Chapter 14
Gameplay Metrics in Game User Research: Examples from the Trenches

Anders Drachen, Alessandro Canossa, and Janus Rau Møller Sørensen

Take Away Points:

1. An overview of the fundamental approaches to working with behavioral metrics.
2. Four case studies of gameplay analyses from the development of the games Tomb Raider: Underworld and Kane & Lynch: Dog Days.
3. Recommendations and considerations on the application of metrics to game user research.

A. Drachen, Ph.D. (✉)
PLAIT Lab, Northeastern University, Boston, MA, USA

Department of Communication and Psychology, Aalborg University, Aalborg, Denmark

Game Analytics, Copenhagen, Denmark
e-mail: andersdrachen@gmail.com

A. Canossa, Ph.D.
College of Arts, Media and Design, Northeastern University,
Boston, MA, USA

Center for Computer Games Research, IT University of Copenhagen,
Copenhagen, Denmark
e-mail: a.canossa@neu.edu

J.R.M. Sørensen
Crystal Dynamics/Square Enix, Redwood City, CA, USA
e-mail: januss@crystald.com

M. Seif El-Nasr et al. (eds.), *Game Analytics: Maximizing the Value of Player Data*,
DOI 10.1007/978-1-4471-4769-5_14,

14.1 Introduction

Games User Research (GUR) has gained a lot of ground recently in game development. It has become apparent just how beneficial knowing your target audience really is. More and more, studios and publishers integrate user-oriented methods in their development process and report direct benefits from this practice, above and beyond simple focus group feedback. This because GUR is a field that is all about the user: their experience, their game, their feeling of fun, etc. As users are also customers, it has become a widely accepted notion that GUR helps to improve games and thereby sales, so the investment and growth in this area is currently considerable (Pagulayan et al. 2003; Pagulayan and Keeker 2007; Isbister and Schaffer 2008; Kim et al. 2008; Drachen and Canossa 2009a, b; Lewis-Ewans 2012).

One of the newer developments in the game development in general, as of the time of writing, is the increased use of game metrics to support development (Kennerly 2003; Thompson 2007; Kim et al. 2008; Drachen and Canossa 2011), inspired by the application in e.g. web analytics (Sterne 2002). Quantitative measures of players, processes and performance have a long history in game development and software development in general, but it is only in the past roughly 5 years that the application of metrics, calculated from client telemetry data, have become more mainstream to game user research and game development.

From the perspective of GUR, which is the focus in this chapter, the unique value of game metrics lies in the objective tracking of user behavior and subsequent translation into data that can be quantified and manipulated – and jointly analyzed with other sources of user data, e.g. from usability- and playtesting. Using telemetry, it is possible to evaluate designs and debug user experiences to a degree of detail that for example observational methods does not allow.

Game metrics can be obtained from a variety of sources, but the most common, and the one we focus on here, is telemetry. Game telemetry data are the raw units of data that are derived from e.g. an installed game client. Code embedded in the game client transmits data to a collection server about how a player interacts with the game; or alternatively telemetry data are collected from game servers (as used in e.g. online multi-player games) (Derosa 2007; Kim et al. 2008; Canossa and Drachen 2009; King and Chen 2009).

The challenge with the goldmine of data that game metrics comprise is to know what to look for, why to look for it and how to make it valuable for different stakeholders, so it becomes more than a basic tool but a basis for analysis. While there are some gameplay metrics that are universally useful to track and analyze, irrespective of the game – such as playtime, player progress through a games´ levels, player ID, asset use etc. – the choice of what to track and how to track it is highly varied across games and specific features such as whether or not the game is persistent, single- or multiplayer, etc. Likewise, the design and research ecology that the metrics analysis becomes a part of differs from publisher to publisher and studio to studio. This makes it hard to generalize specific solutions across publishers, studios, and games, and it poses the inevitable question: What should we track, and how should we work with the data?

In this chapter, we will outline a few case studies from work at IO Interactive and Crystal Dynamics, both subsidiary studios of Square Enix Europe, showing some of the specific ways in which we have utilized gameplay metrics in practice, in the specific context of GUR (e.g. evaluation, playtesting, usability testing, and so forth). We try to illustrate how gameplay metrics are a useful source of data on player behavior during the development process, focusing on the games *Tomb Raider: Underworld* (2008, Eidos) and *Kane & Lynch: Dog Days* (2010, Eidos/Square Enix), as well as the multi-player form: *Fragile Alliance 2*. The cases presented are drawn partly from earlier research publications (Drachen and Canossa 2009a, b, 2011). We focus on methodologically simple, straight-forward analyses that do not require advanced statistical or data mining expertise (for more on game data mining see Chap. 12). The cases are all from the third-person action adventure- and shooter genres, but many of the ideas can potentially be applied across game forms.

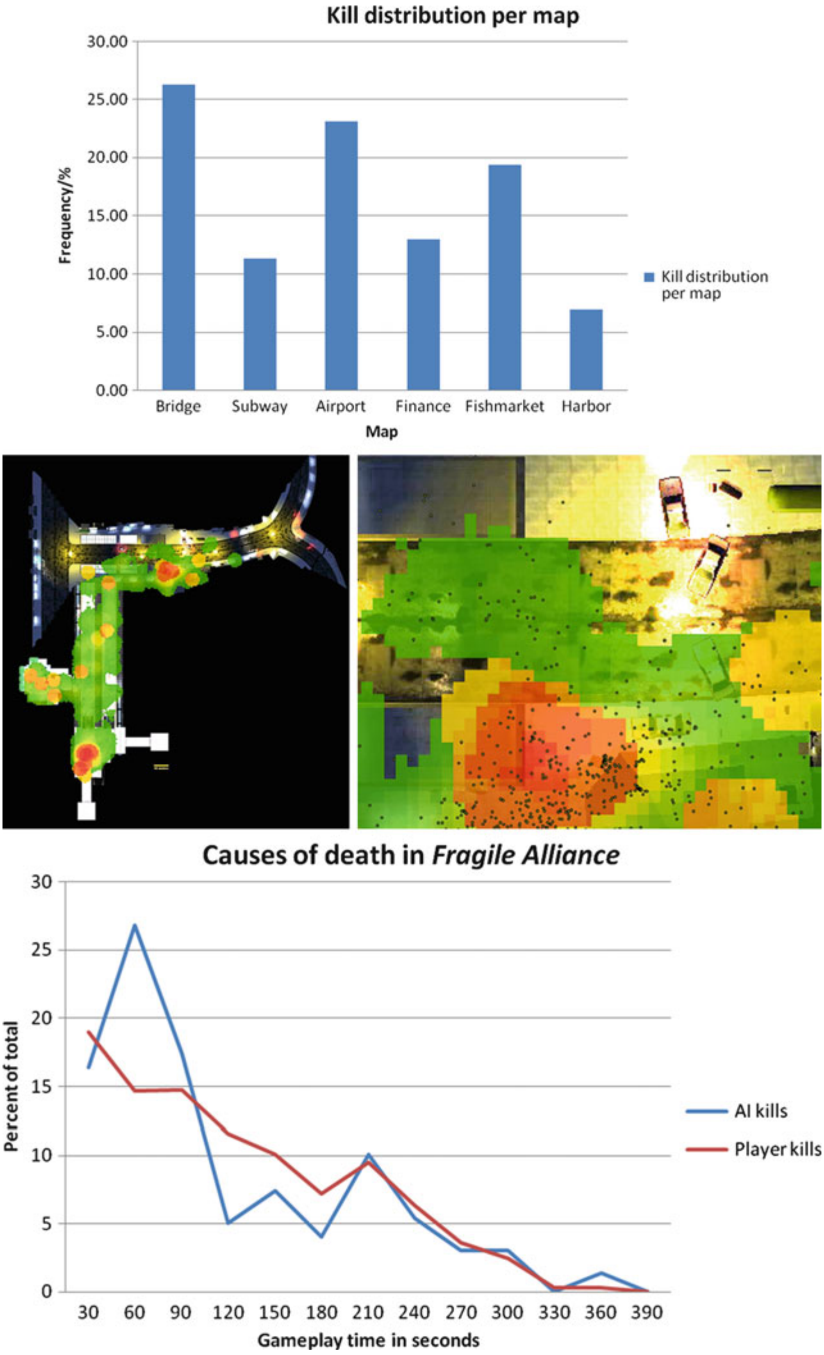
We will not be arguing for the superiority of game metrics above other sources of user-oriented data or methods in GUR and user-oriented testing, nor are we asserting that data mining has more value than other user research methods. In fact, we will describe how the practice of metrics analysis often evolves interplay with other research methods and the context of the research (the game development process). Towards this aim, we will focus on how the designers and GUR people at IO Interactive and Crystal Dynamics employ game metrics analysis with other sources of user-oriented data. This topic is also discussed in Chaps. 21 and 22 of this book.

14.2 Gameplay Metrics: Capturing User Behavior

Game metrics are discussed in more depth in Chap. 2, but it is worth mentioning that we view them as a specific form of business intelligence data, derived from the people, customers and processes involved in the business of games. Game metrics present the same potential advantages as other sources of business intelligence, i.e. support for decision-making in companies, at all levels of a company and practice. As will all other forms of business intelligence data that we use to develop and research games, the fundamental way to work with game metrics is through analysis (Vercellis 2009).

When game metrics are employed in Game User Research, the type of metrics we work with are generally referred to as **gameplay metrics**. These are measures of player behavior, e.g. navigation, item- and ability use, jumping, trading, running and whatever else players actually do inside the virtual environment of a game (whether 2D or 3D).

Player behavior measures can be anything happening inside the game environment, and gameplay metrics can be viewed as the “breadcrumbs”, the tracks, left by players in the games. Any action the player undertakes while playing can be tracked: every time a door is opened, a gun is fired, a treasure uncovered or a level completed, a telemetry tracking system can note down where and when that action happened (see Figs. 14.1 and 14.2 for some examples). Gameplay metrics are the most important form of game telemetry when the purpose is to evaluate design and user


Kill distribution per map
30.00
25.00
20.00
15.00
10.00
5.00
0.00
Frequency/%
Bridge
Subway
Airport
Finance
Fishmarket
Harbor
Map
Kill distribution per map
Causes of death in *Fragile Alliance*
30
25
20
15
10
5
0
Percent of total
30
60
90
120
150
180
210
240
270
300
330
360
390
Gameplay time in seconds
AI kills
Player kills

experience, and are furthest from the traditional perspective of the revenue chain in game development, and hence are generally under-prioritized (Pagulayan et al. 2003; Pagulayan and Keeker 2007; Isbister and Schaffer 2008; Kim et al. 2008; Drachen and Canossa 2009a, b). However, analysis of gameplay metrics provide the opportunity to address key questions, including whether any game world areas are over- or underused, if players utilize game features as intended, or whether there are any barriers hindering player progression. This kind of instrumentation data can be recorded during all phases of game development, as well as following a launch.

As a data source, gameplay metrics – and the methods used to obtain knowledge from them, such as data mining, machine learning and statistical analysis (Chap. 12) – supplement other types of user-oriented data and the methods used to analyze them, e.g. *usability evaluation* (measuring ease of use of the game) and *playability* and *user experience evaluation* (other methods exploring if players have a good experience playing the game), by offering insights into how people are actually playing the games being studied – i.e. their behavior – in detail. This has led numerous developers and publishers to combine gameplay metrics analysis with other sources of information on the player experience, for example questionnaires, interviews and gameplay observations and recordings (also discussed in Chaps. 21 and 22, see also Isbister and Schaffer 2008; Kim et al. 2008; Seif El-Nasr and Zammitto 2010; Lameman et al. 2010; Lewis-Ewans 2012).

14.3 Approaches to Working with Gameplay Metrics

When examining the available literature on game analytics from the different areas of the game industry, as is evident in the various chapters in this book (e.g. Chaps. 4, 7, 12, 17, 18, 19, 21, and 22) as well as various conference presentations (Zoeller 2011) and articles and books (Kim et al. 2008; Isbister et al. 2008; Drachen et al. 2009; Lews-Ewans 2012), and the older and much more developed fields of web analytics and Business Intelligence (e.g. Sterne 2002; Vercellis 2009), it becomes apparent that there are different ways to approach the kind of detailed user behavior work that gameplay metrics permit.

Fig. 14.1 Examples of various non-spatial and spatial syntheses of gameplay metrics data from *Fragile Alliance 2*, capturing various features of user behavior. Note that the visualizations were generated during development, so the relative distributions of event frequencies and map designs are different in the launched version of the game: (*top*) Kill distribution (*in percent*) across six maps from the team multi-player shooter *Fragile Alliance 2*, comparing lethality. (*middle left*): Heat map (developed using a density kernel function) from the Subway map of the same game, and a (*middle right*) closeup from the top of the map, showing the pinpoint spatial accuracy of individual death events. (*bottom*) A graph showing the overall causes of death for a group of playtests on the Subway map, indicating a strong AI influence in the first 60 s of the game at that point during production – in line with the designers intention

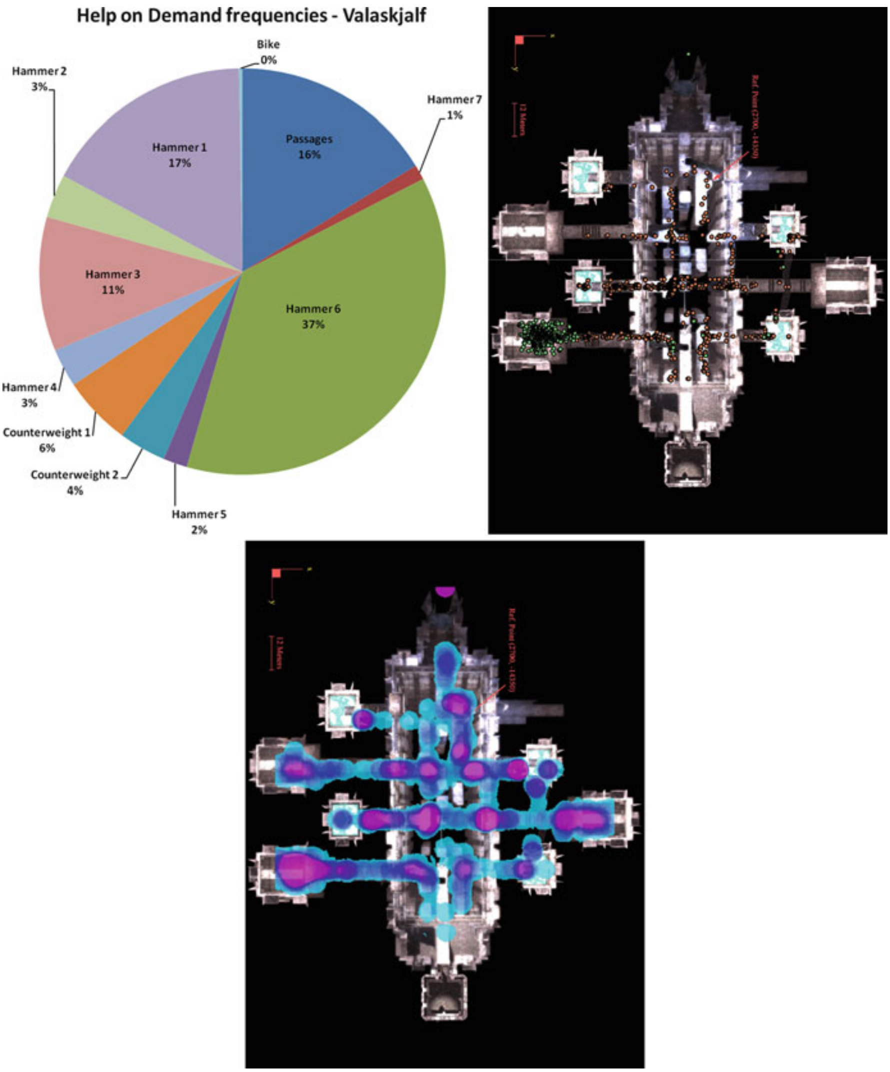


Fig. 14.2 Examples of various non-spatial and spatial syntheses of gameplay metrics data from *Tomb Raider: Underworld*, capturing various features of user behavior. (*top left*) pie chart showing the distribution of help requests for the different puzzles or challenges in' the Valaskjalf map unit. (*top right*) A map of the position of players in the Valaskjalf map unit of *Tomb Raider: Underworld*, when they request help for two of the puzzles in the map unit (*brown* and *green* points respectively) (the game features a built-in help system to assist players with puzzle solutions). (*bottom*) a heat map of the player death events (*purple* most events, *light blue* fewest)

Fundamentally, game analytics – despite the term – is either performed via **analysis** or by **synthesis**, both classic scientific methods. The difference can be subtle in practice, but basically *analysis* is when we break down a complex whole into parts or components, whereas *synthesis* is the opposite procedure, i.e. combining separate elements or components in order to form a coherent and complex whole. For example, breaking down behavioral data into smaller parts (e.g. time spent per checkpoint, number of button presses) to gain a better understanding of the individual components is analysis, while a chart showing the number of daily active users is a synthesis of individual components (time, number of users etc.). As such, the two methods go hand in hand and complement each other.

Both analysis and synthesis can be initiated by fairly open-ended or specific questions, roughly correlating with the concepts of **explorative** vs. **hypothesis** driven research from scientific theory (see also Chap. 12). What this means is that the approaches we use to find the answers to questions are either of a type where we are looking to confirm some idea we have and are looking for confirmation, or alternatively have a pretty good idea about the possible answers (hypothesis-driven); or alternatively more open, where we are not sure what the answer to a given question is, or have a hard time predicting the possible answers.

Explorative metrics research is when the possible answers cannot or are hard to predict from looking at the game design. For example, in a free-to-play MMORPG there can be hundreds of different reasons for why a player decides to spend money on a virtual item or –currency. It can therefore be hard to hypothesize over which of these reasons that cause players convert from being non-paying users to paying users. Similarly, in a Real-Time Strategy game (RTS), we might wonder which order of constructing buildings for a base that is the most effective? In any type of game, it is interesting to know what types of behavior our players exhibit. These are all examples of explorative questions.

A typical data-driven method for explorative research is the **drill-down analysis**, where you examine the gameplay metrics data at more and more detailed levels until an answer is found (see Chap. 12 for details, or: Han et al. 2005).

Hypothesis-driven metrics research is when we are looking to confirming conclusions or ideas, or when we can predict the answer. For example, we may think that Spiders are way too powerful on level 12, and perform metrics analysis in order to confirm this suspicion, finding that either we are right or wrong in our hypothesis (wrong in the case of the only spider in the level being an arthritic tarantula with bad eyesight). Alternatively, we could have a hypothesis stating that the amounts of player deaths on a certain map correlates to the perceived difficulty level of the map. Checking metrics data on player death events with feedback from research study participants can either lead to confirmation or rejection of the hypothesis, possibly leading to the formulation of a new hypothesis. A commonly applied method in game data mining to answer these kinds of questions is prediction analysis – the application of specific algorithms to predict something, e.g. which users that will convert to paying users, or when a person will stop playing (Mahlman et al. 2010) (see Chap. 12 for an example).

In practice, as soon as you move outside of the kind of questions that can be answered with synthesis, a quick analysis or standard algorithms, e.g., "what is the number of active users today?" or "what is the average playtime for level 22?" (see also Chap. 4 for examples of common metrics based on synthesis), you end up mixing hypothesis-driven and explorative work, something that is also evident in the case examples presented here.

In our personal experience – and we fully acknowledge that others may have a different opinion – the explorative questions are usually more time-consuming to answer and more often requires analysis than the hypothesis-driven, specific questions, which can more often be handled using synthesis (or very simple statistical analysis) of the relevant gameplay metrics data.

Purely explorative questions are, again in our personal experience, not as common in game development. This both because there is usually some ambition with an investigation, and because we cannot be certain that explorative questions lead to a result that impacts on the quality of the game. This is not to say that purely explorative analysis of gameplay metrics data cannot be useful, but it is often a kind of blue-water research that it can be hard to justify expenditure on. This however presents an excellent opportunity for game companies to collaborate with academic institutions like universities, who are able to focus on research and can bring expertise to the table which companies may not be interested in or cannot afford, to contain on a permanent basis.

Another fundamental question in gameplay metrics work is whether the analysis in question is initiated and driven by **designers** (including leads and producers) or by **games user researchers**. As outlined in Chap. 3, often designers require feedback on specific questions related to game design or the user experience. In comparison, user researchers also conduct more explorative evaluations. There are other groups of stakeholders that can drive metrics analysis, but usually these are not as focused on gameplay. For example, marketing differentiates from designers and user researchers by focusing on players as customers, i.e. sources of revenue.

Finally, gameplay metrics analyses can operate with **small** or **large** amounts of data – these both in terms of the number of players involved in the analysis and the number of variables (or features) for each player. The goals of small-scale analyses are typically more detailed than large-scale work, where the sheer amount of data and players lends itself better towards drawing broad conclusions and evaluating overall patterns in player behavior.

The reason that these considerations about the fundamental ways we can approach gameplay metrics analysis in games are important and relevant for GUR is threefold:

1. It provides the means for classifying methods.
2. It provides a terminology to use when we talk about gameplay metrics work on a daily basis.
3. It provides guidance on which questions that are useful to answer when planning to address a particular problem or task – for example, considering whether a problem is best solved analytically or using simple synthesis, if a spatial approach will provide added depth or be irrelevant, etc.

In this chapter, we have included four cases that represent some of the fundamental differences in the approaches that can be selected when working with gameplay metrics. These are outlined in Table 14.1, which describes how each case example varies along the dimensions outlined in this section (complexity – synthesis or analysis, approach – explorative or hypothesis-driven, stakeholder initiating the investigation, scale of the dataset used, and timing in relation to launch of the game).

Table 14.1 Classification of the four case studies presented in this chapter. This way of classifying approaches towards gameplay metrics analysis is based on concepts that are foundational in science theory, e.g. synthesis and analysis. The above approaches are not exclusive to gameplay metrics; they could be applied to other research practices and methodologies as well. The column "Initiated by" refers to the stakeholder group in a game development company who initiates the metrics study in question. The stakeholders who benefit from the work can be another group than those initiating a study (e.g. marketing, management, design, user-research, producers)

Number	Case	Complexity	Approach	Initiated by	Scale	Timing
1	Weapon usage metrics in Kane & Lynch: Dog Days	Synthesis	Explorative	User research	Small	Pre-launch
2	Gameplay analysis in Fragile Alliance 2	Analysis	Hypothesis	Design	Large	Pre-launch
3	Frustration analysis in Kane & Lynch: Dog Days	Analysis	Explorative	User research	Small	Pre-launch
4	Causes of death in Tomb Raider: Underworld	Analysis	Explorative	Design	Large	Post-launch

On a final note, gameplay metrics work is often either **spatial** or **non-spatial**, i.e., we commonly work with data that are separated from the gaming environment, or with data that contain some sort of spatial reference. Chapter 17 discusses spatial game analytics in more detail. A similar point can be made about **static** vs. **dynamic** visualizations. Metrics reports – and GUR reports in general – can be represented either as static or dynamic reports, the latter being interactive so the stakeholder it is intended for can manipulate the report. Chapters 18 and 19 discuss visualization in more detail.

14.4 Case 1: Weapon Usage Metrics in Kane & Lynch: Dog Days

Kane & Lynch: Dog Days is a third-person shooter developed by IO Interactive in 2010. This case study shows how metrics was utilized during mid-development as a direct part of the development process and joined with several other sources of user-data. The case study represents an initially explorative approach, with synthesis of data from a small sample size and a situation where the user researcher initiates and runs the investigation.

In *Kane & Lynch: Dog Days*, as in other shooter games, controlling the hero's weapon range is only one of many ways of controlling the player's behavior – and experience. For instance, hypothetically, short-range weapons against far-away enemies will drive the player forward, while long-range weapons typically afford a more sniper-like and less roaming behavior. Since short-range weapons lead to short-range kills, the killing experience is potentially more intense (closer to the frontline), while long-range weapons can give a more marksman-like experience, which can be equally satisfying. The trick is to; of course, enable the right experience at the right time. Balancing the weapon design in this regard can be hard to do based on observation or hypothesis alone, and players can have difficulties describing their 'weapon behavior' in abstract terms, since hopefully players will use the weapons as tools in a ready-to-hand fashion. Here, weapon usage gameplay metrics is useful to visualize allowing us to explore how different weapons can increase the likelihood of different behaviors.

During mid-development of *Kane & Lynch: Dog Days*, the user research team at IO Interactive wanted to explore how weapon attributes affected behavior on a specific prototype level. This intention was based to some degree on different vague hypotheses; however, the approach was to explore and learn. Playsessions were run with a small sample, only three participants, and various gameplay metrics captured via the Square Enix Europe Metrics Suite (Canossa and Drachen 2009), the system in place at Square Enix to log, transform and store gameplay metrics data.

The metrics data was plotted on a map of the prototype level. The level consists of two sectors (1 and 2). The map was produced using a native metrics visualization system developed in-house at IO Interactive, called *QuickMetrics*. The visualization system is hard-coded, and thus less advanced than some commercial-grade visualization systems and applications, e.g. most major Geographical Information Systems (GIS) packages such as *ArcGIS* and *QGIS* (Longley and Goodchild 2005; Drachen and Canossa 2011), or systems such Square Enix' other metrics visualization system *Amber* (developed by Square Enix Montreal) or visualization systems used by other publishers, e.g. EA Games, Microsoft Studios Research (Kim et al. 2008). However, *QuickMetrics* provided a very fast visualization process, enabling data (synthesis) immediately after a user-testing session.

At first glance, it looks like the behavioral pattern in the beginning of the map (Fig. 14.3, bottom left corner: *Sector 1*), is different from the behavioral pattern at the end of the map (Fig. 14.3, top-right corner: *Sector 2*). Some kills in Sector 1 (orange lines) are done at shorter distances and from more locations than the kills in Sector 2 (red lines). Other kills in Sector 1 (turquoise lines), on the other hand, looks like they were almost exclusively made from the same spot, though at a relatively short distance. In order to understand the behavior in the two sectors, we added additional information to the synthesis, namely movement paths; health level and player deaths (example shown in Figs. 14.4 and 14.5).

Comparing the beginnings of Sector 1 and Sector 2 (Figs. 14.4 and 14.5), we see that the player seems to be much more under pressure or at least vulnerable (the path color is more yellow and red) in Sector 1 than in Sector 2. This is also reflected in the number of player deaths in the two sectors. Exploring the synthesis of data represented by the spatial visualizations creates a backdrop for making a new hypothesis. This hints at a possible advantage in grouping data points permanently via the *QuickMetrics* tool, for instance by creating a new permanent metric called

Fig. 14.3 Enemy kills on a *Kane & Lynch: Dog Days* level. Green dots are the player positions at the time killing shots were made; the *colored lines* represent the lines of fire (different colors are different weapons), and *red dots* are locations of enemies at the time of death (Reprinted from Drachen & Canossa 2011; image is © Inderscience Enterprises Ltd.)

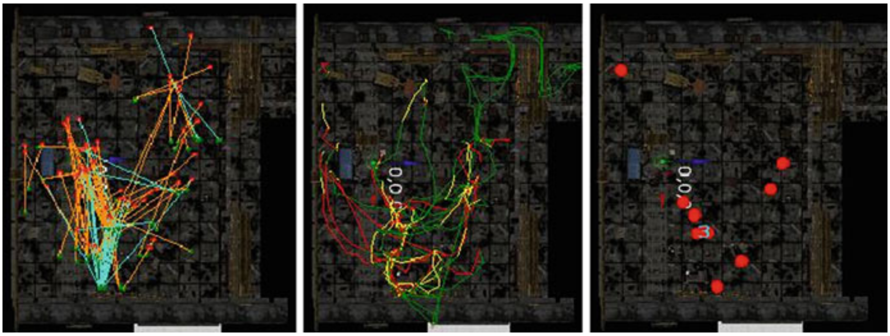


Fig. 14.4 Sector 1: (*left*) Enemy kills; (*middle*) location vs. health; (*right*) locations of death. The location vs. health map shows the hero moving around. If the line is *green*, the hit-point loss is from 0 to 80%, if the line is *yellow*, the hit-point loss is between 80 and 95%, and if the line is red, the hit-point loss is between 95% loss and death (Reprinted from Drachen and Canossa 2011; image is © Inderscience Enterprises Ltd.)

'kills/location ratio', i.e. how many kills are made per spot in a given area. If there are many kills per spot, this means that the behavior hypothetically is much like in the start of Sector 1 and 2. If we also add kill distance as a variable in this metric, then we can also highlight where on the levels we are more likely to have 'under

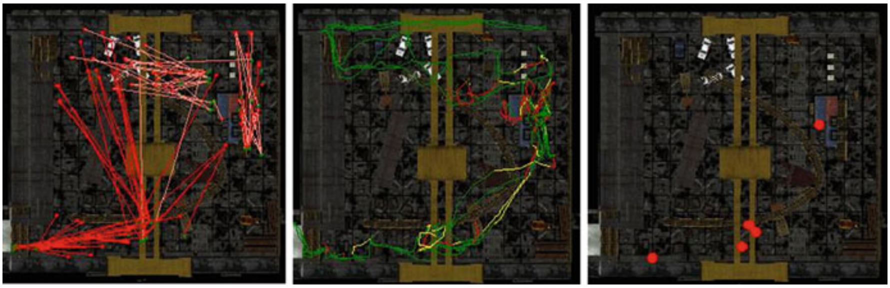


Fig. 14.5 Sector 2: (*left*) Enemy kills; (*middle*) location vs. health; (*right*) locations of death. The location vs. health map shows the hero moving around. If the line is *green*, the hit-point loss is from 0 to 80%, if the line is *yellow*, the hit-point loss is between 80 and 95%, and if the line is *red*, the hit-point loss is between 95% loss and death (Reprinted from Drachen and Canossa 2011; image is © Inderscience Enterprises Ltd.)

pressure'-experiences, rather than 'sniper/marksman'-experiences. This hypothesis could then be validated with a larger sample size. Obviously, the end-result of an analysis is not a nice looking heat map. The real goal is understanding the player's experience. Hence, any metrics analyses need to be validated with other methods as well to confirm the hypothesized link between behavior and experience. Likewise, AI movement (as a parameter) was not part of the synthesis, so this would also be beneficial to include in the work.

In this case, it was the designers' intention to put the player under extreme pressure in the beginning of Sector 1. In the ensuing analysis (which also drew on more qualitative data from the user test in question) it became apparent that the combination of the high-intensity, under pressure, frantic short-distance kills and high level of roaming in Sect. 14.1 and the less roaming, "pwnage killing" in the start of Sect. 14.2 created a compelling game experience coupling high stress and challenge with an opportunity for the player in Sector 2 to feel competent and de-stress by just staying in the same spot.

14.4.1 Mini-case: The Perfect Path

During development of *Kane & Lynch: Dog Days*, the lead level designer asked the games user research-team at IO Interactive to investigate whether players generally followed the path through the levels in the game as intended by the designers. This is an example of an explorative question in the spatial domain of gameplay metrics analysis, and a topic that can be tackled either at large or small scales.

The initial challenge was to figure out how to answer the question in the first place. Initially, the level designers were asked to play through the game in the way they intended, and their in-game behavior was logged via telemetry. This provided

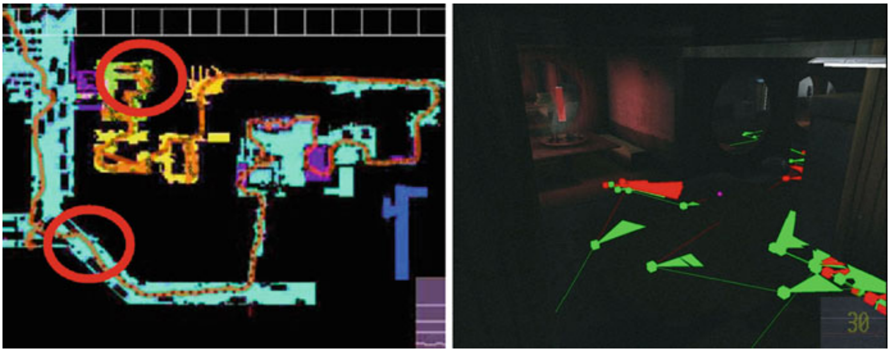


Fig. 14.6 (*left*) a section of a *Kane & Lynch: Dog Days,* level during development of the game. The *brown* line is a 2 m wide "perfect path", i.e. the ideal or intended path through the level. Overlain the second-by-second location of a playtester, with *red* circles highlighting where the player deviates from the intended path (Reprinted from Drachen and Canossa 2009a with permission; image is © 2009 by ACM, Inc.)[1] (*right*) Metrics added directly into the game editor. *Green* color marks the player being in good health, *red* color heavily damaged. The *cones* indicate the direction the player is watching at the time. Individual points are 1 s of playtime apart

a baseline to compare the behavior of players with what was intended in the design. However, obviously some deviation from the "perfect playthrough" of the designers was expected, notably in terms of weapon selection, strategies for overcoming enemies, etc. It was chosen to focus on the physical path described by players as they navigate the level. The path of the designers was overlain on level maps, and extended to be 2 m wide to allow for minor deviations (the brown path in Fig. 14.6, left). Subsequently, paths from user research sessions were overlain the path, and deviations from the designer-intended path located via overlay analysis (the paths were put on top of each other. Places, where the players' path deviated within a 2-m wide measure form the designers', were marked). The result pointed to specific areas where players strayed considerably from the ideal path.

This kind of explorative analysis showcases the potential of behavioral data to enable modeling of player navigation behaviors through the game environment. As a follow-up analysis, we used the same approach to locate areas where players were at low health, jumped, crouched, etc. (Fig. 14.6, right) This is of key interest to game designers because it allows them to observe from different perspectives how their games are being played. On a final note, navigation data can also be displayed directly inside the game editor, allowing designers to see through the eyes of the player, with the added benefit of having recorded metrics mapped in the game environment.

14.5 Case 2: Gameplay Analysis in Fragile Alliance 2

One of the key design concerns when creating scenario-based FPS-levels is to ensure that the right events take place in the right locations. During the development of *Fragile Alliance 2* – a team-based multiplayer game mode from *Kane & Lynch 2: Dog Days* – the designers at IO Interactive wanted to know if players died in the right locations with the right frequency as intended by the design of the game. The hypothesis driving this piece of GUR work was that the design worked as intended in terms of the spatial and temporal behavior of the players. The investigation performed to address this question represents a hypothesis-driven analysis (either the players die in the right locations with the right frequencies or they do not). We had a large sample size to work with (thousands of death events from playtests). In this situation, the designers of the game initiated the investigation with an overall focus on map dynamics in a 3D team-based FPS context.

Fragile Alliance 2 (Fig. 14.7) pitches players as either mercenaries trying to accomplish a heist, or police trying to prevent this from happening. A game session will typically consist of multiple rounds being placed on the same map (scenario) and/or different maps. The winner of a round is the player who leaves with the most money, irrespective of how these rounds were obtained. *Fragile Alliance 2* features scenarios deeply integrated in the individual maps, and has a twist unusual in

Fig. 14.7 Screenshot from *Fragile Alliance 2*, showing a Traitor clearly marked to the remaining mercenary players (Reprinted from Drachen and Canossa 2009a with permission; image is © 2009 by ACM, Inc.)

team-based shooters, which is that mercenaries can kill each other and steal each other's loot. If, for example, a mercenary player managed to secure a sum of money from a bank vault, another mercenary could kill the first, and steal his/her money. If a mercenary kills another mercenary, he becomes a "traitor", and is clearly labeled so to the other players, but is allowed to keep all the money collected without sharing. If a mercenary dies, they respawn (are reinstated in the game universe) as police officers, working along with a group of AI-bots.

The game design is intended to create a situation where the balance of power initially will typically be on the side of the mercenaries, but shift towards the police (AI and players), putting increasing pressure on the mercenary team. After the second death, the player will typically not respawn, but will have to wait for the end of the game round (usually after a few hundred seconds depending on the map). Mercenaries run the risk of being killed by AI-police, police players and other mercenaries. If a player kills a traitor they receive an instant reward; however, if a police player kills the traitor that killed him as a mercenary, he will reap a bigger revenge-reward. If a police player secures an amount of loot money from a mercenary, he keeps a percentage as a finder's fee. The purpose of the mercenaries is to escape with as much money as possible – either by working independently or together.

In order to address the question posed by the designers at IO Interactive (did players die in the right locations with the right frequency, as intended by the design of the game), we ran a series of analyses on metrics data obtained from comprehensive user-testing. Focusing on death events (where players die on the *Fragile Alliance 2* maps and why) provided a means for investigating spatial behavior of the players, which is easier to work with computationally than path analysis. When, where and why a player dies holds a substantial amount of information about the spatial dynamics of a team-based shooter level, as is known from traditional applications of heat maps, and this consideration formed the basis for the investigation.

One of the maps (or scenarios) in *Fragile Alliance 2* is the "Subway". The map gives the mercenary team the job of reaching and breaking into a subterranean vault, and then escape through a street-level area. The police team is tasked with preventing the heist. For the "Subway" map to work as intended, we would expect the mercenary team (or survivors) to at least in some cases reach the exit and complete the mission, rather than being gunned down by the police early in the map.

The dataset used for this analysis contained roughly 38,000 death events, obtained from sessions where the designers played the map "for fun", without any specific testing purpose in mind. They thus represented the closest we could get to "natural" player behavior data at the time. Locations of the death events were extracted along with various other variables deemed appropriate (e.g., role of the player at the time of death, role of the killer).

A central advantage of the *QuickMetrics* tool that was mentioned earlier is that the visualizations are pre-defined, and data from user-sessions can, therefore, be rapidly displayed and used in context with other data sources – or even discussed with the player. The system (in its current version), however, does not provide summative in-depth data analysis. It can provide visualizations of spatial data, but it does not provide opportunities for more complicated processes. In order to

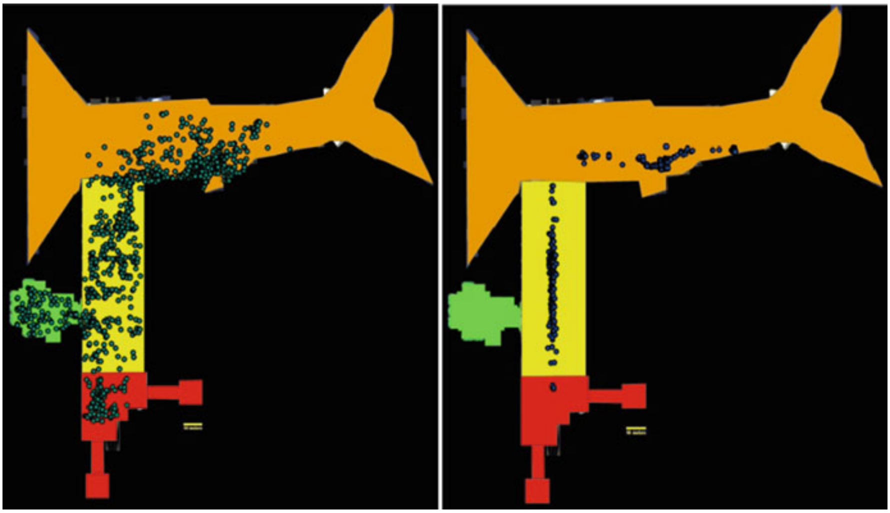


Fig. 14.8 The "Subway" map divided into four sub-sections: *Red* = spawning area for mercenary players; *Yellow* = subway; *Green* = vault area; *Orange* = road/exit area (+ spawning area for police players and police AI). (*left*): The locations where police officers were the cause of death. A broad distribution is apparent indicating that AI police officers can reach the entire map. (*right*): Locations of environment-caused death events. The events in the *yellow* sector of the map are caused by players being run over by a subway train while crossing a set of tracks, while the death events in the orange zone are caused by exploding scenery (e.g. cars that explode after becoming too damaged by weapons fire) (Reprinted from Drachen and Canossa 2009a with permission; image is © 2009 by ACM, Inc.)

perform custom spatial analyses, we worked with geodata (gameplay metrics with a spatial component – point, line or polygon), and used a Geographical Information System (GIS) as the front-end tool. A GIS is inherently flexible, but also time-consuming to use (see Chap. 14 for more on GIS software). Thus, it is ideal to have any analysis/visualization that can be pre-defined moved into a custom application, in our case *QuickMetrics*, for ease of use and to increase the speed with which visualizations can be generated (see Chap. 7 for another example of a visualization tool).

We used *ArcGIS,* an off-the shelf package. In *ArcGIS*, spatial data are added as individual layers (see Chap. 17 for more on the process of generating data layers in spatial analytics). In the current case, one layer was developed for the zonal division of the "Subway" map and for the locations of death events as a function of player role at the time of death (e.g. mercenary, police, traitor etc.). The "Subway" map was divided into four sections according to the level design (show in Fig. 14.8). In the map the mercenaries spawn in the bottom of the map, the police AI agents to the top right (Fig. 14.8). The objective of the mercenaries is a vault, located to the left in the map, and thereafter to reach the level exist, in the top right corner, behind the spawning area of the police. The game level consists of four sub-sectors:

- The **spawning area**, where the mercenary players enter the game and begin addressing the heist objective (red in Fig. 14.8).
- The **vault area**, where the money they need to steal are located (green in Fig. 14.6), a **subway station** area approximately in the middle between the spawning area of the police (AI and players) and mercenaries (yellow in Fig. 14.8).
- The **exit area**, an area at street level (orange in Fig. 14.8), through the rightmost side of which the mercenary players go if they want to escape the map (i.e. complete the heist or run away).

Dividing the level map into sub-sectors permits a more detailed analysis of the distribution of the death events. By combining visualizations of the spatial behavior of players with statistics of their temporal (and spatial) behavior, a more thorough understanding of the player behavior is gained. The analysis we ran showed that mercenaries primarily turn traitor in the beginning of the game in the spawn area, but most commonly (55.72%) in the road/exit area – i.e. when the mercenaries are close to getting out with their stolen booty. For the mercenaries, the majority of the kills occur in the spawning area, where mercenaries enter the game (red zone in Fig. 14.8). The AI police kill and death events are spread across the entire map, indicating that their search & destroy behavior is working excellently (Fig. 14.8 shows the locations where AI police kill opponents – distributed across the map). Suicides, i.e. when a player dies for a reason not related to another player, occur in the vast majority of cases (76.04%) in the road/exit area, where a series of cars are placed which can explode, doing damage. A smaller part takes place in the metro station area, where players can be hit by metro trains while crossing the tracks coming from the vault to the exit/road area to the north in the map (shown in Fig. 14.8 (right) as a line of death events up the middle of the yellow area).

That does it for the role of the killers, but what about the role of the players when they die? Police (recall that these can include both players and AI) are often killed in the road/exit area where they spawn (69.32%) (shown with purple bars in Fig. 14.9), and very few are killed in the spawn and vault areas, where instead the mercenaries are under pressure from the police (44.17%, shown in Fig. 14.10). Interestingly, traitors are typically killed in the spawn area (61.25%) (show in Fig. 14.10), but rarely in the road/exit area (8.81%) (show in Fig. 14.10), which in combination with spatiotemporal analysis indicates that it is a much more risk-filled endeavor to turn traitor early in the game rather than later (associated spatiotemporal analysis shows that mercenaries turning traitor outside of the spawning area rarely move into the spawning area again – by this point the action has moved to the other segments of the map).

We also considered these patterns in a temporal context. In *Fragile Alliance 2*, one of the key design components is that there should be a shift in balance as the game progresses. Initially, the mercenaries are strong, however, as more and more mercenaries are eliminated, the police force – augmented by AI agents – became stronger. In order to evaluate if this shift in power actually manifests, a time-series analysis was performed using temporal slices, finding to the pleasure of the

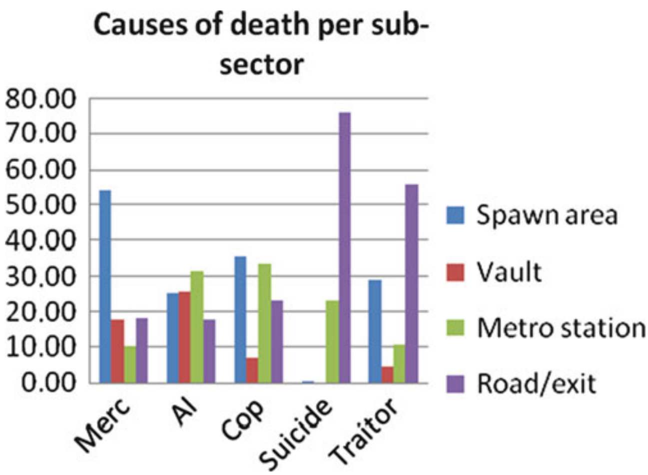


Fig. 14.9 The causes of player death as a function of the four sub-sectors (Reprinted from Drachen and Canossa 2009a with permission; image is © 2009 by ACM, Inc.)

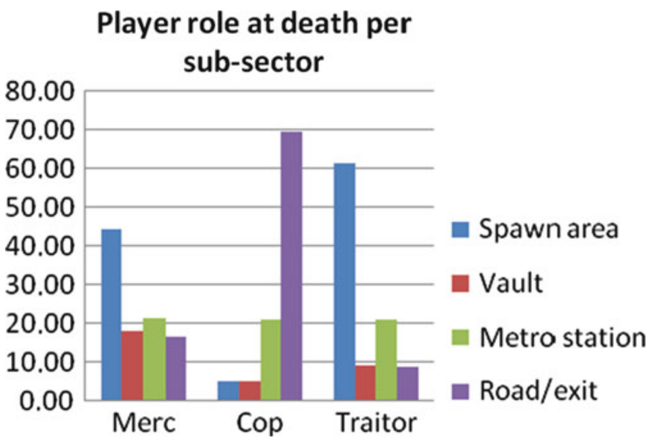


Fig. 14.10 The role of the player at the time of death as a function of the map sub-sector (Reprinted from Drachen and Canossa 2009a with permission; image is © 2009 by ACM, Inc.)

designers that there is a significant shift in who the dominant killers are from the early to the middle of the typical game round. This is illustrated in Fig. 14.11, where the left pie chart shows the breakdown of the causes of death (killer roles) summed for the first 45 s of play, and the right pie chart the causes of death, but following 90 s of play until the end of the game round. A distinct shift in the causes of death is apparent.

While the behavioral patterns of the players indicate the manifestation of the designer's intent, a somewhat larger amount of death events occur in the spawn area (lowermost part of the map) than is ideal, which could indicate that

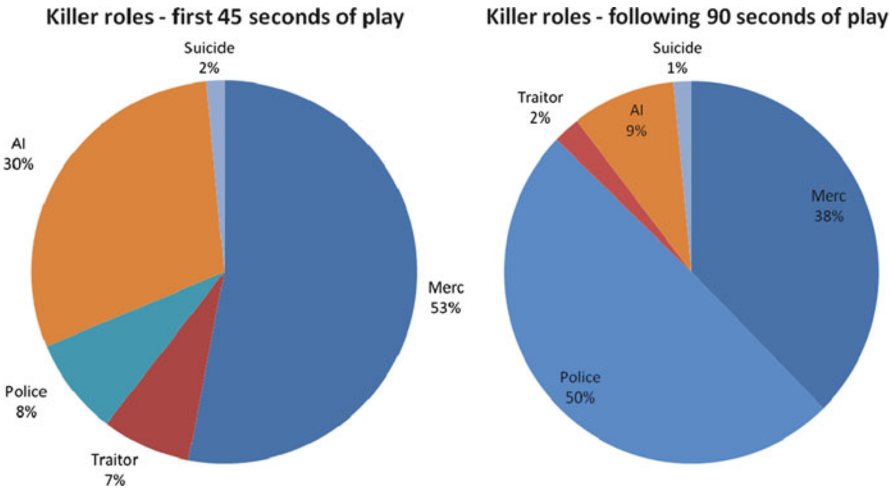


Fig. 14.11 Roles of players who kill another player or AI-bot during the first 45 s vs. following 90 s of play on the "Subway" map. A distinct shift is noted as mercenaries are eliminated and the police force gains correspondingly in strength

mercenaries are perhaps a bit too eager to turn traitor early in the game. This is exemplified in Fig. 14.12, where death event data from just one sessions are shown. Out of 253 kills, 119 appear in the spawn area for the mercenary team (blue area in Fig. 14.12). This kind of behavioral pattern is however not necessarily a problem to the user experience – it can be the opposite: the pattern observed may be fun to the players even though it is not what was expected. To find out requires the use of other GUR methods than telemetry analysis, for example play-testing using a think-aloud protocol or surveys (Isbister and Schaffer 2008; Lewis-Ewans 2012). For an introduction to user research methods in general, including think-aloud testing, see Kuniavsky (2003).

14.6 Case 3: Frustration Analysis in Kane & Lynch: Dog Days

The third case is an example of a small-scale, explorative analysis where the user research team at IO initiated and drove the investigation based on a problem observed during user testing: During the regular user-oriented testing of *Kane & Lynch: Dog Days*, in the later development stages of the game where vertical slices were available and the game is mostly functional, the user research team at IO Interactive observed several participants becoming obviously frustrated while playing a specific segment of the game (e.g. groaning, throwing the controller away, agitated, angry etc.). This case study describes the work conducted subsequently to identify which

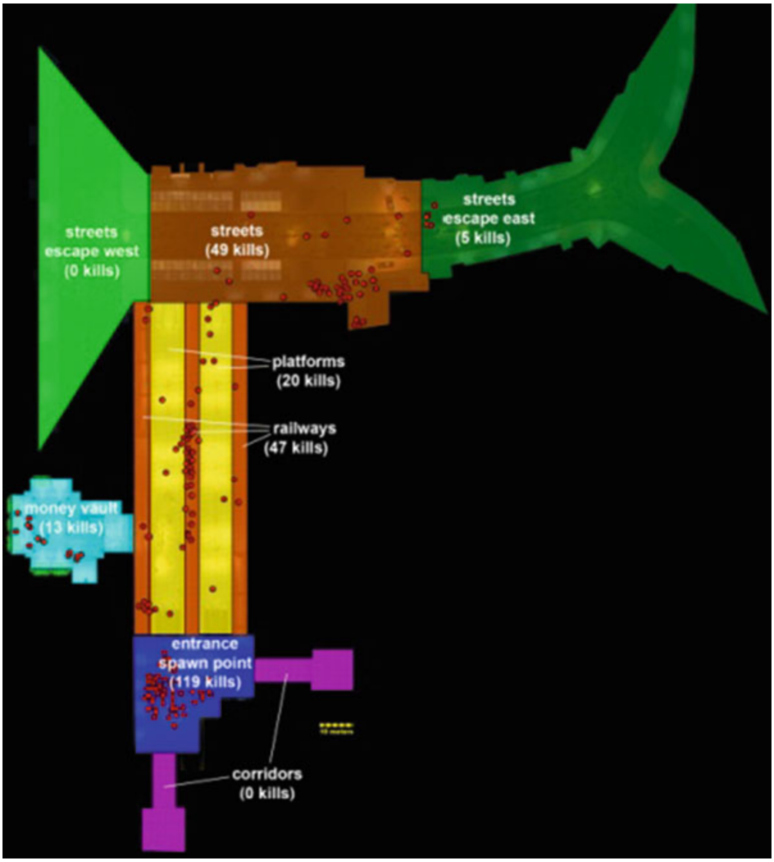


Fig. 14.12 Visualization of death events from the "Subway" map. The map shows the distribution of 253 player death occurrences (all from a specific play session) overlain the level map, and has added explanations to guide the interpretation of the map (Reprinted from Drachen and Canossa 2009a with permission; image is © 2009 by ACM, Inc.)

behaviors led to the exhibited frustration. The case study is described in more depth in Canossa et al. (2011).

A central limitation of gameplay metrics analysis is that the data can inform what players are doing, but cannot directly answer questions about 'why' players are exhibiting that behavior, although sometimes we can make decent guesses (inference). The same is the case for the user experience, we cannot directly tell what the player feels like. This is precisely the reason why the more design-oriented use of gameplay metrics-based analysis cannot always stand alone in user research, and is ideally combined with other approaches for GUR, e.g., observation, interviews or questionnaires as discussed in Chaps. 21 and 22. The following case study presents another example of how game analytics and traditional user testing can go hand-in-hand to inform game development.

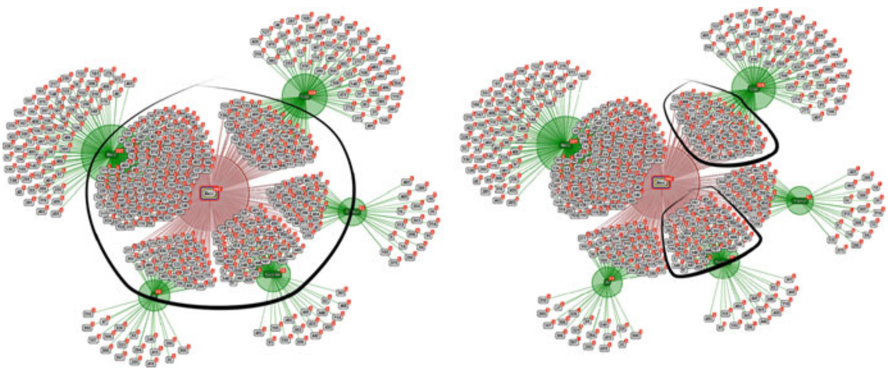


Fig. 14.13 "The Flower of Death". This is an example of a more interactive way of working with gameplay data that was tested back in 2008 at IO Interactive. TouchGraph Navigator is a simple visualization tool which allows the user to visualize connections between data points. In this case, 253 death events are shown, including the role of the killer and victim. TouchGraph Navigator permits the manipulation of the groups using drag and drop, which makes the data easily navigable. In this case, the tool was used to discover that a significant percentage of the mercenaries (the *central mass*, *circled* in the diagram to the *left*), were killed by suicides (i.e. environmental effects) (the *lower right* mass of data points marked on the *right diagram*). Checking the spatial distribution of these events, it was found that most of these kills occurred because the players were hit by a train in the railways area of Fig. 14.12. A signal was added to the train, warning of its arrival, which solved the problem. The *upper right* group marked in the right diagram above are mercenaries killed by police AI or players. The amounts of deaths is about the same as the suicides, suggesting that this early stage of development, the danger to the mercenary players needed adjustment

In many user-testing situations, the causes of player frustration are related to bugs, unfinished designs – malfunctioning items, checkpoints etc. Thus, finding the reason for player frustration can be a straight-forward process of observations and interviewing, ideally assisted with videos of screen capture and similar tools. However, what the user researcher team observed in this case was different, they identified a trace of specific patterns of behavior connected to frustration, which they believed was connected to a 'real' design issue. For example, they observed players rushing forward within the game after dying over and over, not really looking around for danger, increasing the risk of dying again, causing even more frustration, after which the participants slowly regained their composure and concentration and took a more careful approach.

When running user tests with one participant, detecting such behaviors and querying the player about them is possible. However, in the context of larger-scale tests (e.g. 5 or 25 players), it becomes cumbersome for resident GUR personnel to detect the various hints in the behavior of players that point towards frustration. This prompted the idea of investigating if particular expressions of frustration were related to particular behaviors while playing the game (avatar/character behaviors) and subsequently using gameplay metrics data to pinpoint the exact locations in the game world where such behaviors occur, across multiple players.

The first step was to examine the detailed telemetry data for one of the frustrated players (Fig. 14.14) in conjunction with the video recordings of the screen and the

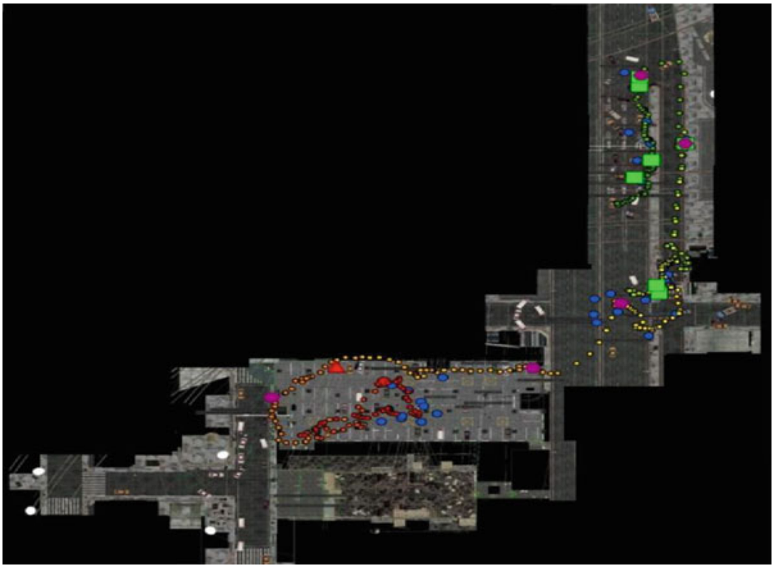


Fig. 14.14 Visualization of the spatial behavior of a player through the environment of a *Kane & Lynch: Dog Days* level, between two death events. The location of the player is plotted at each second of playtime, and a color scale applied to show the dimension of time along the path. Various events are plotted as symbols: enemy kills (*blue dots*), weapon pickups (*red triangle*) and taking cover (*green squares*). Spatial metrics visualizations such as this one are highly useful for the detailed evaluation of gameplay and balancing in shooter games (Reprinted from Canossa et al. 2011 with permission; image is © 2011 by ACM, Inc.)

player from the testing session. This was done in order to examine what the player actually did during the periods where he exhibited frustration, identified through body movements, facial expression and verbalizations. The metrics data were fairly comprehensive. We used a temporal sampling interval of 1 s. The data includes the location of the players, the vector of the avatar, the vector of the virtual camera, the health of the player, movement modifiers (walking, running, sprinting), whether the players were crouching or not and whether the players were "in cover" or not. In addition, triggered events (events the logging of which are triggered by specific player actions) were analyzed, which include checkpoint activation, picking up weapons and ammo, making use of exploding objects in the level, being "down but not dead", killing of Non-Player Characters (NPCs) and player deaths. Examining the data generated a list of indicators of frustration (Fig. 14.14), as follows:

1. The player dies repeatedly in the same location or even regresses in terms of progress made between each death event.
2. The number of enemies killed decreases considerably with each successive attempt to progress in the game following a death event.
3. The pace of movement becomes considerably faster for each attempt, and the same route is taken each time.

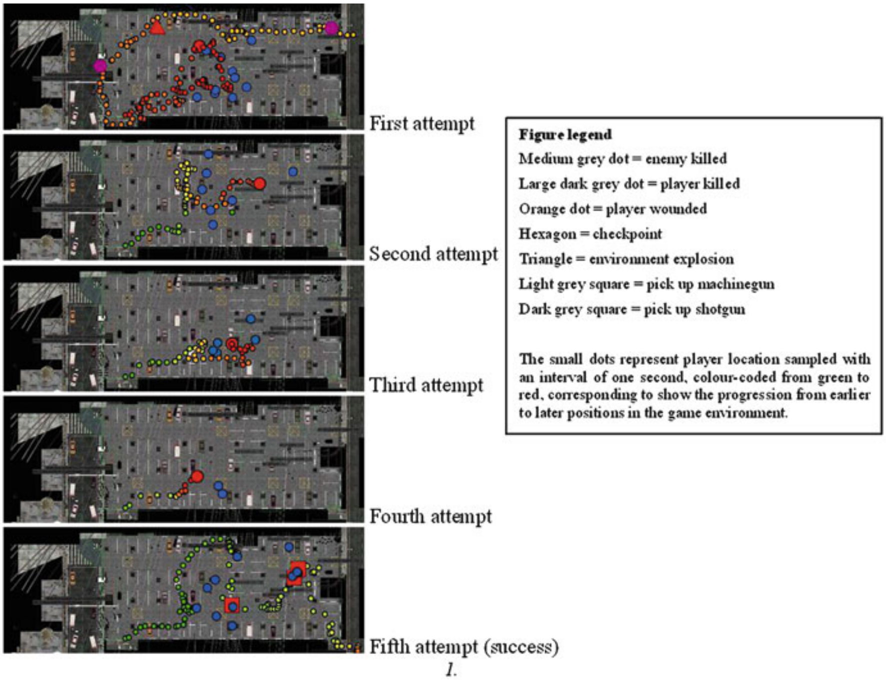


Fig. 14.15 The images show the path of a single player (participant) and specific events that occurred during the gameplay session used in the case study. Each image represents the time segment from one instance of player death to the next, showing decreasingly less progress in the game from death 1 to 4; indicative of a behavioral pattern pointing towards player frustration (*red dots*: player deaths, *blue dots*: NPC kills, *red square*: weapon or ammo pickup, *red triangle*: environment explosion, *purple dots*: checkpoints, *small dots*: player position in time) (Reprinted from Canossa et al. 2011 with permission; image is © 2011 by ACM, Inc.)

4. There is minimal or no use of special abilities, picking up of weapons or using the environment for help, e.g. triggering explosions.
5. The vector of the camera increasingly coincides with the direction of movement of the character – the higher the frustration the less interest in examining the environment.

A second and a third set of metrics data were also examined, based on reports from the participant about experienced frustration. The behavioral patterns in these situations matched those located initially (Fig. 14.15).

Following the identification of the above list of possible indicators of frustration, we analyzed metrics data from a sample of 22 randomly selected players among the testers used by IO Interactive. We used procedural algorithms to confirm the initial results and establish whether the pattern of behavior identified signified frustration within the 22 selected play sessions. It should be mentioned that it is, of course, possible that other players might also have experienced frustration but reacted very differently.

We found a match in the behavioral patterns of 6 out of the 22 players. One or more times during the playtest, these six players exhibited the same kind of behavioral profile as defined by the five points listed above. This profile is radically different from the kind of behaviors observed in the remaining 16 players.

The behavioral variables were found to correlate, e.g., if a player death event (Pd) occurs within 2 min, the average pace (speed) of movement of the player (Pm) will increase compared to the average movement pace for the entire game, and the number of NPCs killed decrease progressively between each death happening. Similarly, the number of weapons and ammunition supplies picked up (WApu) decreased progressively as players continuously died within shorter time intervals.

Based on the behaviors of the six players and the initial tester, we developed a model specifying the timing and frequency of the behaviors identified, specifying the value of the key parameters indicating player frustration. The model was presented as follows:

```
tn <tf<tn+1
Pd>=2
0<Pdl<20
Pmf>Pm
NPCd(tfn)>NPCd(tfn+1)
WApu(tfn)>WApu(tfn+1)
```

Where:

- Timestamp (t). The timestamp is set to zero the moment a new play session begins. <tf> describes a time interval that has been identified as "frustrated"
- Number of player's deaths (Pd), <Pdl> expresses location of player deaths in world units.
- Player's pace of movement (Pm) measured as distance in space travelled in 1 s, averaged for the whole playsession. <Pmf> defines the average pace of player movement during an interval of time identified as "frustrated"
- Number of NPCs killed (NPCd).
- Number of weapons or ammo picked up (WApu)

Importantly, all conditions need to occur simultaneously for the model to contain all the indications of player frustration reported – i.e., frustration is not indicated by any single behavioral variable, but the occurrence of a set of behaviors (e.g. fast movement and many rapid death events).

As the final step in the investigation, the six players who exhibited frustration were called in for open interview sessions. Using video recordings of their play sessions and visualizations of their metrics data, we attempted to uncover if the players felt frustrated during the intervals identified by the model. During the interviews, we used a custom browser-based tool (Fig. 14.16), "G-player" to show animated replays of test sessions, which is very useful when reconstructing play experiences with players. The participants confirmed that in all of the segments of play identified by the model, they had experienced undesirable frustration (i.e. frustration contrary to the user experience).

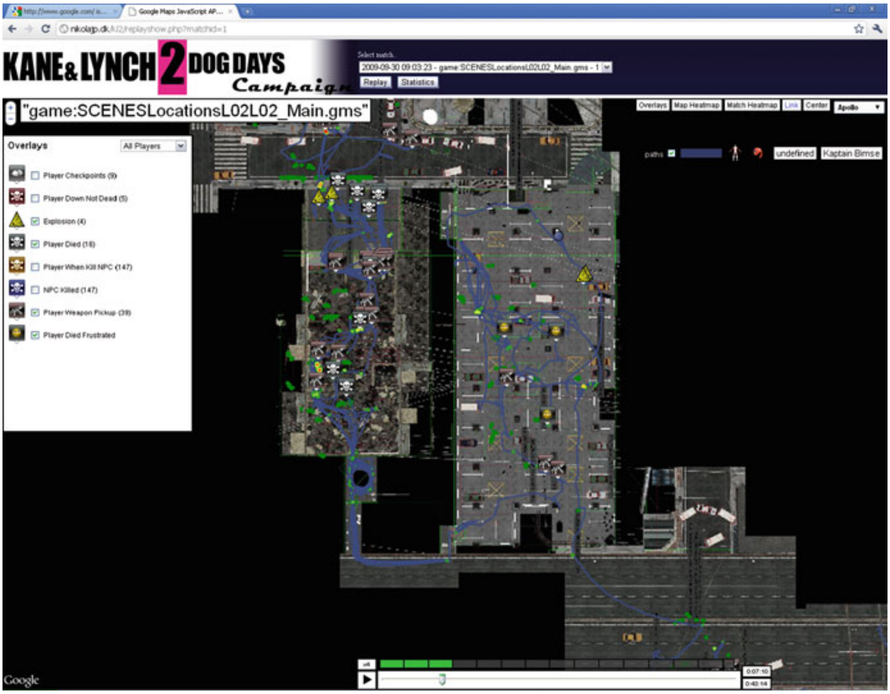


Fig. 14.16 The G-Player dynamic visualization tool, which allows replaying game sessions from a *top-down* perspective, showing the behaviors of different players and events as icons as they move around the playfield. A real-time version, where metrics data are being fed directly into G-player during the research study, is currently being developed (Reprinted from Canossa et al. 2011 with permission; image is © 2011 by ACM, Inc.)

The case study is an example of a relatively comprehensive metrics-based examination, more time-consuming than the day-to-day use of metrics data to evaluate game designs, but nonetheless potentially highly useful. The initial exploration led to a hypothesis that frustration is quantifiable and visible in the metrics. This led to the analysis of data and partial validation of the hypothesis.

There is much further work to be done to validate the utility of the model and its generality across different games. However, the case study does showcase the potential fruitful systemic interplay between hypothesis and exploration which potentially can make it possible for the user research team to more automatically detect and evaluate frustration in user research sessions, by analyzing the behavioral metrics data from the testers. This saves valuable manpower and provides a means for the development of automated frustration detection systems. Considering how fast and inexpensively automated the problem detection system operates, it provides a concrete benefit, because in a real-life user testing situation, it is not realistic to expect user researchers to keep track of all of these variables while running user tests. This highlights the usefulness of automated gameplay metrics tracking and -recording as a tool for user-oriented testing. At the same time, it also highlights the challenge of automating. In other words, just because it works on this game and with these users, does not necessarily mean it is universal.

14.7 Case 4: Causes of Death in *Tomb Raider: Underworld*

This case study, originally conducted following the launch of *Tomb Raider: Underworld*, represents a designer-driven, large-scale and fundamentally explorative example of gameplay metrics analysis. The focus of the case study is challenge. This is historically one of the key objectives that a game user research team investigates; the game should provide the exact right amount of challenge to the target audience. One way to get an initial grasp of this key question in a game like *Tomb Raider: Underworld* is to consider the locations and causes of player death. In essence, investigating areas where players die consistently and repeatedly may signify imbalance in terms of the challenge posed by the areas. Identifying such areas via gameplay metrics analysis provide valuable information about potential problem areas, directing user research on challenge.

This kind of design problem can be addressed during production, as well as post-launch. It also can be studied as a non-spatial and/or a spatial angle. In this example, the metrics data were collected following the launch of *Tomb Raider: Underworld*, which allowed the tracking of the entire population of players. Post-launch data analysis is excellent as it informs us about patches as well as provides information for future game productions. In an MMOG (massively multiplayer online game) context, post-launch analysis is essential to the continued development of the game (as is also discussed in Chaps. 4 and 7; and by Mellon 2009).

In *Tomb Raider: Underworld*, each game level is comprised of multiple "map units" for the purpose of metrics logging, about 100 in total. The Valaskjalf map unit is one of the more complex puzzle/trap locations in the game, featuring multiple different challenges to the players' skills. In analyzing the patterns of death in map, the first step was to produce a heat map based on locations of player death (Fig. 14.17).

Heat maps can be produced in different ways, e.g., using density functions or simply summing the number of deaths occurring within grid cells (as is done here). Such heat maps are excellent for informing the lethality of different game areas. However, it is unspecific as to the nature of the deaths. In order to evaluate where different causes of death, such as death by falling, death caused by different kinds of environmental dangers, or death caused by AI enemies (and if they occurred as intended by the game's design), a series of visualizations showing the areas where players died of different causes (Fig. 14.18).

We wanted to know where players died from a large number of different threats (and died a lot). Such areas potentially represent sites of high challenge for the

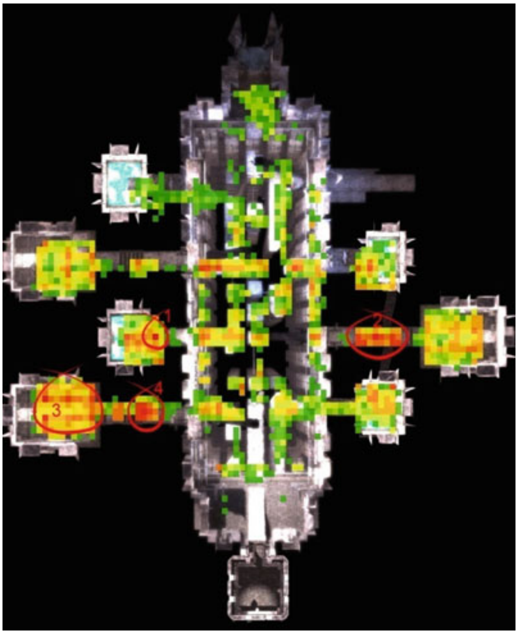


Fig. 14.17 Grid-based heat map of the locations of player death in the Valaskjalf map unit of *Tomb Raider: Underworld*. Scale ranges from *light green* (low numbers of death) to *red* (high numbers of death). Locations with no color have zero deaths. Four of the most lethal areas are marked with *red circles* (Reprinted from Drachen and Canossa et al. 2009b with permission; image is © 2011 by ACM, Inc.)

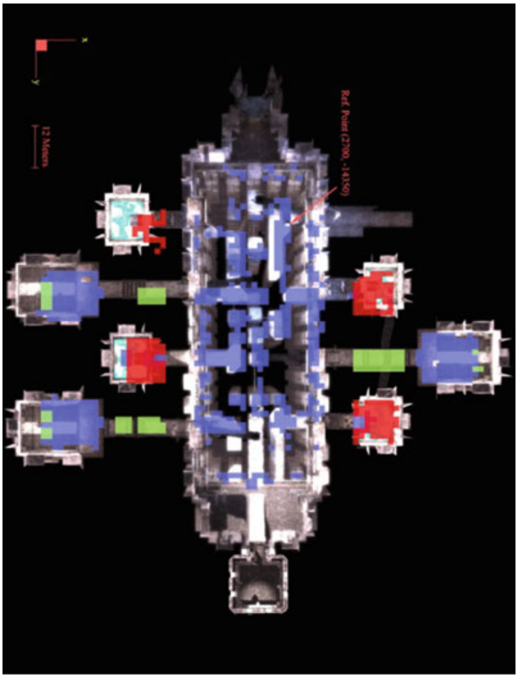


Fig. 14.18 The Valaskjalf map has been overlain with three layers showing the extent of three separate causes of death: Falling (*light blue*), traps (*green*) and water volume [players drowning by being submerged in rising water] (*red*) (Reprinted from Drachen and Canossa et al. 2009b with permission; image is © 2011 by ACM, Inc.)

Fig. 14.19 Overlay analysis showing the areas of the map where the highest number of different causes of death occur, on a scale from *light green* (1–2) to *red* (6). The area with the most variety in causes of death is also one of the places with the highest overall death count (Fig. 14.18) (Reprinted from Drachen and Canossa et al. 2009b with permission; image is © 2011 by ACM, Inc.)

players, and therefore form targets for evaluation about whether their challenge level is too high. To some degree, these areas can be predicted from the game design. However, if there is one thing user research testing has shown game developers, it is that players are very hard to predict. So, in this case, we added a series of layers on top of the Valaskjalf map, each layer containing the spatial distribution of one cause of death. Using GIS, we performed a simple count across the layers (8 in total) (Fig. 14.19).

The result of the overlay analysis shows the distribution of death causes within the map. Four areas, however, showed several different causes of death (see Fig. 14.19). For Area 1, a high number of deaths occurs in one specific grid cell (about 5*5 m) caused by a low variety of causes: the attack of a Thrall (an AI-enemy, third row in Fig. 14.20) combined with a tricky jump (death by falling, fourth row in Fig. 14.20). If the number of deaths occurring in this area is deemed to be high (i.e. prevents or diminishes player enjoyment), the analysis suggests two ways of solving the problem: making the jump easier or eliminating the Thrall enemy.

Area 2 (second column, Fig. 14.20) also shows a high number of deaths and even though there are only two different causes: tarantulas (third row) and traps (fourth row), the distribution of tarantula kills on the *Valaskjalf* map is not spread enough to justify all the deaths displayed. This means that most of the deaths are caused by the traps. This could suggest that the traps should be more lenient. Area 3 displays a high number of deaths, which is motivated by a varied array of

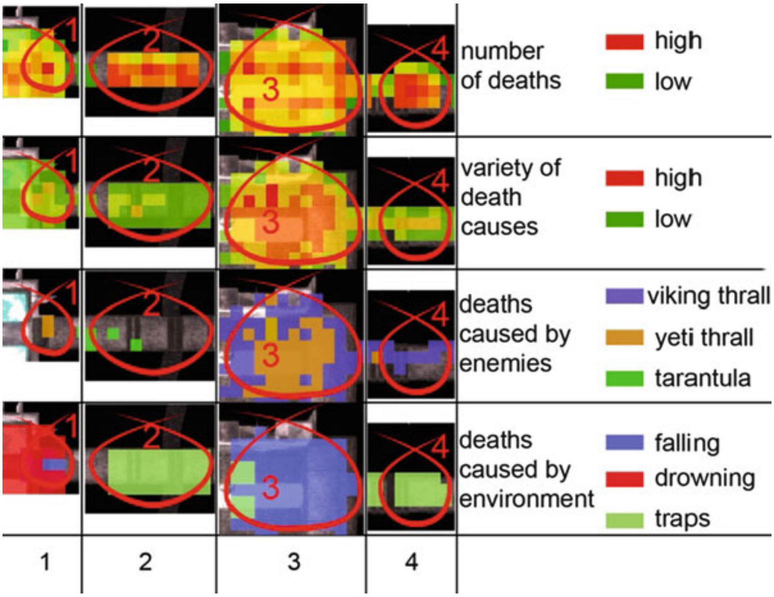


Fig. 14.20 Detail of the overlay analysis with a breakdown of the four targeted areas with multiple causes of death. Four *ArcGIS*-derived layers included: Aggregated death count, aggregated causes of death (*top two rows*) and deaths specifically caused by enemies or environment effects (*bottom two rows*) (Reprinted from Drachen and Canossa et al. 2009b with permission; image is © 2011 by ACM, Inc.)

causes: enemies, environment effects and falling – this is the climax of the level and clearly the toughest part to get through without dying. As with Area 1, a revision of the challenge level might be useful here – further attention is required. Area 4 displays very similar characteristics to Area 2 with similar implications in terms of the play experience.

The (admittedly rather simple) spatial analysis has thus identified potential trouble spots in the *Valaskjalf* map design, which subsequently can be analyzed in further detail. For instance, it is possible to compare the data with user-satisfaction feedback from the level to evaluate whether there is a problem or not. GIS permits different layers to be turned off and on flexibly, and also that specific layers be given different weights in the analysis – if, for example, players dying of electrocution is an unwanted occurrence in the game design, this can be given a greater weight and thus show up stronger in the analysis. Additionally, maps can be exported using an extension as dynamic reports, which permits the user to add or remove layers dynamically, forming the perfect reporting tool for giving feedback to e.g. designers. This type of analysis, even though explorative and time consuming to begin with, has the potential for being operationalized into automated metrics queries in the user research of new titles.

14.8 Working with Gameplay Metrics

There are a few experiences that we would like to pass on from our work with gameplay metrics and other approaches to GUR. Perhaps the key experience we have made is that a healthy user research system has been the continual interplay or feedback between different methods for investigating and researching games – and this includes game analytics.

Managing user research, to us, is very much concerned with keeping the movement and information exchange flowing between the different parts of the user research process, and allowing the user research system to learn. The same goes for working with gameplay metrics – it provides the biggest ROI when there is a continual openness to and constant interplay between work based on synthesis and analysis, exploration and hypothesis, user research and design, and small and large scale data sets, etc. For example, evaluating data from a single user research participant can provide important new information (as is known from usability testing where a small number of people can typically find the majority of interface errors), but will often need to be validated with data from a larger group. Similarly, the research-driven needs should be aligned with the designer (and business) goals, and conversely, designer-driven questioning should be aligned with GUR goals and methods. Additionally, method interplay is important, i.e., how the interplay between different approaches should be handled. This is not a mundane challenge since the theoretical foundations of different methods often clash significantly, as is for instance the case for usability (positivism) and participatory design (interpretive social science) (Silverman 1993).

Apart from these overall considerations, there are three other issues we would like to highlight:

- **Remember that gameplay metrics inform what players are doing, not always why.** Gameplay metrics provide information only regarding actions undertaken in-game by players, it is not directly possible to assess reasons and motivations behind the action, unless additional user data are captured – although inferences can be drawn. Gameplay metrics do not inform whether the player is having a good day, or what the player thinks of the game experience. In short, gameplay metrics cannot provide any contextual data. A metrics tracking tool can only record information from the specific game software. When an analysis of a set of metrics data point to a specific player behavior, it is therefore almost always a good idea to combine the approach with observations or user feedback (even if just some simple questions like how fun an encounter was, what they found to be the most frustrating/challenging, etc.). The same point is emphasized by our GUR colleagues in most of the conference presentations and talks we are aware of (see e.g. Chaps. 21 and 22; or Isbister and Schaffer 2008). This is part of the feedback loop that keeps the metrics tools relevant and useful. In addition to utilizing gameplay metrics in user-oriented testing, *IO Interactive*, *Crystal Dynamics* and other *Square Enix Europe* developers potentially involves a battery of methods, including audiovisual recording and analysis, survey-based approaches, expert testing, different forms of usability testing, etc., depending on the specific requirements of the user test. The combination of gameplay metrics analysis with existing methods for user-oriented game testing provides the ability to probe player behavior and its causes in detail.

- **Find the right metrics to track at the start of the process.** Our recommendation is to involve the consideration of which gameplay metrics to track as early as possible. The earlier that the design and user research teams sit together and figure out what information to track, the better, with the understanding that you will always learn something new about this on the way. Also, you should considered whether data are to be logged post-launch, and if/how metrics should tie in with community feedback. There are no general rules about what should be logged; it all depends on the specifics of the game design. It is, however, vital to ensure that your logging system is flexible to accommodate the adding of new variables as the development progresses so as to make it more likely that exploration and hypothesis work is actually possible with the available data.
- **Manage the allure of numbers.** Gameplay metrics present hard numbers about player behavior, convincing diagrams and what not. However, critical thinking should always be applied – sometimes the analysis will show one thing through a flashing red color on a heat map or another suspicious pattern in the data, but the problem may actually rest in some minor detail in the design. The expertise of the designers is important to spot these kinds of problems, and this is another argument for why user-research and designers should work closely together. Also, and even more importantly: just because it looks good, does not mean it is true. Heat maps and graphs look cool and travel better in organizations than two pages of text with detailed explanation of a specific finding from a comprehensive user test. Heat maps, data visualizations and diagrams are deceptively easy to understand, but, also, they make it easy to ignore other factors that could potentially hold an impact on whatever is being investigated, but which is not included in the metrics-based analysis in question. Heatmaps can be printed out, provide valuable feedback on design, and also used as trophies on the wall of an office, or they can be powerful tools in the politics behind game development. So what is wrong with that? Nothing necessarily, except it can potentially take a life on its own and be used out of context, thereby escaping the feedback loop that is supposed to keep it in check.

14.9 Final Considerations

In this chapter we have attempted to provide some insight into a few simple ways to work with gameplay metrics in practice during mid-late production, in the specific context of single/multi-player, third-person 3D-games like *Kane & Lynch: Dog Days, Tomb Raider: Underworld* and *Fragile Alliance 2*. There are many ways to work with and utilize this highly useful source of user behavior data both during production and post-launch, and ours is just one of these. Apart from the differences between single and multiplayer games, the degree of non-linearity and whether the game in question supports a persistent world or not, game type is also highly important. In essence, working with traditional boxed single/multi-player games is somewhat different from persistent-world massively multi-player online games and social online games (see also Chaps. 4 and 5), where a substantial focus is on the application of gameplay metrics analysis and synthesis to tune game design on a running basis, as well as for monitoring churn rates, average revenue per user and similar business metrics. Even within the confines of the shooter genre, there are different approaches to game analytics, but there appears to be a general consensus that gameplay metrics mesh well with other user-oriented approaches for evaluating and testing games.

Our experience has shown that more exploratively-oriented enquiries initiated during production – case 2 about finding out if the kill balance worked – being a good example, have a tendency to become part of the daily practices, and topics/issues that were regularly consulted by games user researchers and designers alike. While explorative work is often more time consuming and less certain to produce actionable results on than work driven by specific questions or hypotheses, the potential benefit can be substantial. It is in the interplay between different approaches where the best procedures and methods are developed. We emphasize an adaptive and flexible approach, having developed in-house tools and appropriated off-the-shelf software for the purpose, because we acknowledge that games are different. For example, there is a big difference in the metrics that are of interest in an RTS as compared to a soccer management game. The user research and analyses that we conduct during the production of different games varies.

In summary, gameplay metrics analysis addresses one of the major challenges to GUR, namely that of tracking and analyzing user behavior when interacting with the very complex systems that make contemporary computer games. As a user-oriented approach, it complements existing methods utilized in the industry very well, providing detailed and quantitative data to supplement qualitative and semi-quantitative data from other user research methods.

14.9.1 Next Steps

If interested in reading more about practical work with game telemetry in a game development context, or GUR in general, in addition to all the chapters in this book, we suggest the following:

- Article from industry and research: Pagulayan et al. (2003), Thompson (2007), Kim et al. (2008), Isbister and Schaffer (2008), Mellon (2009), Drachen and Canossa (2011) and Zoeller (2010).
- Conference presentations: In general, there have, for the past 5–6 years, been a series of excellent talks at the yearly Game Developers Conference on the topic of game telemetry analysis, both in the main conference and the associated summits, notably the Social Games Summits.

- Websites: Another source of information is Gamasutra.com, whose features-section has been host to several GUR-oriented articles by some of the best people in the industry working in the area.
- The GUR-SIG: the International Game Developers Association hosts a Special Interest Group on Game User Research, which hosts a collection of GUR-literature and –references on the website: http://www.mendeley.com/groups/758231/gur-sig/ (the collection may be moved in the future).

About the Authors

Anders Drachen, Ph.D. is a veteran Data Scientist, currently operating as Lead Game Analyst for Game Analytics (www.gameanalytics.com). He is also affiliated with the PLAIT Lab at Northeastern University (USA) and Aalborg University (Denmark) as an Associate Professor, and sometimes takes on independent consulting jobs. His work in the game industry as well as in data and game science is focused on game analytics, business intelligence for games, game data mining, game user experience, industry economics, business development and game user research. His research and professional work is carried out in collaboration with companies spanning the industry, from big publishers to indies. He writes about analytics for game development on *blog.gameanalytics.com*, and about game- and data science in general on *www.andersdrachen.wordpress.com*. His writings can also be found on the pages of Game Developer Magazine and Gamasutra.com.

Alessandro Canossa, Ph.D. is Associate Professor in the College of Arts, Media and Design at Northeastern University, he obtained a MA in Science of Communication from the University of Turin in 1999 and in 2009 he received his PhD from The Danish Design School and The Royal Danish Academy of Fine Arts, School of Architecture. His doctoral research was carried out in collaboration with Io Interactive, a Square Enix game development studio, and it focused on user-centric design methods and approaches: prototypical player behaviors are described procedurally and from those profiles game environments emerge that are able to accommodate all of the possibilities for action. In the course of his research, Dr. Canossa has published more than 30 articles, book chapters, and journal contributions, he has presented at several international conferences including Future Play, GDC San Francisco, GDC Canada, Mindtrek, IFIP Interact, IEEE Conference on Computational Intelligence and Games, ACM Foundations of Digital Games and DiGRA discussing topics from game user research, HCI, game metrics analysis and player experience. He has also received a Best Paper award at the largest media conference in Northern Europe, MindTrek, in 2009. Beside his academic activities he still maintain contact with the industry: his work has been commented on and used by companies such as Ubisoft, Electronic Arts, Microsoft, and Square Enix. Within Square Enix he carries on an ongoing collaboration with IO Interactive, Crystal Dynamics and Beautiful Games Studio, where he has worked on titles such as Kane & Lynch: Dead Men, Tomb Raider: Underworld and Kane & Lynch: Dog Days.

Janus Rau Møller Sørensen: User Research Manager at Crystal Dynamics and IO Interactive (Eidos/Square Enix), managing games user research activities in the US and Europe. Janus has a multidisciplinary academic background in Information Studies and Ethnography & Social Anthropology, and he has worked in the game industry since 2004 – first in Denmark at IO Interactive, and since 2009 in the US at Crystal Dynamics – on titles such as Hitman Absolution, Deus Ex: Human Revolution, Tomb Raider, and Lara Croft & the Guardian of Light. Janus's chief work-objective is to help enable great game experiences by putting a deep understanding of real people at the center of design decision-making; seeing gamers as human beings, not just behaviors or numbers in a spreadsheet.

References

Canossa, A., & Drachen, A. (2009). *Patterns of play: Play-personas in user-centered game development*. *DIGRA*. London: DIGRA Publishers.

Canossa, A., Drachen, A., Sørensen, J. R. M. (2011). Arrgghh!!! – Blending quantitative and qualitative methods to detect player frustration. In *Proceedings of the 2011 foundations of digital games conference*. Bordeaux: ACM Publishers.

DeRosa, P. (2007, August 7). Tracking player feedback to improve game design. *Gamasutra*. http://www.gamasutra.com/view/feature/1546/tracking_player_feedback_to_.php

Drachen, A., & Canossa, A. (2009a). Towards gameplay analysis via gameplay metrics. In *Proceedings of the 13th MindTrek*. Tampere: ACM-SIGCHI Publishers.

Drachen, A., & Canossa, A. (2009b). Analyzing spatial user behavior in computer games using geographic information systems. In *Proceedings of the 13th MindTrek*. Tampere: ACM-SIGCHI Publishers.

Drachen, A., & Canossa, A. (in press). Evaluating motion. Spatial user behavior in virtual environments. In O. Stammaa, A. Lugmayr, H. Franssila, P. Näränen, & J. Vanhala (Eds.), *Special issue on ACM Academic MindTrek 2009: Everyday life in the ubiquitous era*. Inderscience Publishers. *International Journal of Arts and Technology (IJART)*, 4(3), 294–314. doi:10.1504/IJART.2011.041483 (2011 issue).

Drachen, A., Canossa, A., & Yannakakis, G. (2009). Player modeling using self-organization in Tomb Raider: Underworld. In *Proceedings of the IEEE computational intelligence in games* (pp. 1–8). Milan: IEEE Publishers.

Han, J., Kamber, M., & Pei, J. (2005). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufmann.

Isbister, K., & Schaffer, N. (2008). *Game usability: Advancing the player experience*. San Francisco: Morgan Kaufman. Available from Amazon here: http://www.amazon.com/Game-Usability-Advancing-Player-Experience/dp/0123744474

Kennerly, D. (2003, August 15). Better game design through data mining. *Gamasutra*. Available from: http://www.gamasutra.com/view/feature/2816/better_game_design_through_data_.php

Kim, J. H., Gunn, D. V., Suhuh, E., Phillips, B. C., Pagulayan, R. J., & Wixon, D. (2008). *Tracking real-time user experience (TRUE): A comprehensive instrumentation solution for complex systems*. Florence: Computer–Human Interaction (CHI).

King, D., & Chen, S. (2009). Metrics for social games. In *Proceedings of the social games summit,* San Francisco.

Kuniavsky, M. (2003). *Observing the user experience: A practitioner's guide to user research*. San Francisco: Morgan Kaufman Publishers.

Lameman, B. A., Seif El-Nasr, M., Drachen, A., Foster, W., Moura, D., & Aghabeigi, B. (2010). User studies – A strategy towards a successful industry-academic relationship. In *Proceedings of future play 2010* (pp. 1–9). Vancouver: ACM Publishers. doi:10.1145/1920778.1920798.

Lewis-Ewans, B. (2012, April 24). Finding out what they think: A rough primer to user research, parts 1 and 2. *Gamasutra*. http://www.gamasutra.com/view/feature/169069/finding_out_what_they_think_a_.php

Longley, P., Goodchild, M. F., Macquire, D., & Rhind, D. (2005). *Geographic information systems and science*. Chichester: Wiley.

Mahlman, T., Drachen, A., Canossa, A., Togelius, J., Yannakakis, G. (2010, 18–21 August). Predicting player behavior in Tomb Raider: Underworld. In: *Proceedings of the 2010 IEEE symposium on computational intelligence and games, CIG'10* (pp. 178–185), Copenhagen.

Mellon, L. (2009). *Applying metrics driven development to MMO costs and risks*. Versant Corporation. http://www.maggotranch.com/gdc2004.ppt

Pagulayan, R. J., & Keeker, K. (2007). Measuring pleasure and fun: Playtesting. In C. Wilson (Ed.), *Handbook of formal and informal interaction design methods*. San Francisco: Morgan Kaufmann Publishers.

Pagulayan, R. J., Keeker, K., Wixon, D., Romero, R. L., & Fuller, T. (2003). User-centered design in games. In *The HCI handbook*. Mahwah: Lawrence Erlbaum Associates.

Seif El-Nasr, M., & Zammitto, V. (2010). User experience research for sports games. In *GDC submit on games user research*.

Silverman, D. (1993). *Interpreting qualitative data*. London: Sage Publications Ltd.

Sterne, J. (2002). *Web metrics, proven methods for measuring web site success*. London: Wiley.

Thompson, C. (2007). How Microsoft labs invented a new science of play. *Wired Magazine 15*(9). URL: http://www.wired.com/gaming/virtualworlds/magazine/15-09/ff_halo?currentPage=all

Vercellis, C. (2009). *Business intelligence: Data mining and optimization for decision making*. Chichester: Wiley.

Zoeller, G. (2010). Game development telemetry. *Presentation at the game developers conference*, San Francisco

Zoeller, G. (2011). MMO rapid content iteration. *Presentation at GDC Online 2011*, San Francisco, CA. URL: http://gdc.gulbsoft.org/