Tracking Real-Time User Experience (TRUE): A comprehensive instrumentation solution for complex systems

Conference Paper · January 2008

DOI: 10.1145/1357054.1357126 · Source: DBLP

Daniel V. Gunn
Microsoft

Eric Schuh
Microsoft

Bruce Phillips
Microsoft

Randy J. Pagulayan
Microsoft

Tracking Real-Time User Experience (TRUE): A comprehensive instrumentation solution for complex systems

Jun H. Kim, Daniel V. Gunn, Eric Schuh, Bruce C. Phillips, Randy J. Pagulayan, and Dennis Wixon
Microsoft Game Studios
1 Microsoft Way Redmond, WA, USA
{junkim, dgunn, eschuh, bphil, randypag, denniswi}@microsoft.com

ABSTRACT

Automatic recording of user behavior within a system (instrumentation) to develop and test theories has a rich history in psychology and system design. Often, researchers analyze instrumented behavior in isolation from other data. The problem with collecting instrumented behaviors without attitudinal, demographic, and contextual data is that researchers have no way to answer the 'why' behind the 'what.' We have combined the collection and analysis of behavioral instrumentation with other HCI methods to develop a system for Tracking Real-Time User Experience (TRUE). Using two case studies as examples, we demonstrate how we have evolved instrumentation methodology and analysis to extensively improve the design of video games. It is our hope that TRUE is adopted and adapted by the broader HCI community, becoming a useful tool for gaining deep insights into user behavior and improvement of design for other complex systems.

Author Keywords

Instrumentation, User Initiated Events, Usability, TRUE.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

User initiated events (UIEs) refer to events that occur when a user directly interacts with a system. For example, when a user clicks the print button in a word processor, that UIE triggers the "print" algorithm within the application. Similarly, when a player kills a dragon in a game, the dragon death sequence begins. While such events are being monitored by the application or game, and progression within the application or game is frequently contingent on them, often little is done with UIEs outside of the system.

One methodology that has recently begun to show promise within the HCI field is the automated tracking or logging of UIEs to better understand user behavior [6, 9, 11, 21]. While users are interacting with an application, the systems log all of their UIEs. This logging of UIEs, or *instrumentation*, enables the usability practitioner to automatically record specific user behaviors and more accurately calculate usability metrics of interest (e.g., time to completion, errors). Without these tools, these measurements require user researchers to meticulously hand-code behaviors of interest. By having the system automatically keep track of these behaviors, researchers can reduce errors and save significant time.

While the idea of automatically recording what users are doing while interacting with a system and using that information to improve the system design may seem simple, the actual process is not. Consider the sheer number of possible UIEs that could be captured, for example, in a flight simulator (such as user initiated changes in airplane pitch, yaw, roll) over the course of a flight, or in a tax software application (such as the user switching between deduction worksheets, W2 forms, and audit checking subroutines) over the course of preparing a tax return. Moreover, once all the data are captured, there are important problems to solve in order to effectively transform, analyze, and interpret the deluge of information so that design improvements can be made. The complexity of today's systems makes successful instrumentation a formidable task. Despite these difficulties, researchers have developed successful instrumentation solutions in a variety of domains.

Classic Instrumentation

The recording of UIEs and subsequent analysis to understand an organism's behavior within a system is not a new idea. Indeed, the method has a long history in the field of psychology, specifically within Behaviorism. One of the most well known examples is the work of B.F. Skinner [22, 23]. Skinner constructed a device known as a "Skinner Box" which would automatically log the UIEs of an animal (e.g., a lever press by a rat or a key peck by a pigeon). This was unlike earlier experimental apparatus used to

CHI 2008, April 5–10, 2008, Florence, Italy.
Copyright 2008 ACM 978-1-60558-011-1/08/04…$5.00.

investigate animal behavior, such as Thorndike's puzzle boxes [25, 26]. These used a discrete trial procedure. This procedure required the experimenter to be present throughout each trial in order to record behaviors, record the animal's time to complete the trial, and reset the apparatus. In contrast, Skinner's apparatus *itself* kept track of the animal's behavior. This attribute meant that Skinner's instrumented solution had several key advantages over many of the previous methods used to study animal behavior including:

1.) It did not require the experimenter to be present during the session to manually record the behavior (more efficient)

2.) It helped improve reliability by eliminating direct experimenter interaction with the animal during the session (improved reliability)

3.) Continuous behavior over long periods of time could be accurately recorded and depicted to generate new insights about behavior (larger snapshot of animal behavior could be looked at)

Skinner's methods produced a revolution in the analysis and understanding of behavior. Specifically, it enabled scientists to see patterns in behavior that were impossible to capture with other methods. These methods, in turn, created a breakthrough in the understanding of animal behavior.

Instrumentation in HCI

Over the last 25 years HCI researchers have used instrumentation to better understand users and, consequently, to improve applications. For example, Whiteside et al., [27] collected keystroke data from a text editor while users performed their actual work in order to investigate users' natural usage patterns. Using these data, the investigators were able to derive an overall architecture for a new editor and to convince the keyboard designers to choose the inverted T configuration of arrow keys (the standard on today's keyboard) that was superior the others they were considering. In addition to strict log analyses, researchers have not only recorded UIEs but also played them back through the system to recreate the user's interaction with the system [16]. Such a setup is useful for analyzing and interpreting user data after the session.

These early HCI instrumented studies, while conducted with systems that were relatively simple when compared to today's applications, produced an enormous amount of low-level UIE data (e.g., thousands of single keystrokes). However, the tracking of UIEs in many of today's systems is considerably more complex. Yet, as systems have become more complex, there has been a greater need to come up with a solution to deal with the deluge of data collected, and turn it into something easily interpretable, in order to help improve the system design [15, 20].

Compared to the confines of a Skinner Box, the tracking of UIEs in many of today's systems is considerably more complex. Indeed, extracting usability information from today's applications often requires sophisticated software built specifically for that purpose [6]. Within the 1990s, there was a burgeoning set of tools dedicated to helping industry professionals collect instrumentation data from their systems (e.g., CHIME [1]; EMA [2]; KRI/AG [12]; and AUS [3]). Yet, after an extensive survey of these tools, Hilbert and Redmiles [6] concluded that they fell short. None of the techniques fully supported the practitioner's goal of having an instrumentation solution that can easily map between: UI events and application features; lower level and higher level events (e.g., typing, deleting, moving); and events and the context in which they occur. As Hilbert and Redmiles pointed out, performing transformations and analyses automatically, while providing contextual information, reduces the amount of data that needs to be reported, thereby reducing the possibility of data overload.

There have also been tools utilized for data logging on the web [10]. These methods have been effective for some types of analyses. However, many are limited in their ability to easily tally large sets of data and provide contextual information in conjunction with the data they gather.

Additional UCD methodologies

Just as instrumentation has evolved over the last 25 years, a number of non-instrumentation methods have also evolved. Each method has its strengths and limitations. Three common methods are traditional usability testing, ethnographic methods, and surveys.

In the typical usability test, small numbers of participants perform defined tasks in a lab while thinking out loud [5]. A usability professional watches the participant and records their actions and comments. These types of tests often uncover usability problems quickly and at relatively low cost. They require a working system or simulation (e.g., a paper prototype). Their limitations often include small sample size, limited testing time (usually less than two hours), and the possibility that results could be tainted by interaction between the researcher and the participant.

In ethnographic studies, researchers observe or interview participants while they perform typical activities. Some methods (e.g., Contextual Inquiry [8]) require that the researcher interacts with the participants in order to probe into the assumptions that users make about software and the workarounds they use to accomplish their tasks. The methods produce ecologically sound data and can generate profound insights regarding user activities and their meaning. However, there are a variety of ways to interpret these data [14]. Interpretation can be time consuming and can be difficult to communicate to those who were not directly involved in the initial data gathering. Like traditional tests, these methods require highly skilled researchers, and the interaction between the interviewer and participant may affect the results.

In contrast, surveys [19] usually produce data with little or no direct interaction between the researcher and the participant. They can be done cost-effectively with large samples. However, such research captures self-report data only and the quality of that data is completely dependent on the quality of the question set, the motivation of the respondent to answer honestly and their ability to report accurately. A survey may include open-ended questions, but its ability to uncover unanticipated and contextual aspects of user behavior is limited.

For each of these methods, researchers have developed variants that address the limitations. In addition, researchers often advocate using the methods in combination [4, 13, 17, 18]. However, none of the methods simultaneously combines large sets of behavioral data with deep qualitative data and collect user evaluation all within the same study. By synthesizing existing methods, such as logging and surveys, by conducting them in either controlled (please do the following tasks) or naturalistic (act as you normally would) environments, and by collecting rich contextual data (video recordings), researchers would have a tool of great flexibility and power. If that method were also coupled with well-developed analysis tools, the time involved in analysis could be minimized and the method could be employed by teams to make rapid iterative changes. Finally, by standardizing the method, predetermining the analysis, and eliminating interaction with the researcher, the method would have better reliability.

TRUE – A holistic instrumentation solution

We feel that the HCI field is standing on the threshold of an instrumentation revolution not unlike that wrought by Skinner over 75 years ago. Researchers have made great strides in implementing instrumentation techniques. However, there is not yet a tractable instrumentation solution for the complex systems of today. Over the past 5 years, we have been iterating and refining an instrumentation solution in the context of the most complex systems imaginable for a usability practitioner – human interaction with video games. Our Tracking Real-Time User Experience (TRUE) solution builds upon the strengths of instrumentation and uniquely combines the strengths of the other common UCD methodologies. Moreover, TRUE is highly adaptable and works extremely well in game development, which requires rapid turnaround of results to meet tight production schedules. We feel our solution will not only push user research in games to a new level but will also be equally useful in understanding user interaction with other computer systems.

DEFINING TRAITS OF TRUE

TRUE shares traits in common with other instrumentation and logfile analysis techniques. Like those approaches, users interact with systems that automatically record application events of interest into logfiles, which the system uploads to a server for further analysis.

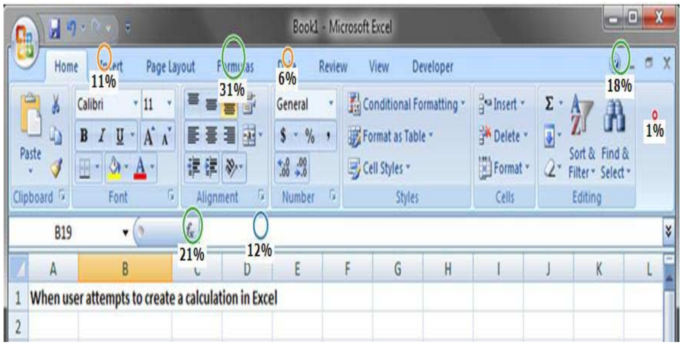
Events in context

One important difference between TRUE and other approaches is that we look at *streams* of data rather than aggregated frequency counts. TRUE systems log sequences of events along with a time stamp for each event. This is different from other approaches that simply increment a counter every time an event occurs. For many research questions, understanding the sequence of events is important for producing actionable recommendations. For example, knowing that a user of a word processing application started a Mail Merge, hit F1, spent 5 minutes in Help, returned to the application, selected a data source, hit F1, spent 3 minutes in Help, selected a different data source, hit F1, spent 5 more minutes in Help, then quits is more useful than simply recording that there were 3 instances of accessing Help for a duration of 13 minutes during the session.

Another difference between TRUE and other instrumentation approaches is the type of information collected. Rather than simply collecting a series of low-level system events (mouse coordinates, generic ‘open dialog’ calls, etc), we advocate collecting *event sets* that contain both the event of interest as well as the contextual information needed to make sense of that event. For example, when using TRUE in a racing game we do not just record a ‘crash’ event every time a player runs into a wall. Instead, we record the car the player was using, the track they were racing on, the difficulty setting they were using, whether the racetrack was wet, and so on every time we record the ‘crash’ event. This event set contains the crucial information needed to determine the root cause of why players keep crashing; this information allows us to get at the ‘why’ behind the ‘what.’

Attitudinal data

An additional defining aspect of the TRUE approach is the inclusion of *attitudinal data* alongside the behavioral data tracked in the event sets. This allows us to capture information that is typically missed with traditional systems that only track UIEs [7, 10]. When testing a racing game using TRUE, we display a brief survey at the end of each race asking the participant whether they had fun and how difficult the race was. If we had simply recorded whether the player had won or lost the race, we would likely make wrong inferences about whether there is a problem that needs fixed. Sometimes failing a race is motivating and part of the fun, while winning on the first time can indicate that the race is too easy or boring. By combining attitudinal data with behavioral we get a much clearer picture of how users experience our products.



When user attempts to create a calculation in Excel

Figure 1: A visualization of where people clicked in a spreadsheet when doing a calculation.

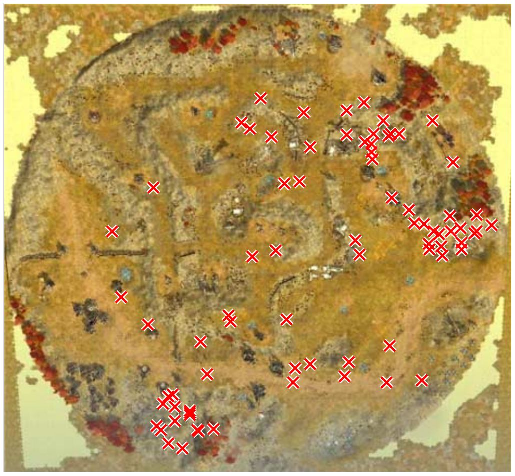


Figure 2: A map displaying where people died in a Real Time Strategy Game

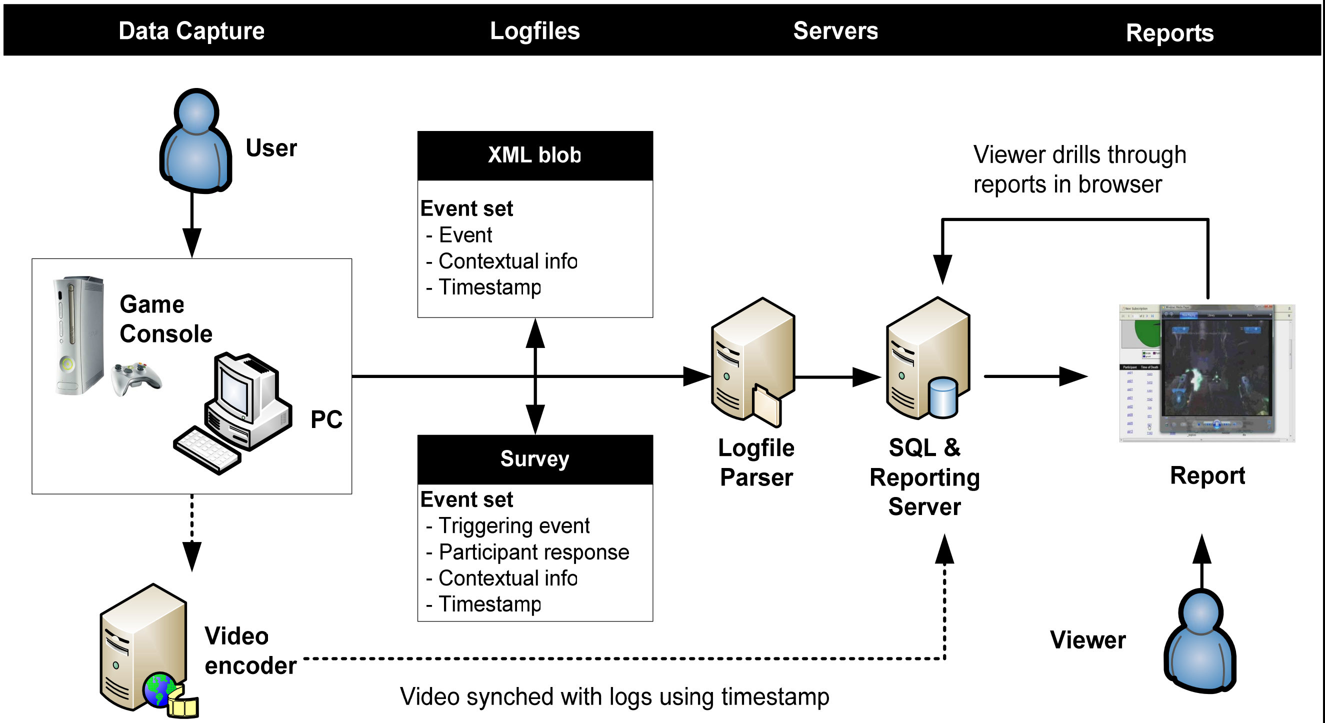


Figure 3: The TRUE architecture

Visualization

A hallmark of the TRUE approach is the use of powerful ways of visualizing and transforming the data, making it easy to spot problems and drill down to determine their root causes. Although how we use TRUE to present the data varies with the research questions we have (see Figure 1 and Figure 2 for some examples), the general approach to viewing and interacting with the data is the same. First, we create a series of graphs and tables that contain all of the those specific goals. Third, we use a meaningful hierarchical scheme. For an Office suite of applications, this could be the entire suite, then specific application, then functionality within an application, then an individual's experience of that functionality within that application. Fourth, our presentation of the data supports drilling down to details from a high-level view and also supports cross-links so that viewers can quickly navigate to desired information.

Video

One of the most powerful forms of data visualization we employ with TRUE is video. We capture digital video of users interacting with our products, which we then synchronize with the timestamp for each event. This creates an automatically indexed video that allows us to jump to positions in the video that we are interested in. We have found this approach of linking video to the logged events extremely powerful for understanding the context in which the users' behavior occurred. Further, TRUE creates these links automatically, bypassing the manual creation of video clips. See Figure 3 for a schematic diagram of the TRUE architecture.

We have found the ability of TRUE to record streams of data along with attitudinal data, each linked to video of the user, to be a very effective approach for understanding how users interact with systems. TRUE enables us to detect the existence of a problem, verify that the problem is detracting from the overall experience of the application, and then drill down to specific occurrences to understand why the problem is occurring [24]. With just a few clicks in real time, we can go from an overview of how people experience an entire product to an understanding of what problems people are encountering, why they are encountering them, and how they feel about it.

The next section outlines how we have used the TRUE system to improve two different games. The first case study demonstrates how we used TRUE in a laboratory setting, taking full advantage of the rich contextual information, drill down reports, attitudinal data, and video to identify problems and verify fixes. The second case study illustrates how we deployed TRUE in a beta context, enabling us to observe naturalistic behaviors of thousands of users over extended periods, using it to identify issues that emerged over time.

CASE STUDIES

Halo 2

The primary goal of this research was to identify and address problem areas participants encountered in the single player component of Halo 2. In particular, we were concerned with unintended increases in difficulty introduced into the game through previous design changes. This example illustrates the ability of TRUE to navigate through large data sets, from high-level visualizations to specific examples. Further, it illustrates how we used attitudinal data alongside the behavioral data we collected.

Halo 2 is a first-person shooter video game released in 2005. There are twelve single player missions that take approximately ten to twelve hours to complete. The missions are linear, containing between ten and thirty smaller encounters. Overall, there are over two hundred encounters in the Halo 2 single player campaign.

Laboratory Set-up

Our lab facility contains 51 individual stations, each with a television monitor, headphones, Xbox development kit, and Xbox controller. The stations also have a PC for encoding game-play video.

Method & Procedure

We describe below a subset of data from two research sessions where we used TRUE for studying the difficulty of the Halo 2 single player campaign.

There were 44 participants who came to our lab and played the Halo 2 single player campaign. Participants were broken into two groups (Session 1 and Session 2). All participants had prior experience with other first-person shooter games. We instructed participants to play through the single player campaign as if they were playing at home, but instructed not to change the default difficulty setting. Most participants were able to finish the campaign during the testing sessions. Data were available for analysis immediately following the end of a testing session.

Behavioral Event Set & Attitudinal Variable

We collected data on several UIEs in the game, including player deaths, the time at which the deaths occurred, who killed them, how they were killed, and other contextual information to help us better understand the UIE.

In addition to the automated collection of UIE we collected attitudinal data. Every three minutes the game was paused and an in-game prompt appeared on the players' screens (see Figure 4).

Select one. This part of the game is…

Too easy

About right, I'm making progress

Too hard, I don't know what to do next

Too hard, I don't know where to go

Too hard, I keep getting killed

Figure 4. Attitudinal on-screen prompt – participants made a selection approximately every three minutes while playing the game.

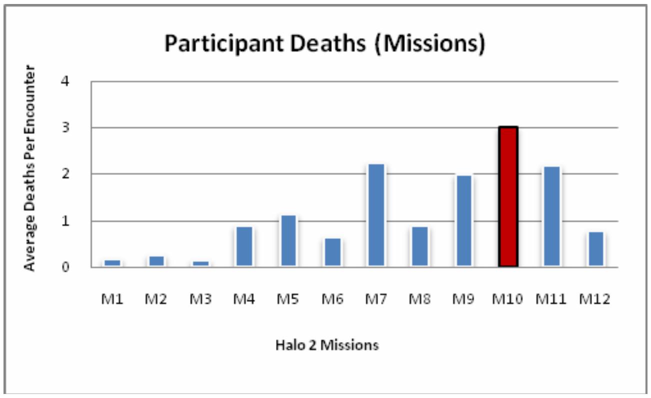


Figure 5. First level of analysis for Session 1. Data represent average number of encounter participant deaths for each mission.

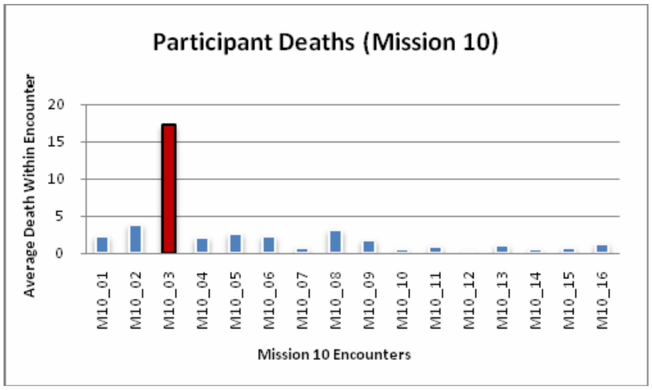


Figure 6. Second level of analysis for Session 1. Data represent average number of participant deaths for each encounter in Mission 1.

Difficulty: Death Identification - Session 1

Using TRUE, we were able to quickly view participant performance at a high level across the individual missions comprising the single player campaign (see Figure 5). In this example, we were interested in the number of times a player died in each mission.

As shown in Figure 5, there were more player deaths in Mission 10 (M10) than in the other missions -- more than we had expected. However, viewing total deaths across missions did not tell us how participants died and whether participants found this frustrating.

To better understand what was happening, TRUE enabled us to drill down into the specific UIEs of the mission to see how participants died in each of the encounters comprising this mission (see Figure 6). Using this drill down, we observed that there may be a problem in the third encounter of the mission. Although these data helped us locate the area of a potential issue with the mission difficulty, it did not provide sufficient information to explain what in particular was causing participants difficulty. We knew that during this particular encounter in the mission, participants were fighting successive waves of enemies in a large room. However, we did not know what in particular was causing them to die.

At this point, TRUE enabled us to drill down even further into specific details of this encounter. Specifically, we were able to break out deaths into the particular causes. In this example, the Brutes (one of the enemies the player had to defeat) were responsible for 85% of participant deaths.

Drilling down into the data even further, we identified three primary ways participants were dying; Brute Melee (27%), Plasma Grenade Attach (the grenade sticks to the player) (22%), and Plasma Grenade Explosions (19%). Being able to isolate the exact cause of deaths was important, because in Halo 2 there are numerous ways enemies can kill a player. This successive drill down approach allowed us to quickly discover the main causes of participant deaths.

However, we still did not completely understand how this was killing the participants. Because the combat system in Halo 2 is complex, we turned to the designers of that system to provide further insight.

TRUE gave us the ability drill down to yet another level of detail. For each death we could link to a video that showed us exactly what happened. With the game designers by our side, we viewed several of the participant deaths, in particular, deaths where participants died due to direct hits from a plasma grenade.

After watching these videos, the designers were able to immediately pick up on a subtle nuance in the game mechanics that only they were able to identify. The Brutes in this section of the game threw grenades faster and with less of an arc, which gave participants less time to react.

Using this information, the designers made several changes to reduce difficulty. Specifically, they prevented Brutes from throwing Plasma Grenades altogether. In addition, they reduced the overall number of enemies and spawned enemies in front of the player so that the enemies would not melee them from behind.

Design Verification - Session 2

To confirm that the design changes from Session 1 worked, we ran a follow up session with a new group of participants playing the same missions. The results from Session 2 confirmed a dramatic decrease in participant deaths at the problem points in the mission. In the first session, there were 311 participant deaths. In session two, this number fell to 34 deaths, with no plasma grenade deaths at all. However, we were still concerned that these changes may have made the game too easy. TRUE enabled us to investigate this by looking at the attitudinal data paired with the UIEs. For comparison, 43% of the participant responses in Session 1 indicated that the third encounter was “Too Hard, I keep getting killed,” 43% of responses indicated “About right, I’m making good progress,” and no participants reported their experience as being “Too Easy.” In Session 2, after we made the design change and reduced the number of participant deaths, the attitudinal data confirmed that we did not make the game too easy as only 4% of the participant responses indicated that the encounter was “Too easy,” 74% of responses indicated it was “About right, I’m making good progress,” and 9% was “Too Hard, I keep getting killed.”

Summary

TRUE gave us the ability to navigate large data sets representing hours and hours of participant game play to identify potential problem areas in the game. When we found indicators, at an aggregate level, that there were issues with mission difficulty, TRUE enabled us to drill down into specific details of the mission, understand the issue, and use this information to provide recommendations to fix the problems. Using the video associated with the UIEs enabled the designers to understand the issues players were facing. Finally, the pairing of attitudinal data with the behavioral data in the event sets confirmed that design changes were consistent with the designers’ intent and player expectation.

Shadowrun

The research we conducted on Shadowrun using TRUE similarly illustrates some of the strengths of this methodology. Specifically, it allowed us to study consumers in a naturalistic setting interacting with the game over an extended period of time. Further, the data we collected allowed us to model player activity with a great deal of precision, which would not have been possible with most observational methodologies.

Shadowrun is a multiplayer, round-based, first-person shooter game that allows players to customize their character in several ways. At the beginning of the game, players are able to choose from one of four different character classes, each with different abilities. In addition, at the beginning of each new round players are able to purchase weapons and additional skills with money collected during prior rounds. This design permitted players a great deal of customization for their character. However, this also created a challenge for the designers: How to balance the game so that one character customization path would not dominate while making all customization paths enjoyable to play.

We had many research goals under investigation at this time, two of which we briefly discuss here. These examples illustrate data collection procedures that differentiate TRUE from other user research techniques. First, we used TRUE to collect naturalistic data over extended periods of time. Most games involve a significant amount of learning and experimentation. By collecting data from players in their homes over the course of months (rather than a few hours in a usability lab), we were able to look at longer-term trends and patterns in player behavior.

Second, TRUE enabled us to collect detailed data that would have been difficult or impossible to gather using observational or questionnaire methodologies. For example, asking participants to record frequent game events manually, such as their choice of weapon or character class, would undoubtedly introduce significant error into the data. Players will misremember, forget, or neglect to engage in accurate behavior logging. Further, much of the data we used would have been impossible to capture through other means. For example, for each combat encounter resulting in a player death we were able to record the players’ precise coordinates in the game world. This precision enabled us to discover patterns in game play encounters and weapon effectiveness that players would not have been able to self-report.

Method

We collected data from ten thousand players who downloaded and played Shadowrun from their homes with each other on their Xbox 360 using the Xbox Live online service. Using TRUE we had a real-time stream of data about players’ choice of character classes and weapons, as well as many other game events and behaviors. The game designers used these data to tweak game parameters and then deliver an updated version of the game to the participants. This iterative process of collecting and interpreting data and updating the game took place over the course of four months.

Character Class Selection

Our first research question concerned the popularity of the character classes. There were four classes in the game, each with different abilities. The intention of the designers was that each character class should provide strengths and weaknesses that differentiate it from the other classes and therefore appeal to different play-styles. However, what we found by tracking character choices using TRUE was that over time one character class was clearly preferred to all of the others. Figure 7 shows the character classes participants selected over the course of a month. The Elf class -- the top line in the graph -- was substantially more popular than the other classes. TRUE enabled us to drill down further into the data to discover that those selecting Elf were also more successful in their games. This was not the intention of the designers.

Over the course of the beta, the designers used these data to tweak the attributes of the various character classes. Through several iterations of the game, the design team was able to realize their original design intention for a balanced selection of character class. Each character class had its strengths and weakness, but no class was clearly dominating the others, as had been the case prior to this research.

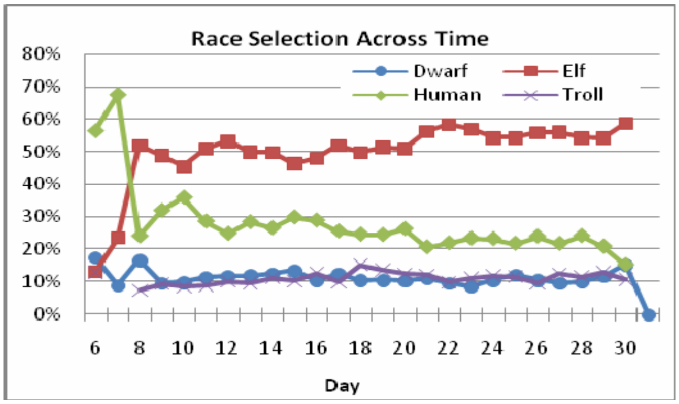


Figure 7. Race selection in Shadowrun

Weapon Usage

Our second question concerned the use and effectiveness of the various weapons in the game. There were several different weapons, each intended to serve different purposes. There was a shotgun for short-range combat, a sniper rifle for long-range encounters, and several weapons in between. Using TRUE we were able to track weapon purchases, overall weapon effectiveness, as well as the effectiveness of weapons at different ranges. Each time a player killed another player we recorded the coordinates of both players in the game world and the weapons they were using. This allowed us to determine the relative effectiveness of each weapon at different distances.

Ideally, the various weapons would enable players to engage in a variety of different combat strategies, therefore enabling different play-styles. Instead, however, we found that the majority of combat encounters were short-range battles with the same weapons. Further, the data revealed that several of the weapons were ineffective at nearly all ranges. Using these data, the designers were able to tweak the damage and range settings for several of the weapons and create more variety in combat encounters.

Summary

These examples illustrate several of the advantages of using an automated logging system like TRUE. First, as illustrated by the investigation of character class preferences, TRUE allowed us to collect data over extended periods of time. The fluctuating popularity of the character classes shown on the left of Figure 7 occurred during a time period when we introduced 1000 new participants into the beta. The popularity of the races changed as the new participants explored each of the races. A few days later, however, the preference for Elf once again emerged. A methodology that limited our investigation to a few games or even a few days of gaming would have given us a misleading picture of character class preferences.

Second, TRUE allowed us to collect very precise data. In this example, the game world coordinates of characters in combat.

Third, TRUE enabled us to do relatively quick iteration of game parameters with participants playing at home. We collected data from thousands of players, used it to tweak game settings, and then uploaded a new version of the game. This iterative process enabled the design team to balance the complex class and weapon systems using consumer data in a timely manner.

SIGNIFICANCE AND DISCUSSION

The use of instrumentation as part of a UCD methodology has the potential to revolutionize the way usability professionals think about, measure, and act on user data. Instrumentation has a long history in psychology and its use in simple systems helped to advance the study of animal learning and revealed new patterns of behavior. Over the last 25 years, usability professionals have used instrumentation more frequently to improve sophisticated systems for users. However, as systems have become more complex, most existing instrumentation solutions have suffered from shortcomings that hinder their ability to transform data into actionable findings within a cross-discipline team and tight development schedules.

In an attempt to improve games, we needed a method that would enable us to 1) detect issues and understand root causes in the same way usability testing does, 2) to support design iteration to the extent that RITE [13] testing affords, 3) to incorporate attitudinal behavior in the manner of surveys, and 4) to understand the naturalistic use of our products as is found with ethnographic methods. We needed to do all of this in the context of the videogame industry, where development budgets are big, business risk is high, schedules are tight, and demonstrating value in a timely way is a necessity.

To that end, we developed TRUE, an instrumentation system that combines the strengths of all of these HCI methods. By recording streams of data and pertinent contextual information in event sets, we retain the ability to detect issues and determine their underlying cause, as is the hallmark of usability testing. By using powerful visualizations, hierarchically organized reports that afford user researchers to drill down to specific details, and linking directly to game play video, we are able to locate quickly the problem spots, understand the issues in context, make fixes, and test again to verify fixes, much like RITE. By including attitudinal data, we tap into the power of surveys in helping understand how people emotionally experience our products. By deploying TRUE in beta tests of products, we are able to make unobtrusive naturalistic observations of usage over time in a manner inspired by ethnography.

We developed TRUE out of the necessity to understand how people use the complex systems that we call games. But games are not alone in their complexity, or in their need to create seamless user experiences. Using software to manage the tangled webs of investments, online bill payments, and tax calculations is similarly complex. As is collaborating synchronously and asynchronously with 6 co-authors using 5 different applications to pull together a paper with arcane formatting rules under a deadline. The truth is that many in the HCI community are struggling to keep apace of the growing complexity of modern systems.

We believe that, because of this increasing complexity, some form of instrumentation is an increasingly important tool in the UCD toolbox. Instrumentation can supplement existing methods, extending their reach to get information about events that happen outside of what can be observed in a laboratory setting, or at a greater granularity of precision, or without the biases associated with having an observer present. It also enables us to see things that were previously difficult to uncover – extended usage patterns, data from large numbers of participants, collection of data that is difficult to observe but has an impact on the user experience.

The TRUE form of instrumentation has been an invaluable tool for us. We have used it for the past 5 years to improve more than 20 games in a variety of genres (racing, shooters, action adventure, role playing, casual), platforms (Xbox 360, PC, web), and phases of development (production, beta, post-release). Moreover, other businesses in our company (productivity applications, developer tools, and mobile devices) have modified their existing instrumentation efforts to incorporate elements of TRUE. Indeed, an instrumentation revolution has begun within our company. It is our hope that TRUE breaks outside of our corporate confines and is adopted and adapted by the broader HCI community, becoming a useful tool for gaining deep insights into user behavior within complex systems.

REFERENCES

1. Badre, A.N. and Santos, P. J. *CHIME: A Knowledge-Based Computer-Human Interaction Monitoring Engine.* Tech Rept. GIT-GVU-91-06, 1991.
2. Balbo, S. *EMA: Automatic Analysis Mechanism for the Ergonomic Evaluation of User Interfaces.* Tech Rept. CSIRO-DIT 96/44, 1996.
3. Chang, E. and Dillon, T.S. Automated usability testing. In *Proc. INTERACT 1997*, IOS Press (1997), 77-84.
4. Davis, J., Steury, K., and Pagulayan, R. A survey method for assessing perceptions of a game: The consumer playtest in game design. *Game Studies: The International Journal of Computer Game Research*, 5 (2005).

   http://www.gamestudies.org/0501/davis_steury_pagulayan/
5. Dumas, J.S. and Redish, J.C. *A practical guide to usability testing.* (Rev. ed.). Intellect Books, OR, USA, 1999.
6. Hilbert, D.M. and Redmiles, D.F. Extracting usability information from user interface events. *ACM Comput. Surv. 32*, 4 (2000), 384-421.
7. Hochstein, L., Basili, V.R., Zelkowitz, M.V., Hollingsworth, J.K., and Carver, J. Combining self-reported and automatic data to improve programming effort measurement. In *Proc. ESEC/FSE 2005*, ACM Press (2005), 356-365.
8. Holtzblat, K. and Beyer, H. *Contextual Design: Defining Customer Centered Systems.* Morgan Kaufmann, San Francisco, CA, 1998.
9. Hurst, A., Hudson, S.E., and Mankoff, J. Dynamic detection of novice vs. skilled use without a task model. In *Proc. CHI 2007*, ACM Press (2007), 271-280.
10. Ivory, M.Y. and Hearst, M.A. 2001. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv. 33*, 4 (2001), 470-516.
11. Kort, J., Steen, M.G.D., de Poot, H., ter Hofte, H., and Mulder, I. Studying usage of complex applications. In *Proc. Measuring Behav. 2005*, Noldus Information Technology (2005), 266-269.
12. Lowgren, J. and Nordqvist, T. Knowledge based evaluation as design support for graphical user interfaces. In *Proc. CHI 1992*, ACM Press (1992), 181-188.
13. Medlock, M.C., Wixon, D., Terrano, M., Romero, R.L., and Fulton, B. Using the RITE method to improve products: a definition and a case study. In *Proc. UPA 2002*, UPA (2002).
14. Miles, M.B. and Huberman, M.A. *Qualitative Data Analysis: An Expanded Source Book*. Sage Publications, Thousand Oaks, CA, 1994.
15. Misanchuk, E.R. and Schwier, R. Representing interactive multimedia and hypermedia audit trails. *Journal of Educational Multimedia and Hypermedia 1,* 3 (1992), 355–372.
16. Neal, A.S. and Simons, R.M. Playback: A method for evaluating the usability of software and its documentation. In *Proc. CHI 1983*, ACM Press (1983), 12-15.
17. Pagulayan, R., Gunn, D., and Romero, R. A Gameplay-Centered Design Framework for Human Factors in Games. In W. Karwowski (Ed.), *2nd Edition of International Encyclopedia of Ergonomics and Human Factors*, Taylor & Francis (2006), 1314-1319.
18. Pagulayan, R., Keeker, K., Fuller, T., Wixon, D., and Romero, R. User-centered design in games (revision). In J. Jacko and A. Sears (Eds.), *Handbook for Human-Computer Interaction in Interactive Systems*, Lawrence Erlbaum Associates (In press).
19. Rea, L.M. and Parker. R.A. *Designing and Conducting Survey Research: A Comprehensive Guide (3rd ed.).* Jossey-Bass Publishers, CA, USA, 2005.
20. Reeves, T.C. and Hedberg, J.G. *Interactive Learning Systems Evaluation*. Educational Technology Publications, Englewood Cliffs, NJ, 2003.
21. Renaud, K. and Gray, P. Making sense of low-level usage data to understand user activities. In *Proc. SAICSIT 2004,* ACM Press (2004), 115-124.
22. Skinner, B.F. On the rate of formation of a conditioned reflex. *Journal of General Psychology*, 7 (1932), 274-86.
23. Skinner, B.F. *The behavior of organisms: An experimental analysis.* Appleton-Century, NY, USA, 1938.
24. Thompson, C. Halo 3: How Microsoft Labs Invented a New Science of Play. *Wired*, 15.09 (2007). http://www.wired.com/gaming/virtualworlds/magazine/15-09/ff_halo
25. Thorndike, E.L. Animal intelligence: An experimental study of the associative processes in animals. *Psychological Review Monograph Supplement 2*, 4 (1898), Whole No. 8.
26. Thorndike, E.L. *Animal Intelligence.* Macmillan, NY, USA, 1911.
27. Whiteside, J., Archer, N.P., Wixon, D., and Good, M. How Do People Really Use Text Editors? In *Proc. SIGOA Conference on Office Information Systems*, ACM Press, (1982), 29-40.