) in which two or more aggregators had the same number of up-votes (draws). PC, RL and CL had most (or all) of their up-time only because they were randomly selected in such situations. This shows that the crowd was not uniform. EC instead was preferred clearly as it was active in 17 turns (with only two draw wins). EC also dominates in the second match, as PC for example was never activated after the 16th turn. This dominance was also visible in the vote distributions, as four participants wanted EC, while only two wanted other aggregators towards the end of the study (4:1:1). Fourth, the conformity-based aggregators were not really used (even though CC is about as good as EC, as shown in Table 2). As the participants also stated that they were not interested in the conformity, they either did not understand the functionality of the value itself, did not understand the aggregator, or they found the expertise-based aggregators more appealing. As seven participants stated they did not feel part of the group, they might have thought that CC and CL do not provide coherent results, as they are "outside" the group. We also checked for every aggregator how many players voted for it at least once: PC (3x), CC (1x), EC (7x), RL (3x), CL (1x), EL (0x). This again shows that CC, CL and EL were not appealing for the players and that EC dominated.

Overall, these results should be seen in the context that only seven users did aggregator switches at all. As we had a question on infrequent aggregator switches in the questionnaire (and allowed multiple selections) we could investigate this further: considering the players that never used the aggregator change command, the players with ID 10 and 12[4] reported that they did not understand the aggregators, players 03 and 11 that they had too little time and players 03 and 07 that they thought switching the aggregator would not have led to better moves and that they were satisfied with the current aggregator. Considering the players that voted for aggregators, 01, 04 and 05 reported that they had too little time, 02 and 06 that switching would not have led to better outcomes and 01, 04, 08 and 09 that they were satisfied with the current aggregator.

Participants were also asked why they did change the aggregator, and five participants responded: 06 stated that RL was more interesting as he/she had not expected to make an impact in EL and CL. 08 stated that he/she was interested in PC and EC and that the game was too fast to switch the aggregator more often. 01 reported that he/she switched "*when moves were executed that I thought were bad and a switch would favor mine*"; 09 stated "*partially, I have selected expertise, because I'm not so good myself*". 02 answered that he/she wanted to prevent a move being executed randomly. This shows that the motives for why changes happen are different. None of the participants thought that *CrowdChess* needed further aggregators, and four thought that there were too many, but the answers for which ones should be removed were inconclusive (CC (1x), EC (1x), RL (2x), CL (2x), EL (2x)).

We also considered the chat to learn which social interactions towards decision-finding happened that might be a further explanation for why the aggregator system was not used more often. 15 turns were discussed in terms of which move should be done next (the chat was used in both matches for these discussions): either by a user giving a concrete suggestion (e.g., 06: "*f8 - b5*" or 01: "*I recommend to move the queen*"), a player starting a discussion (e.g., 03: "*g8-e7?*"), a player asking for help (e.g., 09: "*suggestions from pros, please*") or discussing more general plans (e.g., 06: "*cover the e5*" or 09: "*killllllit - with the pawn?????*"). Seven players participated in these discussions. One time, a player asked which aggregator should be activated and two other players responded with their preferences. Similar to TPP [18], even in this small user base, we found trolling tendencies by one user (e.g., 06: "*we should always do the same like the AI*" or 06: "*randomize it*").

Discussion

In this first study, we only tested *CrowdChess* with a small number of participants in an exploratory fashion (see also the limitations below). Our results should thus not be overestimated, but they already give valuable insights on how small groups of viewers interact in such a scenario: first of all, we found that the aggregators were not used by all players. As we could show, EC performed well in this setting, and the audience mainly activated this aggregator. Viewers might have voted for EC because they liked its move selection, they approved our expertise metric and/or they liked the idea of the aggregator. As no other aggregator significantly outperforms EC, we cannot decide which hypothesis should be accepted, but we will explore this further, for example with the addition of a "fake aggregator" that is purely based on the chess engine and objectively excels the other aggregators. In contrast, PC—often used in the live-streaming context—was not often favored (otherwise it would not have come to so many draws) and had no up-time later on. The conformity-based aggregators were not interesting for the audience, even though the performance of CC was slightly better than EC's. As many participants reported not feeling part of the group, this might be an explanation of why the audience had no trust in an aggregator that uses one's similarity in relation to the group. Lessel et al. [12] also reported that the conformity-based group aggregator was not interesting anymore after an expertise-based aggregator became available. The difference here is that in their experiment the expertise values were viewer-based (viewers could up-vote others) instead of computed objectively; however, both show that conformity-based aggregators are less interesting as soon as other aggregators become available. Furthermore, EL was never activated, indicating that either the audience has more trust in the "wisdom of crowds" [29], or that they simply did not want to enable one player to decide the moves alone. Both is also supported by considering that the "leader" aggregators had less up-time in comparison to the "crowd" aggregators.

From the qualitative answers, we learned that there were different reasons why participants did not vote for aggregators (more often): besides too little time, they often reported that they were already satisfied with the active aggregator or thought that other (inactive) aggregators would not perform better. This seems to be a reasonable explanation and hints that this needs to be considered in such types of audience-driven games. Not participating in something can also mean that the users agree with what is happening, and today, when polls are used

[4]Indicates the ID in Table 1.

in streams, the number of non-voting viewers is not considered. We also learned that there are different motivations for why aggregators should be changed. Even though EC was selected, other tendencies were also revealed that are not primarily helpful for the goal of winning the game (such as being able to contribute a move with RL in the presence of a conformity/expertise aggregator where a viewer with low conformity/expertise values has little influence). Considering TPP, this might explain why so much chaos happened there and led to the introduction of the majority-based aggregator. Although participants reported problems of time, it seems that some still had enough time to discuss moves in the chat, instead of relying on the aggregators alone. We hypothesize that this has less impact in larger channels, as work already exists that shows that the chat in larger channels is hard to maintain [6] or that the chat dynamics change [5].

For audience-driven games, our results hint that the time restriction is an issue. We will increase the turn time and will offer a *!ready* command in the future: if entered by enough players (e.g. 70% of the active players) the turn ends early. Additionally, regular chess players who are used to playing with time constraints should also be able to deal better with the time restriction in *CrowdChess*. The main issue that the participants reported with the short turn time was that they did not have enough time to understand/consider all UI elements of *CrowdChess*. In a live-streaming situation this might be different than in our study: as our participants knew that they would only play for a limited time overall, they apparently invested more in the move suggestions than in getting familiar with the interface. The chat-based interactions were treated as suboptimal and users wanted a more direct interaction. This is in line with Lessel et al. [13], in which the direct interaction with the live-stream was found to be a beneficial feature. Unfortunately, the current live-streaming platforms do not offer enough technical features to realize such an interaction that at the same time provides sufficient individual feedback. Thus, we reason that the current platforms need to improve to allow audience-driven games to truly emerge.

Our study had several limitations: first, the small number of participants, which seems acceptable for a first exploration with *CrowdChess*. Further studies will be done with more participants and we expect that a larger sample will change the dynamics that happen in the chat. Also, the selection of aggregators might be different when more users in a larger sample have trolling tendencies or want to follow their own agenda. Here, it will be interesting to see whether participants that have not voted so far start to vote to reduce the impact of such players. Second, restricting the study time to only 45 minutes meant participants were only able to play two matches and were keen on participating in both, instead of interacting with the other *CrowdChess* features. Especially the perception of such an audience-driven game might change during a longer time-span. Third, regarding the selection of our sample, only four participants had live-streaming experiences, and no one was a regular chess player. Conducting a study with chess players only, or players that are regular consumers of live-streams, might reveal different results in relation to the aggregators (for the chess group) or towards audience-driven games (for the stream viewers). Fourth, we investigated the board game chess instead of digital games. Even though we explicitly wanted to use a less complex game in terms of its game actions to evaluate aggregator decisions, it is currently unclear how these findings map to other (video) games. Nonetheless, as findings are in line with reports of TPP (e.g., such as the different player tendencies) [9, 18] and also the lab study of Lessel et al. [12] (e.g., similar up-times of the aggregators), it seems that the results can be extended to different settings. Fifth, from a technical point of view, some participants reported disconnects on Beam or refreshed their browser window. In both cases, *CrowdChess* cleared their votes and it is unclear if they were aware of this. In the future votes will be persisted briefly before they are erased.

FUTURE WORK

Besides the aspects for future work already stated throughout the paper (e.g., investigating how dynamics change in a streamer vs. audience setting) and conducting studies that overcome the mentioned limitations (e.g., a larger sample size, integrating more chess players), several further research directions can be followed with *CrowdChess*. For the aggregators, it would be interesting to see whether the usage patterns remain stable across samples (size and composition). For example, as the “Anarchy” aggregator was often used for entertainment reasons in TPP [25], it might happen that a sample of regular live-stream viewers behaves differently. The exact rationale for how we derived the conformity and expertise values was not disclosed. In upcoming studies this could be explained in detail, and different approaches on how these values are updated could be compared in respect to the viewers’ perception. At the same time, effects on the viewers’ perceived agency and fairness could also be considered. Another valuable direction could be to analyze the social dynamics that happen in the chat in relation to the audience size and how this affects the move suggestions and aggregator selections. This could be contrasted with a study in which an aggregator is dictated by the system (and not by the viewers) for every turn. Because of the technical situation of live-streaming platforms, the chat interaction was chosen as the primary interaction paradigm. Analyzing the viewers’ perception of other input options (e.g., via an external web page) in such a context seems important as well. Taken together, such research will eventually lead to design recommendations for audience-driven games.

CONCLUSION

We presented *CrowdChess*, a game to investigate shared game control in live-streams. With it, we follow the goal to investigate the emerging experience of audience-driven games in the live-streaming context, which is not yet researched in depth. Besides elaborating on the features of *CrowdChess*, we contribute a first user study. Our results can help other researchers to inform the design of their audience-driven projects. Furthermore, we contributed insights on how a small-scale audience interacts when their goal is to win the game and aggregators are offered. We found that the audience selected an expertise-based group aggregator primarily, but did not want to let the most experienced player play alone. As discussed, *CrowdChess* can now be used to pursue a range of further research directions that will help to inform future audience-driven games.

REFERENCES

1. Alexa Internet, Inc. 2017. Top Sites in United States. (2017). http://www.alexa.com/topsites/countries/US, last accessed: 31/07/2017.
2. Pablo Cesar and David Geerts. 2011. Understanding Social TV: A Survey. *Proc. NEM Summit 2011* (2011), 94–99.
3. Gifford Cheung and Jeff Huang. 2011. Starcraft from the Stands: Understanding the Game Spectator. In *Proc. CHI 2011*. ACM, New York, NY, USA, 763–772. DOI: http://dx.doi.org/10.1145/1978942.1979053
4. Matthew DiPietro. 2014. Twitch is 4th in Peak US Internet Traffic. (2014). https://blog.twitch.tv/twitch-is-4th-in-peak-us-internet-traffic-90b1295af358, last accessed: 31/07/2017.
5. Colin Ford, Dan Gardner, Leah Elaine Horgan, Calvin Liu, a. m. tsaasan, Bonnie Nardi, and Jordan Rickman. 2017. Chat Speed OP PogChamp: Practices of Coherence in Massive Twitch Chat. In *CHI 2017 Extended Abstracts*. ACM, New York, NY, USA, 858–871. DOI: http://dx.doi.org/10.1145/3027063.3052765
6. William A. Hamilton, Oliver Garretson, and Andruid Kerne. 2014. Streaming on Twitch: Fostering Participatory Communities of Play Within Live Mixed Media. In *Proc. CHI 2014*. ACM, New York, NY, USA, 1315–1324. DOI: http://dx.doi.org/10.1145/2556288.2557048
7. Hsi-Mei Hsu and Chen-Tung Chen. 1996. Aggregation of Fuzzy Opinions Under Group Decision Making. *Fuzzy Sets and Systems* 79, 3 (1996), 279–285.
8. 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 *Proc. WWW 2012 Companion*. ACM, New York, NY, USA, 1181–1188. DOI: http://dx.doi.org/10.1145/2187980.2188259
9. Harris Kyriakou. 2015. Twitch Plays Pokémon: An Exploratory Analysis of Crowd Collaboration. (2015). http://www.innqui.com/wp-content/uploads/2015/11/Twitch-Draft.pdf, last accessed: 31/07/2017.
10. Walter S. Lasecki, Christopher Homan, and Jeffrey P. Bigham. 2014. Architecting Real-Time Crowd-Powered Systems. *Human Computation* 1, 1 (2014).
11. Walter S. Lasecki, Kyle I. Murray, Samuel White, Robert C. Miller, and Jeffrey P. Bigham. 2011. Real-Time Crowd Control of Existing Interfaces. In *Proc. UIST 2011*. ACM, New York, NY, USA, 23–32. DOI: http://dx.doi.org/10.1145/2047196.2047200
12. Pascal Lessel, Michael Mauderer, Christian Wolff, and Antonio Krüger. 2017a. Let's Play My Way: Investigating Audience Influence in User-Generated Gaming Live-Streams. In *Proc. TVX 2017*. ACM, New York, NY, USA, 51–63. DOI: http://dx.doi.org/10.1145/3077548.3077556
13. Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. 2017b. Expanding Video Game Live-Streams with Enhanced Communication Channels: A Case Study. In *Proc. CHI 2017*. ACM, New York, NY, USA, 1571–1576. DOI:http://dx.doi.org/10.1145/3025453.3025708
14. Jing Liu, Young-In Song, and Chin-Yew Lin. 2011. Competition-based User Expertise Score Estimation. In *Proc. SIGIR 2011*. ACM, New York, NY, USA, 425–434. DOI:http://dx.doi.org/10.1145/2009916.2009975
15. Anna Loparev, Walter S. Lasecki, Kyle I. Murray, and Jeffrey P. Bigham. 2014. Introducing Shared Character Control to Existing Video Games. In *Proc. FDG 2014*.
16. Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. 2011. How Social Influence can Undermine the Wisdom of Crowd Effect. *Proc. National Academy of Sciences* 108, 22 (2011), 9020–9025.
17. Albert E. Mannes. 2009. Are We Wise About the Wisdom of Crowds? The Use of Group Judgments in Belief Revision. *Management Science* 55, 8 (2009), 1267–1279. DOI:http://dx.doi.org/10.1287/mnsc.1090.1031
18. Michael Margel. 2014. Twitch Plays Pokemon: An Analysis of Social Dynamics in Crowdsourced Games. (2014). http://www.cs.utoronto.ca/~mmargel/2720/paper.pdf, last accessed: 31/07/2017.
19. Winter A. Mason, Frederica R. Conrey, and Eliot R. Smith. 2007. Situating Social Influence Processes: Dynamic, Multidirectional Flows of Influence Within Social Networks. *Personality and Social Psychology Review* 11, 3 (2007), 279–300. DOI: http://dx.doi.org/10.1177/1088868307301032
20. Dan Maynes-Aminzade, Randy Pausch, and Steve Seitz. 2002. Techniques for Interactive Audience Participation. In *Proc. ICMI 2002*. IEEE Computer Society, Washington, DC, USA, 15–. DOI: http://dx.doi.org/10.1109/ICMI.2002.1166962
21. Gustavo Nascimento, Manoel Ribeiro, Loïc Cerf, Natalia Cesario, Mehdi Kaytoue, Chedy Raïssi, Thiago Vasconcelos, and Wagner Meira Jr. 2014. Modeling and Analyzing the Video Game Live-Streaming Community. In *9th Latin American Web Congress*. 1–9. DOI: http://dx.doi.org/10.1109/LAWeb.2014.9
22. Karine Pires and Gwendal Simon. 2015. YouTube Live and Twitch: A Tour of User-generated Live Streaming Systems. In *Proc. MMSys 2015*. ACM, New York, NY, USA, 225–230. DOI: http://dx.doi.org/10.1145/2713168.2713195
23. Sam Prell. 2014. Twitch Plays Pokemon Final Stats: 1.1 Million Players, 36 Million Views. Internet. (2014). http://www.engadget.com/2014/03/01/twitch-plays-pokemon-final-stats-1-1-million-players-36-millio, last accessed: 31/07/2017.

24. Nguyen Quoc Viet Hung, Nguyen Thanh Tam, LamNgoc Tran, and Karl Aberer. 2013. An Evaluation of Aggregation Techniques in Crowdsourcing. In *Web Information Systems Engineering (WISE 2013)*, Xuemin Lin, Yannis Manolopoulos, Divesh Srivastava, and Guangyan Huang (Eds.). Lecture Notes in Computer Science, Vol. 8181. Springer Berlin Heidelberg, 1–15. DOI:http://dx.doi.org/10.1007/978-3-642-41154-0_1

25. Dennis Ramirez, Jenny Saucerman, and Jeremy Dietmeier. 2014. Twitch Plays Pokemon: A Case Study in Big G Games. In *Proc. DiGRA 2014*. 3–12.

26. Tord Romstad, Marco Costalba, and Joona Kiiski. 2017. Stockfish Chess. (2017). https://stockfishchess.org, last accessed: 31/07/2017.

27. Thomas Smith, Marianna Obrist, and Peter Wright. 2013. Live-Streaming Changes the (Video) Game. In *Proc. EuroITV 2013*. ACM, New York, NY, USA, 131–138. DOI:http://dx.doi.org/10.1145/2465958.2465971

28. Studio Bean. 2015. Choice Chamber. (2015). http://www.choicechamber.com, last accessed: 31/07/2017.

29. James Surowiecki. 2005. *The Wisdom of Crowds*. Anchor.

30. Crispin Thurlow, Laura Lengel, and Alice Tomic. 2004. *Computer Mediated Communication*. Sage.

31. Wikipedia Community. 2017. Infinite Monkey Theorem. (2017). https://en.wikipedia.org/wiki/Infinite_monkey_theorem, last accessed: 31/07/2017.

32. Mu Xia, Yun Huang, Wenjing Duan, and Andrew B. Whinston. 2009. Ballot Box Communication in Online Communities. *Communications of the ACM* 52, 9 (Sept. 2009), 138–142. DOI: http://dx.doi.org/10.1145/1562164.1562199

33. Cong Zhang and Jiangchuan Liu. 2015. On Crowdsourced Interactive Live Streaming: A Twitch.tv-Based Measurement Study. In *Proc. NOSSDAV 2015*. ACM, New York, NY, USA, 55–60. DOI: http://dx.doi.org/10.1145/2736084.2736091

34. Cong Zhang, Jiangchuan Liu, and Haiyang Wang. Towards Hybrid Cloud-assisted Crowdsourced Live Streaming: Measurement and Analysis. In *Proc. NOSSDAV 2016*. ACM, New York, NY, USA, Article 1, 6 pages. DOI:http://dx.doi.org/10.1145/2910642.2910644