Data Analytics for Game Development (NIER Track)

Kenneth Hullett
UC Santa Cruz
Santa Cruz, CA, USA
khullett@soe.ucsc.edu

Nachiappan Nagappan
Microsoft Research
Redmond, WA, USA
nachin@microsoft.com

Eric Schuh
Microsoft Game Studios
Redmond, WA, USA
eschuh@microsoft.com

John Hopson
Bungie Studios
Bellevue, WA, USA
jhopson@bungie.com

ABSTRACT
The software engineering community has had seminal papers on data analysis for software productivity, quality, reliability, performance etc. Analyses have involved software systems ranging from desktop software to telecommunication switching systems. Little work has been done on the emerging digital game industry. In this paper we explore how data can drive game design and production decisions in game development. We define a mixture of qualitative and quantitative data sources, broken down into three broad categories: internal testing, external testing, and subjective evaluations. We present preliminary results of a case study of how data collected from users of a released game can inform subsequent development.

Categories and Subject Descriptors
D.2.8 [**Software**]: Metrics, K.8.0 [**Personal Computing**]: Games

General Terms
Design, Measurement

Keywords
Game design, Game development, Game metrics

1. INTRODUCTION
PricewaterhouseCoopers (PwC) report on *Global Entertainment and Media Outlook: 2007-2011* estimates that the video game market will increase from $31.6 Billion in 2006 to $48.9 Billion in 2011 [12]. Software engineering research has often focused on software systems ranging from the traditional telecommunication systems to more recent web services. There has been little research on the software engineering aspect of digital games (a.k.a. video games, computer games, electronic games, etc.; referred to simply as games for the remainder of this paper). Games require a significant software engineering effort and have become increasingly complex as games become more sophisticated [2].

Many of the issues in the development, production, and testing of games reflect those of the general software engineering community, and in many cases represent the state of the art. Research communities exist for specialized aspects of game development, such as SIGGRAPH's game track for graphics or AAAI's Artificial Intelligence and Interactive Digital Entertainment for game AI, but none exists for software engineering in games. That said, games are a significantly wide field and in this paper our goals are twofold:

ICSE'11, May 21–28, 2011, Waikiki, Honolulu, HI, USA

(i) Identify a specific area of research and characterize its operation in the gaming community

(ii) Investigate via data analytics the ability to improve game design

A hidden agenda is to introduce this topic to the software engineering community and expose the potential for research in games.

In recent years there has been a rise in interest in the collection and analysis of game metrics, and how they can be used to inform the game development process. As games have gotten larger and more complex, the need for such metrics to make sense of player behavior has increased. The number of reachable states in a modern commercial game title is enormous; without some way to simplify and represent collected data development teams would be unable to act on it in a timely matter.

This paper is organized as follows. In section 2 we discuss the related work. In section 3 we characterize data analysis in games via quantitative (testing) and qualitative (subjective evaluations) aspects to characterize the process by describing how the game industry handles these activities. Section 4 describes our preliminary case study and section 5 our conclusions.

2. RELATED WORK

2.1 Academic
There are some examples of academic work that uses data analysis from games. Dixit et al. performed user tests and created visualizations of the collected data to better understand where users' attention was focused during gameplay [8][9]. This has direct implications for game design by helping designers understand the best places to place clues for players.

Kim et al. presented TRUE, a system for collection and visualization of data from user studies, and presented a case study of its use in Halo 2 [5], a popular First-Person Shooter (FPS) game. They specifically were looking for unintended difficulty increase introduced during development. Through user tests, they collected data on player deaths and opinions on difficulty. They were able to identify several unbalanced elements in the game and correct them before release.

Weber et al. and Lewis et al. both used data mining techniques on large amounts of collected data. In the former, over 5000 replays of expert StarCraft matches were used as training data for a machine learning algorithm for predicting strategy [16]. The strategy predictor became a component of a StarCraft playing bot, thus helping to improve game AI. The latter work presents a case study of large-scale data collection and interpretation of World of Warcraft repositories for better understanding of player behavior [6]. They analyzed how long it took players from each class to reach level 80 (the highest level) in order to empirically evaluate whether the game design is balanced, and confirm or refute common folklore surrounding the game.

2.2 Industry

Articles in industry-focused publications like Gamasutra suggest that the use of data in the game industry as a means to improve design is increasing. Some key examples are presented below.

Russell examined the combat design in *Uncharted 2* [10][11]. They studied both the previous game in the series as well as iterated on the design of their current game. Levels were played repeatedly, and the data collected informed design changes.

Adent discussed the development of *Forza Motorsport 3*, particularly the importance of always having a stable, playable build, and how that feeds into the iterative development of the game [1]. Constant playability means a constant stream of data for the designers to study and make changes accordingly.

Van der Heijden examined the usability testing done for *Swords & Soldiers* [15]. They describe the key questions the developers hoped to answer, the set up and testing process, and what they learned. In particular they were interested in improving the interface design and used eye-tracking data to see where players' attention was focused.

Another example of usability testing is in Thompson's article on *Halo 3* development [14]. They describe the extensive playtesting performed to improve the playability and balance of the game. Large numbers of players were observed and data was collected about how well they performed, leading designers to make adjustments. Players were also asked subjective questions about their level of enjoyment.

Another game in the Halo series, *Halo: Reach*, was subjected to a large beta test – over 2.7 million players and 16 million hours of testing [7][13]. The result was not only finding and fixing bugs, but also significantly tweaking the gameplay by adjusting factors such as weapon damage, reload times, shield recharge rates, etc.

3. SOURCES OF DATA

3.1 Internal Testing

One of the earliest sources of data for game development teams is from their internal testing. This includes informal testing by the developers themselves and more formal testing by the QA team.

3.1.1 Developers

The earliest testers of a game are the development team itself, and therefore are the earliest creators of useful data about the game. In the early development, teams create small prototypes to explore new ideas. While these prototypes are generally discarded once the main development cycle begins, the lessons learned are an important in learning what works and doesn't work in the game.

Once the game is fully in development, the team will continuously be testing the game. Of particular interest to designers is the play balance of the game. Level designers will play levels to ensure that they have the correct difficulty level for where they appear in the game. Matching increasing difficulty to the players' increasing skill as they learn the game is key to keeping players engaged.

3.1.2 QA

The main objective of the QA team is to find bugs and report them to the development team. Statistics from reported bugs are used to make production decisions in much the same way as they are used in traditional software development.

Many bugs are straightforward problems that the programmers, designers, and artists can easily address, but the QA team will often find problems with the playability of the game, including play balance issues. QA testers are often highly skilled game players, and continuously evaluate aspects of the game for difficulty, play time, and balance. Data collected from this playtesting can be used by the developers to make adjustments while the game is still in development.

3.2 External Testing

External testing is testing done by players from the community, rather than members of the development or QA teams. Releases of the game used for external testing are generally instrumented to collect data about the players' actions in the game.

3.2.1 Usability Testing

Usability testing is done with selected members of the target audience to better understand interactions with and reactions to the game. It is generally done under controlled psychological research protocols. To be effective, usability testing must be done late enough in the development cycle so that the game is representative of its final state, but not so late that it's costly to make changes.

In most cases, usability testing is the first time someone outside the organization plays the game. As the development and QA teams have been involved in the project for a long time, they are familiar with how the game is intended to be played and may not realize what is obvious or not to players. By putting a subject in a room and observing them play without instruction or interference, the development team can better gauge their expectations of how players will react to the finished game.

Typical outcomes of usability testing include the need for better tutorials to teach new players and clearer interfaces. Besides the qualitative assessment of players' reactions to the game, quantitative data about the players' specific actions can also be gathered.

3.2.2 Beta Tests

A beta test is a release of a nearly-complete version of a game to a limited set of players. Beta testers are generally selected from a pool of players of previous games.

In the past, beta tests consisted of sending copies of games to members of the pool, waiting for them to play, and receiving back questionnaire responses and comments. However, with the increasing ubiquity of internet connected game machines, the beta version can be downloaded directly to the tester's machine and play data can be reported directly to the development team.

Beta tests can also be contribute to the marketing of a game by giving players a preview of the game and building excitement about the release.

3.2.3 Long-term Play Data

While not actually testing per se, data gathered from players after a game's release can be an important source of data. Due to the increasing ubiquity of internet-connected game, development teams can easily collect player data indefinitely after release. If problems are found, teams can make changes and deliver a new version to players even after release.

Examples of useful data that can be obtained from long-term play include what achievements are earned, how quickly players progress, or prefered game play modes. One well known example of long-term play data are the Halo heat maps [3]. These show the locations of player deaths and kills by different weapons across all

multiplayer maps. By examining these, the team can make adjustments for future releases.

Data from long-term play is particularly useful for maintaining play balance. A lack of balance may not have been appeared in earlier testing, but only becomes apparent after many months of play. An example would be an unanticipated dominant strategy. If, by observing play data, a team sees that a particular weapon has become favored, then they may want to adjust the balance to counter this.

Long-term data can also help teams plan the release of expansion content. When interest in a game starts to wane, developers can release new downloadable content that will entice players to continue playing. Also, examining at what point in their progress players start downloading new content can drive recommendation systems for future players.

3.3 Subjective Evaluations

3.3.1 Surveys

While much of game metrics is focused on quantitative data, qualitative data is also important. Survey data can be collected along with the quantitative data collection during usability and beta testing. This data can be open ended, such as general questions about players' reactions to the game, or structured, such as rating various aspects of a game on a Likert scale.

3.3.2 Reviews

One source of expert data is reviews of games written by professional or non-professional journalists. The games industry is a large, international industry with hundreds of games released each year; game buyers consult reviews to determine what games are most worth spending their money on. By looking at reviews of their own and similar games, developers can decide what aspects to focus on to increase the likelihood of good reviews.

3.3.3 Online Communities

Gaming culture is increasingly involved and worldwide. Gamers don't play games in isolation; they comment upon and read other player's comments on various message boards and blogs dedicated to the subject.

Another aspect of online communities is expert players writing guides for new players. These guides, often called FAQs (from Frequently Asked Questions), are published at websites like GameFAQs.com [4]. Information found in FAQs includes complete walkthroughs of games, strategy guides, maps, and character creation guides.

By monitoring the online communities populated by their players, development teams can get a sense of how their game has been received by the gaming community and how the audience's view of the game matches the design. If the walkthroughs miss some important aspect, then it was too hard to find. If players' assessment of the game's balance doesn't match the team's expectations, then their play balancing may need adjustment.

3.3.4 Post Mortems

It is becoming increasingly common for industry-focused publications to publish game developers' post mortems after a game is released. This is a summary of what went right and wrong in the development process. By studying areas of development that were problematic in other projects, developers can better anticipate and avoid problems in their own projects.

4. CASE STUDY

To illustrate the potential of data for informing future development, we present an example analysis of long-term play data from a released game. We looked at data from *Project Gotham Racing 4*, an XBox 360 game developed by Bizarre Creations and published by Microsoft Game Studios in 2007, and made recommendations for future game development.

This auto racing game features 134 vehicles in 7 classes, 121 routes in 9 locations, and 29 event types in 10 game modes. For the portion of our analysis presented here, we mined 3.1 million entries of game log data, from several thousands of users who played the game. Each entry in the data set represents each time a player started a race in either multi- or single player mode. Data was captured about both the race and the player. Data about the race includes the type of event, the route, the vehicle used and the number of vehicles in the race. Data about the player includes their career rating and the number of events they've completed.

For our analysis we looked at usage patterns for three features of interest to the development team: game modes, event types, and vehicles. We looked at overall usage, as well as usage in single player races and multiplayer races separately.

Our analysis showed that much of the content was underused. In the three features we looked at, 30-40% of the content was used in less than 1% of races. Significant savings could be realized in future versions of the game by reducing the amount of content while still presenting more than enough variety to satisfy players.

4.1 Game Modes

As shown in Table 1, OFFLINE_CAREER (a single player mode) is the most commonly used game mode by far.

Table 1. Game Mode (Reduced)

Game Mode	Races	% of Total
OFFLINE_CAREER	1479586	47.63%
PGR_ARCADE	566705	18.24%
NETWORK_PLAYTIME	584201	18.81%
SINGLE_PLAYER_PLAYTIME	185415	5.97%
….		
NETWORK_TOURNAMENT_ELIM	2713	0.09%

PGR_ARCADE and NETWORK_PLAYTIME (multiplayer modes) are also used in a significant percentage of races. The other modes were used in far fewer races, with NETWORK_TOURNAMENT_ELIM (elimination rounds in a network tournament when racing against a large pool of players) being used in less than 0.1% of races.

4.2 Event Types

When looking at event types, we again see noticeable differences between the most and least popular types. A reduced version of this data is shown in Table 2.

Single player street races were the most popular event type, followed by multiplayer street races and elimination races (knock out stages in tournaments), whereas 12 of the 29 event types were used in less than 1% of races. The underutilization of content is

even more pronounced when looking at event types in multiplayer races only (7 of 16 event types used in less than 0.1% of races).

Table 2. Event Type (Reduced)

Group	Races	% of Total
STREET_RACE	795334	25.60%
NET_STREET_RACE	543491	17.50%
ELIMINATION	216042	6.95%
HOTLAP	195949	6.31%
…		
TESTTRACK_TIME	7484	0.24%
NET_CAT_AND_MOUSE_FREE_ROAM	3989	0.13%
CAT_AND_MOUSE	53	0.00%

4.3 Vehicles

Similarly, out of 134 unique vehicles, 50 were used in less than 0.25% of races, and 16 in less than 0.1%. Each vehicle represents a significant investment: a 3d artist must model it, a texture artist has to decorate it, and a designer has to tweak its performance. The number of vehicles could be reduced by more than 20% and the marketing copy on the back of the box would still be able to say the game contains more than 100 vehicles.

5. CONCLUSIONS

Based on our analysis, we were able to make several recommendations for future development. In particular, our examination of unused content suggests that considerable savings could be achieved.

Across all races:

- 2 of 9 game modes were used in < 0.5% of races
- 12 of 29 event types were used in < 1% of races
- 50 of 134 vehicles were used in < 0.25% of races

When looking solely at multiplayer races:

- 2 of 4 game modes were used in < 2% of races
- 7 of 16 event types were used in < 0.1% of races
- 53 of 133 vehicles used in < 0.25% of races

Our analysis shows that many game modes, event types, and vehicles did not appeal to the players, and were not necessary to hold their interest. Asset creation is a significant expense in modern game development, so a 25% reduction in this area would noticeably reduce costs and development time.

Additional analyses not presented here led to other useful recommendations, such as encouraging new players to play in career mode, thus increasing their engagement and likelihood of continuing play, and encouraging new users to stay with F Class for longer, rather than moving into the more difficult to control A Class vehicles as soon as they are available.

6. REFERENCES

[1] Adent, D. *Forza Motorsport 3* And Predictable Development. Gamasutra.com. http://www.gamasutra.com/view/feature/6182/forza_motorsport_3_and_predictable_.php

[2] Blow, J. 2004. Game Development: Harder Than You Think. *Queue* 1, 10 (February 2004), 28-37.

[3] Bungie.net. Global Heatmaps. http://www.bungie.net/online/heatmaps.aspx

[4] GameFAQs.com. http://www.gamefaqs.com/

[5] Kim, J., Gunn, D., Schuh, E., Phillips, B., Pagulayan, R., and Wixon, D. 2008. Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (CHI '08). ACM, New York, NY, USA, 443-452.

[6] Lewis, C. and Wardrip-Fruin, N. 2010. Mining game statistics from web services: a World of Warcraft armory case study. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (FDG '10). ACM, New York, NY, USA, 100-107.

[7] Nutt, C. *Halo: Reach* - The Beta Story. Gamasutra.com http://www.gamasutra.com/view/feature/5875/halo_reach__the_beta_story.php

[8] Priyesh N. Dixit and G. Michael Youngblood. 2007. Optimal information placement in an interactive 3D environment. In *Proceedings of the 2007 ACM SIGGRAPH symposium on Video games* (Sandbox '07). ACM, New York, NY, USA, 141-147.

[9] Priyesh N. Dixit and G. Michael Youngblood. 2008. Understanding information observation in interactive 3D environments. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games* (Sandbox '08). ACM, New York, NY, USA, 163-170.

[10] Russell, B. A Deeper Look Into The Combat Design Of *Uncharted 2*. Gamasutra.com. http://www.gamasutra.com/view/feature/5883/a_deeper_look_into_the_combat_.php

[11] Russell, B. Designing Combat Encounters In *Uncharted 2*. Gamasutra.com http://www.gamasutra.com/view/feature/5945/designing_combat_encounters_in_.php

[12] Scanlon, J. The Video Game Industry Outlook: $31.6 Billion and Growing. BusinessWeek.com. http://www.businessweek.com/innovate/content/aug2007/id20070813_120384.htm

[13] Sofge, E. How *Halo: Reach* Was Created: Insider's Guide. PopularMechanics.com. http://www.popularmechanics.com/technology/gadgets/video-games/how-halo-reach-was-created-insiders-guide

[14] Thompson, C. *Halo 3*: How Microsoft Labs Invented a New Science of Play. Wired.com. http://www.wired.com/gaming/virtualworlds/magazine/15-09/ff_halo

[15] Van der Heijden, J. Successful Playtesting In *Swords & Soldiers*. Gamasutra.com http://www.gamasutra.com/view/feature/5939/successful_playtesting_in_swords__.php

[16] Weber, B., Mateas, M. A data mining approach to strategy prediction. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on* , vol., no., pp.140-147, 7-10 Sept. 2009