HKUST SPD - INSTITUTIONAL REPOSITORY

Title

Authors

Source

Version

DOI

Publisher

Copyright

This version is available at HKUST SPD - Institutional Repository (<https://repository.ust.hk/ir>)

If it is the author's pre-published version, changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published version.

A Multi-Phased Co-design of an Interactive Analytics System for MOBA Game Occurrences

Quan Li[†], Ziming Wu[†], Peng Xu[‡], Huamin Qu[†], and Xiaojuan Ma[†]

[†] The Hong Kong University of Science and Technology, Hong Kong

[‡] User Experience Center, NetEase, Inc., Hangzhou, Zhejiang, China

{qliba, zwual}@connect.ust.hk, hzxupeng@corp.netease.com, {huamin, mxj}@cse.ust.hk

ABSTRACT

To ensure the playability of Multiplayer Online Battle Arena (MOBA) games, designers strive to balance different game occurrences. Although machine learning (ML) can help classify matches into different occurrence categories, designers demand more flexible input, interpretable output, and interactive collaboration with ML to facilitate analysis in breadth and depth. To this end, we work closely with a game company to design a visual occurrence analytics system through a stepwise co-design process. We first identify bottlenecks in game designers' conventional practices and their concerns about ML via an observational study. Then, we develop the single-match module of the visualization system to familiarize users with interactive analytics. Next, we incorporate ML models to recommend match segments of interest during occurrence classification and streamline the cross-match analysis. Empirical studies confirm the efficacy of our system. Experts' feedback suggests that our stepwise co-design process indeed helps them better embrace collaboration with machines.

ACM Classification Keywords

K.8.0. General: Games; H.5.m. Information Interface and Presentation (e.g., HCI): Miscellaneous.

Author Keywords

Game analytics; co-design; visualization; machine learning.

INTRODUCTION

Multiplayer Online Battle Arena (MOBA) games create immersive virtual gaming worlds in which two teams compete with each other to take down the opponent's base. Existing MOBA games such as *DOTA*, *League of Legends (LoL)*, *King of Glory*, etc. attract millions of players to invest their time and money, making MOBA one of the most profitable markets in the digital game industry. To succeed in fierce business competitions, game designers strive by all means to attract and keep players in the game to ensure its playability [17], which, in a multiplayer setting, particularly relates to teamwork satisfaction, a sense of fair competition, and mastery of complex gameplay interactions [19]. This motivates game designers to

identify the critical factors that may influence game dynamics by closely inspecting individual and team gameplay behaviors.

While game outcomes (*Win* or *Lose*) indicate the level of players' gaming skill and the effectiveness of gaming strategies, **game occurrences** capture game dynamics as the changes of a team's advantage (or disadvantage) over its opponent [25]. In this paper, we focus on three major occurrences: *snowballing*, *comeback*, *back and forth*. *Snowballing* occurs when a team easily achieves and maintains dominating advantages over its adversary throughout a match. *Comeback* occurs when a team overcomes a substantial disadvantage and eventually wins the battle. *Back and forth*, in comparison, defines the situations in which there is no clear winner until the end. To a great extent, the distribution of different occurrences of a game reflects its playability [23]. A game dominated by *snowballing* (or *comeback*) occurrences may inevitably raise concerns about its fairness. If the majority of the matches are *back and forth*, players may feel that there is no adequate differentiation or the sense of mastery. In either situation, the game engagement is at stake. Therefore, it is critical for game designers to identify and understand common patterns of key gaming behaviors that contribute to each type of game occurrences. They can modify game design settings accordingly to balance different types of game occurrences and optimize playability and engagement.

Conventionally, game experts classify game occurrences based on designed rules and then manually inspect a selected half-hour match for gameplay behaviors and events that possibly lead to its occurrence [25]. Nevertheless, manually designing rules to distinguish occurrences and inspecting individual matches are both time- and labor-intensive, and the experts have conceived ML to help them conduct game occurrence analysis. Previous research on game analytics employs ML as an effective means to predict the game outcomes, or extract the patterns that lead to such outcomes from aggregated data [2, 9, 14, 34, 36]. Particularly, for game designers intending to improve an existing game or even conceive a new one for better experiences, ML can enable the game to automatically customize its offers to the players and gaming context. However, given the characteristics of gameplay data, game designers may struggle to directly communicate with ML. Adapting general-purpose ML methods to solve a particular application problem, such as game occurrence reasoning, often requires significant domain knowledge. Furthermore, there has been little empirical study on the extent to which a design process can resolve the breakdowns in game designers' communication with ML analysis, bridge the expert knowledge with the investigation of MOBA game occurrences in large-scale data.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DIS '18, June 9–13, 2018, Hong Kong

© 2018 ACM. ISBN 978-1-4503-5198-0/18/06...\$15.00

DOI: <https://doi.org/10.1145/3196709.3196771>

In this paper, through a close collaboration with a game team, we explore how their conversation with our proposed visual occurrence analytics system evolves. We begin with an observational study of their current approaches, and identify their initial needs and concerns. Then, we build a single-match module of the system to familiarize the team with interactive analytics and co-design the input and output data representations to be inherited in the full system. Based on their feedback and derived insights, we further incorporate ML models in the full system to facilitate result reasoning and in-depth analysis based on multiple game matches. This system contributes to a commercial game that is already in the market and attracts the interest of the other MOBA game developing teams in the company. Our primary contributions include: 1) an interactive game occurrence analytics system that supports human-ML collaboration, and 2) insights derived from the stepwise co-design process into how to establish better communication with ML. The remainder of the paper is structured as follows. After providing the background about the game and reviewing related work, we present the three phases of our co-design process: Phase I about the observational study, Phase II about the co-design of the single match module, and Phase III about full system with a ML model incorporated. Finally, we detail our exploratory case study and conclude with the takeaways of designers and our reflection on the entire process.

BACKGROUND AND RELATED WORK

MOBA is a genre of real-time strategy games that involves two teams of players battling against each other with the objective of destroying the opponent's base. The game map is usually symmetric and each side contains defense towers and a base with two to three paths connecting the two camps. Each player manipulates a "hero" with specified skills. A hero can level up or be better equipped by attaining experience and earning cash through destroying an enemy character/structure or standing near such as incident. The game we showcase is *Wildfire*, the 4th commercial MOBA game released by *NetEase Games*¹ on July 20th, 2017. In this game, there are two special designed units besides the conventional heroes, monsters, towers, and base. One is *light tower*, which can heal its associated team's avatars. However, if taken over by the opponent, its field of view for detecting tactical actions of the adversary will be transferred to the occupant. The other is *trooper*, a neutral unit until it is hit by one camp, making it belong to this camp.

Game Analysis

Game Analysis identifies gameplay patterns for understanding and optimizing game designs. Previous research mainly investigates social interactions among players, e.g., teamwork, chatting, and trading [1, 4, 6, 7, 20], as it is easier to infer users' intention and attitude from such data. In contrast, gameplay data such as movements, tactic choices, and collective interactions, etc. are hard to make sense of compared to social interaction data, due to a lack of subjective feedback. In practice, game designers study MOBA game playability through surveys in online communities, interviews with players, and observations of trial play in the lab [15, 22], which tend to be on a limited scale and are rather costly [13].

¹<https://game.163.com/en/>, <http://wf.163.com/>

Visualization Techniques are effective to aid manual inspection of gameplay data, including subgraph matching and pathfinding [35], game metric changes [28], player pattern analysis [12], game behavior summarization [18], and game performance analysis [26]. Li et al. proposed a visual analytics system to enable the analysis of occurrence in a single MOBA game match data [25]. It only focuses on a single match, leaving it challenging to extract general gameplay patterns for domain experts. Additionally, many existing game data visualization systems have limited scalability and generality, and still require labor-intensive expert scrutiny [24, 25, 26]. *Machine Learning* has been introduced to ease game analysis in a larger scale. Supervised learning approaches are utilized, e.g., modeling combats as a sequence of graphs and extracting patterns by decision trees to predicate MOBA match success [38], introducing low-level topological clues to predict MOBA match outcomes by decision trees and SVM, etc. [32], classifying players based on roles of *Dota2* [14], analyzing and visualizing player behaviors based on the trajectory data [21], etc. These approaches require domain knowledge in feature engineering, which is hard and expensive in terms of time and expertise while in our work, we leverage the visual insights to infer ML input design. In addition, most of them aim at predicting the win/loss outcome rather than differentiating game phases and revealing the game dynamics. Some work utilize unsupervised methods to cluster players, and encode player's actions into sequences to mine sequential patterns [2, 13, 31]. *Deep Learning* (DL) has been recently applied to game analytics, which further reduces the task of manual specification of features [8]. More advanced ML such as *Reinforcement Learning* (RL) has achieved promising results in game AI design [30, 37]. However, it is primarily used to infer the optimized strategies for agents for the purpose of winning the game. Besides, adopting RL calls for extensive machine power, long training hours, and it is hard to define policy/reward functions. In our paper, the target is to understand human players' strategies. Most existing ML-based game analysis methods lack an intuitive interpretability without involving perception abilities of humans. Different from their work, we extract useful features through visual analytics that embeds human knowledge, and involve human's perception in explaining model output.

Human-in-the-loop Machine Learning

In HCI community, two general approaches are proposed to make ML more accessible. The first involves a set of ML algorithms in a specific domain and then building a tailored application and interface. While most efforts are devoted to make the system easier to use, the flexibility is limited [29]. Many systems were proposed to support users in image classification through decision trees [16], vision algorithms [27], building context-aware systems based on pre-defined events [10], etc. Another approach is to support a wider range of ML algorithms in a more general way and rely on users to configure them for specific domains [3]. This is more flexible but it requires users to decide which ML algorithms and features to use. Different from making a sole system to access ML, our approach leverages expert domain knowledge for scaffolding the system building process. Experts can get a taste of how to engage their domain knowledge in designing ML inputs and making sense of ML outputs through visualization.

PHASE I: OBSERVATIONAL STUDY

About the Game Team

We worked with a team of experts from *NetEase Games*, including one user experience analyst (E.1), one data analyst (E.2), and two game designers (E.3-4) (a typical setup for a game product in the company) to improve *wildfire*, a MOBA game already in the market. All of them have been in the game industry for more than four years and they have been involved since *wildfire*'s preliminary inception. One of the main responsibility of the game team after the game is launched is reducing the churn rate. According to the game team's analysis (Figure 1), the lowest churn rate can be achieved when the ratio of *snowballing* occurrence is about 0.42.

Figure 1. The 15-day churn rate first decreases and then increases as the percentage of *snowballing* occurrences goes up. The players are 7-day retained and have all played over more than 20 matches. The calculation of *snowballing* ratio is: if player A has played 20 matches, and 10 of them are *snowballing*, then the *snowballing* ratio for player A is 50%.

Study Procedure

Figure 2. User interviews and observational study with the game team.

Before the official launch, the game *wildfire* had gone through three small scaled trial runs from 9th, Feb to 26th, June of 2017. The purpose was to collect feedback from players, helping designers develop some initial assessment of their designs and predict distribution of the game occurrence. During the trial run period, with consent, we shadowed the team's daily working process, including video-taping how they observed players experiencing the game and conducted on-site interviews with the players, etc. (Figure 2). Later, we carried out retrospective analysis together with the game team on their conventional practices. The findings are summarized as follows.

Identified Findings

Designers' Conventional Practice and Bottlenecks

The overview of the conventional practice is shown in Figure 3. To study different game occurrences, the game team takes a rule-based approach to differentiate occurrences [25]. For a selected match under each occurrence, E.1 reviews the video playback or studies players battling in the observation room. E.2 manually selects several matches under this occurrence, inspects the details, and translates the gameplay statistics into human insights. E.2 especially pays attention to the cumulative changes of player behaviors and trajectories over time. Then, combining findings from both sources, E.3-4 infer the critical factors that may result in a certain game occurrence,

and improve the game settings accordingly. "Although the designed rules can classify some game occurrences, they can be expensive and cumbersome, and uncertainty exists whether more effective rules could be extracted to cover more occurrences," said E.1. To envision a more effortless way, the game team made an attempt to leverage ML to automatically classify different occurrences. However, they were not confident with the further step to reason the factors that may lead to different occurrences: "Currently, it largely depends on our expertise to elicit important events and timeframes and this is time-consuming," "we may spend 20-30 minutes on the elaborate observation and statistical analysis for each match," said E.1-2. Moreover, regarding the size of the game matches, the experts desire a more automatic way like ML to reason the occurrences beyond just classifying occurrences.

When we discussed with them on the possibilities by involving ML in game occurrence analysis, we further identified the following difficulties: **(1) Communication Barrier and Vague Requirements.** Game designers tend to have difficulties articulating the factors of interest to the ML specialists, as many concepts are ill-defined. At the initial stage of collaboration, the team only elicited needs as rather high-level concepts: "What did players do in real-time?", "What factors lead to different game occurrences?", "What is the relationship between game events and final game occurrence?" These requirements would be of no help to narrow the scope of design space. Besides, the complex/ambiguous collaborative activities exhibited by the game data require sophisticated analysis that cannot be handled by game designers alone, who are more accustomed to work with small-scale data [11]. **(2) Skepticism about Performance.** E.1-2 work together to make sense of game matches through observations and statistical analysis. In the initial discussion, they felt that their understanding of crucial factors behind each game occurrence may help design the input of ML models, but were not all confident in ML's ability to further their understanding in a large scale. They also had concerns about the understandability of ML model outputs. E.3-4 stressed that "ML is still like a baffling mystery to laymen like us." More complicated ML models like reinforcement learning are even more difficult for them to understand as they involve too many intelligent agents' decisions. They questioned the capability and credibility of ML outputs. On the one hand, designers usually visualize data and search for any correlations or patterns that 'make sense': that "fit with their understanding of how the virtual world should and does work", while ML in contrast finds machine-recognizable patterns of data relations that "can appear simplistic and even stupid to a person trying to make sense of the patterns" [11]. On the other hand, as indicated in [11]: "If ML is utilized, how 'deep' does it go... Can it be trusted to make decisions or take actions on its own?" **(3) Unclear Division of Responsibilities.** The team felt that given the scale and complexity of gameplay data, both ML and their domain knowledge can play an important role in the game occurrence analysis. However, the game designers had little experience working collectively with ML, especially on extracting ill-defined game behavior patterns that involves multiple dimensions (e.g. time, position, events, etc.), handling action sequences with high spatiotemporal variances, and differentiating co-occurrence from collaboration.

Figure 3. Conventionally, rule-based methods classify game occurrences and experts randomly select matches for analysis. The multi-phased co-design process scaffolds the system development: a single-match module to better inform ML model input and familiarize users with interactive analysis, back-end ML models are trained to identify segments of interest and cluster matches, front-end interface to facilitate interactive analysis.

Needs and Expectations of Collaborating with ML

Since there is a lack of tools for game designers to quickly implement game analytics with ML on their own [11], after several rounds of discussion with the game team, we realize the need for a preliminary prototype to ground the conversation and help the game team construct and articulate their thoughts. However, immediately providing the game team with an interactive analytics system with full-function ML may expose them to too many possibilities and uncertainties [11]. Therefore, we take multiple steps to achieve this. First, we develop a single-match module to assist the team’s current practice: that is, marking player activities and gaming events in a single match that can potentially differentiate the three types of MOBA game occurrences from a single match. Then, once the game team gets to see how visualizing gameplay data can streamline their routine analysis, we bring ML models into play. This visual occurrence analytics system leverages the visual insights from the first step to guide the ML model input design. What is more, it ranks potential match segments of interest and returns match clusters that share similar patterns. We present the multi-phased co-design process and how it scaffolds the communication with ML, visualization and users that result in the final analytical system in Figure 3.

PHASE II: DESIGN OF THE SINGLE-MATCH MODULE

Motivation and Design Considerations

The single match module assists the team’s current practice, to build trust, and get some initial insights into game occurrence-related factors, as well as verify potential visual designs of these factors. To elicit a clear requirement and narrow the design scope, we conducted a detailed discussion with the game team. “I want to see what the players did in the game and the associated resources outcomes,” said E.2, “for example, ‘jungling’-oriented players often kill the creeps and conduct ‘gank’, i.e., want someone to come in behind their enemies and kill them.” “I want to know whether the game rhythm meets my expectation,” E.3 added. We further enquired about the definition of “rhythm”. E.3 explained, “game rhythm depicts the advancements of key game events, such as destroying a tower, killing a hero, etc.” He gave us some straightforward examples, “launch an attack upon skill and key equipment upgrades.”, “react quickly upon the opponent’s mistake.” We

extract a set of features of interest from the comments: **F.1 Match Replay Simulation.** It should reveal the changes in the players’ movements and tactics, e.g., how each player plans to launch an attack, retreat or proceed. **F.2 Gameplay Events Summarization.** A clear record of time, location, and details, as well as the frequency of occurrence of important game events should be displayed. It allows analysts to better understand the effects of events on the game dynamics. **F.3 Gameplay Statistics Comparison.** The comparison of the total amount of *cash*, *experience*, and other achievements between the two teams along the timeline should be provided.

Visual Design

In general, MOBA games share the similar above-mentioned features. Although some MOBA games differ in the number of players, the variation is relatively small [25]. Therefore, the basic design principle is to cover the core features that can be shared or easily extended across multiple MOBA games, and to exploit or augment familiar visual metaphors to guide analysis and enable the experts to easily interpret the module output. Based on this, we design three following components: trajectory, rhythm and resource/equipment to visualize the above F.1-3 features, respectively.

Trajectory. We simulate the actual movement between two discrete positions on the game map to reconstruct a continuous trajectory [25]. Furthermore, players from the same team will form a bubble if they are close to each other. When the opponents meet, the bubble border will intensify (Figure 4).

Rhythm. Heat maps are often used to show aggregated data over a time period but fail to show details related to a specific timestamp. We use a “Y-axis position mapping” along the timeline [25]. The Y-positions of players are shown as smooth curves in a time series throughout the game (Figure 4 (1)). Noted that we use Y-axis because of the map design, since in this game, positions along the Y-axis that distinguishes the bases of the two camps can serve as a cue on which camp is marching towards the opponent’s base and which camp is in a defensive position. In addition, several critical events suggested by the experts are defined: bars on the timeline encode the scores of the results of *team combat* (Figure 4 (5)); rectangles on the top indicate the periods of *hitting towers*

Figure 4. (1) Y-axis positions are mapped along the time. (2) *Hitting towers* events from both teams are encoded by rectangles along the timeline. (3) A time slider to select the interesting time period by brush and drag. (4) Glyphs indicate when the occupying *light towers* happen. (5) *Team combats* are encoded by two vertical bars with scores indicating the results. (7) Critical events option enables users to click and display on the “Y-trajectories”. (8) Resource option consists of *cash*, *experience*, and *kills*. (9-10) Cumulative resource values are visualized by two areas with a black curve (6) indicating the difference between them. (11, 13) Different types of equipment upgrades and player-level resource changes are shown along the timeline (12).

(Figure 4 (2)); glyphs along the timeline show the occupying *light towers* events (Figure 4 (4)). Users can freely select other events by clicking the legend buttons (Figure 4 (7)).

Resource/Equipment. We visualize the changes of the cumulative resources (e.g., *cash*, *experience*, and *kills*) of both teams as areas (Figure 4 (9,10)), with a black curve in the middle showing the value discrepancy between two teams (Figure 4 (6)). We further visualize the resource changes at the player level to facilitate the analysis of player performance. Analysts can discover any sudden increases through the bars in the timeline (Figure 4 (13)). Equipment upgrades can also level up avatars’ skills. To capture this, an equipment upgrade timeline is presented with different type of equipment encoded by different glyphs and the level of equipment visualized by the size of the glyphs (Figure 4 (11)).

Interactions are provided to facilitate an efficient in-depth analysis. For example, when analysts select a time period by the time slider in the Rhythm, the detailed actions, values of the obtained resources, etc. will be shown in the GeoTimeline, which serves as an amplified information area of the selected time period. When users select certain data entry in one view, the corresponding information in other views will be highlighted. Besides, gameplay simulation presents an animation of players’ movement on the game map. When the time elapses during the animation, the time windows in other views will shift at the same pace.

Feedback and Discussion

Figure 5. Before *comeback*: (1) The *early advantaged team* won in *team combats*. (2) Two *inner towers* were destroyed. (3-5) The “purple” players assembled and proceeded, but were defeated.

E.1-2 conducted a case study of *comeback* occurrence by the single-match module using one *wildfire* game match that they

Figure 6. *Comeback*: The *comeback team* (1) killed the *troopers*, (2) destroyed one tower and (3) the *early advantaged team* was defeated and returned. The *comeback team* grasped the chance, (4) destroyed another *outer towers*, (5, 6) obtaining resources and (7) upgrading equipment.

themselves had experienced. They found that in the early stage of the match, the *early advantaged team* mainly overwhelmed the *combats*. The experts observed that before *comeback* occurred, after the *early advantaged team* destroyed two *inner towers*, they then assembled on one side of the map and decided to proceed, but were defeated by the *base towers* (Figure 5). After around 960 seconds (Figure 5 (*comeback time*))), it was shown that the *comeback team* started to gain advantages and marched towards their opponents’ bases (Figure 6). They killed the *troopers*, and then proceeded to destroy two *outer towers*. There were increases in the *comeback team*’s levels and equipment upgrades, which made the *comeback team* win *combats* in the later stages.

“My takeaway is the imbalanced proportion of rewards between *outer towers* and *inner towers*,” said E.1, “as you can see the *comeback* occurred after two *outer towers* were destroyed.” E.2 observed that “after two *inner towers* were destroyed, the *high attacking capability* blocked the *early advantaged team*’s advance.” They were excited to see that the insight obtained from this *comeback* case analysis was consistent with their subjective feelings when playing. “When we were playing, we felt that the gain from previous stage of the match could not compensate the loss, and this consequently let the *comeback team* able to outweigh us in the following battles”.

They were also happy to see that the prototype could largely reduce their analysis workload. “Previously, I needed to observe players playing games and took notes on site, which was quite stressful and time-consuming,” said E.1. Now he can quickly digest a match and easily detect when a team retreats or proceeds by Rhythm. “The interface is very concise and easy to

understand.” E.2 appreciated that the system makes their findings more compelling to the game designers, “*Previously I had to use statistical methods to extract latent features that were hard to understand by other people*”. They also appreciated the provided interactive mechanism, “*it helps us conduct an efficient and proactive analysis*,” said E.1. However, they also indicated that the GeoTimeline is not that necessary, “*this view might be too detailed and sometimes causes visual clutter for game occurrence analysis in a larger scale and we are clearly aware of the settings of values, like how much cash you gain after destroying a tower*.” Therefore, we may safely ignore these details in the full system design.

Expectations for the Full System

The game team all agreed that the single-match module brings them closer to the detailed match analysis. When we asked them to envision the possible need for inclusion of ML models, they shared an interesting and rather important observation made through our module. As shown in Figure 7, the cumulative resources discrepancy curves in terms of *cash*, *experience* and *kills*, as well as the movement trajectories across game occurrences are visually different. “*In snowballing, you can clearly see that the cash discrepancy curve climbs almost monotonically as the match progresses. The same trend occurs at the experience/kills discrepancy curves. However, in comeback/back-and-forth, the trend fluctuates and the gap even disappears at some timestamps*.” Similarly, the “Y-trajectories” presented in Rhythm reveal different patterns for the three type game occurrences. “*in snowballing, you can really see the trajectories are sweeping away towards the losing team’s camp, while in comeback they reverse their trend in the later stage*,” “*this is an exciting finding that we did not notice before*.” They wondered if such information could be leveraged in ML models to handle large-scaled match analysis.

Figure 7. The cumulative resources discrepancy curves in terms of *cash*, *experience* and *kills*, as well as the movement trajectories across *snowballing*, *comeback*, and *back and forth* are visually different.

General purpose ML can only classify matches into different occurrences. The experts still have to get back to one-by-one manual inspection to dig deeper into the results. Obviously, the single-match module is insufficient for larger-scaled gameplay pattern detection and verification, as it still requires experts to manually go through a match. “*It would be great to know which matches are more likely to contain typical patterns*,” said E.2. The experts felt that their choice of match for inspection was rather subjective, even random, which may not guarantee that the patterns extracted from the samples could adequately cover the most essential keys to a game occurrence, and that they could be generalized to other matches. “*It would be great to attain the occurrence patterns based on many matches*,” said E.3. “*I want to know how many patterns contribute to a certain occurrence, and the corresponding ratios*,” said E.2, “*then we can design different game mechanisms to tackle the issues*.” In sum, we should make better sense of ML output, and resolve the issues of “suggesting segments of interest” and “covering sufficient patterns”.

PHASE III: DESIGN OF FULL SYSTEM

To address the aforementioned issues, we design a full visual occurrence analytics system that incorporates ML models with interactive analytics. This system consists of a back-end ML model and a front-end interactive interface. The back-end model extracts and ranks critical data segments of interest deriving from a large set of gameplay data, as well as compiles a list of matches that potentially share similar patterns for further investigation. The front-end interactive interface reuses some confirmed visual designs from the single-match module to assist users to summarize possible patterns behind the candidate segments, and further explore the reasons why they lead to the associated occurrences.

Back-end ML Model Design

In this section, we introduce the back-end model design of our system that embraces the experts’ domain knowledge and facilitates result reasoning with human-in-the-loop. We first illustrate how the single-match module infers the ML model inputs. Then, we automatically identify critical segments that, with high probability, contain determinant patterns shared across many matches in a high-level feature space, in order to alleviate the effort of manual, clueless search for player behavior patterns among a big volume of gameplay data. Thus, the experts can summarize possible patterns, and further explore the reasons why they lead to the associated occurrences.

Classifying Game Occurrences

Conventional practice relies heavily on expert-designed rules to classify matches. ML models can make this classification more effective and cover sufficient occurrences. For ML models, we need to transform MOBA raw data into a representative feature space, and then apply general classifiers like *KNN*, *Decision Forest*, *SVM*, etc. to categorize matches.

Preparing MOBA Raw Data for Models. From the single-match module, the experts found that the cumulative amount of *cash*, *experience*, and *kills*, as well as player *positions* can significantly differentiate game occurrences. Therefore, we take them as the MOBA raw data for occurrence classification and the segments of interest determination. The raw data for each match has 16 dimensions: *cash*, *experience*, and *kills* of both teams contribute 6 dimensions while the rest is the 10 players’ *position*. The gaming period for each match is normalized to 3000s, suggested by the experts. Therefore, each game match is transformed into a 16×3000 matrix, serving as the input for our models.

Training Models. We collect 3000 real matches from the game company, of which 2500 matches are used for training while the rest for testing. All matches are labeled into three categories: *snowballing*, *comeback* and *back and forth* by the rule-based method by the experts [25]. Taking them as inputs, we train the classification models and we also compare the performance when only using *cash*, *experience*, *kills*, and *positions*. As shown in Table 1, although *RBF-SVM* achieves the best result by only using *kills*, due to the requirements that we want to study more information that could be involved as much as possible, and also the 16-dim feature has a comparatively good enough performance across all these classifiers, we finally select *RBF-SVM* as our model in this paper and

conduct the subsequent analysis based on the full-dim feature. Then, given an input in the testing phase, we obtain the 3-dim classification scores from the models, which correspond to the confidence that the input belongs to a corresponding category.

Classifier \ Feature	KNN	Decision Forest	RBF-SVM
Full (16-dim)	85.5%	85.5%	87.2%
Cash (2-dim)	84.2%	83.4%	84.8%
Exp (2-dim)	78.5%	80.2%	82.5%
Kills (2-dim)	81.0%	82.3%	88.0%
Pos (10-dim)	84.5%	85.8%	86.2%

Table 1. The performance of different classifiers on low-level features.

Identifying Segments and Clustering Matches

After training a ML model to classify game occurrences based on a large MOBA dataset, our goal then is to specify the key game segments that dominate the final occurrence. Specifically, we need to identify which segments significantly contribute to the final occurrence, denoting as segments of interest. However, lacking the label information that indicates the contribution of different segments of a match data makes it hard to estimate. Borrowing the idea from [5], we apply a patch-based noise perturbation to address this problem, in which, a certain amount of noise is added iteratively on the MOBA raw data within a pre-defined temporal range. Then we identify the perturbed patches that lead to severe performance decay in the classification task. Those patches with active response would be taken as the segments of interest. Figure 8 shows this step in detail. We identify which parts of the data contribute most to the classification outcome and output them as the segments of interest based on the trained discriminative ML model.

Figure 8. Identifying segments of interest: MOBA raw data are iteratively permuted by a certain amount of noise, which are then fed into a pre-trained ML model. With the classification score output by the model, we attain weights by measuring how much the original input’s classification score differs from that of the permuted samples, and finally weight time segments by averaging all matches of the same occurrence.

We propose a patch-based noise permutation strategy to infer which part of the input data significantly affects the classification result. To put it formally, we first define a segment as the continuous entries in patches of the input data with the predefined size of $W * H$, where W and H denote the width and the height of a segment, respectively. Concretely, H corresponds to the length of a temporal period while W is the dimension of the involved features. In our paper, H is 30 to cover the average time length (30s) of a game event suggested by the experts and W is 16 to cover the input dimensions. The permuted area that results in significant **performance decay** is taken as the segments of interest. To attain the estimation,

a patch-based noise perturbation process is conducted iteratively by generating additive noises on a $W * H$ patch with a certain stride. The noise follows uniform distribution as: $noise \sim uniform(-C * \sigma, C * \sigma)$ where σ denotes the standard deviation of x and C is a constant. Similar to [33], we take the perturbation process that follows the ordered sequence of input data. Specifically, the patch perturbation process can be considered as performed consecutively along timestamps. It can be modeled as: $x'_{ij} = x_{ij} + noise$ ($x_{ij} \in p_k, \forall p_k \subset x$), where p_k is the k^{th} patch in x with the size of $W * H$ and x_{ij} is the i^{th} data point of p_k . Then the weight of p_k can be attained by: $W_k = \frac{1}{N} \sum H(f(p'_k), f(p_k))$ where W_k corresponds to the weight indicating how much the performance decay resulted from a permuted patch p'_k . The larger W_k is, the more important p_k is. $f(x)$ is a 3-dim output from the designated model in which the i^{th} dimension scores the confidence that the model believes the input x belongs to the i^{th} class. $H(A, A')$ is an estimator which measures the difference between the classification results of the original data A and permuted data A' . The most intuitive way to measure the difference is using metric distance, e.g. Euclidean distance. However, it would lead to bias since the sum of the score difference of the outer classes would dominate the overall distance while the score of the i^{th} dimension is actually more crucial if X belongs to the i^{th} class. To eliminate the bias, we propose a class-based estimator: $H(A, A') = |(a_i - a'_i) - \max(a_j - a'_j | \forall a_j \in A, a'_j \in A' \text{ and } j \neq i)|$ where i denotes the index of the category that A belongs to. We also leverage the 16-dim raw MOBA data and takes them as the feature representation of a match, which is then projected to a two-dimensional space using the principle component analysis (PCA) for visualization. Each occurrence is further clustered by k-means algorithm (default k is 5).

Front-end Interaction Design

To bridge designers with the back-end model, we extract two additional features that our system should include: **F.4 Segment Summarization**. This enables the experts to observe the overall distribution of the significance of effect on the confidence for occurrence classification throughout the unified gaming period. **F.5 Match Recommendation**. To confirm the “generality” of a pattern, matches that share a similar pattern should be clustered and recommended. Therefore, we re-designed the interface (Figure 3): (1) **Weight View** visualizes the weight of each occurrence output by the ML model (F.4). Segments with higher weights contribute more to the final occurrence. To understand the major behaviors in these segments, we extract the *combat*, *occupying light towers*, and *equipment upgrades* in terms of two teams (**win** and **loss**) for each occurrence and display the changes of their values and discrepancies over time. (2) **Cluster View** clusters matches to investigate the reasons for a certain game occurrence (F.5). It consists of a PCA subview for occurrence selection and k-means subview for match selection. Matches represented by nodes within a cluster share similar features. We incorporate the single-match module but remove GeoTimeline suggested by the experts. User can select one occurrence and further click a node representing a match for detailed analysis.

Figure 9. For *snowballing*, the first 1/6 stage and the last 1/10 stage of the entire gameplay time contribute most to the final occurrence. For *comeback*, four “comeback time-points” are witnessed. The *back and forth* is quite similar to *comeback* at these “comeback periods” except for the last stage.

SYSTEM EVALUATION WITH A COMMERCIAL GAME

To evaluate the efficacy of the system, we conducted two case studies with the game team (E.1-4), including comparing the segments of interest of different game occurrences by Weight View, and investigating the match clusters for each occurrence by Cluster View, followed by an interview discussion. The two cases are based on the results of the designated trained ML model on the 500 testing matches.

Breadth: Segment Analysis Across Occurrences

From Figure 9 (A), the experts noticed that in the first 1/6 stage, the weights are high for *snowballing* occurrence, followed by a gradual decreasing. Another peak happens at the last 1/10 stage. This gives the experts some confirmation that “in the initial stage, the *snowballing* team quickly accumulated the resources and enlarged the gap between the two teams”, as they found that the *snowballing* team had more *team combats* events (Figure 9 (1)), leading to quick equipment upgrades for the winner side (Figure 9 (2)). They also witnessed that nearly all the *light towers* of the losing team were occupied over the entire gaming period (Figure 9 (3)). For *comeback* occurrence, the experts witnessed some “twists and turns” and four “comeback periods” (around 800s, 1800s, 2400s, 2800s), “there should be several moments that the two teams were fighting for some resources,” said E.2. Around 800s, the *comeback* teams mainly worked on earning resources, e.g. *cash* and this may further lead to an increasing number of *defense equipment* upgrades (Figure 9 (5)) more than its opponent. From the trend of the number of *team combats* (Figure 9 (4)), the *comeback* took place. The experts observed that at around 1800s, the *comeback* team started to have an overwhelming performance on *team combats*, occupying *light towers* of the other team (Figure 9 (6)). In the last two “comeback periods” (2400s, 2800s), this *comeback* advantage continued to the end as all

the *light towers* of the losing team were occupied, reversing the previous trend (Figure 9 (7)). For *back and forth* occurrence, the experts found except for the initial and final stage, it is quite similar to *comeback* as those “comeback periods” also exist, which means that “both teams were struggling for opportunities for success”. “In the initial and final stage, the weights of segments of interest are much higher than that in *comeback*”. They found that there was an increasing *team combats* and equipment upgrades events (Figure 9 (8)) in the final stage of the game in *back and forth*.

Depth: Cluster Investigation for Each Occurrence

We leveraged the clustering results of matches derived from the MOBA raw data in the testing data set. The system recommends similar matches and the experts mainly focus on those critical segments suggested by the ML model.

Comeback Investigation. The experts verified the patterns found by the single-match module that the imbalanced proportion of rewards received between destroying *outer* and *inner towers*, and the high attacking capability of *base towers*. They agreed that lowering the attack of the *base tower* and the reward for destroying the *outer tower* could be one of the solutions for improvement. They also observed (See Figure 10) some unwise actions of certain players (players’ own misjudgment) in the early-advantaged team (happened around first 1/3 stage), as shown by the trajectories. Around 3/5 stage, players were seizing the precious resources, e.g. *troopers*, to help them revise the occurrence. “The neutral resources such as *troopers* should contribute significantly to create *comeback opportunities*,” said the experts. Later at the final 1/7 stage, the experts saw a conservatively defensive strategy of the *comeback* team: “the *comeback* team made concessions in order to gain advantages,” said E.2. These

Figure 10. Three occurrences are well separated. *Comeback* matches are further clustered. Typical *comeback* pattern set includes “unwise actions of players”, “defensive strategies of *comeback* teams” and “seizing the precious resources”: (1) two “green” players return, but an opponent (the “purple” player) chases them alone and gets killed. (2) Then, “green” players destroy two towers (indicated by two green dots), (3) followed by equipment upgrades, and (4) win in the following team combats and finally make the *comeback* happen. (5) The *comeback* team wins in combats, (6) defeats the opponents in their field and finally revises the match with (7) equipment upgrades. In the early stage, (8) the *comeback* team is defeated in most of the team combats. However, (9) they kill many troopers, which helps them win in the following team combats.

observations are also in accordance with the previous four “comeback time-points”. Also, the ratio (around 11.1%) for this defensive strategy makes sense, “since it requires highly close collaboration, while in reality, players are often ‘unwise’ and ‘impulsive’,” said E.3-4.

Back and Forth Investigation. The experts found that this occurrence is quite similar with *comeback*, except for more team combats on the victory team side in the final stage. In *back and forth*, the well-matched competition between the two teams lasts for a longer time than that in *comeback* until the final stage of the match: “both teams have a good chance for winning the game, and the final pushes settle the match”, while in *comeback*, “the early advantaged team is unable to reverse the already established occurrence and loses the game.”

Snowballing Investigation. Through the similar approach, apart from two quick findings (cumulative advantages by winning team combats and equipment upgrades, and overwhelming light tower occupation), the experts also identified that two representative clusters are due to passive resistance of the disadvantaged teams in terms of “on-hook behaviors” and “giving up the defending effort”. “I see a lot of abnormal trajectories of some players in disadvantaged teams, and these demonstrate the players are ‘Away From Keyboard’ (one’s absence from the computer when still appearing to be present),” said E.3. Besides, “all the hitting towers are conducted by the snowballing teams, indicating that the disadvantaged teams present less efforts on destroying opponents’ towers.” Another frequent pattern is the huge gap between the performance of the teams, and even an unbalanced number of players in the beginning of the match. “The unbalanced performance of players can be resolved by modifying the player matching mechanisms,” said E.4. Besides, the design of offering the weaker some incentives at the early stage could be a possible way to avoid overwhelming snowballing. And they reached a consensus that light towers are too favorable for the snowballing teams, making it difficult for the disadvantaged teams to compete for the resources for the future development. “Actually, we are considering modifying some mechanisms of the light towers in our next game version,” said E.4.

DISCUSSION AND REFLECTION

System Performance and Experience

We first asked the experts to evaluate the results of ML models, such as the accuracy of the extracted segments, since we did not have labeled ground truth. E.3-4 reported that “the four ‘comeback periods’ make sense since the teams may contend for some resources, but the details are not very intuitive to judge at first.” Through the cluster analysis, they confirmed that in the initial stage some unwise player behaviors of the early-advantaged team may contribute to the *comeback* and the conservatively defensive strategy in the final stage is another way for the *comeback*. In a word, the further analysis can provide designers with more concrete information to establish explanation of ambiguous segments generated from the ML model. Besides, our manner demonstrates the capability of revealing meaningful game segments that contributes to game analysis and the extracted key segments of the typical game occurrences are meaningful compared to what the experts can identify manually: “the coverage of potential segments can sufficiently facilitate sense making of ML output”.

Conventionally, the game team needs to watch video replays, and manually marks the critical events: “the time we spent on the entire process was about 30 minutes, since sometimes we had to replay a certain video segment many times,” said E.1. Our system can display the visualization of one game-play match in real-time (about 3-5 seconds with 2.5 GHz Intel Core i7 processor and 16 GB memory), making it easier for the experts to interact with the data rather than go through the whole match, and largely shorten the analysis time (now 5-10 minutes for each match). The system further streamlines the process as the experts no longer need to randomly select matches from a large collection and specify segments in each match by hand. Instead, all possible segments of interest are automatically retrieved from a list of matches and similar matches are clustered, even though it still requires the experts’ proactive investigation at the segment level. E.2 was satisfied with the system’s ability to extend the scale of analysis and achieve sufficient patterns for each occurrence, “it helps me learn which segments play an important role in classifying occurrences and quickly find similar matches to confirm my

assumptions.” With this system, the team was able to identify key reasons for typical occurrences. They continuously optimized the design, e.g., balancing the rewards of towers, monsters, etc. to maintain a balanced occurrence ratio, and the satisfaction of online gaming experience improved: “the weekly retention rate rose by 1.1%,” said E.2.

Generalizability

The successful deployment of the system captured interests from other MOBA game developing teams and they all indicated that it can be potentially beneficial to their initial game designs as well. To further explore such needs, we subsequently invited two designers from other teams, and found that they all called for the features identified earlier (F.1-F.3): “the system can unify the MOBA analysis process, and we can follow the similar procedure to analyze a new game”. We then discussed which component(s) of our system can be directly applied to other games and which one(s) need customization. Three findings arose: (1) **Requirement Generalization.** The designers all agreed that occurrence analysis is meaningful as it sheds light upon game playability. “A successful MOBA balances itself well on each occurrence and we can learn from it when designing our own game.” They appreciated that we have generalized this concern into a measurable problem. (2) **Visual Design Generalization.** Most MOBA maps are diagonally symmetrical rather than longitudinal like *wildfire*. Thus, the “Y-position mapping” is not applicable. One intuitive solution is to rotate the map by 45 degrees. However, designers found it no longer intuitive to read, “the diagonal design is so classic”. Therefore, we measure the distances from trajectories to the diagonal as “Y-positions”. Besides, typical MOBAs have two units that are missing in *wildfire*: *Buff*, a beneficial boost to performance, and *Dragon*, a powerful neutral monster, slaying which grants the team *buffs*. Hence, we need to update the set of critical game events. (3) **Model Generalization.** ML models classify occurrences and extract segments of interest in the testing dataset. “We can just follow the same way and train the models by using the e.g. extracted 16-dim data from the new game,” said one designer. However, they also indicated that the cost of training an optimal model for each MOBA game to classify game occurrences is comparatively high, “making a set of unified rules across multiple games may be easier adopted and cost-effective for promotion.” Another issue is a lack of enough training data degrades the performance of the model, “we can add a feedback option indicating game occurrences on the game interface. When players finish a match, they can help us label the match,” said E.2.

Reflection

Balance Between Value and Cost. We realize that game teams have different levels of conservative decisions against the new technique due to several reasons. First, developing such a ML-based system demands costly collection of massive labeled data. Second, the iterative co-design intensively occupies the human resource and time of the game teams. Therefore, while we argue that game occurrence analysis is vital and worth the initial investment, it, in some cases or at a wrong time, might conflict the primary design goal for game development that may focus on the initial game design and avoid significant cost, e.g. log data storage and management.

Involvement of all Parties of Interest. Despite that we work mainly with the designers, we realize the importance of also involving programmers at an early stage, who are responsible for logging match data. Although we designed a decent log format as a guideline for programmers, some fields were still missing or mistaken, causing difficulty in data parsing. We also spent a lot of time fixing mapping issues due to low map resolution and incorrect map coordinates given by programmers. If we had brought them into the conversations earlier, we would have avoided these issues. Besides, if we had anticipated that our system would attract a wide range of game teams, we could engage them earlier to ensure the flexibility of our designs as well.

Communication Tactics. Designers often have misconceptions about what ML can actually do and produce, as many of them manage it much like magic, or “being collaborative with other professions” [11]. We need to realize that ML and designers treat data quite differently. Therefore, a preliminary system that imitates the conventional practice taken by the experts may be a good start. Thus, they would not be overwhelmed and intimidated by working with something unknown to them. Instead, they have time and space to gain knowledge, build trust, and ultimately embrace innovation. Meanwhile, we can take the chance to test the initial visual designs for later use. In our case, the single-match module improves designers’ understanding of and interaction with low-level input features and the ML output results via visual aids.

Limitation

First, the system may fail to capture some critical incidents due to the fixed parameter setting, or weighting segments by averaging over all matches of the same occurrence, etc. Second, the model currently extracts segments as if they were independent of one another. Third, only a few experts (E.1-4) are involved in the evaluation and we cannot provide a quantitative assessment of the ML models, e.g. the accuracy of segments based on labeled ground truth.

CONCLUSION AND FUTURE WORK

MOBA game designers strive to ensure the playability by reasoning and balancing different game occurrences. Through a multi-phased co-design process with a commercial game team, we scaffold an interactive analytics system to tackle this issue. Empirical studies verify the efficacy of our system. For the future work, we will implement mechanisms to optimize the fixed parameter setting, take potential inter-event attribute dependency into account, and experiment with joint perturbation across two or more segments. Also, we will perform a systematic evaluation involving more experts. With more analytical needs in an increasing number of MOBA games, this system can be refined continuously.

ACKNOWLEDGMENTS

The authors would like to thank the game experts for providing the feedback and the anonymous reviewers for their valuable comments. This paper was supported by funding from the Theme-based Research Scheme of the Hong Kong Research Grants Council, project number T44-707/16-N.

REFERENCES

1. Chee Siang Ang and Panayiotis Zaphiris. 2010. Social roles of players in MMORPG guilds: A social network analytic perspective. *Information, Communication & Society* 13, 4 (2010), 592–614.
2. Guillaume Bosc, Mehdi Kaytoue, Chedy Raissi, and Jean-François Boulicaut. 2013. Strategic pattern discovery in rts-games for e-sport with sequential pattern mining. In *Proceedings of the ECML/PKDD 2013 Workshop on Machine Learning and Data Mining for Sports Analytics (MLSA2013)*.
3. Jamie Bullock and Ali Momeni. 2015. ML. lib: robust, cross-platform, open-source machine learning for max and pure data.. In *NIME*. 265–270.
4. Chien-Hsun Chen, Chuen-Tsai Sun, and Jilung Hsieh. 2008. Player guild dynamics and evolution in massively multiplayer online games. *CyberPsychology & Behavior* 11, 3 (2008), 293–301.
5. Li Chen, Dmitry B Goldgof, Lawrence O Hall, and Steven A Eschrich. 2007. Noise-based feature perturbation as a selection method for microarray data. In *International Symposium on Bioinformatics Research and Applications*. Springer, 237–247.
6. Vivian Hsueh-hua Chen, Henry Been-Lirn Duh, and Hong Renyi. 2008. The changing dynamic of social interaction in World of Warcraft: the impacts of game feature change. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. ACM, 356–359.
7. Helena Cole and Mark D Griffiths. 2007. Social interactions in massively multiplayer online role-playing gamers. *CyberPsychology & Behavior* 10, 4 (2007), 575–583.
8. Marjolein de Vries and Pieter Spronck. 2016. Deep learning for classifying Battlefield 4 players. In *International Conference on Intelligent Technologies for Interactive Entertainment*. Springer, 161–166.
9. Marjolein de Vries and Pieter Spronck. 2017. Deep learning for classifying Battlefield 4 players. In *Intelligent Technologies for Interactive Entertainment: 8th International Conference, INTETAIN 2016, Utrecht, The Netherlands, June 28–30, 2016, Revised Selected Papers*. Springer, 161–166.
10. Anind K Dey, Raffay Hamid, Chris Beckmann, Ian Li, and Daniel Hsu. 2004. a CAPpella: programming by demonstration of context-aware applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 33–40.
11. Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 278–288.
12. Anders Drachen and Alessandro Canossa. 2009. Analyzing spatial user behavior in computer games using geographic information systems. In *Proceedings of the 13th international MindTrek conference: Everyday life in the ubiquitous era*. ACM, 182–189.
13. Anders Drachen, Christian Thureau, Julian Togelius, Georgios N. Yannakakis, and Christian Bauckhage. 2013. *Game Data Mining*. Springer London. 205–253 pages.
14. Christoph Eggert, Marc Herrlich, Jan Smeddinck, and Rainer Malaka. 2015. Classification of player roles in the team-based multi-player game dota 2. In *International Conference on Entertainment Computing*. Springer, 112–125.
15. Magy Seif El-Nasr, Heather Desurvire, Bardia Aghabeigi, and Anders Drachen. 2013. Game analytics for game user research, Part 1: A workshop review and case study. *IEEE computer graphics and applications* 33, 2 (2013), 6–11.
16. Jerry Fails and Dan Olsen. 2003. A design tool for camera-based interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 449–456.
17. Alessandro Febretti and Franca Garzotto. 2009. Usability, playability, and long-term engagement in computer games. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 4063–4068.
18. Nate Hoobler, Greg Humphreys, and Maneesh Agrawala. 2004. Visualizing competitive behaviors in multi-user virtual environments. In *Visualization, 2004. IEEE. IEEE*, 163–170.
19. William H Huber. 2009. Epic spatialities: the production of space in Final Fantasy games. *Third person: Authoring and exploring vast narratives* (2009), 373–384.
20. Daniel Johnson, Lennart E Nacke, and Peta Wyeth. 2015. All about that base: differing player experiences in video game genres and the unique case of moba games. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2265–2274.
21. Shin-Jin Kang, Young Bin Kim, Taejung Park, and Chang-Hun Kim. 2013. Automatic player behavior analysis system using trajectory data in a massive multiplayer online game. *Multimedia tools and applications* 66, 3 (2013), 383–404.
22. Tihana Lapaš and Tihomir Orehovački. 2015. Evaluation of user experience in interaction with computer games. In *International Conference of Design, User Experience, and Usability*. Springer, 271–282.
23. Ryan Leigh, Justin Schonfeld, and Sushil J Louis. 2008. Using coevolution to understand and validate game balance in continuous games. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 1563–1570.

24. Quan Li, Qiaomu Shen, Yao Ming, Peng Xu, Yun Wang, Xiaojuan Ma, and Huamin Qu. 2017. A visual analytics approach for understanding egocentric intimacy network evolution and impact propagation in MMORPGs. In *Pacific Visualization Symposium (PacificVis), 2017 IEEE*. IEEE, 31–40.
25. Quan Li, Peng Xu, Yeuk Yin Chan, Yun Wang, Zhipeng Wang, Huamin Qu, and Xiaojuan Ma. 2017. 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.
26. Quan Li, Peng Xu, and Huamin Qu. 2015. Fpsseer: Visual analysis of game frame rate data. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*. IEEE, 73–80.
27. Dan Maynes-Aminzade, Terry Winograd, and Takeo Igarashi. 2007. Eyepatch: prototyping camera-based interaction through examples. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 33–42.
28. Ben Medler, Michael John, and Jeff Lane. 2011. Data cracker: developing a visual game analytic tool for analyzing online gameplay. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2365–2374.
29. David A Mellis, Ben Zhang, Audrey Leung, and Björn Hartmann. 2017. Machine Learning for Makers: Interactive Sensor Data Classification Based on Augmented Code Examples. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. ACM, 1213–1225.
30. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).
31. Aline Normoyle and Shane T Jensen. 2015. Bayesian clustering of player styles for multiplayer games. *Proc. AAAI AIIDE* (2015).
32. François Rioult, Jean-Philippe Métivier, Boris Helleu, Nicolas Scelles, and Christophe Durand. 2014. Mining tracks of competitive video games. *AASRI Procedia* 8 (2014), 82–87.
33. Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems* (2016).
34. Marjolein De Vries and Pieter Spronck. 2016. *Deep Learning for Classifying Battlefield 4 Players*. Springer International Publishing.
35. Günter Wallner and Simone Kriglstein. 2014. PLATO: A visual analytics system for gameplay data. *Computers & Graphics* 38 (2014), 341–356.
36. Hao Wang, Hao-Tsung Yang, and Chuen-Tsai Sun. 2015. Thinking style and team competition game performance and enjoyment. *IEEE Transactions on Computational Intelligence and AI in Games* 7, 3 (2015), 243–254.
37. J Willich. 2015. Reinforcement learning for heroes of newerth. *Bachelor thesis, Technische Universität Darmstadt* (2015).
38. Pu Yang, Brent E Harrison, and David L Roberts. 2014. Identifying patterns in combat that are predictive of success in MOBA games.. In *FDG*.