

A Visual Analytics Approach for Understanding Reasons behind Snowballing and Comeback in MOBA Games

Quan Li, Peng Xu, Yeuk Yin Chan, Yun Wang, Zhipeng Wang, Huamin Qu, *Member, IEEE*, and Xiaojuan Ma

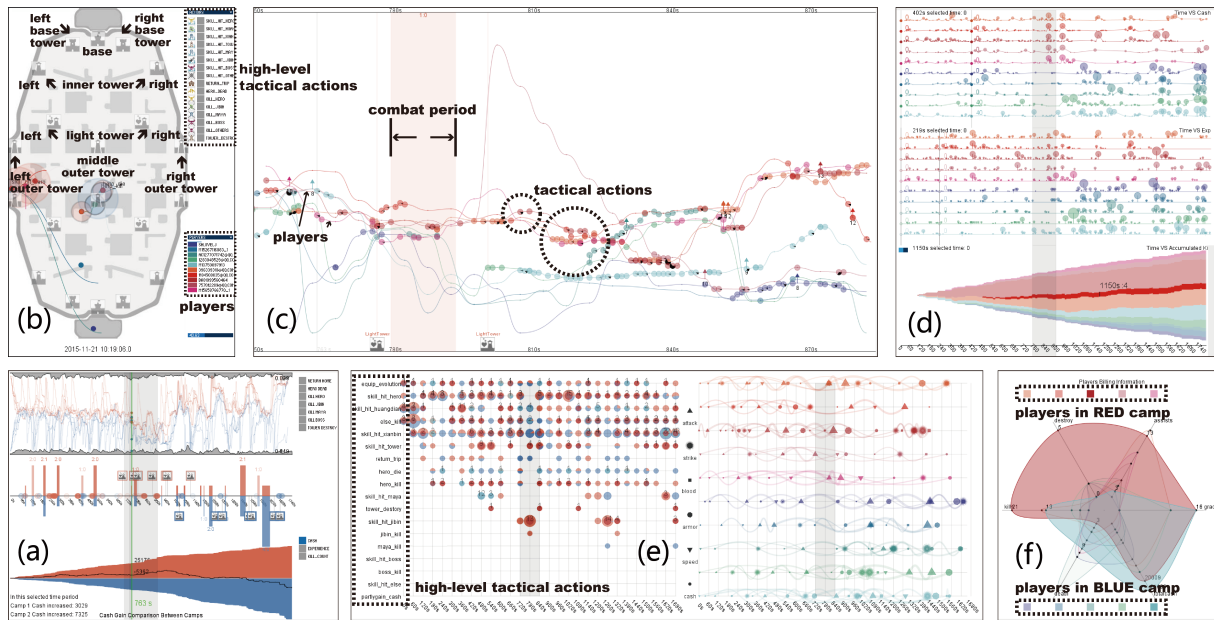


Figure 1. A match with *comeback* occurrence. (a) Trend View discloses the trend of game play during a match. (b) Trajectory View simulates the game replay. (c) Tactic Geographical Timeline View presents details of players' behavior in the time period of interest. (d) Resource Time Sequence View displays the accumulated resources and changes in resources of each player. (e) Tactic Comparison View (Left) unfolds the temporal dynamics of all the tactical actions in two camps while Equipment Evolution View (Right) shows the equipment evolution hierarchies. (f) Player Billing Radar View represents the statistical information of each player.

Abstract— To design a successful Multiplayer Online Battle Arena (MOBA) game, the ratio of *snowballing* and *comeback* occurrences to all matches played must be maintained at a certain level to ensure its fairness and engagement. Although it is easy to identify these two types of occurrences, game developers often find it difficult to determine their causes and triggers with so many game design choices and game parameters involved. In addition, the huge amounts of MOBA game data are often heterogeneous, multi-dimensional and highly dynamic in terms of space and time, which poses special challenges for analysts. In this paper, we present a visual analytics system to help game designers find key events and game parameters resulting in *snowballing* or *comeback* occurrences in MOBA game data. We follow a user-centered design process developing the system with game analysts and testing with real data of a trial version MOBA game from NetEase Inc.. We apply novel visualization techniques in conjunction with well-established ones to depict the evolution of players' positions, status and the occurrences of events. Our system can reveal players' strategies and performance throughout a single match and suggest patterns, e.g., specific player' actions and game events, that have led to the final occurrences. We further demonstrate a workflow of leveraging human analyzed patterns to improve the scalability and generality of match data analysis. Finally, we validate the usability of our system by proving the identified patterns are representative in *snowballing* or *comeback* matches in a one-month-long MOBA tournament dataset.

Index Terms—Game play data visualization, visual knowledge discovery, visual knowledge representation, and game reconstruction

1 INTRODUCTION

- Quan Li, Yeuk Yin Chan, Yun Wang, Huamin Qu, and Xiaojuan Ma are with the Hong Kong University of Science and Technology. E-mail: {qliba, yychanae, ywangch}@connect.ust.hk, {huamin, mxj}@cse.ust.hk
- Peng Xu is with NetEase, Inc.. E-mail: hzxupeng@corp.netease.com.
- Zhipeng Wang is with China Academy of Art. E-mail: wackwang007@gmail.com

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxxx

Multiplayer online battle arena, also known as MOBA, is a subgenre of video games in which two teams launch coordinated attacks on each other's base. Since the release of League of Legends, MOBA games have become the most played online games after the surge of Massively Multiplayer Online Games (MMOGs). Its market is huge but highly competitive with popular ones including Warcraft III, DOTA, League of Legends, SMITE and DOTA 2. It typically involves intense actions, fast-paced decision making, team-oriented play, and skillful use of character abilities. Failure to provide good experiences in any of the areas would easily lead to a quick loss of players [18]. Therefore, understanding and creating an engaging gaming user experience (GUX) are crucial to the research and development of MOBA games.

Previous research shows that MOBA games contain the most distinct playing experience among different genres of video games: the players' primary attractiveness comes from enjoyment and sense of reward from three aspects of challenges: competition and winning, mastering the challenges and difficulties presented by the games, and teamwork and fun with friends [13]. While the last aspect is more relevant to the quality of social interaction during the game, winning and encountering challenges are related more to the game design, such as resources, upgrades, conflicts and maps. Furthermore, research identifies that competition brings enjoyment to video games because it involves social-competitive situations as well [36]. Thus, for game designers, the considerations are how to strike a balance between delivering satisfactions for winning and achieving difficult challenges.

In MOBA games, game design and associated player performance are reflected in several types of game occurrences. We focus on two main occurrences: *snowballing* and *comeback*. *Snowballing* occurs when a team achieves and maintains certain advantages over their opponents without much effort throughout the remaining game. If it happens too frequently, players will tend to only undergo a limited set of actions instead of performing more complicated yet more enjoyable cooperation, combat and decision. Also, too much *snowballing* will encourage a player to dropout mid-game. *Comeback*, as in sports, occurs when a team overcomes a substantial disadvantage, particularly when this results in the disadvantaged team winning. If it happens too often, players may suffer from a misalignment between expectation and reality. However making the game impossible for *comeback* will also cause similar drawbacks to *snowballing*. Understanding when *snowballing* or *comeback* occurs during the game and the key features and events that lead to these occurrences can help game designers modify the existing game settings and maintain a good balance between the enjoyment derived from winning and that from difficult challenges for players. This also reduces dominant strategies that stereotype the gameplay and reduce players' perceived freshness of the game [20]. Overall, keeping a good ratio of *snowballing* and *comeback*, among all the matches is crucial to creating a pleasant, sustainable gaming experience: players tend to play a fair game with believable causes and outcomes more than the ones which fail to exhibit these qualities [31].

Using data visualization [22] to understand the occurrences of *snowballing* and *comeback* in MOBA games is particularly effective for the following reasons. First, a MOBA game is played and coordinated in a spatiotemporal way. Each player controls an in-game unit called "hero", travels within a fixed set of coordinates, and performs a fixed set of actions. All of these can generate precise information, which can be extracted in a structural way. Second, MOBA games share the same objective as team sports [5] and other collaborative games [11], i.e., individuals are organized into opposing teams that compete to win. Gameplay data analysis provides a better understanding of team strategies, game situations, and individual player's behavior. Without a visual analytics system, the data is usually analyzed in aggregated statistics; thus, valuable insights into local details and trends are often missing [30]. Finally, we believe that visual analytics is preferable for satisfying our research objective, which is to study the said occurrences during the gameplay. Although artificial intelligence and machine learning have been introduced to predict the outcomes of MOBA games or extract the patterns that lead to such outcomes [1, 8, 26, 29, 39, 41], we are interested in observing the patterns that may come from a single event or a sequence of events and that result in *comeback* or *snowballing*, which are loosely defined. Automatically deriving meaningful rules or patterns using pure statistical analysis approaches is difficult without a clearly delineated definition of occurrence, which in turn makes automated solutions difficult to achieve. Visualization here becomes a convenient tool to define situations and explore useful patterns. It follows a Human-in-The-Loop (HTL) approach [4] to apply one's own knowledge and interact with the results. Consequently, the domain knowledge can be included in the processes of visual analytics, instead of deriving the answers by automated analysis techniques.

To achieve the objective of visualization, we build a visual

analytics system to help game designers discover patterns of occurrences and allow a new kind of exploration of MOBA games. We produce a full gameplay visualization demonstrating detailed information of team building, team combat and team tactics. In addition, we provide an overview of gameplay statistics to compare each team's performance in terms of different indicators to aid the discovery of trends for potential situations. To provide additional insights into interesting patterns, we also propose a visual analytics workflow that enables experts to understand the characteristics of the two main occurrences and their overall distribution in a large collection of gameplay data before commercial public beta game testing. The techniques we propose have been used in the analysis of gameplay data from an unreleased commercial MOBA game offered by NetEase Inc.¹, a China-based listed Internet technology company. The major contributions of this paper are as follows:

1. We address the challenges of improving gaming experience by modifying game design and present the visual design requirements to identify the reasons behind *snowballing* or *comeback*.
2. We develop a suite of interactive visualization techniques enhanced with new features to support visually assisted knowledge discovery and sense making from gameplay data, thereby helping game analysts explore the major reasons that contribute to the two occurrences.
3. We leverage the results obtained from a single game to discover valid patterns applicable to multiple matches.
4. We showcase an experience of working with real world game designers to iteratively design a visual analytics system, actual deployment, finding elicitation, and expert feedback.

2 BACKGROUND AND REQUIREMENT ANALYSIS

2.1 Background

2.1.1 Rules of MOBA Games

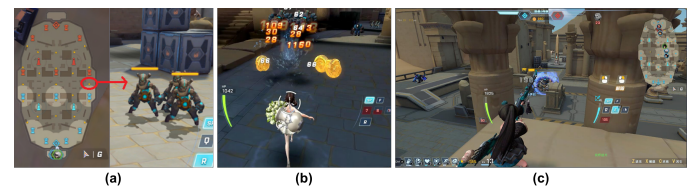


Figure 2. Screenshot from a MOBA game: Golds are rewarded for killing hostile computer-controlled units (b) (i.e. yellow points in the map) (a). Heroes in two teams are in a combat (c).

MOBA games involve two opposing teams whose goal is to destroy the other team's base to win. It normally consists of 5 vs. 5 players and no more than 10 vs. 10 due to certain game design considerations (e.g., optimizing network latency and role-playing features). Typically, alongside two or three main pathways, both sides contains one main structure (base) that must be destroyed for a team to win, although destroying other structures (towers) (Fig. 1b) within the opposing team's base provides other benefits. Automatic defensive structures are in place to prevent this, as well as relatively weak computer-controlled units that are periodically spawned at each base and travel down predefined paths toward the opposing team's base. For the MOBA game analyzed in this work, two special units are available. The first one is the *light tower*, which can heal its own team's avatars. However, if they are occupied by the opposite camp, the opposite camp will gain their views, which assists them to detect the tactical actions of the opposite camp. Another one is the *trooper*. This unit is neutral until it is hit by one camp, which makes it belong to another camp.

A player controls a single powerful in-game unit generally called a "hero". When a hero stands near a killed enemy unit or kills an enemy unit, it gains experience points, which allows the hero to level up. When a hero levels up, it gains the ability to learn more powerful skills and abilities. When a hero dies, it must wait for a designated

¹<http://game.163.com/en/>

time, which generally increases as it levels up, before it revives at its base. Each player receives a small amount of gold per second from its base. Moderate amounts of gold are rewarded for killing hostile computer-controlled units (Fig. 2b) and larger amounts for killing enemy heroes. Heroes use gold to buy a variety of items that are different in price and impact to upgrade their equipment. If heroes of one team become stronger, they can gain advantages in terms of securing objectives, killing enemy heroes and farming gold by killing computer-controlled units. As the stronger a team becomes, the more capable it becomes at destroying the enemy team and its base.

2.1.2 Complexity of Gameplay Data

The gameplay is recorded as spatiotemporal data among different characters with several time-varying attributes, such as equipment or experience. The combat between camps can be categorized based on the following three groups of attributes in the data:

A.1 Character Position. It provides the time-stamped positions of the players and computer-controlled units on the field. Each position is defined by x,y coordinates in the game map (Fig. 2a). It also facilitates ad-hoc analyses of the match dynamics, in which we can observe how each team is faring vis-à-vis the opponent's base.

A.2 Character Status. This provides information on equipment level and health of players or computer-controlled units. The level determines how strong the player is and how fast the gold and experience of the whole team will grow. The distribution of level among players and its comparison with that of other teams can help identify possible occurrences patterns or biases in the game.

A.3 Skill Hits. This provides the information of "hits" during the game, including the targets e.g., other players (Fig. 2c), computer units or towers, as well as the time and location of these hits. This information enables us to capture more details of the game dynamics.

Timestamp plays the main role in aligning these three groups of information. To identify the hidden correlation between different attributes of data, such as the relationship between the equipment of players and game events, further data processing is needed. Generally, the events and states for MOBA games fall into the change of these categories: health point, cash, equipment status and character status. Each MOBA game has their own unique sets of behaviors and simulations in the gameplay, but most of the outcomes can be explained from these categories.

2.2 Domain Expert Expectations

2.2.1 Working with Domain Experts

We have worked with a team of experts from NetEase Inc.: one gaming user experience (GUX) analyst (E.1), one data analyst (E.2), and two game designers (E.3-4) to analyze a MOBA game that has undergone a one-month trial period with the public. In conventional practices, the GUX analyst understands game situations by observing real players, and the data analyst translates the gameplay statistics into insights. Then combining the GUX analyst's impressions and the data analyst's findings, the game designers decide how to improve the game settings.

More specifically, to begin with, data analysts (E.2) first divide a single match into nine periods of time, in which the first three can be treated as the initial phase, the middle three are the development phase, and the last three as the final phase. Then they compute the difference in cash between the two camps in each period (Δ_t) in which the progress of the game can be classified into three categories according to the differences: Gap Narrowing (N_n), if the decrease in cash difference between two periods exceeds 10%; Gap Increasing (N_i), if the increase in difference exceeds 10%; and Stable Gap, if the cash difference is stable (N_s).

The game analysts we collaborate with have provided some rule-of-thumb to decide whether a *snowballing* or *comeback* occurs. The formula they used are as follows:

$$S = \begin{cases} N_n \leq \theta_{N_n} \\ \max(\Delta_t) > \theta_{\max_s} \\ \min(\Delta_t) > \theta_{\min_s} \\ \text{avg}(\Delta_t) > \theta_{\text{avg}_s} \end{cases} \quad C = \begin{cases} f_i \geq \theta_f \\ i_i \leq \theta_i \\ \min(\Delta_t) \leq \theta_{\min_c} \end{cases}$$

f_i is the number of N_i in the final stage, and i_i is the number of N_i within the initial stage. Each θ gives the threshold for each variable on the left and is dependent on an individual MOBA game. For the game we analyze, a *snowballing* game should have no more than one Gap Narrowing period (N_n). Also, the amount of cash difference should not descend to certain values as well. For a *comeback* game, there should be more than two N_n in the final stage but no more than one N_n in the initial stage. Moreover, the minimum cash difference should not exceed a certain amount. To determine the values for each threshold, game analysts use trial-and-error to see which values divides the proportions of occurrences into a reasonable percentage, and after that they discuss with designers who can work on the game design to see whether the proportions can be improved based on these thresholds.

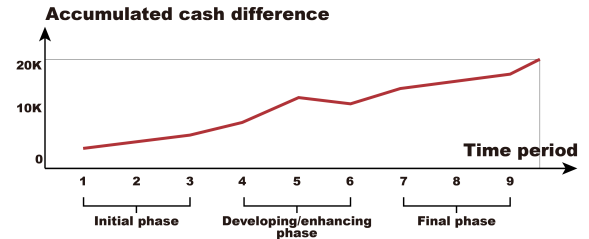


Figure 3. Quantitative analysis of cash difference of a typical game match: X-axis represents the nine time periods; Y-axis indicates accumulated cash difference between the two camps.

2.2.2 Abstracting User Requirement

After determining the matches that are of interest for our further research from the traditional methods, we develop a visual analytics system to analyze single match one by one, in which we could observe the dynamics of gameplay in each game and identify each action each player makes. Subsequently we can discover the spatiotemporal events and therefore understand how players' actions affect occurrences and game situations. To achieve these objectives in terms of user requirement, our main target users, GUX analyst and data analyst (E.1-2), indicate the presence of the following features:

R.1 Displaying Comparative Overall Gameplay Statistics. Our system has to display the change of each team's statistics (A.1.2), apart from the time and players' locations throughout the match. That is, our data analyst (E.2), has to be able to observe each team's total amount of gold, experience and hit rate well aligned by the time-stamps, so that comparisons can be made. Moreover, the team performance should also be divided into the statistics of each player. This is to allow the users to identify biases of performance for any individual player that heavily influences the outcome or dynamics of the game. Addressing these concerns can enhance the gaming experience of all the players.

R.2 Interactive Filtering to Select Timeframe for Simulating the Match Replay. One traditional approach for GUX analysts (E.1) to investigate user behaviors in MOBA is to watch the whole gameplay from each player's screen. To realize the outcome of such observation, our system must provide a simplified view of match replay to observe the change in the players' movement and tactics behaviors (A.1.3). Simulating the game in a simplified setting can allow users to understand how each player plans to launch an attack, retreat or proceed in one screen. Moreover, interactions should be provided to narrow down the game to important timeframes to search for insights (E.2) and seek for details from their impressions (E.1).

R.3 Summarization of Gameplay Events to Facilitate Search. Given various events that strongly determine game occurrences (e.g., destroying towers or accumulating kills of enemies), our system should display a clear record of time, location and details for these important events. It mainly leverages the time and location attributes (A.1) in data logs to arrange different events to provide insights for our data analyst (E.2). By contrast, our GUX analyst also wants to cluster the events (A.3) to search for the relevant time and locations, so that arranging the events in a manner that helps locate time and positions is important as well.

3 RELATED WORK

Since MOBA games can be considered both as team sports and video games, our related work can fall into the following categories: 1) research on MOBA player's behavior; 2) visual analytics of sports and; 3) data driven approaches on research of game data.

3.1 Research on MOBA Player's Behavior

Research on the behavior of MOBA players often focuses on the social interactions among players in the team competition environment. Chen et al. showed that game features affect in-game social interaction, including interpersonal relationships, community size and social alienation [3]. Shim et al. provided a domain-knowledge based approach to propose an efficient and automatic abnormal player decision support scheme using PageRank and normal distribution to find and judge bad players [33]. Johnson et al. found that MOBA games stimulate less immersion and presence for players. Additionally, although challenges and frustration are significantly higher in this genre, players derive a sense of satisfaction from teamwork, competition and mastery of complex gameplay interactions [15]. These kinds of work identify the characteristics of MOBA players, instead of the characteristics of game designs, which affect the group of players in terms of the interests explored in the above-cited works and which are considered the key elements of game design in our work. Lastly, for the user studies and evaluation done related to MOBA games, questionnaires from online community, interviews with computer players and trial play are popular methods [9, 19, 17, 23].

3.2 Visual Analytics on Sports

Similar to analyzing MOBA as a team game, there are general frameworks available for sports visual analytics [14] [40], but they only reveal general descriptions about matches rather than discovering problems on game design. Also there are visual analytics systems for professional sports such as soccer, ice hockey, baseball and tennis.

SoccerStories [27] helps analysts explore soccer data by focusing on game phases and connected visualizations for specific actions such as a series of passes or a goal attempt. Janetzko et al. presented a system to analyze high-frequency position-based soccer data for analyzing movement features and game events [14]. Manuel et al. proposed a visual analysis system for interactive identification of soccer patterns and specific situations [34]. These works are useful when a specific occurrence (i.e. phase) can be clearly defined, which is not our case. Baseball4D [6] reconstructs the whole baseball game in 3D from the raw position data and provides statistical methods to inform baseball analysis. For us, revealing the whole gameplay in 3D is not as important as trends with features (i.e., powers, abilities or levels). Visualizing the whole action context is insufficient to realize our goals.

SnapShot [28] integrates hockey intelligence gathering process to support the exploration of dataspace, sharing of hypotheses and communication of findings. It aims at providing free discussion of ice hockey situations, whereas we are more target-oriented to search for the reasons behind specific behavior of *snowballing* and *comeback*.

TennisVis [30] applies basic data such as score, point outcomes, point lengths, service information, and match videos from camera to provide visualizations for tennis coaches and players to quickly gain insights into match performance. Although it has demonstrated easily learnable visualizations, ad hoc hypothesis generation, and evaluation and facilitation of result sharing, it only demonstrates how sports visualization for a one-on-one match should be done. We want to summarize overall team performance, which means our visualization techniques are more tailor made with team comparison and group actions.

3.3 Data Driven Approaches on Research of Game Data

The research works of game data include: prediction of match outcomes, team matching mechanism and visualization of game data.

Firstly, for researches in feature extractions to predict match outcomes, Yang et al. modeled combat from game logs as a sequence of graphs to identify patterns in combat [41]. Rioult et al. introduced low-level topological clues, allowing for characterizing the space structure of a MOBA [32]. Bosc et al. identified encoding player

actions into sequences, mining sequential patterns, and computing the balance of each resulting strategy as steps to extract patterns in a reasonable time [1]. Though they demonstrated good results and predictions, they cannot reveal the dynamics of intermediate results, since there are no clear labels of such results for supervised learning.

For team matching mechanism, Veron et al. presented a database for the matchmaking service [35]. Drachen et al. presented zone changes, distribution of team members and time series clustering via a fuzzy approach as data-driven measures of spatiotemporal behavior and indicated such behavior is highly related to team skill and collaborations [7]. Kim et al. explored how users negotiate the proficiency-congruency dilemma and proved player proficiency increased team performance more than team congruency [16]. Since matchmaking on MOBA game is comprehensive now, we focus on factors affecting game occurrence assuming the skills between teams are matched.

Lastly, for visualization of game data, Wallner provided a literature review [37] which stated the first visual design scheme for such analytics is to define target audience and users, who are game analysts in our case. Wallner et al. proposed PLATO [38] composed of subgraph matching, pathfinding, data comparison, clustering and several visualization techniques to analyze game data not only limited to MOBA games. However to study MOBA games occurrence, it is too general. Hoobler et al. presented a system for enhancing observations of user interactions in virtual environments and focused on analyzing player behavior patterns [12]. It allowed spectators to visualize large-scale behaviors, team strategies and semantic information of specific actions to reduce the information overload from traditional overview visualizations. For us, besides visualizing semantic information and team strategies, we would also like to explore and identify potential flaw in game design. Besides, some game data management systems [10, 21, 24, 25] aid game designers gain insights on game data, but they focus on game technical specifications only.

4 WORKFLOW AND SYSTEM OVERVIEW

In this section, we present a visual analytics workflow (Fig. 4) that facilitates analysis of the game occurrences based on discussions with domain experts. It helps translate domain knowledge into visualized patterns for further investigation on data at a bigger scale.

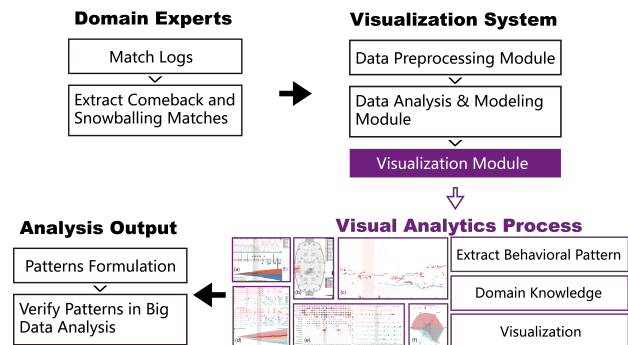


Figure 4. Workflow for identification and exploration of patterns for *snowballing* and *comeback* in MOBA matches. Users apply visual reasoning to extract patterns which will be verified by high volume of data.

The proposed workflow consists of three components: a data preprocessing module, a data analysis/modeling module, and a visualization module containing six major linked views (Fig. 1). Users are free to explore and select any match periods with timeline sliders in the Trend View and the Trajectory View. The Tactic Geographical Timeline View will automatically switch to the corresponding time frame. When using the system on their own, experts tend to start with inspecting the Player Billing Radar and Tactic Comparison View to gain an overall impression of team performances. After that they proceed to the Trend View and Resource Time Sequence View to locate the important time frames. Then they undertake a more detailed analysis of the events happening during these time frames in the Tactic Geographical Timeline View and Trajectory View.

4.1 Data Preprocessing

Processing the raw data described in Section 2.1.2 is necessary prior to analysis. In particular, to model and visualize the data, we need to classify and summarize them by respective features as follows:

F.1 Aligning Log Data: We align the log data by the timestamps, which provides spatiotemporal information of different game units (A.1) and their corresponding actions (A.3).

F.2 Game Event Extraction: We then extract game events such as combat and destroying towers from the activity logs by calculating the individual hits (A.3) throughout the time.

F.3 Summarizing Events: Based on aggregated game statistics (A.2-3), we can construct a table summarizing each of the matches, with which we can better understand individual player's performance.

F.4 Classify Players' and A.I. Activities: We need to differentiate activities conducted by heroes vs. those by computer-controlled units as well to extract players' involvement for their game status (A.2).

4.2 User Interface Design

Our visual analytic system contains six interactive views to enable free exploration of MOBA gameplay data as requested by our experts (R.1-3). For the color encoding, we use red and blue to represent two teams, respectively and warm versus cool colors to encode the players in the corresponding teams.

4.2.1 Trend View

The Trend View (Fig. 5) provides an overview of the game progress during a match (R.1). We categorize the information on team statistics into three sub charts: the position dynamics of players (F.1), the periods of time when important events happen (F.2); and the accumulation of resources in each team throughout the match (F.3).

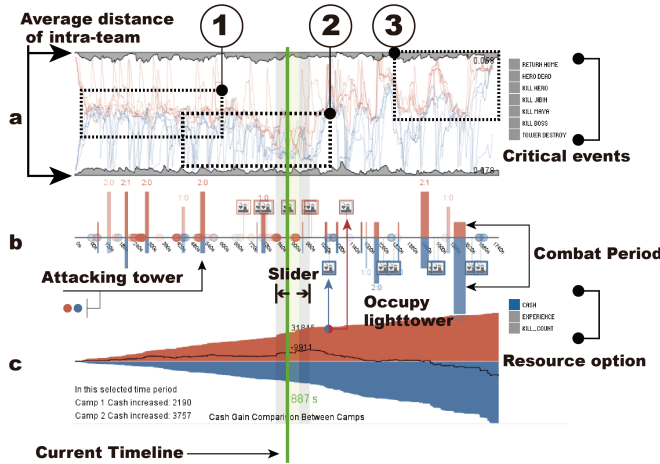


Figure 5. (a) A time sequence view of Y-trajectory of all players enables selection of important stages (1,2,3); two curves represent the average distance of intra-team. (b) timeline with bars indicates combat periods, circle dots representing timestamps when attacking towers and glyphs showing timestamps when starting to occupy towers; (c) comparison view of resource changes of both camps.

Traditionally, heat maps are applied to reveal the overall distribution of players and their matching trajectories in a selected time window on a geographical map. However, heat maps show aggregated data over a period of time and thus it is difficult to extract information related to a specific timestamp for feature alignment. Here, we propose a Y-axis position dynamics mapping, along with the traditional timeline. The positions of players are shown in a time series view that illustrates the Y-trajectories of each player throughout the game (F.1). Each player is represented by a smooth curve. Noted that we use Y-axis because of the map design. In all scenarios of MOBA games, positions along the axis that distinguishes the bases between the two camps is far more meaningful than the other one. This is because it can serve as cues on

which camp is marching towards the opponent's base and which camp is in a defense position. Another way to investigate team movement is to consider the distances between each member of a team [7]. In other words, the degree of team distribution over the map. To obtain a baseline metric for intra-team distance, the average over all pairs of distances between the players on a team $T = p_1, \dots, p_n$ is computed. Formally, this is described in the following formula:

$$D_t = \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n \|p_{i_{pos}} - p_{j_{pos}}\|_2$$

where p_1, \dots, p_n are the players in team T , n is the size of T , and D_t is the average Euclidian (physical) distance between any pair of team member positions $p_{i_{pos}}$. To obtain the fluctuation ratio of intra-team distance in the selected time period as requested by the analyst, we define N as the observation samples, and S_i as the average distance of intra-team at i timestamp. The standard deviation of change rate of each sample s is described in the following formula:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N u_i^2 - \frac{1}{N(N-1)} \left(\sum_{i=1}^N u_i \right)^2}$$

in which u_i is the change rate between two subsequent time periods, defined as $u_i = \ln(\frac{S_i}{S_{i-1}})$. Since we have N samples in the selected time period, the fluctuation ratio of intra-team distance δ is defined as [2]: $\delta = \frac{s}{\sqrt{(2N)}}$, which reflects the relative comparison between the average intra-team distance of the two opposite teams. Two curves are used to represent the average intra-team distances of the teams respectively along the entire match game in the Trend View.

Furthermore, we define several important events (F.2) based on the discussion with domain experts. Analysts can interactively select certain events and display them on the position dynamics mapping to select the suitable timeframe (R.2). Among all these important events, *combats* and *tower attack/occupy towers* need special handling. We need to determine the size of time window in which each *combat* or *tower attack* takes, as shown in Fig. 6a. Tradeoffs in window size are as follows. Combats defined by a small time window are generally shorter and less complicated, but may include more unnecessary combats and attacking times as noises. A large time window in contrast can capture more complicated combats, but consequently reduces the number of events to consider. After discussion with the analysts, we use a default time window of 10 seconds, with adjustment enabled for analysts. After this, to display the periods of time when important events occur (F.2), we provide a timeline view with bars suggesting the duration of combats, circle dots representing timestamps of tower attacking and glyphs indicating the starting time of the occupation of *light towers* (See Fig. 6b). Note that the heights of the bars are determined by the number of combats won in each time period of each team.

Lastly, the comparison view (Fig. 5c) displays the change of accumulated resources of each team in terms of cash, experience (F.3) and killing counts (F.4), with a black line in the middle showing the value difference between the two teams. Tooltips showing the resource values at the current timestamp indicated by the vertical green line and the value changes of both camp before and after this selected time period are available to facilitate the search (R.3). This view is further divided by player-level (Fig. 1d) to facilitate performance analysis of individual player. Through an accumulation of dots in the timeline, users can discover any sudden increase in the above-mentioned resources.

4.2.2 Trajectory View

We display the gameplay dynamics in a game map in the Trajectory View (Fig. 7) to simulate real matches (R.2). Considering the quantity and transfer load of the movement data, the game company we work with currently samples players' position and status every five seconds. Thus, we combine trajectories with the temporal dimension (A.1) to add a dynamic quality for a better understanding of the flow of gameplay.

After discussion with our GUX analyst (E.1), we use a moving dot to simulate how a hero moves between two discrete positions on the

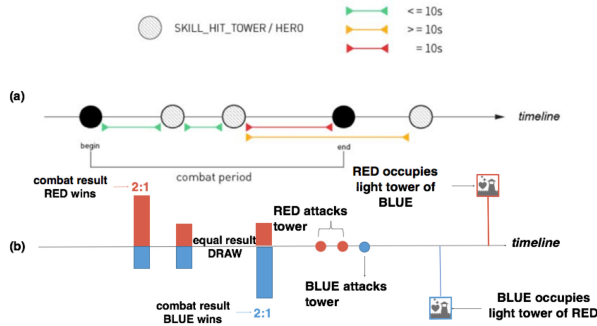


Figure 6. (a) Action time periods construction: If the timestamp difference between two same successive events is less than 10 seconds, it extends the length to next event until the timestamp difference exceeds 10 seconds. End time will be the last event's time plus 10 seconds. (b) Combats constructed as bars, attacking towers as circle dots and occupying light towers are aligned along the timeline.

game map in the actual gameplay: accelerating from a stand still until it reaches halfway, and then decelerating reaching the end point. The purpose of such tweening algorithm, which addresses the above need, is to deliver a position at a specific time and present a smooth animated transition, as shown in the following equation:

$$p_t = \begin{cases} t_t = t_t / (d/2) \\ p_t = c/2 * 2^{10*t_t-1} + b & \text{if } t_t < 1 \\ p_t = c/2 * (-2^{-10*t_t} + 2) + b & \text{if } t_t \geq 1 \end{cases}$$

where p_t is the object position at timestamp t_t , and d is the interval between the two successive timestamps; c is the acceleration parameter in the moving speed, and b is the starting position of the whole trajectory. The tweening algorithm is able to reconstruct a continuous trajectory from discretely sampled movement logs, which is more noticeable and traceable than “blinking” positions of each player.

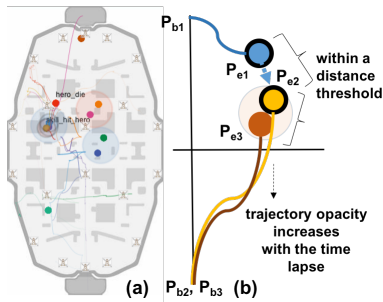


Figure 7. (a) Simulation display of the real gameplay scenario. (b) Visual encoding of the simulation: Players from the same camp form a bubble if they are close to each other. Border intensifies when opponents meet. And attack actions are represented by a moving arrow from the attacker. The transparency of trajectory lines increases with the time lapse (b). P_{b1} , P_{b2} , P_{b3} and P_{e1} , P_{e2} , P_{e3} represent the beginning positions and ending positions of the three players, respectively.

4.2.3 Tactic Geographical Timeline View

This view serves as an extension of the Trend View and displays more detailed information on resource within the user-selected timeframe (F.2). By enlarging and only displaying the time sequence data of a specific time window, the view can alleviate and even remove the possible visual clutter in the Trend view. Users can adjust the window size according to their needs. A big time window allows them to study the overall game dynamics, while a smaller time window helps them focus on some particular actions or tactics. The system also allows game analysts to interactively select the appropriate tactical actions to display in search for meaningful patterns and clear flows. In addition, we design a glyph icon for each tactical action and group them based on our data analyst's suggestions (E.2). Note that we still display members

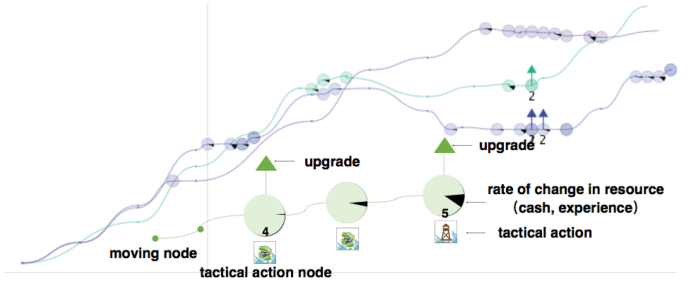


Figure 8. Encoding scheme: moving nodes and action pie nodes. Each pie node provides a summary of the current rate of change of resource of the player. An arrow indicates that an upgrading behavior occurs.

of the action groups individually as they are independent of one another. Furthermore, to summarize the effect of a tactical action on its executor's status (F.3), we design a pie node in which the black region represents the change of resources (e.g., cash or experience) as a result of the associated action: the larger ratio the black region is, the bigger the change of the resources is. We use an arrow to explicitly indicate upgrading events (Fig. 8). Each line represents one player with the same color used in other views.

Design Alternatives We develop the Tactic Geographical Timeline View through an iterative process with our domain experts. The first design alternative (DA1 in Fig. 9) is a two-layer hierarchy timeline view, with the horizontal axis being the traditional timeline and each vertical block indicating the tactical actions of the two opposite camps in a one-minute period. In each block, the five columns represent five players in the corresponding camp, and each row starting from the side closer to the timeline represents the sequence of tactical actions the player performs. Note that the actions across different team members are aligned step by step. DA2 is very similar to DA1, except that the players' tactical actions in each block are distributed based on their real timestamps: the shorter the distance is vertically, the closer they are in time. For the other two design alternatives (DA3 and DA4), we extract the unique tactical actions in each block, line them up horizontally, and use separate curves to connect the series of actions performed by individual players. The difference between DA3 and DA4 is also the temporal alignment of the tactical actions across players.



Figure 9. Design alternatives based on a two-layer hierarchy timeline.

Our GUX analyst and data analyst (E.1-2) have found that the above mentioned design alternatives are cumbersome for memorizing the dynamic evolution of players' tactical actions. Among the four designs, they prefer DA2 for the following reasons: (1) it reflects the real game situation, as the layout of tactical actions on the y-axis suggests the actions' actual temporal correspondence, providing a sense of realism especially for our GUX analyst (E.1). (2) There is no need to track the players' curves, since they fail to identify any movement pattern and may cause visual clutter and confusion. Still, they have the following concerns regarding DA2: (1) two hierarchies of timeline (i.e., traditional horizontal timeline and vertical timeline within a one-minute period) is highly ambiguous. (2) Sometimes successive tactical actions are performed in such a short period of time that they overlay together and lead to visual clutter. Moreover, when the number of tactical actions increases in a single block, the screen space becomes a bottleneck. Therefore, we propose the Tactic Geographical Timeline View, which combines the traditional timeline and players' geographical positions (A.1) together. It provides an overview of how players are distributed across the battle field over time which is easier to obtain than using the interactive approach in the four design alternatives.

4.2.4 Tactic View & Equipment Evolution View

We develop the Matrix View (Fig. 1e) to unfold the temporal dynamics of all the tactical actions (F.2) in the two camps, providing an overview of the match from the tactics' perspective (R.1). The X-axis of the matrix diagrams represents the timeline, grouped by one-minute interval. Valuable on the Y-axis are high-level categories of tactical actions as suggested by our data analyst (E.2) in the Tactic Geographical Timeline View. The pie glyph in each cell in the matrix diagram indicates the ratio of the two opposite camps in terms of the tactical actions performed in the corresponding category. The total number of tactical actions in that time period is mapped to the size of the glyph. In addition, if selected, a particular player's tactical action distribution shows up in each cell (F.4). When hovering over a cell in the matrix, its portion of the pie glyph and tooltips get highlighted.

All equipment of players can evolve that allows a significant improvement of the abilities for each character. To capture this, we design an Equipment Evolution View, which different types of equipment are encoded by different glyphs (A.2) and the equipment levels are visualized by the sizes of the glyphs. Since the impact of equipment evolution on different character attributes are interdependent, it is necessary to show changes in all the attributes for a holistic overview. To understand how the time when the equipment evolves may affect the occurrence, we plot the equipment evolution events along the timeline with the players. The evolution hierarchy is illustrated using a curve connecting the parent equipment to the child. Its width represents the cost in cash. Moreover, relevant equipment can be related together by hovering over a particular equipment.

To provide a summary of the results of such evolution, a Player Billing Radar on the right (Fig. 1f) displays the overall statistics of each player. The dimensions are preset according to our experts' needs, including the number of kills, towers destroyed, death, assists, cash earned and levels up. It is drawn with a Bezier curve since the curvature enables a perception of an area even when shapes overlap, and thus the performances of all ten players can be more easily seen and compared.

4.2.5 Interactions Among the Views

Our system provides users with rich interactions to facilitate an efficient in-depth analysis (R.2). Besides typical user interactions, such as filtering (e.g., filtering by players, tactical actions, etc.) and brushing (e.g., brushing on the timeline, etc.), we also support the following interactions: (1) **Linking**. The system enables automatic linking among different views mainly through temporal correspondence. Information updates are triggered by clicking the buttons in the legends in each view. That is to say, all the legends in the system serve not only as labels, but also act as data-filter buttons. (2) **Highlighting**. When users select certain data entry in a view, our system automatically highlights the corresponding information in all the other views. (3) **Animation**. The simulation of real gameplay environment with media control is presented as animation. When the time elapses in the simulation, the time window in other views shifts at the same pace.

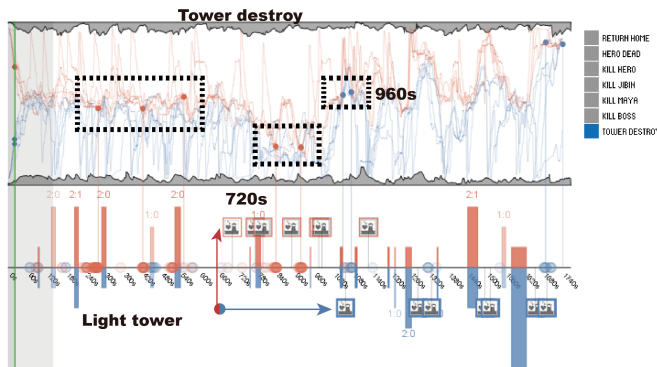


Figure 10. Early advantaged team (RED) occupies and destroys one light tower and three inner towers at around 12 minutes. However after 2 minutes, the comeback team (BLUE) destroys two base towers.

5 USE CASES

This section explains the game occurrences identified by our game designers (E.3-4) using the analysis of our GUX analyst (E.1) and data analyst (E.2). They identify key reasons due to players' decisions or biases in the game settings. To demonstrate the functionality of our system, we present how design issues are discovered, verified and improved and the "common mistakes" that lead to such occurrences.

5.1 Case One: Get Back on the Winning Track

The following sequences of activities occur when our GUX analyst and data analyst analyze each of the matches.

5.1.1 A First Glance: Observing the General Performance and Match Dynamics

Our data analyst (E.2) first gives attention to the Player Billing Information Radar View (Fig. 1f) to understand players' performance from an overview perspective (R.1) to discover players whose overall performances largely overwhelm their teammates. Then, he proceeds to the Tactic View (Fig. 1e) to obtain a main idea of the evolution of tactical actions conducted by both sides. In the initial stage, both camps focus on killing creeps reflected by the release of "skill hit creep", since these are the most basic way to earn cash and gain experience. After that the two sides meet and fight with each other as shown in the increase of "skill hit hero", "hero die" and "hero kill" tactical actions, with some starts of main events like "skill hit tower" and "tower destroy". These symbolize the entering of the developing/enhancing stage of the match.

5.1.2 After the Glance: Identifying Important Timestamps

The analyst then takes a detailed analysis of the Trend View to learn about the trajectory evolution of both camps and the changes in their resources accumulation. In the early stage of the match, the early advantaged teams mainly overwhelm all the combats in the midfield. However, the main outcome they have achieved is only occupying or destroying the several relatively small towers in Fig. 10). After around 1 minute, it is shown that the comeback teams start to gain advantages and march towards their opponents bases. After identifying the time of such occurrence, the expert filters out the important events to observe their timestamps and the corresponding resources changes before and after these events.

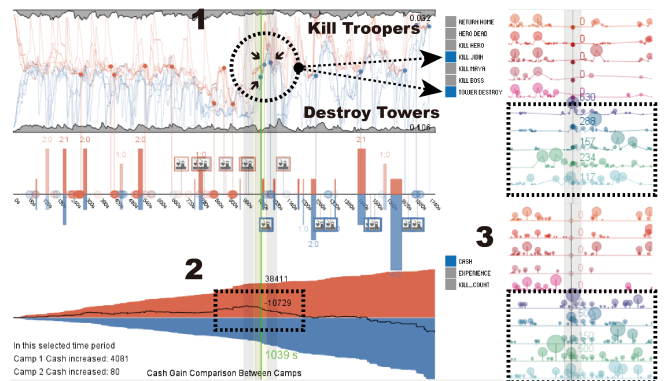


Figure 12. Large resource difference between early advantaged team (RED) and the comeback team (BLUE) before (2). Resource difference narrows after BLUE kills troopers (1), and plummets to 0 after BLUE destroys two outer towers (1), due to a large award of resources (3).

5.1.3 Narrow Down to Collect Evidences in Simulations

The analyst observes that after the early advantaged teams have destroyed an inner tower, they are busy defending the troopers and some eventually stay to fight against the opposite camp. They all assemble on one side in the map and destroy another inner tower. After that, they decide to proceed and choose not to go back to get healed or wait for their equipment upgrade, which leads to defeat due to the lack of

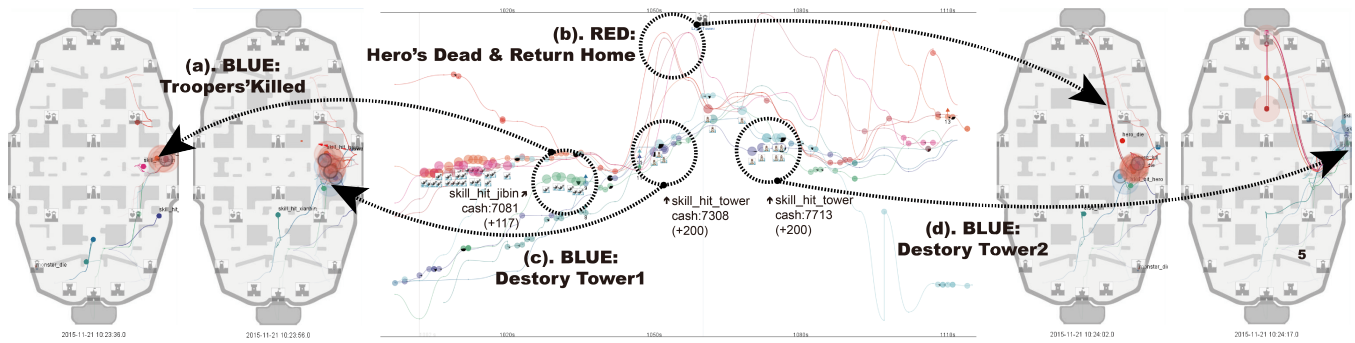


Figure 11. How the disadvantaged team (BLUE) does a *comeback*: (a) They kill *troopers*; (b) Heroes of the early advantaged team (RED) are defeated and return home; (c) BLUE destroys a tower of RED camp; (d) BLUE destroys another tower of RED camp.

health points. Thus, the *comeback* occurs when the *comeback* team take such advantage and kills the *troopers*. The team then proceeds to destroy two *outer towers*, as shown in Fig. 11. The expert clearly observes in the Resource View (Fig. 12) that the resource difference between two teams decreases to a great extent after such towers are destroyed. Furthermore there are increases in the *comeback* teams' levels and equipment evolution shown in Fig. 13, which make the *comeback* teams win combats in the later stages.

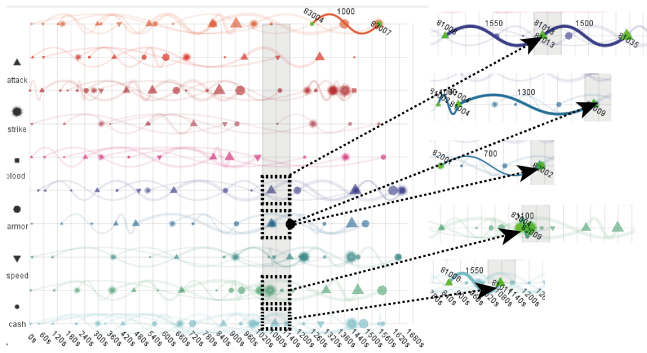


Figure 13. There are four players in BLUE camp who upgrade their equipment due to the large awards by destroying towers.

5.1.4 Summarizing the Takeaways and Verification

Our game designers' (E.3-4) takeaway for the game design is the imbalanced proportion of reward received between destroying *outer towers* and *inner towers*, and the high attacking capability of *base towers*. The gain of destroying an *inner tower* can not compensate the loss of health for the early advantaged players, consequently letting the *comeback* team able to outweigh them in the following battles.

To verify the hypothesis of imbalanced towers' rewards, we take the positions of all players after the *comeback* team's *inner towers* are destroyed by the early advantaged team in 100 sampled matches, and plot them into heat maps. This is because such group spatial-temporal information (i.e. players accumulating near the pathways, towers or enemy bases at a specific time) can give us a hint of the team skills and collaborations [7]. The results show a positive feedback. After the success of destroying *inner towers* and attempting to proceed to destroy *base towers*, the early advantaged teams show two traces of movements. Either they retreat or they are killed near the opponents' bases. Oppositely, the *comeback* teams seize the opportunity to march to the *outer towers* of the early advantaged teams, implying upgrades their equipment from the reward by destroying the *outer towers*. The contrast shows that it is undesirable for the early advantaged team to proceed after destroying the *inner towers*, which is supposed to be legit from the perspective of game design. Therefore, our game designers agree that lowering the attack of the *base tower* and the reward for destroying the *outer towers* could be the solutions for improvement.

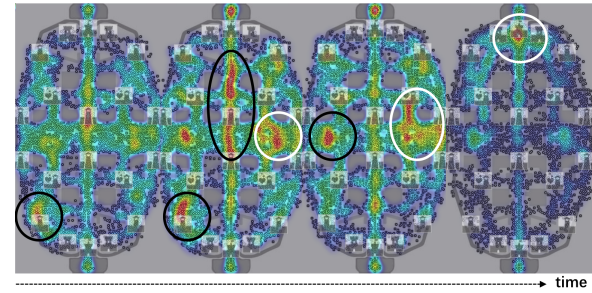


Figure 14. Heat maps of positions from the time when the early advantage teams successively destroy the *inner tower* of the opposite teams and proceed to when the *comeback* teams win. Black circles represent the retreat of players of the early advantaged teams, while the white circles represent the proceeds of the *comeback* teams.

5.1.5 Other Comebacks Caused By Players' Misjudgment

We briefly explain some reasons of *comeback* occurrences caused by players' own misjudgment. Our experts list out positions and hits as two factors derived from the system that attribute to the possibility of *comeback*. Position factor represents the retreat actions done by some players while their teammates are still in combats though they are in an advantaged positions, which can be seen in Trajectory/Geographical Timeline View. Hit factor represents the over focus on killing heroes or creeps without destroying towers, which can be seen in the performances metrics. As killing does not contribute much to equipment evolution that increases the overall power of the team, these teams become disadvantaged after their opponents have destroyed some towers.

5.2 Case Two: Snowballing on a Roll

5.2.1 Dominating Performance in Several Key Attributes

Following a similar observation process, the analysts first notice that the *snowballing* teams have a much better performance than the opposite camp through the views related to the overall performances. They observe through the Trend View that, in the initial phase, the *snowballing* team accumulates advantages by winning in the combats (Fig. 15 (2)) and destroying towers (i.e., two towers in around 3 minute (Fig. 15) (1)). Then they continue to overpower the opposite team by occupying *light towers* and interval combats until the end.

5.2.2 Destroying Key Structures Leading to Initial Success

Our GUX analyst (E.1) then wonders "what makes the BLUE camp lose all the combats?" By selecting the first four-minute period, he discovers that after two combats the *snowballing* teams win (Fig. 15 (5)), they accumulate sufficient cash and experience to upgrade the players' equipment, while the opponents' still remains the same. Thus the *snowballing* teams easily push down another two towers (Fig. 15 (6)) and continue to upgrade the defense equipment (Fig. 15 (3,4)). At this stage, all the winning teams keep holding a large gap by upgrading their equipment to higher levels, maintaining an overwhelming advantage in the following combats. The analyst also discovers another reason why the disadvantaged teams cannot accumulate resources in other ways such as sweeping creeps: nearly all of their light towers

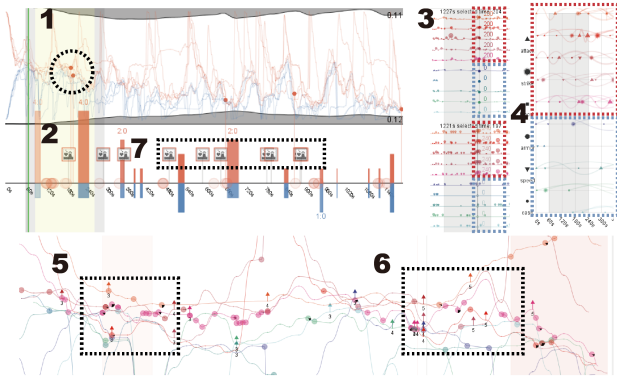


Figure 15. (1) Tower destroying events (2) RED wins in combats. (3) Resource accumulation differs. (4) Equipment evolution events occur. (5, 6) The resource gained from successful combats and tower destroying events (7) All *light towers* are occupied by the *snowballing* team (RED).

are occupied ((Fig. 15 (7))), in which the *snowballing* teams can easily get to know the opponents' tactical actions, thus reach and combat with the opposite camp to sustain their advantages.

5.2.3 Summarizing the Takeaways and Verification

Our experts conclude that, the culprit from game design causing *snowballing* is the over rewarding attacking equipment after evolution. After several combats, while the experience and reward from kills and hits of game units can accumulate some comparable assets between the teams, the first team able to evolve their equipment can easily break the tie and trump their opponents. If they can maintain the equipment advantages for a long enough time, the *snowballing* will become inevitable. To verify this hypothesis, we examine the combats in which win by the *snowballing* teams and their opponents before and after one side leveled up their attacking equipment among 100 sampled games, again. The result (Fig. 16) shows that the turning point of dominating combats starts in the 90 seconds after the attacking equipment evolution. Therefore, lowering the power difference between the first level up of attacking equipment could be a possible remedy to improve the *snowballing*. Meanwhile, the field of view of *light towers* should be narrowed to give the underdogs sufficient room to gain resources.

5.2.4 Other Snowballing Occurrences

Our analysts find another pattern that frequently occurs in *snowballing* occurrence. It is often that certain players from one side contribute to most of the killing events in the initial phase before destroying towers. Thus the *snowballing* occurrence is mainly due to the overwhelming performance of individual players. Analysts indicate that this pattern can be reduced by modifying the Match Making Rating (MMR) mechanism to provide a better fairness in the chances of winning.

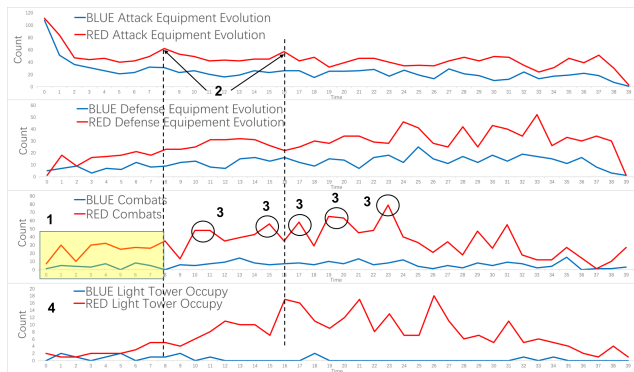


Figure 16. (1) *Snowballing* teams (RED) win in combats in the initial phase, and (2) they level up the attacking equipment. (3) RED dominates combats (circles) after equipment evolution. (4) RED nearly occupies all the *light towers* of BLUE.

5.3 Experts Review and Discussion

First, our GUX analyst (E.1) is keen on using the system to reduce his workload in User Experience Evaluation. Previously, he needed to gather a group of user experience researchers in the company together, standing behind the players to observe them playing the game and take notes of the game statistics changes in each match. This is labor intensive and time consuming. With our system, the experts can explore different but interconnected information about the match in various views, some of which resemble the actual gameplay to facilitate recall and interpretation. Once our experts become familiar with the interactions in the system, they start to develop a path through the system for single match inspection, which boosts their analysis efficiency.

Furthermore, our system effectively records changes in players' behavior changes and resources throughout the match. For example, our GUX analyst (E.1) points out that he can easily observe when a team decides to retreat or proceed using our Tactic Geographical Timeline View. The large and clear display allows analysts to quickly skim through trivial matches and identify the ones that arouse their interests. The experts then dive in and investigate the consequences of the decisions in the Trend View. In the end, by analyzing several representative matches with respect to each type of occurrence, our game designers (R.3-4) can compare the conclusions obtained from each game and establish a consensus on the reasons behind the occurrences. Moreover, our data analyst (E.2) enjoys using the comparison views to deliver a more compelling stories of statistical findings to the game designers. Before that they have been using statistical methods to extract latent features that are hard to understand by the others. Also, the animation in the Trajectory View removes the special visual effects in the MOBA gameplay that confused our GUX analyst (E.1) in the past. Showing the timeline, movement and actions of each character in a simplified view does facilitate our analysts to conduct a better comparison of the teams' performances (R.2).

In the interview, we ask about the envisioned applicability of our system for analyzing other MOBA games. All of them agree that only a slight change in the glyph design would be sufficient. This is because although MOBA games may differ in the number of players, the variation is relatively small (max. 10 vs. 10). The characters' tactical actions may be different in names and visual effects as well, but in general can still be grouped by the categories mentioned in section 2.1.2. In a word, our system can be easily extended by making minor modifications. For the improvements, our experts plan to add more game features that increase the engagement of analysts to understand player's emotions. However, they point out that the emotional factors are not essential in the analysis of occurrences in MOBA games.

6 CONCLUSION AND FUTURE WORK

We present a visual analytics system designed to enable the analysis of occurrences in a MOBA game. Our system consolidates the multi-variate gameplay data into insights of trends, game replay and players' tactics. By narrowing down the inspection into specific match periods and locations, we demonstrate that patterns can be observed and the design of the MOBA game can easily be studied.

For future development, we plan to provide a simulation of "what-if" conditions, so that users can change some attributes in the games to see how the new designs can possibly change the game situations. This will allow the game analysts to fine tune their results and therefore foster more desirable outcomes. For example, analysts can know whether a comeback will be easier by changing some parameters. We also plan to aggregate multiple matches into one display to see whether or not comparing performance across matches at the same can generate other insights as well. This kind of work will cope with the challenges of solving the display of increasing game data.

ACKNOWLEDGMENTS

The authors would like to thank the game experts in NetEase Inc. for providing the feedback and the anonymous reviewers for their valuable comments. This research was supported in part by HK RGC GRF 16208514.

REFERENCES

- [1] G. Bosc, M. Kaytoue, C. Raissi, and J.-F. Boulicaut. Strategic pattern discovery in rts-games for e-sport with sequential pattern mining. Technical report, Tech. Rep. RR-LIRIS-2013-012, 2013.
- [2] C. T. Carlstrom, T. S. Fuerst, et al. Forward-looking versus backward-looking taylor rules. Technical report, Citeseer, 2000.
- [3] V. H.-h. Chen, H. B.-L. Duh, and H. Renyi. 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*, pages 356–359. ACM, 2008.
- [4] L. F. Cranor. A framework for reasoning about the human in the loop. *UPSEC*, 8:1–15, 2008.
- [5] M. Csikszentmihalyi and I. S. Csikszentmihalyi. *Optimal experience: Psychological studies of flow in consciousness*. Cambridge university press, 1992.
- [6] C. Dietrich, D. Koop, H. T. Vo, and C. T. Silva. Baseball4d: A tool for baseball game reconstruction & visualization. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 23–32. IEEE, 2014.
- [7] A. Drachen, M. Yancey, J. Maguire, D. Chu, I. Y. Wang, T. Mahlmann, M. Schubert, and D. Klabajan. Skill-based differences in spatio-temporal team behaviour in defence of the ancients 2 (dota 2). In *Games Media Entertainment (GEM), 2014 IEEE*, pages 1–8. IEEE, 2014.
- [8] C. Eggert, M. Herrlich, J. Smeddinck, and R. Malaka. Classification of player roles in the team-based multi-player game dota 2. In *Entertainment Computing-ICEC 2015*, pages 112–125. Springer, 2015.
- [9] M. S. El-Nasr, H. Desurvire, B. Aghabeigi, and A. Drachen. Game analytics for game user research, part 1: A workshop review and case study. *Computer Graphics and Applications, IEEE*, 33(2):6–11, 2013.
- [10] M. S. El-Nasr, A. Drachen, and A. Canossa. *Game analytics: Maximizing the value of player data*. Springer Science & Business Media, 2013.
- [11] L. Elliott, A. Golub, G. Ream, and E. Dunlap. Video game genre as a predictor of problem use. *Cyberpsychology, Behavior, and Social Networking*, 15(3):155–161, 2012.
- [12] N. Hoobler, G. Humphreys, and M. Agrawala. Visualizing competitive behaviors in multi-user virtual environments. In *Proceedings of the conference on Visualization'04*, pages 163–170. IEEE Computer Society, 2004.
- [13] W. H. Huber. Epic spatialities: the production of space in final fantasy games. *Third person: Authoring and exploring vast narratives*, pages 373–384, 2009.
- [14] H. Janetzko, D. Sacha, M. Stein, T. Schreck, D. A. Keim, and O. Deussen. Feature-driven visual analytics of soccer data. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 13–22. IEEE, 2014.
- [15] D. Johnson, L. E. Nacke, and P. Wyeth. 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*, pages 2265–2274. ACM, 2015.
- [16] J. Kim, B. C. Keegan, S. Park, and A. Oh. The proficiency-congruency dilemma: Virtual team design and performance in multiplayer online games. *arXiv preprint arXiv:1512.08321*, 2015.
- [17] H. Korhonen, M. Montola, and J. Arrasvuori. Understanding playful user experience through digital games. In *International Conference on Designing Pleasurable Products and Interfaces*, pages 274–285. Citeseer, 2009.
- [18] R. Koroluk. Heroes of the storm vs. league of legends ? a game design breakdown & comparison, jul 2015.
- [19] T. Lapaš and T. Orehovački. Evaluation of user experience in interaction with computer games. In *Design, User Experience, and Usability: Interactive Experience Design*, pages 271–282. Springer, 2015.
- [20] R. Leigh, J. Schonfeld, and S. J. Louis. Using coevolution to understand and validate game balance in continuous games. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1563–1570. ACM, 2008.
- [21] Q. Li, P. Xu, and H. Qu. Fpsseer: Visual analysis of game frame rate data. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*, pages 73–80. IEEE, 2015.
- [22] S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: recent advances and challenges. *Visual Computer*, 30(12):1373–1393, 2014.
- [23] T. W. Malone. Heuristics for designing enjoyable user interfaces: Lessons from computer games. In *Proceedings of the 1982 conference on Human factors in computing systems*, pages 63–68. ACM, 1982.
- [24] B. Medler. Generations of game analytics, achievements and high scores. *Eludamos. Journal for Computer Game Culture*, 3(2):177–194, 2009.
- [25] D. Moura, M. S. el Nasr, and C. D. Shaw. Visualizing and understanding players' behavior in video games: discovering patterns and supporting aggregation and comparison. In *ACM SIGGRAPH 2011 Game Papers*, page 2. ACM, 2011.
- [26] T. Nuangjumnong and H. Mitomo. Leadership development through online gaming. 2012.
- [27] C. Perin, R. Vuillemot, and J.-D. Fekete. Soccerstories: A kick-off for visual soccer analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2506–2515, 2013.
- [28] H. Pileggi, C. D. Stolper, J. M. Boyle, and J. T. Stasko. Snapshot: Visualization to propel ice hockey analytics. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2819–2828, 2012.
- [29] N. Pobiedina, J. Neidhardt, M. C. Moreno, L. Grad-Gyenge, and H. Werthner. On successful team formation. Technical report, Technical report, Vienna University of Technology, 2013. Available at <http://www.ec.tuwien.ac.at/files/OnSuccessfulTeamFormation.pdf>.
- [30] T. Polk, J. Yang, Y. Hu, and Y. Zhao. Tannis: Visualization for tennis match analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2339–2348, 2014.
- [31] J. Portnow. The fine art of balance @ONLINE, Jan. 2008.
- [32] F. Rioult, J.-P. Metivier, B. Helleu, N. Scelles, and C. Durand. Mining tracks of competitive video games. In *AASRI Conference on Sports Engineering and Computer Science (SECS 2014)*, pages 6–p, 2014.
- [33] J. Y. Shim, T. H. Kim, and S. W. Kim. Decision support of bad player identification in moba games using pagerank based evidence accumulation and normal distribution based confidence interval. *International Journal of Multimedia & Ubiquitous Engineering*, 9(8), 2014.
- [34] M. Stein, J. Häußler, D. Jäckle, H. Janetzko, T. Schreck, and D. A. Keim. Visual soccer analytics: Understanding the characteristics of collective team movement based on feature-driven analysis and abstraction. *ISPRS International Journal of Geo-Information*, 4(4):2159–2184, 2015.
- [35] M. Véron, O. Marin, and S. Monnet. Matchmaking in multi-player online games: studying user traces to improve the user experience. In *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, page 7. ACM, 2014.
- [36] P. Vorderer, T. Hartmann, and C. Klimmt. Explaining the enjoyment of playing video games: the role of competition. In *Proceedings of the second international conference on Entertainment computing*, pages 1–9. Carnegie Mellon University, 2003.
- [37] G. Wallner and S. Kriglstein. Visualization-based analysis of gameplay data—a review of literature. *Entertainment Computing*, 4(3):143–155, 2013.
- [38] G. Wallner and S. Kriglstein. Plato: A visual analytics system for gameplay data. *Computers & Graphics*, 38:341–356, 2014.
- [39] H. Wang, H.-T. Yang, and C.-T. Sun. Thinking style and team competition game performance and enjoyment. *Computational Intelligence and AI in Games, IEEE Transactions on*, 7(3):243–254, 2015.
- [40] W. Wang, J. Zhang, and T. Liu. Matchvis: A generalized visual multi-scale analysis framework for competitive sports. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 287–288. IEEE, 2014.
- [41] P. Yang, B. Harrison, and D. L. Roberts. Identifying patterns in combat that are predictive of success in moba games. *Proceedings of Foundations of Digital Games*, 2014.