Bugs and Emotion: A Content Analysis of Quality Assurance Player Feedback

Extended Abstract

Luke Thominet
Florida International University
Miami, Florida
luke.thominet@fiu.edu

ABSTRACT

This study uses content analysis to explore how players provide quality assurance (QA) feedback of games in progress. It identifies both the most commonly reported bugs and the bug types that caused the most positive and negative experiences. In doing so, it seeks to build stronger connections between QA and UX research.

CCS CONCEPTS

• **Human-centered computing → HCI design and evaluation methods**;

KEYWORDS

User Experience, Video Games, Content Analysis, Quality Assurance

ACM Reference Format:
Luke Thominet. 2018. Bugs and Emotion: A Content Analysis of Quality Assurance Player Feedback: Extended Abstract. In *SIGDOC '18: The 36th ACM International Conference on the Design of Communication, August 3–5, 2018, Milwaukee, WI, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3233756.3233934

1 INTRODUCTION

There are conceptual and procedural divisions between quality assurance (QA) testing and playtesting in games. While playtesting, a closer analogue of general user experience (UX) research, usually involves players/users, QA testing is primarily the work of internal, specialized testing teams [13]. Yet QA issues have a significant impact on UX [10], suggesting the need to build connections between these two areas. This study ties these testing areas together by examining player feedback tickets from an open development game.

Open development games are publicly released during the development process in order to gather player feedback and iterate on design. These games provide valuable data on how users can engage in technology development. Elsewhere, I have argued that open development projects show how we can design prolonged UX research projects as an engaging experience [2].

SIGDOC '18, August 3–5, 2018, Milwaukee, WI, USA

ACM ISBN 978-1-4503-5935-1/18/08.
https://doi.org/10.1145/3233756.3233934

Notably, open developers have also debated the role of players in QA testing. Some developers argued categorically that players should not be used for QA testing [12], and others argued that players are ineffective at finding bugs [15]. Still, many developers implicitly supported using players as QA testers by recommending centralized locations for bug reporting [7] or acknowledging how bugs affect players' experiences [3].

This study responds to the debate by arguing that rhetorical discourse methods offer insight into how game bugs affect player experience. As such, this study contributes to technical communication's ongoing work to use rhetorical discourse analysis to analyze UX practice [11] and video game play and development [4, 5].

2 METHODS

This study used a data set of 300,000 player feedback tickets for *Subnautica*, a scuba diving, exploration game. The tickets were created through an in-game widget where players submitted textual feedback and reported their emotional state (very unhappy, unhappy, happy, very happy). In a previous study, I analyzed a sample of 3,250 feedback tickets to trace broader UX topics and found that the majority (63.3%) were focused on QA topics [1]. These 2,296 QA player feedback statements were the sample for the current study.

This study sought to answer two research questions: 1) What types of bugs did *Subnautica* players frequently report? and 2) What types of bugs most negatively affected players' experiences?

This study used conventional content analysis [6] to identify the types of bugs that players reported with the initial coding categories drawn from existing game bug taxonomies [8, 9]. This first coding system was developed further during an initial coding pass. The final coding system can be seen in Table 1 below.

To analyze the results, overall frequencies of each bug type were calculated. Then the frequency distribution of players' positive and negative emotions was calculated for each bug type. These distributions were analyzed using a chi-squared test to determine if bug categories showed non-random variation in the reporting of emotions. Results with a p value of $<.05$ were considered significant.

3 RESULTS AND DISCUSSION

The full range of bug types identified in the sample, as well as the frequency of each type is shown in Table 1 below.

Two bug types dominated QA player feedback: game object bugs and visual defects. Since *Subnautica* is an exploration/survival game where players commonly craft new items, these two types of errors were probably apparent during regular gameplay. However, the prevalence of level design reports also suggested that some

Table 1: Code Descriptions and Frequency

Name	Description	Frequency
Game Object	Errors when interacting with in-game objects (e.g., unbreakable coral)	718
Visual	Graphics errors (e.g., rendering)	501
Level Design	Defects with the environment (e.g., holes in the map)	180
Physics	Illogical physics simulation (e.g., flying submarine)	150
User Interface	User interface errors (e.g, missing menu)	150
Player Character	Player character status errors (e.g., sudden health loss)	139
Performance	Game optimization defects (e.g., lag)	124
Stability	Defects in core functions (e.g., crashes)	121
Artificial Intelligence	Illogical non-player characters actions (e.g., poor pathfinding)	112
Audio	Sound errors (e.g., wrong noises)	70

players actively tested the game. For example, one player used the QA concept of verifying a bug when reporting a map hole: "Can replicate issue by being in hatch then deconstructing."

On average, players submitting QA tickets were significantly less happy (38.2% reported being happy) than those submitting non-QA tickets (73% happy). This finding supports industry advice on the importance of polish before a game is distributed to a broad open development community [14].

However, players did not respond to all types of bugs identically. There were significant trends related to 1) entertaining bugs, 2) stability and performance, and 3) core game interactions.

First, players reported being happier than expected when submitting physics (49% happy, p=.009) and visual (44% happy, p=.007) bugs. Notably many players who were very happy when reporting these bugs described them as humorous. For example, "LOL textures EVERYWHERE are goofed."

Conversely, players responded very negatively to stability errors like crashes (20% happy, p=.00003). These bugs are often described as critical errors in QA literature since they are a full disruption of gameplay [8]. Yet players had a more positive response to disruption caused by performance errors (54% happy, p=.0003). This was the only bug category where the majority of players reported being happy. Players did not necessarily enjoy lag, but they were willing to accept it as part of an otherwise positive experience. In fact, players frequently reported performance bugs in tickets that otherwise praised the game: "Getting some stuttering while swimming. Pretty good game! :D."

Game object bugs also had more negative tickets than average (33% happy, p=.0002). In unhappy game object bug reports, the objects were often related to core game interactions (e.g., "I can't access the fabricator. Now I'mma die") or the errors were unrecoverable (e.g., "my Seamoth disappeared"). Hence, these game object bugs could be seen as more critical to positive player experience.

4 CONCLUSION

This study largely confirmed tacit knowledge from open development and QA testing while providing additional empirical evidence linking QA and UX. Bugs negatively affect player experience, and critical bugs related to core functions create even worse experiences. The study also offers new insight into how players find and react to bugs: most players seem to operate from a play rather than testing mindset, even when they submit QA feedback. Because of this, their most common types of feedback align with readily-apparent errors. On the other hand, this mindset also meant that players were more likely to forgive errors that were entertaining. Ultimately, these findings support the idea that players can contribute to QA testing even if they cannot fully replace professional QA teams.

REFERENCES

[1] Author. 2017. Tracing Player Experience: A Content Analysis of Player Feedback Tickets. In *Proceedings of the 35th ACM International Conference on the Design of Communication (SIGDOC '17)*. ACM, New York, NY, USA, 11:1–11:9. https://doi.org/10.1145/3121113.3121218
[2] Author. Forthcoming. How to Be Open: User experience and technical communication in an emerging game development methodology. *Communication Design Quarterly* (Forthcoming).
[3] Jim Brown. 2015. The Impact of Open Development on Unreal Tournament. In *GDC Vault*. Cologne, Germany. http://gdcvault.com/play/1022760/The-Impact-of-Open-Development
[4] Jennifer deWinter and Ryan Moeller (Eds.). 2014. *Computer Games and Technical Communication: Critical Methods and Applications at the Intersection.* Ashgate, London.
[5] Douglas Eyman. 2008. Computer Gaming and Technical Communication: An Ecological Framework. *Technical Communication* 55, 3 (Aug. 2008), 242–250.
[6] Hsiu-Fang Hsieh and Sarah E. Shannon. 2005. Three Approaches to Qualitative Content Analysis. *Qualitative Health Research* 15, 9 (Nov. 2005), 1277–1288. https://doi.org/10.1177/1049732305276687
[7] Trent Kusters. 2016. 'The Armello' Postmortem. In *GDC Vault*. San Francisco, CA. http://www.gdcvault.com/play/1023416/-The-Armello-Postmortem-A
[8] Luis Levy and Jeannie Novak. 2009. *Game QA & Testing.* Cengage Learning.
[9] Chris Lewis, Jim Whitehead, and Noah Wardrip-Fruin. 2010. What went wrong: a taxonomy of video game bugs. In *Proceedings of the fifth international conference on the foundations of digital games.* ACM, 108–115.
[10] Jakob Nielsen. 2103. QA & UX. (Feb. 2103). https://www.nngroup.com/articles/quality-assurance-ux/
[11] Janice Redish and Carol Barnum. 2011. Overlap, Influence, Intertwining: The Interplay of UX and Technical Communication. *Journal of Usability Studies* 6, 3 (2011), 90–101. https://uxpa.org/jus/article/overlap-influence-intertwining-interplay-ux-and-technical-communication
[12] Jeff Spock. 2014. Bringing the Community into the Dev Team - A Look into Open Development. In *GDC Vault*. Los Angeles, CA. http://www.gdcvault.com/play/1021475/Bringing-the-Community-into-the
[13] Maurice Tan. 2012. Understanding User Research: It's Not QA or Marketing! (April 2012). http://www.gamasutra.com/view/feature/168114/understanding_user_research_its_.php
[14] Raphael van Lierop. 2015. There Be Monsters: Harnessing the Power of Community-Informed Development for The Long Dark. In *GDC Vault*. San Francisco, CA. http://www.gdcvault.com/play/1022333/There-Be-Monsters-Harnessing-the
[15] Swen Vincke. 2015. Divinity: Original Sin Postmortem- Success Stories and Lessons Learned. In *GDC Vault*. San Francisco, CA. http://www.gdcvault.com/play/1021934/Divinity-Original-Sin-Postmortem-Success