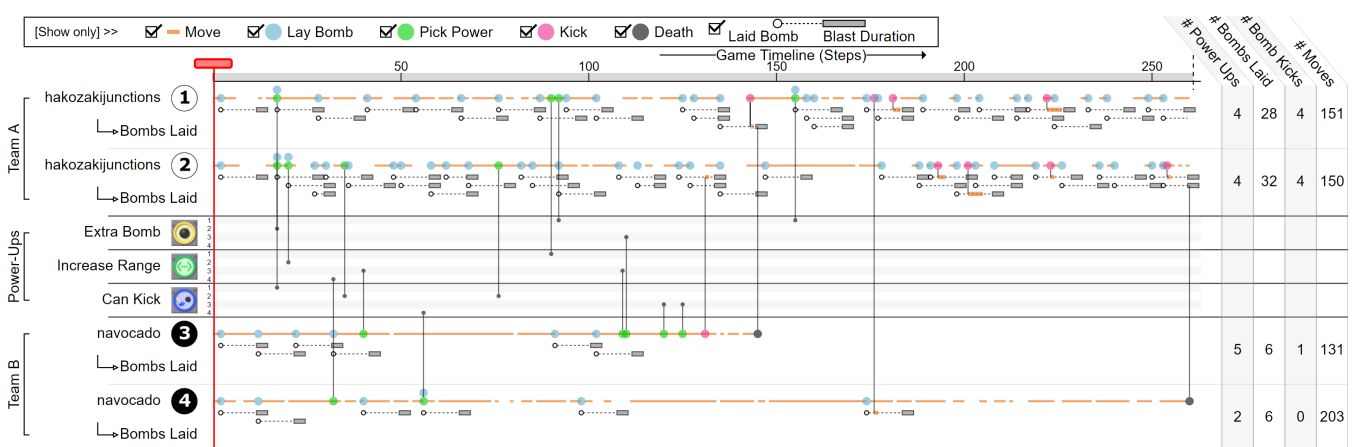# Bombalytics: Visualization of Competition and Collaboration Strategies of Players in a Bomb Laying Game

Shivam Agarwal[1][†] , Günter Wallner[2][‡][§] and Fabian Beck[3][†]

[†] University of Duisburg-Essen, Germany
[‡] Eindhoven University of Technology, The Netherlands
[§] University of Applied Arts Vienna, Austria

Figure 1: *Timeline visualization to analyze a game of* Pommerman, *where the horizontal axis represents time and the entities of the game—four players and three types of power-ups—are shown in individual rows. The players are divided into two teams. Events are encoded in color. Vertical lines between rows denote interactions between the corresponding entities. Columns at the end show different game metrics.*

Abstract

Competition and collaboration form complex interaction patterns between the agents and objects involved. Only by understanding these interaction patterns, we can reveal the strategies the participating parties applied. In this paper, we study such competition and collaboration behavior for a computer game. Serving as a testbed for artificial intelligence, the multiplayer bomb laying game Pommerman *provides a rich source of advanced behavior of computer agents. We propose a visualization approach that shows an overview of multiple games, with a detailed timeline-based visualization for exploring the specifics of each game. Since an analyst can only fully understand the data when considering the direct and indirect interactions between agents, we suggest various visual encodings of these interactions. Based on feedback from expert users and an application example, we demonstrate that the approach helps identify central competition strategies and provides insights on collaboration.*

1. Introduction

Behavioral science widely acknowledges that, in order to best understand dynamic processes of interactions, these need to be viewed from a sequential perspective. For instance, Bakeman et al. [BG97] state that a *"defining characteristic of interaction is that it unfolds in time."* Indeed, the importance of visualizing temporal event data to better understand complex processes has long been recognized within the visualization community (cf. [SP19]). Also, in games research the chronology of actions forms an important basis for understanding player behavior (cf. [CMTD18, MG11, Wal15]). Existing work has employed various sequence mining [KKK14, LJ14]

---

[1] e-mail: shivam.agarwal@paluno.uni-due.de
[2] e-mail: g.wallner@tue.nl
[3] e-mail: fabian.beck@paluno.uni-due.de

and statistical techniques [Wal15, Hou12] to obtain insights into behavioral sequences. Visualization solutions to help analyze and explore sequences in games are, however, much more scarce (e.g., [OSM18, LFLB19]). These approaches focus on sequences performed by individual players without considering how choices may be based on actions made by teammates or opposing players. Yet, these interactions between players are essential to understand aspects of competition and collaboration. Moreover, actions are commonly linked to specific entities of the game. Showing these connections can provide further context for behavioral analysis.

To help fill this gap, we propose a timeline-based visualization that displays and contrasts actions performed by a small number of players (see Figure 1) and also relates those to specific in-game entities. Such a visualization can serve different purposes. Among others, it allows players to analyze gameplay and strategies to help them improve their skills (cf. [Haz14]). In this paper, however, we focus on how developers of autonomous agents can benefit from such a visualization by exploring strategies of their agents. Computer games form an active area of Artificial Intelligence (AI) research, accounting for about 50% of all published work in the field [YT18]. Game-based competitions (e.g., [Hof19, TSKY13, GHLKT19]) have become a useful environment for testing, training, and benchmarking new AI algorithms [Tog16]. Creating autonomous agents for such (multi-agent) environments has several challenges as they need to successfully compete, cooperate, or do both to complete their objectives. Understanding the strategies (i.e., the sequences of actions) learned by the AI agents can help improve their performance. Building upon research in game analytics and visualization, we propose a novel visualization for the exploration of strategies executed by the agents.

While we envision the basic concept to be adaptable to a variety of games, we have tailored the approach to the game *Pommerman* [REH∗18]. *Pommerman* is a variant of the classic *Bomberman* [Hud83] game series and serves as a popular testbed environment for game AI researchers (e.g., [GHLKT19, PLGD∗19]). A constraint on real-time decision making (an agent has only 100 microseconds to decide) makes it more challenging to develop agents. The game environment was specifically designed to assess competition and collaboration among agents and features an active research community, thus serving as an ideal application to demonstrate our visualization. We also collected feedback from members of the *Pommerman* community to assess the usefulness of the approach. Hence, our contribution is threefold: We present (1) a visualization approach called *Bombalytics* (analytics for bomb laying games) for exploring sequences of actions of multiple players, (2) an evaluation of the approach with expert participants from different domains, and (3) an interactive web-based tool called *PomVis*[1] that implements the proposed approach [AWB20]. The supplementary material [AWB20] also includes the questionnaire and responses of participants from the user study.

2. Related Work

Sequential analysis of behavioral patterns is not only common in behavioral and social sciences (e.g., [BQ11, GMN77]) but also in

---
[1] Hosted at: https://vis-tools.paluno.uni-due.de/pom/

games research to better understand player actions. Soppitt and McAllister [MG11] video-recorded players while playing and then coded the videos based on exhibited behavioral states (e.g., boredom, engagement, frustration). Probabilities of behavioral transitions were then calculated based on the resulting state sequences. Others, in turn, used more large-scale datasets and relied on sequence mining techniques to analyze gameplay. For instance, Kang et al. [KKK14] analyzed how abilities get concatenated by players in *League of Legends* [Rio09]. Leece and Jhala [LJ14] employed sequential pattern mining to derive common action patterns and build orders from *StarCraft: Brood War* [Bli98]. Wallner [Wal15] argued that frequently occurring patterns may not necessarily be the most interesting ones as certain patterns may naturally occur more frequently than others and thus focused on statistically significant patterns identified through lag-sequential analysis. Hou [Hou12] followed the same approach to better understand behavioral patterns in an educational online role-playing game. While not exhaustive, this set of papers reflects the large interest in unraveling behavioral sequences in games research. The work at hand, contributes to this area by proposing a novel visualization tool that not only allows exploring sequential patterns but also viewing these actions with respect to activities performed by other players.

Visualization of Event Sequences in Games. Visualization of behavioral player data has gained considerable momentum within games user research and analytics during the last decade. These visualizations can be roughly classified into two groups: a) visualizations that spatially situate the data with respect to the game environment and b) visualizations that use more abstract representations of the data. As our approach also follows an abstract approach, we will focus on the latter category in the following. A comprehensive overview of gameplay visualizations can be found in the survey by Wallner and Kriglstein [WK13].

Many abstract in-game data visualizations use node-link diagrams to offer summary views of behavioral sequences across a multitude of players. Examples include *Playtracer* [ALA∗10], which visualizes transitions through game states. Multidimensional scaling is used to create a two-dimensional embedding of the states and to convey the different traces players took through the state space. *PLATO* [WK14] uses a similar graph-based representation to formally describe gameplay but extends previous work by including a variety of interaction and analytics methods such as clustering and subgraph matching. Similarly, *Glyph* [NENC15] also uses node-link diagrams to provide an aggregated view of play traces while, at the same time, allowing for inspection and comparison of individual traces through highlighting. In contrast to the work above, our solution is focused on preserving details of individual games rather than only aggregating data over multiple games.

Osborn et al. [OSM18], focused on sequences of in-game actions as opposed to game states, which may be difficult to define. Each play trace is represented through an individual line with color-coded circles—indicating different actions—placed along the lines. Similar play traces are arranged in proximity. Li et al. [LFLB19] used a horizontal timeline and plot actions performed by players using color-coded dots. Likewise, Li et al. [LXC∗17] also used timelines to convey concurrent actions of multiple players as part of a larger system to investigate gameplay patterns in multiplayer on-

line battle arena games. Our work shares some similarities with the above works, that is, a) using an action-centric approach like Osborn et al. [OSM18] and b) using a horizontal timeline to represent the timing of actions similar to Li et al. [LXC*17, LFLB19]. However, these works take a player-centric perspective as sequences are visualized independent of other players' actions. The former aggregates traces over a large number of players while Li et al. focused on how sequences change over repeated runs to assess players' skill growth and strategy adjustments over time. Wang et al. [WGSY19]—also analyzing agents—proposed a visual analytics system for deep reinforcement learning models, but their work focuses on better understanding the training phases of a solitary agent. Our focus, in contrast, is on how (groups of) agents act and react to each other to better understand aspects of competition and collaboration between agents.

Visualization of Event Sequences across Domains. Beyond games, visualization of event sequences has also attracted attention in Human-Computer Interaction and other domains. For example, works concerned with visualizing dynamics between multiple actors, such as in conversations (e.g., [EAGA*16]) or interactions in interior spaces (e.g., [SH17]), also have to deal with understanding interactions among agents. In sports visualization, relationships among the actions of players form a key aspect as well. Ono et al. [ODS18], for instance, created small multiples to convey player positions and movements during interesting events in baseball matches and coupled this with a space–time diagram to display player positions with respect to the bases. Polk et al. [PJHY20] used space–time charts to show interactions of tennis players allowing to observe player and ball positions at the same time. Wang et al. [WZD*20] proposed a system to simulate and analyze table tennis matches, putting a strong focus on rallies (i.e., sequences of ball hits). These works share similarities with ours, given their emphasis on showing how actions unfold over time but are not directly applicable in our scenario due to, for instance, only considering two players or not taking interactions with different items into account.

Taking an even more general view, Guo et al. [GXZ*18] segmented event sequences into groups of fixed-length time intervals with similar segments being grouped into clusters to help better understand progression patterns within the sequence data. Subsequent work by Guo et al. [GJG*19] utilizes this approach within a larger visual analytics system and relaxes the fixed-width time interval restriction of the original approach. Chen et al. [CPYQ18], also dealing with progression analysis, extracted and visualized frequently occurring subsequences. Burch et al. [BBD08] coupled timelines with a tree representation to support comparisons within a hierarchy of event sequences. Nguyen et al. [NTA*19], like us, made use of summary histograms and a detail view to visualize multiple sequences of actions. These approaches, however, differ from our approach in that we explicitly show interactions with objects and link event sequences to each other to help better understand temporal action–reaction relations among multiple agents.

Visualization of Game AI Behavior. We especially target AI developers to assist in analyzing the behavior of agents in games. While visualizations have been employed for this purpose before, they are—as the works above—usually not concerned with show-

ing the interactions between multiple agents. To give some examples: Chang et al. [CAS19] employed dot distribution maps to compare areas of a level traversed by an AI agent to those of a human player. Pfau et al. [PSM17]—concerned with automated game testing using reinforcement learning—use reward maps to depict the reward of different actions. Karakovskiy and Togelius [KT12] visualize potential future paths considered by an AI agent. Recently, Douglas et al. [DYK*19] presented a three dimensional visualization to better analyze information about AI agents, using—as we do—*Pommerman* as application example. They visualized stacked saliency maps in virtual reality to show which areas are identified by an agent over an entire game. Their approach, however, focuses on salient areas while we focus on succession of actions.

3. The Pommerman Game

Pommerman [REH*18] is a variant of the classic multiplayer game *Bomberman* [Hud83]. A game in *Pommerman* can have a maximum of four players. There are two modes: (a) all players compete against each other or (b) two teams, consisting of two players each, compete against each other. The map of the game is a board with 11×11 tiles where each tile can be a free navigable space, a rigid block, or a wooden wall that collapses when a nearby bomb explodes. The layout of the map is generated randomly for each game, but the starting positions of the players remain the same. Each player can lay a bomb, which explodes after a fixed duration (ten game steps). Flames from the bomb explosion persist for three game steps. Each player has to wait for the previously laid bomb to explode before laying another bomb. There also exist three types of power-ups limited in number and hidden beneath wooden walls, which offer: (i) an increase of the number of bombs a player can place simultaneously, (ii) an increase of the range of the bombs laid by a player, and (iii) the ability to kick bombs. To win a game, players (or teams in team mode) have to eliminate their opponents. In this work we focus on the team mode where players compete and collaborate to win a game.

The *Pommerman* game was built to train agents compete and collaborate in a multi-agent environment [REH*18]. Using the game as a platform, several agents have been trained via different techniques and tested against each other [PPYG18, ZGM*18, HLKT19, OT19, GHLKT19, GKHLT19, KHLGT19]. *Pommerman* competitions are organized to promote research in this field, such as at the *NeurIPS* 2018 and 2019 conferences. Knowledge gained from these competitions has led to a better understanding of the underlying techniques. However, most commonly, performance analysis is done only on the number of games won by the agent, which hides the qualitative aspects of the behavior. This limits the ability of developers to investigate the learned strategies and further improve the performance of the agents. Developers can only watch individual games for a qualitative assessment, which includes checking for competition and collaboration strategies. This was confirmed by a developer of a top performing agent of the *Pommerman NeurIPS* 2018 competition, stating that: *"We find these [learned strategies] by running several battles and recognition by human."*
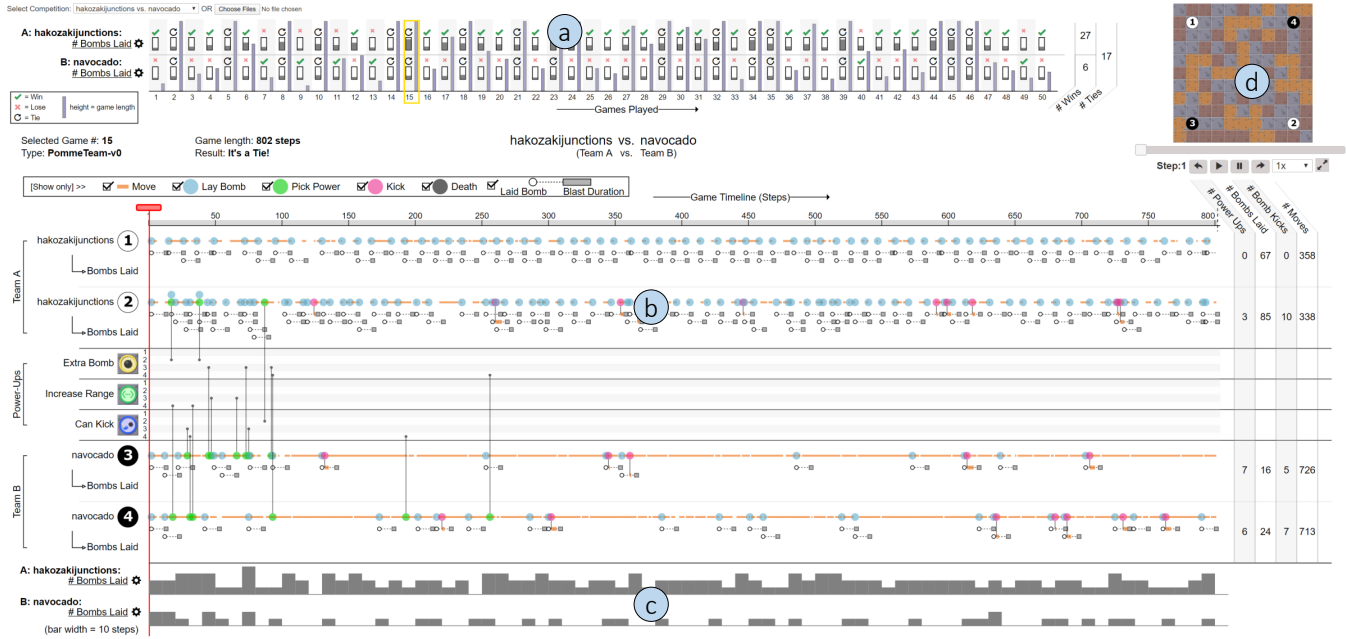
Figure 2: *The* PomVis *interface consists of four components: (a) a summary of all the games in a competition, (b) a detailed timeline visualization of the selected game, (c) histograms to contrast the action densities of two teams, and (d) playback of the selected game.*

4. Design Goals

For investigating agent behavior and comparing the performance of two teams, we first considered the goals that we deemed central for designing the visualization. These design goals are based on informal communication with *Pommerman* community members, our experience in visualizing event sequences, and insights from related approaches. Beyond these specific goals, we tried minimizing visual complexity, using expressive labels, and building an intuitive visualization.

G1: Overview of event sequences in a game. Currently, the developers of *Pommerman* agents usually use playback to analyze the recorded games. While playback is useful in general, developers need to watch an entire animation to get an implicit overview of the event sequences in a game. To reduce time effort and ease interpretation, however, it becomes important to obtain an explicit overview of the events that occurred in the game through a static visualization. The overview should display the distribution of events across the entire game, which could also point out different phases.

G2: Local patterns and repetitions. Collaboration and competition strategies between agents are exhibited by interactions between the agents and specific items, for instance, kicking a teammate's bomb. The design of the visualization should support finding such local patterns. Since the same strategy might be executed several times in a game, the visualization should also show these repetitions. The developers currently rely either on summary game statistics or on the playback to infer behavior patterns. However, aggregated statistics only provide an incomplete picture as they neglect the intermediate processes while identifying multiple occurrence of the same pattern of actions and movements in a playback is tiring.

G3: Overview of a competition in a set of games. To compare two teams in a competition, usually 30–50 games are held. Hence, the visualization should also support statistical comparisons between two teams based on several metrics and provide a basis for selecting the most interesting matches for closer analysis.

5. Visualization Approach

We propose Bombalytics, a novel visualization approach and implement it in a tool called *PomVis*. Figure 2 shows a screenshot of its interface, which consists of four components. Next, we discuss the data required for the visualization followed by a description of each component of the interface.

5.1. Data

The *Pommerman* environment provides a command line option to record the state of a game at each step. Developers of autonomous agents for *Pommerman* use this option to analyze, e.g., number of wins, loses, and ties. To enable easy and widespread use of our tool among the developers, we rely only on this recorded data without further instrumentation of the game. The game states recorded in the data are used to generate a playback and a summary. We extract the actions performed by the agents and identify bomb explosions.

We analyze sample data consisting of six competitions, wich were held between three agents of the 2018 competition (in top 10 final rankings): *hakozakijunctions*, *navocado*, *skynet955*, and the *simpleAgent*, which is the default learning agent provided in the *Pommerman* environment. The executable container images of the
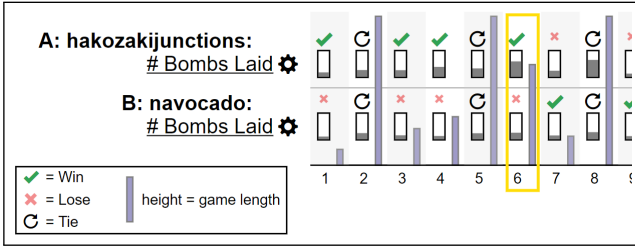
Figure 3: *The summary component showing multiple games in a competition between two teams. For each game, it shows the game result through colored icons, the game duration through the height of thin light purple bars, and a selected game metric for each team through dark gray bars.*

Figure 4: *Vertical lines between rows show associations of a player (here, Player 3) with power-up rows when the player picks the powers and bombs when the player kicks them. In the example, first, the player picks two 'increase range' power-ups, followed by a 'can kick' power-up. Then, the player kicks two bombs and later picks two more 'can kick' power-ups.*

agents were fetched from Docker Hub.[2] A team in our sample data, consists of two instances of the same agent. Each competition consists of 50 games between the two respective teams.
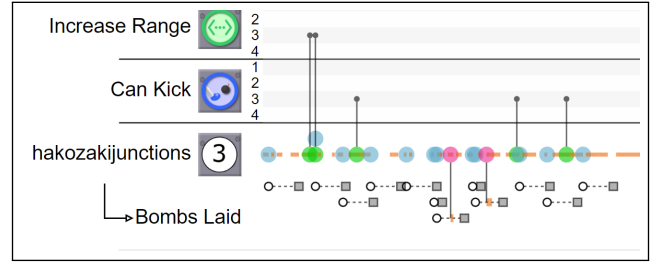
5.2. The Summary Component

The summary component at the top of the interface (Figure 2a) provides a high level overview of all the games in a competition (**G3**). Individual games are represented along the horizontal axis in columns and are numbered, as visible from the enlarged image in Figure 3. The two teams are shown as separate rows. The result of a particular game is represented as icons: *Win* (✔), *Lose* (✘), or *Tie* (↻). We compute seven game metrics for each team in every game, specifically, the number of

1. moves (*#Moves*),
2. bombs laid (*#Bombs Laid*),
3. kicks to bombs (*#Bomb Kicks*),
4. pick-ups for any power (*#Power-ups: Any Power*),
5. pick-ups for 'extra bomb' power (*#Power-up: Extra Bomb*),
6. pick-ups for 'increase range' power (*#Power-up: Increase Range of Bomb*), and
7. pick-ups for 'can kick' power (*#Power-up: Kick*).

The values of one selected metric of a team for individual games are visualized through dark gray bars placed in the respective row. The game metric can be changed by clicking the underlined label of the metric or the gear icon. The length of a game in a competition is encoded by the height of a thin light purple bar. The total number of wins and ties for each team are shown at the end of the rows (Figure 2a). Clicking a particular game column draws the detailed visualization of the corresponding game in the components below, as discussed next.

5.3. The Timeline Visualization of a *Pommerman* Game

The static timeline visualization component (**G1**) is placed in the middle of interface as shown in Figure 2b. The horizontal axis represents the temporal progression of the game (timeline) and shows

each step of the game in sequence from left to right. Each entity (a player or a power-up) is shown as a separate row in the visualization. Rows representing players are split into two parts: the upper part shows actions performed by the player, while the lower part shows bombs laid by the player. Separating the players from the bombs allows identifying more clearly the lifespan of bombs, kicks, and blast duration, as explained later.

For a clear visual distinction between the two teams, rows of players belonging to Team A are placed at the top, while those belonging to players of Team B are placed at the bottom. The rows of power-ups are added in the middle, as they denote common resources that can be utilized by any player. The separation between the rows of the two teams helps differentiate between competition and collaboration interactions among players (**G1** and **G2**).

During a game, players perform different actions, which we represent via color and shape of different glyphs (**G1** and **G2**). A player can move (▬), lay a bomb (●), kick a bomb (●), and pick up a power-up (●). Bomb explosions are important in the game as they might trigger other events, such as the death of a player (●), the destruction of a wooden wall, etc. We represent each bomb by a shape (○┈┈▬) that has an unfilled circle at the head—indicating that the bomb was laid—followed by a rectangular tail—denoting the explosion of the bomb and its duration (i.e., three game steps). The head and tail of the bomb glyph are connected by a dashed line. Since the lifespans of bombs laid by a player can overlap (if a player has an 'extra bomb' power), we place them at different vertical positions in the lower part of the row of the corresponding player if necessary. By visually representing the lifespan of every bomb, it becomes easy to identify actions and events related to each bomb individually (**G2**). Selecting a checkbox of the legend items (placed above the timeline visualization) highlights the corresponding actions, events, and game objects (bombs) in the visualization.

Each row of a power-up is divided into four sub-rows of equal height, each corresponding to a player, as shown in Figure 4. Although this introduces some redundancy, doing so helps in quickly identifying the player associated with the corresponding power-up (**G2**). Also, it becomes easy to follow a sub-row and count the number of dots to infer how many instances of the power-up were picked by the corresponding player (**G1**).

Some events can be associated with multiple entities (players and power-ups) and game objects (bombs). To visualize this association, a vertical line is drawn between rows of the corresponding entities and/or game objects (**G1** and **G2**). Figure 4 shows interactions between Player 3 and different power-ups as well as bombs kicked. The movement of bombs being kicked is shown using orange color in the timeline of the bombs (**G2**).

On the right side of the timeline visualization (Figure 2b), a few game metrics are shown in the columns for each player summed over the entire duration of the game (**G1**). The summed game metric values help in formulating hypothesis about the behavior of teams and individual players. However, the behavior of players might not remain the same for the entire game. For instance, players pick almost all power-ups in the beginning of the game. To visualize the temporal distribution of the game metrics along the progression of a game (**G1**), we draw histograms (two rows, one for each team) as shown in Figure 2c. The game metric can be changed through selection. The bin size (bar width) in the histograms is 10 game steps by default.

5.4. Playback Component

The components discussed before help identify the behavior of players and to formulate hypotheses about strategies executed by them. To verify the formulated hypotheses, it is still essential to watch the actual playback of the game at a specific step of the game. To support this, we integrate a playback component on the top right corner of the interface, as shown in Figure 2d (**G1**). The component includes standard playback controls. Navigation to a specific game step can be done via dragging either the slider placed above the playback controls or the red vertical status line in the timeline visualization (Figure 2b). The playback speed can also be modified.

6. Application Example

In this section, we show the usage of the approach. We present a few strategies and unusual agent behavior identified through visual analysis of the competitions in the sample data.

We select a competition between *hakozakijunctions* and *navocado* consisting of a total of 50 games. The summary component (Figure 2a) reveals that *hakozakijunctions* outperformed the other by winning 27 games and losing only 6 while 17 games resulted in a tie. Looking further into the summary component, we select the '# Bombs Laid' game metric and see that *hakozakijunctions* laid significantly more bombs in most of the games (dark gray bars, **G3**). However, on selecting the '# Power Ups' game metric, we find that *navocado* picked more power-ups in almost all the games. We select game #15, which resulted in a tie (↻), to explore details. Figure 2b reveals that both teams picked power-ups early in the game, inferred from the green dots and vertical lines (**G2**). However, one agent of the *hakozakijunctions* team did not pick any power-up (Agent 1 in the first row), while Agent 4 of *navocado* team continued picking power-ups in the later phase of the game, too (**G2**). The *hakozakijunctions* team moved less (few orange lines) and laid bombs more frequently (**G1**), inferred from the histograms below (Figure 2c) or from the last columns in the timeline visualization (Figure 2b). The *navocado* agents picked a lot of
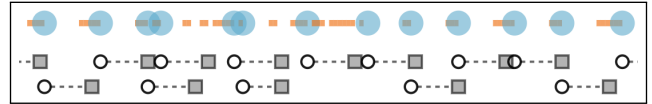
Figure 5: *An excerpt from Game #8 of a competition between* hakozakijunctions *and* navocado*, showing bold and suicidal moves by* hakozakijunctions*. The agent repeatedly lays a bomb, waits, and then moves when the bomb is about to explode.*

extra bomb power-ups but laid fewer bombs (columns at the end). The *navocado* agents moved a lot and seemed to explore the board (orange lines), which was confirmed via playback (Figure 2d) (**G1**). Agents 2, 3, and 4 laid and kicked their own bombs (pink circles and vertical lines) trying to kill the opponents (**G2**), but with no success. Eventually, the game timed out and resulted in a tie (**G1**).

Next, we list the discovered strategies and unusual behavior. Some of these strategies were also found by the participants of the user study (cf. Section 7.2).

Bold and suicidal move: The *hakozakijunctions* agents lay a bomb and stay on top of it. The agents only move when the bomb is just about to explode (**G2**). Figure 5 shows that this behavior is repeated periodically throughout the game. The agents manage to eliminate opponents with this strategy, but in many games get also killed by their own bombs.

Learn to kick bombs: It seems that the power of kicking a bomb makes a difference. In the six games in which the *hakozakijunctions* team was defeated, it was not able to collect 'can kick' power-ups, while *navocado* collected the power-up in these games (**G3**). In general, *hakozakijunctions* and *navocado* often kick bombs ('#Bomb Kicks' game metric). In many games, they also kick bombs laid by the other team (pink circles with lines to the other team rows) (**G2**). This behavior was especially exhibited in competitions with the *simpleAgent*, which does not kick bombs, even after collecting the 'can kick' power-up (**G1**).

Collecting redundant power-ups: The 'can kick' power-up is a binary property that, once picked, persists throughout the game. The *skynet955* agent has learned to avoid redundant collection of 'can kick' power-ups. This can be seen from the summary component in competitions of *skynet955* vs. other teams and selecting the '# Power-up: Kick' metric (**G3**). However, as shown in Figure 4, *hakozakijunctions* collects the power-up more than once; it could be a strategy to prevent opponents from picking it up (**G2**).

Stuck in a loop: Sometimes agents get stuck in a loop repeatedly moving between two tiles. This is visible from long continuous orange lines in the timeline visualization (**G2**). For instance, in Game #28 between *hakozakijunctions* and *navocado*, the *hakozakijunctions* agent was stuck in a loop, while the *navocado* did not do anything in the same duration (white space in the bottom two rows). The same behavior was observed in Game #14 where the *navocado* agent was stuck in a loop while its opponent waited idly. It shows that the agents have not learned to (a) avoid getting stuck in a loop and (b) exploit such vulnerabilities in opponents.

7. Expert User Study

To evaluate the proposed Bombalytics approach, we administered an online questionnaire to AI, visualization, and game analytics experts. The feedback of AI experts verifies the capabilities and usefulness of the proposed technique. However, AI experts in the *Pommerman* community do not typically use visualizations (such as ours) while training the agents. As such, responses of other experts, in particular visualization and game data analysts being more experienced with such interfaces and analysis of player activity in general, verify the visualization design. The questionnaire and responses are provided as part of the supplementary material [AWB20].

7.1. Study Design

The study consisted of an online questionnaire and an online version of the *PomVis* tool. Participants were asked to explore the tool and to optionally go through the help page before starting the questionnaire. The participants confirmed this preparation at the start of the questionnaire. Participants were allowed and reminded to switch back to the tool while filling out the questionnaire. The study was designed to take about 25 minutes, was conducted online, and ran for a period of 10 days. Participation was anonymous and no identifying information was recorded.

Questionnaire: The online questionnaire consisted of seven parts. After explaining the purpose of the study and acquiring consent from participants (Part I), Part II asked participants to provide some background on their domain expertise on a 5-point scale labeled with *no knowledge*, *beginner*, *intermediate*, *advanced*, and *expert*. We also asked about their experience with *Pommerman*, playing *Bomberman* games, and whether they participated in *Pommerman* competitions by submitting autonomous agents. Parts III and IV asked about the summary component and detailed timeline visualization, respectively. Participants were presented with statements in these parts expressing the usefulness of the tool, and were asked to rate them on a 5-point Likert-type scale anchored by *strongly disagree* to *strongly agree*. Optionally, the participants could provide detailed comments regarding what they liked and disliked about the above mentioned aspects of the interface. Part V asked participants to textually mention the competition and collaboration strategies they were able to discover using the tool. It also asked to mention observed differences in gameplay behavior of teams. In Part VI we assessed the usability of the interface regarding four characteristics: efficiency, effectiveness, satisfaction, and overall [Fin10]. We presented four statements for each category which participants answered by selecting *Strongly disagree*, *Disagree*, *Neutral*, *Agree*, or *Strongly agree*. The participants could provide further comments on the usability of the tool. Part VII allowed participants to give additional feedback on tasks for which they would use *PomVis* and missing or unnecessary information in the tool, as well as to provide additional remarks.

Participants: The work presented in this paper aims to assist *Pommerman* AI developers by building a visual tool using research from the fields of visualization and game analytics. Consequently, we invited a diverse group of users to participate in the study and provide their feedback. First, since the tool specifically visualizes the gameplay data of *Pommerman*, we invited users who: (i) make autonomous agents for *Pommerman* environment, (ii) participated in *Pommerman* competition, or (iii) have contributed in building the environment. Second, we invited visualization experts (in Information Visualization and/or Visual Analytics) who have research experience with event-timeline based visualizations. Third, we invited researchers who have expertise in gameplay analytics. Finally, we strove for participants who also had considerable experience in either playing computer games or in programming, and have played Bomberman games before. The invitations were sent via personal e-mail and through the official Discord channel of *Pommerman*.

In total, 20 users participated in the study. We refer to these experts as E1 to E20 in the remainder of the paper. All 20 participants marked their expertise level as *expert* or *advanced* in at-least one of the following five domains: *Artificial Intelligence*, *Playing Computer Games*, *Computer Programming*, *Information Visualization*, and *Game Analytics*. Expert E1 participated in both *Pommerman* competitions of 2018 and 2019, while four experts (E2–E5) participated only in the 2018 competition. Three further experts (E6, E7, and E8) also have experience in developing autonomous agents for the *Pommerman* environment without having participated in a competition. In addition, E6 contributed to the code repository of *Pommerman*. Nine experts (E5, E9–E16) marked themselves as *advanced* or *expert* in *Information Visualization* and/or *Visual Analytics*. Three out of them (E7, E11, and E13) also considered themselves having similar expertise in the domain of *Game Analytics*. We classify the experts in two groups based on their domain of expertise. Group A consists of *Pommerman* developers and AI experts (as the core user group of the tool, E1–E8), while Group B consists of visualization and game analytics experts (providing feedback with respect to visualization design and analytics, E9–E20).

7.2. Results

An inductive thematic analysis was carried out to analyze participants responses per question.

Summary Component: *Pommerman* and AI experts mentioned that essential information is visualized in the summary component (E3, E4, E5, and E7). E2 liked the inclusion of data from multiple games in the tool as it helped to get an overview of a competition. Visualization and computer game experts liked the simplicity of the columns to the right of the timeline showing #wins and #ties (E9, E16, and E17) and the static design of the component (E15). Seven experts liked the compact design of the component and highlighted that it gives a concise summary (E1, E2, E6, E8, E10, E11, and E18). Ratings in Table 1, however, show differences in opinions between the two groups of experts, with Pommerman and AI experts being more critical. Two experts (E1 and E5) did not find that the tool provides a good summary of all games in a competition. E1 noted in the comments the lack of a statistical summary of the games, e.g., average number of bombs. Two experts (E3 and E6) mentioned that it took some time to understand the different encodings used in the summary component. In addition, others reported difficulties with interpreting the game length bars (E11 and E13) and differentiating them from the gray game metric bars (E17). Experts offered suggestions on how to improve the design of

Table 1: *Quantitative expert feedback about the usefulness of interface components; scale from 'Strongly disagree' (1) to 'Strongly agree' (5).*
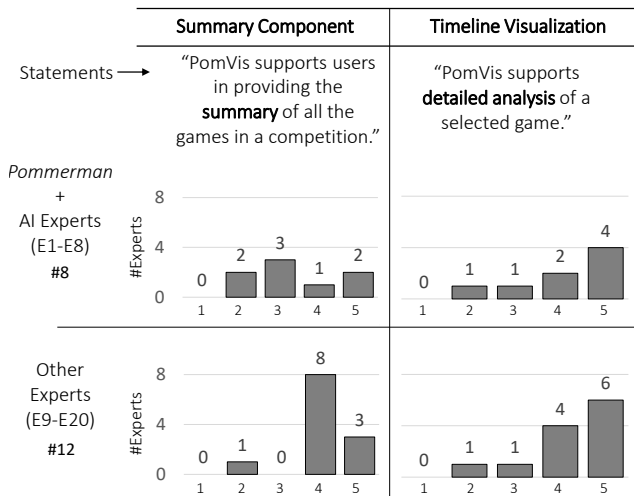
the summary component such as showing details on demand (E12, E18, and E19) and additional statistics (E1, E4, and E9).

Detailed Analysis of a Selected Game: Overall, experts appreciated the timeline visualization (Figure 2b), which is also reflected in their ratings, as shown in Table 1. The experts highlighted that it provides a good overview of the selected game (E6, E13, and E19) in one screen (E4 and E8) and is informative (E17) while at the same time showing details of every action performed by the agents (E1, E3, E9, and E18). They liked the timeline layout and visual encodings (E2, E13, and E20) and commented that it is easy to read and understand (E1 and E6). Visualization expert E20 liked the overall layout of the view with power-up rows being placed in the middle, separate rows showing the lifespan of bombs per agent, and vertical lines connecting bombs and agents for kick events. Three experts (E2, E3, and E20) appreciated the visualization of interactions through vertical lines. They mentioned the usefulness of highlighting events through hovering on legend items (E6, E11, E16, and E20). Experts were also fond of the playback component and its linking with the timeline visualization (E10, E12, E13, and E16). The detailed design and interactions were found useful to explore strategies of agents (E1, E9, and E16). Feedback from *Pommerman* and AI expert E1 summarizes the observations:

> *"[I liked the] extremely detailed but simple and easy to understand visualization! I really like the detailed component. You can quickly identify patterns in an agent's behavior via the timeline visualization and watch them happen in the visual playback." – E1*

While many experts appreciated the amount of details, some mentioned that the timeline visualization is not easily readable (E11) and needs some time to understand (E2 and E19). The visualization contains too many circles (E12 and E15), which overlap (E2, E13, and E20) and make it a bit hard to understand or noisy (E2 and E20). The choice of colors in combination with the trans-

parency of the circles created confusion while reading the timeline (E12 and E16). Two *Pommerman* and AI experts (E2 and E6) highlighted the inability to zoom/scroll on the timeline which would have allowed them to better focus on a specific phase of a selected game. Two experts (E9 and E10) commented on the prominent central position of the power-up rows and instead suggested to use symbols for each power-up in the individual rows of agents. E16 mentioned to have solely relied on the playback component to find strategies, whereas E4 used the playback to uncover interactions between agents. Four visualization or game analytics experts (E14, E15, E16, and E18) suggested that including spatial information in the timeline visualization could be helpful to find position-based strategies. E15 recommended using heatmaps to show the most visited tiles over multiple games. It was also pointed that the histograms provide redundant information (E16) and are difficult to understand (E17) as they lack legends and interactions. With respect to additional features, E2 suggested to also include the option to select multiple actions at once, while computer game expert E18 proposed to show the appearance of a power-up in the timeline.

Competition Strategies: Almost all participants (19 out of 20) reported at least one competition strategy they discovered. Three experts mentioned that picking more power-ups in the early phase of a game gives the team an advantage (E9, E10, and E18). Seven experts (E1, E2, E5, E6, E10, E12, and E19) highlighted that the strategy of kicking bombs helps a team win the game in general, while the three *Pommerman* and AI experts among them (E1, E2, and E6) pointed out that kicking a bomb that is about to explode seems to be more effective. Four experts (E10, E11, E15, and E17) mentioned that laying more bombs helps a team to win more games. *Pommerman* and AI expert E6 was able to discover the strategy to lay a bomb to restrict the movement of opponents. In contrast, E3 observed that the *navocado* team *"places a lot fewer bombs, as bombs also constrain the safety of agents in contrast to the Skynet agent, which places more bombs."* Two experts (E6 and E11) commented that teams moved around a lot in order to avoid being killed. *Pommerman* and AI expert E7 observed two priorities:

> *"This tool makes it easier to understand which agents are using different kinds of reinforcement learning, either more focused on a safe agent or a more aggressive strategy trying to win." – E7*

Sometimes, agents used their own body to block movement of the enemy (E2). One *Pommerman* and AI expert (E4) mentioned that it is hard to see competition strategies speculating that *hakozakijunctions* might not have had sufficient computational resources.

Collaboration Strategies: Experts mentioned that agents of *hakozakijunctions* first engage in one-on-one combat with opponents (E2, E8, E14, E16, and E17) and, after killing one enemy, the two teammates team up against the remaining opponent by moving towards the enemy (E6, E8, E12, E16, E18, and E20). Five experts (E2, E8, E10, E16, and E20) highlighted the collaboration strategy to drive an enemy towards a corner of the board. *Pommerman* and AI experts observed that, when a teammate is near, agents move away (E5) or do not lay a bomb next to their teammate (E6). Expert E15 observed that agents seem to kill themselves while ensuring the death of an opponent. E1 also observed a similar behavior:

"The first hokazaki agent seems to be a lot less aggressive than the second hokazaki agent. It seems like the first agent tries to survive while the second tries to eliminate other agents." – E1

Five experts (E3, E4, E9, E14, E19) highlighted that it is hard to find collaboration strategies from the visualization. However, two *Pommerman* experts among them (E3 and E4) reasoned that the agents might not have learned complex collaboration strategies (*"I think Pommerman agents are still at a reactive strategy level and far from using more complex strategic behaviors." – E3*).

Differences between Behavior of Teams: The questionnaire asked participants to list observed differences between the behavior of teams. One games and visualization expert (E15) provided detailed feedback which summarizes the characteristic behaviors of different teams which were observed by other experts too (specified inside square brackets in the following). In particular, E15 mentioned (with other experts added having similar findings):

"Skynet:
- lay many bombs [E3 and E14] in the beginning [E1], if only one player is still alive or all wooden boxes are cleared → just keep moving to escape bombs (no own bombs are laid)
- defensive game play [E16]
- collecting power-ups is not a goal
- do not try to clear a path to the competing team or the own team mate [E12]
Hakozakijunctions:
- try to collect as many power-ups as fast as possible
- clearing a path just in one direction to one opponent
- lay many bombs throughout the game [E3, E10, E11, E17, and E19]
- use the kick power-up a lot [E3, E8, and E19]
Navocado:
- try to collect many power-ups [E10 and E17]
- clear a path to the opponent but not systematically [E18]
- lay few [E11 and E17] but targeted bombs [E6 and E11]" – E15

Other experts found additional behaviors but mentioned them without naming the teams. These behaviors include: agents idly waiting long times without performing any action or movement (E9), laying bombs on a regular interval (E19), action sequence pattern of lay bomb → kick → move (E13), and taking control of the diagonal field as a winning strategy (E2). *Pommerman* and AI expert E3 highlighted a behavior of the *hakozakijunctions* team—dropping many bombs followed by kicking them away—and mentioned that this is expected as it is a search-based agent.

Usability: The aggregated ratings on four characteristics of usability (self-explanatory, meeting one's requirements, usage being a satisfying experience, and ease of use) for the two disjoint groups of experts is presented in Table 2. All eight *Pommerman* and AI experts agree or strongly agree that the implemented tool is easy to use. Six of them agree or strongly agree that the tool is self-explanatory, meets their requirements, and using it is a satisfying experience. Two of them (E3 and E4) were neutral about the capabilities of the tool meeting their requirements. E3 wanted to see

high level statistics, while feedback of E4 lacks details: *"To check if my agents are working as expected."* One expert (E2) disagreed with the statement that the interface of the tool is self-explanatory which can be explained by a bug in the system he/she encountered and mentioned in the feedback—non-updating team labels and breaking the video player when switching competitions during video playback. The expert was among the first three participants of the study. It was not a critical bug and did not significantly impact the participants' answers, but we fixed the bug to avoid a repetition of a similar experience for the remaining participants.

The ratings of *Pommerman* and AI experts followed largely a similar trend as those of other experts, as shown in Table 2. The majority of them found the tool to be easy to use (#8), self-explanatory (#7), and to provide a satisfying experience (#10). However, two experts (E11 and E19) did not find the tool to be self-explanatory because it is hard to establish the linking of the number images of players between the playback and timeline visualizations (E11) and comparison features are missing (E19). Expert E19 mentioned a bug with the game lengths, but we were not able to reproduce it.

Three experts (E3, E13, and E15) mentioned that the interface contains too much information, which, as remarked by E3, *"is partly due to the nature of the game"*. All three suggested to either show details on demand or only show higher-level statistics. Two experts (E13 and E16) found that the icon used to show the bomb blast duration was unclear. Experts E17 and E19 mentioned that the help page of the tool was useful to understand the encodings in the visualization. Additionally, experts suggested to use a permanent selection of an action (E7 and E19) which we implemented in the follow-up version of the tool.

Additional Feedback: In terms of possible application scenarios, *Pommerman* and AI experts mentioned that they intend to use the tool for analyzing (a) the behavior of the agents they trained (E1, E2, E3, E5, E6, and E7), (b) improving their agent's performance (E1, E2, and E7), and (c) understanding the AI algorithm used for training (E8). Most of the experts commented that the visualizations encoded important information required for analysis. However, two experts (E10 and E19) highlighted that they did not use the histograms, with visualization expert E16 commenting that only one game metric (# Power Ups) was helpful while using the histograms. Four experts (E14, E15, E16, and E18) emphasized the importance of spatial aspects in *Pommerman* and one of them (E15) suggested visualizations such as heatmaps to show the density of player positions and bomb explosions. Experts also suggested to incorporate additional features such as the ability to sort the games in the summary component by any game metric or game length (E14), highlight only associated actions and bombs on selection of an agent (E14), and perform queries based on the strategies/patterns found (E6 and E12). Two experts (E8 and E14) proposed an interaction to jump to a particular game step by clicking on the game timeline rather than dragging the red status line. Experts also suggested to highlight when an agent was not able to make a decision within the 100 microseconds time limit (E4), to provide explanations of the histograms (E17), and to include messages shared between the agents of the same team (E20).

Table 2: *Results from the expert study showing the usability characteristics of the tool. A response on each characteristic was recorded on a five-point scale [Strongly disagree (SD), Disagree (D), Neutral (N), Agree (A), Strongly agree (SA)].*
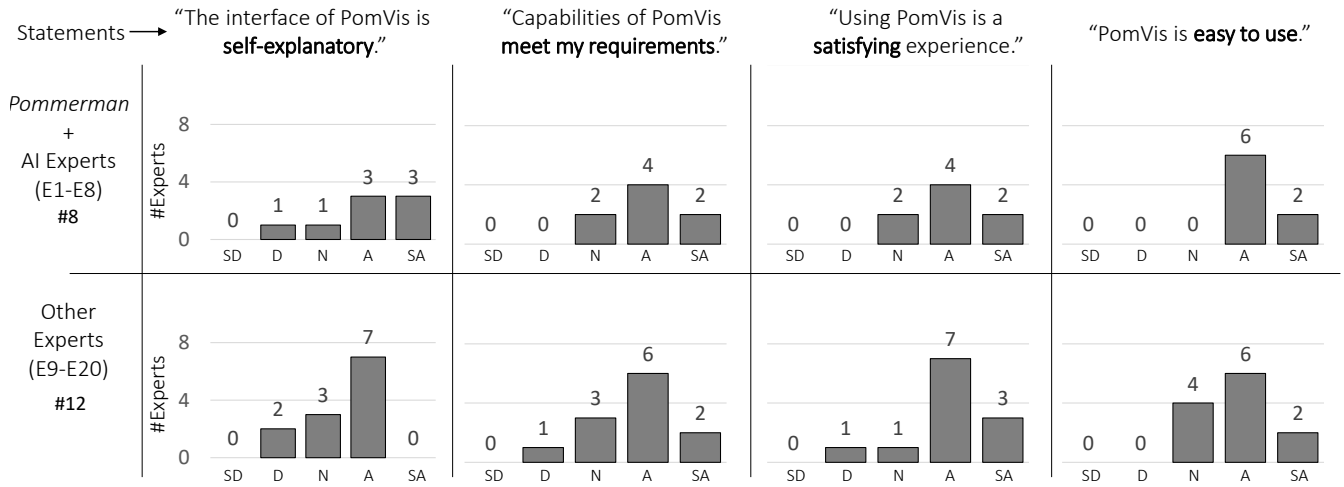
7.3. Validity and Limitations

We strove for participants with varying expertise to ensure evaluation from different perspectives. We also invited participants with high expertise to ensure quality in their feedback. It is, however, important to highlight that the authors had no previous connections with participants from the *Pommerman* community, who are the main target users of the tool. In contrast, the authors had background in visualization and game analytics. The questionnaire did not ask participants to perform any specific task, rather it asked users to explore the tool and describe their observations. Given the exploratory and qualitative focus of our study, we used a mixed-method analysis: qualitative analysis of the free text responses combined with quantitative indicators for usability and usefulness.

8. Discussion and Future Work

The results of the user study show that, in general, the approach is useful for understanding the behavior of agents, to help improve the performance of the agents, and to better understand the underlying AI algorithms. The majority of experts agreed that the tool is usable. But there is also a trade-off between providing all the necessary details while at the same time not being overly complicated. This appears interesting because complex behavior needs certain details to be communicated, but at the same time these details make the visualization more difficult to read.

The participants in the study were able to find many interesting strategies of the three top performing agents from the *Pommerman* 2018 competition. The study showed that by using the tool, they could identify competition and collaboration strategies. Furthermore, they were also able to find characteristic behaviors of different teams by exploring the data using the proposed approach. However, experts also highlighted drawbacks of the approach and provided valuable suggestions on how to address them. Additionally, experts requested more features as discussed next.

Include Spatial Information: A reoccurring theme in the evaluation was the lack of spatial features within the timeline visualization. We did omit such information for simplicity and only included the playback for spatial information. However, the collected feedback indicates that this was not sufficient. Having additional spatial indicators directly within the timeline could ease the identification of strategies based on certain environmental circumstances (e.g., the existence of blocks to hide behind). The highlighting of spatial properties on the timeline might help, such as an agent's proximity to other agents or bombs.

Extending the Approach to a Visual Analytics System: Participants requested additional features for querying, labeling the strategies, and finding occurrences of a pattern over all games in a competition. They also suggested to show higher-level statistics first and then present details on demand. These features point towards the extension of the approach to a visual analytics system.

In addition to the survey responses, some participants and other members of *Pommerman* community also shared informal feedback through Discord. Being able to communicate with a teammate is a new feature in the *Pommerman* 2019 competition. Community members suggested to include this information in histograms to reflect the temporal density of shared messages between teammates, which we implemented for a follow-up version of the tool. Community members also expressed interest in using the tool to illustrate the behavior of agents as part of presentations. The creators of the *Pommerman* environment also used our approach to analyze the behavior of winning agents in the *Pommerman* 2019 competition and to present the final results.[3] The approach was awarded for its usefulness in the *Pommerman* 2019 competition.[4]

The proposed approach is targeted at developers of agents for the *Pommerman* environment. However, going beyond AI agents,

---

[3] http://bit.ly/3cuxTDJ Accessed: March 2020
[4] https://twitter.com/Pommerman/status/1206101858336395264 Accessed: March 2020

the proposed approach could be extended to the analysis of human players. Also it would be applicable to other games where agents (or players) are split into two teams and the number of players is small. For instance, the approach could visualize multiplayer online battle arena games such as *League of Legends* [Rio09] where team coordination is essential. However, the visualization would not scale to many players. While other games may feature many more in-game items than *Pommerman*, in many cases, these can be restricted to a small number that are most important, for instance, capture points in *League of Legends*. More generally, we envision parts of the approach to be applicable for diverse applications where the analysis of interactions between entities (humans, robots, objects etc.) is important in a real or virtual environment, for instance, in a workshop or meeting, for remote assistance, etc.

9. Conclusions

We proposed *Bombalytics*, a novel approach to visualize the games played by autonomous agents in the multi-agent environment *Pommerman*. The approach allows users to explore competitions consisting of several games. It shows a summary of all games and details of a selected game through linked components. We demonstrated the implemented tool by analyzing competitions of top-performing agents and reported observed strategies. We also performed a study with experts from different domains. The results of the study showed that the participants were able to discover competition and collaboration strategies using the tool and rated the approach to be both useful and usable.

Acknowledgments

References

[ALA*10] ANDERSEN E., LIU Y.-E., APTER E., BOUCHER-GENESSE F., POPOVIĆ Z.: Gameplay analysis through state projection. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (2010), FDG, ACM, pp. 1–8. doi:10.1145/1822348.1822349. 2

[AWB20] AGARWAL S., WALLNER G., BECK F.: Supplementary material for Bombalytics: Visualization of competition and collaboration strategies of players in a bomb laying game, 2020. doi:10.17605/OSF.IO/2VXH4. 2, 7

[BBD08] BURCH M., BECK F., DIEHL S.: Timeline Trees: Visualizing sequences of transactions in information hierarchies. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (2008), AVI, ACM, pp. 75–82. doi:10.1145/1385569.1385584. 3

[BG97] BAKEMAN R., GOTTMAN J.: *Observing interaction: An introduction to sequential analysis*. Cambridge University Press, 1997. 1

[Bli98] BLIZZARD ENTERTAINMENT: StarCraft: Brood War. Game [PC], Dezember 1998. Blizzard Entertainment, Irvine, CA, USA. 2

[BQ11] BAKEMAN R., QUERA V.: *Sequential analysis and observational methods for the behavioral sciences*. Cambridge University Press, 2011. 2

[CAS19] CHANG K., AYTEMIZ B., SMITH A. M.: Reveal-More: Amplifying human effort in quality assurance testing using automated exploration. In *2019 IEEE Conference on Games* (2019), CoG, IEEE, pp. 1–8. doi:10.1109/CIG.2019.8848091. 3

[CMTD18] CANOSSA A., MAKAROVYCH S., TOGELIUS J., DRACHENN A.: Like a DNA string: Sequence-based player profiling in Tom Clancy's The Division. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference* (2018), AIIDE, AAAI, pp. 152–158. URL: https://aaai.org/ocs/index.php/AIIDE/AIIDE18/paper/view/18125/17236. 1

[CPYQ18] CHEN Y., PURI A., YUAN L., QU H.: StageMap: Extracting and summarizing progression stages in event sequences. In *2018 IEEE International Conference on Big Data* (2018), Big Data, IEEE, pp. 975–981. doi:10.1109/BigData.2018.8622571. 3

[DYK*19] DOUGLAS N., YIM D., KARTAL B., HERNANDEZ-LEAL P., TAYLOR M. E., MAURER F.: Towers of saliency: A reinforcement learning visualization using immersive environments. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces* (2019), ISS, ACM, pp. 339–342. doi:10.1145/3343055.3360747. 3

[EAGA*16] EL-ASSADY M., GOLD V., ACEVEDO C., COLLINS C., KEIM D.: ConToVi: Multi-party conversation exploration using topic-space views. *Computer Graphics Forum 35*, 3 (2016), 431–440. doi:10.1111/cgf.12919. 3

[Fin10] FINSTAD K.: The Usability Metric for User Experience. *Interacting with Computers 22*, 5 (05 2010), 323–327. doi:10.1016/j.intcom.2010.04.004. 7

[GHLKT19] GAO C., HERNANDEZ-LEAL P., KARTAL B., TAYLOR M. E.: Skynet: A top deep RL agent in the Inaugural Pommerman Team Competition, 2019. arXiv:1905.01360. 2, 3

[GJG*19] GUO S., JIN Z., GOTZ D., DU F., ZHA H., CAO N.: Visual progression analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 417–426. doi:10.1109/TVCG.2018.2864885. 3

[GKHLT19] GAO C., KARTAL B., HERNANDEZ-LEAL P., TAYLOR M. E.: On hard exploration for reinforcement learning: A case study in Pommerman. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (2019), vol. 15 of *AIIDE*, AAAI, pp. 24–30. URL: https://wvvw.aaai.org/ojs/index.php/AIIDE/article/view/5220/5076. 3

[GMN77] GOTTMAN J., MARKMAN H., NOTARIUS C.: The topography of marital conflict: A sequential analysis of verbal and nonverbal behavior. *Journal of Marriage and the Family 39*, 3 (1977), 461–477. doi:10.2307/350902. 2

[GXZ*18] GUO S., XU K., ZHAO R., GOTZ D., ZHA H., CAO N.: EventThread: Visual summarization and stage analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 56–65. doi:10.1109/TVCG.2017.2745320. 3

[Haz14] HAZZARD E.: Data visualization in games, 2014. http://vasir.net/blog/game-development/data-visualization-in-games Accessed: November, 2019. 2

[HLKT19] HERNANDEZ-LEAL P., KARTAL B., TAYLOR M. E.: Agent modeling as auxiliary task for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (2019), vol. 15 of *AIIDE*, AAAI, pp. 31–37. URL: https://wvvw.aaai.org/ojs/index.php/AIIDE/article/view/5221/5077. 3

[Hof19] HOFMANN K.: Minecraft as AI playground and laboratory. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (2019), CHI Play, ACM, pp. 1–1. doi:10.1145/3311350.3357716. 2

[Hou12] HOU H.-T.: Exploring the behavioral patterns of learners in an educational massively multiple online role-playing game (MMORPG). *Computers & Education 58*, 4 (2012), 1225–1233. doi:10.1016/j.compedu.2011.11.015. 2

[Hud83] HUDSON SOFT: *Bomberman*. Game [NES], July 1983. Hudson Soft, Tokyo, Japan. 2, 3

[KHLGT19] KARTAL B., HERNANDEZ-LEAL P., GAO C., TAYLOR M. E.: Safer deep RL with shallow MCTS: A case study in Pommerman, 2019. arXiv:1904.05759. 3

[KKK14] KANG S., KIM Y., KIM S.: Analyzing repetitive action in game based on sequence pattern matching. *Journal of Real-Time Image Processing 9*, 3 (2014), 523–530. doi:10.1007/S11554-013-0347-0. 1, 2

[KT12] KARAKOVSKIY S., TOGELIUS J.: The Mario AI benchmark and competitions. *IEEE Transactions on Computational Intelligence and AI in Games 4*, 1 (2012), 55–67. doi:10.1109/TCIAIG.2012.2188528. 3

[LFLB19] LI W., FUNK M., LI Q., BROMBACHER A.: Visualizing event sequence game data to understand player's skill growth through behavior complexity. *Journal of Visualization 22*, 4 (2019), 833–850. doi:10.1007/s12650-019-00566-5. 2, 3

[LJ14] LEECE M., JHALA A.: Sequential pattern mining in StarCraft: Brood War for short and long-term goals. In *Workshop on Adversarial Real-Time Strategy Games at AIIDE Conference* (2014), AAAI. URL: https://www.aaai.org/ocs/index.php/AIIDE/AIIDE14/paper/view/9082/9011. 1, 2

[LXC*17] LI Q., XU P., CHAN Y. Y., WANG Y., WANG Z., QU H., MA X.: A visual analytics approach for understanding reasons behind snowballing and comeback in MOBA games. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 211–220. doi:10.1109/TVCG.2016.2598415. 2, 3

[MG11] MICHAEL S., GRAHAM M.: Understanding player experience using sequential analysis. In *Proceedings of Think Design Play: The 5th International Conference of the Digital Research Association* (2011), DIGRA, DiGRA/Utrecht School of the Arts. URL: http://www.digra.org/wp-content/uploads/digital-library/11312.20491.pdf. 1, 2

[NENC15] NGUYEN T.-H. D., EL-NASR M. S., CANOSSA A.: Glyph: Visualization tool for understanding problem solving strategies in puzzle games. In *Proceedings of the 10th International Conference on the Foundations of Digital Games* (2015), FDG. URL: http://www.fdg2015.org/papers/fdg2015_paper_64.pdf. 2

[NTA*19] NGUYEN P. H., TURKAY C., ANDRIENKO G., ANDRIENKO N., THONNARD O., ZOUAOUI J.: Understanding user behaviour through action sequences: From the usual to the unusual. *IEEE Transactions on Visualization and Computer Graphics 25*, 9 (2019), 2838–2852. doi:10.1109/TVCG.2018.2859969. 3

[ODS18] ONO J. P., DIETRICH C., SILVA C. T.: Baseball Timeline: Summarizing baseball plays into a static visualization. *Computer Graphics Forum 37*, 3 (2018), 491–501. doi:10.1111/cgf.13436. 3

[OSM18] OSBORN J. C., SAMUEL B., MATEAS M.: Visualizing the strategic landscape of arbitrary games. *Information Visualization 17*, 3 (2018), 196–217. doi:10.1177/1473871617718377. 2, 3

[OT19] OSOGAMI T., TAKAHASHI T.: Real-time tree search with pessimistic scenarios: Winning the NeurIPS 2018 Pommerman Competition. In *Proceedings of The Eleventh Asian Conference on Machine Learning* (2019), PMLR, pp. 583–598. URL: http://proceedings.mlr.press/v101/osogami19a/osogami19a.pdf. 3

[PJHY20] POLK T., JÄCKLE D., HÄUSSLER J., YANG J.: CourtTime: Generating actionable insights into tennis matches using visual analytics. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 397–406. doi:10.1109/TVCG.2019.2934243. 3

[PLGD*19] PEREZ-LIEBANA D., GAINA R. D., DRAGESET O., ILHAN E., BALLA M., LUCAS S. M.: Analysis of statistical forward planning methods in Pommerman. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (2019), vol. 15 of *AIIDE*, AAAI, pp. 66–72. URL: https://www.aaai.org/ojs/index.php/AIIDE/article/view/5226/5082. 2

[PPYG18] PENG P., PANG L., YUAN Y., GAO C.: Continual match based training in Pommerman: Technical report, 2018. arXiv:1812.07297. 3

[PSM17] PFAU J., SMEDDINCK J. D., MALAKA R.: Automated game testing with ICARUS: Intelligent completion of adventure riddles via unsupervised solving. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play* (2017), CHI PLAY, ACM, pp. 153–164. doi:10.1145/3130859.3131439. 3

[REH*18] RESNICK C., ELDRIDGE W., HA D., BRITZ D., FOERSTER J., TOGELIUS J., CHO K., BRUNA J.: Pommerman: A multi-agent playground, 2018. arXiv:1809.07124. 2, 3

[Rio09] RIOT GAMES: League of Legends. Game [PC], October 2009. Riot Games, Los Angeles, CA, USA. 2, 11

[SH17] SHAPIRO B. R., HALL R. P.: Interaction geography in a museum. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (2017), CHI, ACM, pp. 2076–2083. doi:10.1145/3027063.3053146. 3

[SP19] SHNEIDERMAN B., PLAISANT C.: Interactive visual event analytics: Opportunities and challenges. *Computer 52*, 1 (2019), 27–35. doi:10.1109/MC.2018.2890217. 1

[Tog16] TOGELIUS J.: How to run a successful game-based AI competition. *IEEE Transactions on Computational Intelligence and AI in Games 8*, 1 (2016), 95–100. doi:10.1109/TCIAIG.2014.2365470. 2

[TSKY13] TOGELIUS J., SHAKER N., KARAKOVSKIY S., YANNAKAKIS G. N.: The Mario AI Championship 2009-2012. *AI Magazine 34*, 3 (2013), 89–92. doi:10.1609/aimag.v34i3.2492. 2

[Wal15] WALLNER G.: Sequential analysis of player behavior. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play* (2015), CHI Play, ACM, pp. 349–358. doi:10.1145/2793107.2793112. 1, 2

[WGSY19] WANG J., GOU L., SHEN H., YANG H.: DQNViz: A visual analytics approach to understand Deep Q-Networks. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 288–298. doi:10.1109/TVCG.2018.2864504. 3

[WK13] WALLNER G., KRIGLSTEIN S.: Visualization-based analysis of gameplay data – a review of literature. *Entertainment Computing 4*, 3 (2013), 143–155. doi:10.1016/j.entcom.2013.02.002. 2

[WK14] WALLNER G., KRIGLSTEIN S.: PLATO: A visual analytics system for gameplay data. *Computers & Graphics 38* (2014), 341–356. doi:10.1016/j.cag.2013.11.010. 2

[WZD*20] WANG J., ZHAO K., DENG D., CAO A., XIE X., ZHOU Z., ZHANG H., WU Y.: Tac-Simur: Tactic-based simulative visual analytics of table tennis. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 407–417. doi:10.1109/TVCG.2019.2934630. 3

[YT18] YANNAKAKIS G. N., TOGELIUS J.: *Artificial Intelligence and Games*. Springer, 2018. doi:10.1007/978-3-319-63519-4. 2

[ZGM*18] ZHOU H., GONG Y., MUGRAI L., KHALIFA A., NEALEN A., TOGELIUS J.: A hybrid search agent in Pommerman. In *Proceedings of the 13th International Conference on the Foundations of Digital Games* (2018), FDG, ACM, pp. 1–4. doi:10.1145/3235765.3235812. 3