

Experiences in Teaching and Learning Video Game Testing with Post-mortem Analysis in a Game Development Course

Miguel A. Garcia-Ruiz
Algoma University
Canada
miguel.garcia@algonau.ca

Simon Xu
Algoma University
Canada
simon.xu@algonau.ca

Pedro C. Santana-Mancilla
University of Colima
Mexico
psantana@uacol.mx

Adriana L. Iniguez-Carrillo
University of Guadalajara
Mexico
adriana.carrillo@cusur.udg.mx

Abstract: Video game testing is an important topic that is taught in game design and development courses, including quality assurance (QA) testing (testers look for game “bugs” or game software errors) and playtesting (testers evaluate gameplay and analyze how fun the game is). Testing is a valuable activity done in game development projects, because it uncovers game user interface (UI), interaction design, gameplay and software problems. Video game post-mortems (structured documents describing what went well and what went wrong in game development projects) are widely read in the gaming industry. They explain game development activities, including how testing (or the lack of) was conducted in the projects. 27 students from a game development course taught at an undergraduate computer science level analyzed published post-mortems, including the identification of testing practices. The post-mortem analysis was a very relevant educational activity for them, engaging students in knowing how testing is done in the gaming industry and what testing problems, among other aspects, occur in real-world game development projects. Further work includes devising other pedagogical activities such as running practical in-class game testing exercises and letting the students write their own post-mortems about their course projects.

Introduction

Game testing is an essential game development activity, and is of paramount importance because it allows game developers to uncover game design, user interface (UI) and player interaction problems (Redavid et al., 2011) including “bugs” (game glitches or software errors that produce an undesirable and incorrect result when the game is played). User testing methods are run to see if a video game can be played with ease, and whether it is satisfactory and fun to play (Redavid et al., 2011). A game that contains glitches, bad mechanics and user interfaces is likely to yield a poor gaming experience such as player’s frustration and anxiety, affecting the acceptance and long-term usage of the video game (Schultz & Bryant, 2017). These problems could also diminish sales of commercial video games. In this context, the global video game industry is huge and is growing every year. The worldwide sales of games is more than \$100 billions of U.S. Dollars, and is increasing steadily every year (Fullerton, 2018). This makes pressure to game developers to make games faster and with more features, compromising games quality (Redavid et al., 2011). However, there are no specific and standard methods for testing games while they are developed, and not all the testing methods from software engineering are suitable or directly applicable to video game testing.

Obtaining, analyzing and testing requirements for developing games are special and highly subjective (e.g. analyzing how fun a game is), and it has been argued that video games are a very complex type of software (Fullerton, 2018). This paper will address game testing issues described in post-mortems, conducted by students. A post-mortem is a summary of a past game development project that mainly describes what went right and what went wrong (or could have done better) in the project, written by game developers (Dingsoyr, 2005).

Software engineering projects such as game development projects involve processing a large quantity of information distributed over different knowledge domains such as project management, resource management, software requirement elicitation, software quality, software configuration etc. Since video game projects can last for years and need many resources, the project success is very important.

Post-mortems commonly refer to an improvement-oriented analysis by the project team, which may include various users, customers (players), and stakeholders, to evaluate the past experience and to develop “lessons learned” for the benefit for future projects (Dingsoyr, 2005; Myllyaho et al., 2004).

Game development post-mortems are commonly analyzed in the video game industry, which are structured documents written by a game development team or individual that describes important issues and a retrospective analysis of a finished (or cancelled!) game development project, highlighting what went right and what went wrong during its process (Grossman, 2013). They are generally published in the form of book chapters, articles and web pages. One objective of post-mortems is to avoid past projects’ mistakes in current and future game development projects, as well as learning best practices from them. Good examples of post-mortems are reviewed in Wawro (2015).

The objective of this paper is to highlight the importance of game testing described in game post-mortems, and how their analysis can be systematically carried out by students from a game development course, summarizing useful and valuable knowledge and best practices from the gaming industry, which can be applied in the further students’ game development projects.

Literature Review

Washburn et al. (2016) conducted an extensive analysis of 155 post-mortems published in the e-magazine Gamasutra.com, categorizing aspects such as product, development, resources and customer support from the post-mortems, and summarizing findings on what went right and what went wrong in each of them and their sub-categories. Testing was briefly analyzed in their paper as a sub-category of “Development”, and software testing was briefly mentioned in other categories. Interestingly, some of the analyzed post-mortems reported a lack of testing in the game development projects as some aspect of what went wrong in those projects. Grossman’s book (Grossman, 2013) compiled 25 post-mortems of popular video games (at the time of the book was published), showing that testing was a common and important activity in all of them. Some of the post-mortems point out that the time and resources devoted to testing was inadequate, and the project managers wished to have more time allocated for that. McAvoy (2006) analyzed and discussed attitudes and beliefs that project members have towards their own game post-mortems. In it, McAvoy found that in a post-mortem from a game called “The Pygmalion Effect”, project testers’ expectations affect how the rest of the team view software testing, including overly positive beliefs about the game code that may hinder quality assurance testing. This confirms a common effect in software engineering where software testers cannot identify some bugs in their own software. Tschang (2005) reviewed a number of post-mortems published in the magazine *Game Developer*, finding that both positive and negative development features were often described in them, concluding that testing is a very critical activity in video game development projects.

Wang (2009) described how students from a software architecture course wrote post-mortems of their own game development course projects. Students found that writing their own post-mortems was a positive game development activity, learning from their own experiences in order to improve game development. Wang also pointed out that students’ structured brainstorming and root-cause analysis about their post-mortems was a very effective way of revealing positive and negative issues from their game development projects. However, it is not clear whether the students also reviewed already-published post-mortems written by people from the gaming industry. Similarly, Maxim et al. (2007) and Scacchi (2015) found that letting students create and describe their own post-mortems (even if those are informal) can be valuable for sharing their lessons learned with each other and for getting feedback to be used to improve their game designs. In addition, Scacchi (2015) highlighted the benefits of students describing their own post-mortems who participated in game jams (intensive game development sessions), namely sharing common problems within team projects. Volk (2008) described the development of a game

engineering course in the context of a computer science curriculum. Volk explained that students writing game post-mortems in the post-production phase is an essential and important game engineering activity, highlighting game evaluation as a central task.

To our knowledge, there is no academic publication describing a systematic analysis of testing in already-published post-mortems, and particularly applying this analysis in an educational context. This can be very valuable in game development courses, as students learn what went right and what went wrong in game projects from the video game industry, linking what they learn and experience in the course and what is done in real-world game projects. As Poulsen (2011) points out, “When used in education, good post mortems resonate with our desire to evaluate and reflect upon the process, but in a way more closely resembling the way “real” professionals work” (p. 102).

Testing in Game Development Projects

Game testing can be defined as an investigation and activities performed by game development team members, testers, potential players and other project stakeholders, to check whether the actual results of a developed game match the expected results and to identify errors and missing design requirements (Schultz & Bryant, 2017). In the software engineering context, we analyze two main branches of testing: black-box testing, where the video game is seen as a black box and people only revise and review its inputs and outputs (Redavid et al., 2011). An example of black-box testing is quality assurance (QA) testing, where testers look for bugs (game glitches) and report them. Another type of black-box testing is user testing (e.g. usability testing and playtesting), which strives to obtain valuable user feedback that will serve to fix eventual problems from the game’s user interface, mechanics and gameplay (Fullerton, 2018). This type of black-box testing has been described in post-mortems. There are no standard methods for conducting game testing, hence the importance of analyzing them from post-mortems.

This paper shows how undergraduate students from a game development course analyzed published post-mortems, in order to learn how to improve the quality and player experience of video games through testing, among other game development aspects.

Students’ Post-mortem Qualitative Analysis

The first author taught the game testing topic in the Game Software Engineering course from Algoma University’s Bachelor of Computer Science, located at Sault Ste. Marie, Canada. In this course, each of the 27 students selected, analyzed and discussed 3 post-mortems for an individual assignment comparing and contrasting them, focusing on testing aspects. Previously to the assignment, very few of the students knew what a post-mortem is, and most of them were aware of game testing as being an important part of the game development process. Students were taught about main aspects of game testing and post-mortem definitions, their structure and importance in the gaming industry, and a sample of important post-mortems. The post-mortems were obtained from the specialized online magazine Gamasutra.com. In the same class session, students were asked to identify testing issues from the post-mortems samples. This small in-class exercise effectively supported the assignment.

The following were the assignment’s instructions given to the students, having a week to complete it:

- Find on the Internet at least 3 game development post-mortems about shooting games only. You can find lots of post-mortems from the online magazine called Gamasutra (<http://www.gamasutra.com/features/postmortem/>). You can find them from other sources.
- Write a summary about each of them. In addition, pay attention to testing aspects that were explained in the post-mortems.
- Write suggestions about what else game developers could improve or do in the first place in each of the projects described in the post-mortems you chose, in order to avoid the problems they described in the post-mortems.
- Write the sources of the post-mortems (links).

The students analyzed the selected post-mortems by following Miles and Huberman’s (1994) stages for qualitative data analysis, as follows:

I) Preparation (Review and Organization)

In this stage, students searched post-mortems from Gamasutra.com and from other sources such as Grossman (2013), focusing on post-mortems that reviewed popular video games, selecting suitable post-mortems based on shooting games. Students had to find this type of genre in the post-mortems because they were developing a shooting game in their course project.

II) Classification (Data Reduction and Coding Levels)

Students read through the selected post-mortems and identified what parts of the post-mortems should be analyzed such as what went right and what went wrong, including testing issues. It was expected that students could identify them by searching in the post-mortems some key words such as “test”, “testing”, “tester”, “evaluation”, “evaluate”, “QA”, “quality assurance”, “assess”, “requirements analysis”, “gameplay”, “playtesting”, “tried”, “played”, and “feedback”. This could facilitate the post-mortem analysis. Then, they compiled the findings from the analyzed post-mortems.

III) Interpretation (Data Display and Pattern Identification)

In the interpretation stage, students identified important issues in the post-mortems, such as QA testing, playtesting and usability testing, identifying testing best practices and issues. Then, students wrote suggestions about what else game developers could improve or do in the first place in each of the projects described in the post-mortems, in order to avoid the problems they described in the post-mortems. In addition, students prepared a summary of their findings.

IV) Conclusion Drawing (Verification and Presentation)

Students wrote a report on their post-mortem analysis. The report included a summary of the three analyzed post-mortems, a summary of the testing issues described in the post-mortems and students’ suggestions for improving those testing issues. Students were also asked to compare the results from the three analyzed post-mortems.

After students completed the assignment, they were asked to discuss their findings the next class after the assignment’s deadline. This was an interesting activity to gauge, among other game development aspects, the students’ interest and knowledge on the post-mortems and to summarize their main findings.

Results

The students chose very interesting post-mortems (a sample of them is summarized in [Table 1]). Some of them were about popular first-person shooters such as Call of Duty 4: Modern Warfare.

Post-mortem web page:	Game reviewed in the post-mortem:	Testing aspects found:	Students’ suggestions for improvement/comments:
https://www.gamasutra.com/view/news/258315/The_making_of_Call_of_Duty_4_Modern_Warfare.php	Call of Duty 4: Modern Warfare	Playtesting was done alright. However, Beta testing was shorter than planned, meaning some missions didn’t get balance testing well.	Conduct more extensive game balancing (making the game not too hard, not too easy).
https://www.gamasutra.com/view/feature/129929/postmortem_deadline_ga	Chili con Carnage	The game controls were tested very early into the game development project.	Testing sessions were conducted frequently and appropriate changes were made to achieve the final product.

mes_chili_.php			
https://www.gamasutra.com/view/feature/131569/postmortem_epic_games_unreal_.php	Unreal Tournament	Since many story aspects were told using the character skins (e.g. garment), the game didn't provide enough of them. There just very few skins available.	Clearly, the game skins problem could be identified if the game creators could run enough play testing.
https://www.gamasutra.com/view/feature/131829/postmortem_redstorms_rainbow_six.php	Redstorm's Rainbow Six	Lack of testing during the game development due to understaffed project and they did not have enough time to complete project on time.	Game developers had to test the game early in the project.

Table 1: A sample of testing aspects found by the students in the reviewed post-mortems.

After submitting his assignment, a student was very interested in reading more post-mortems, and he was asking for published articles about post-mortems. It is important to point out that students who did the post-mortem analysis were aware of the positive and negative sides of game testing when they were conducting their course projects, and the importance of testing game prototypes early and often, as they found out in some of the post-mortems.

Conclusions

This paper presented teaching and learning experiences from a game development course where students analyzed published post-mortems, focusing on extracting and analyzing development project aspects such as what went well and what went wrong in them, identifying testing issues, methodologies and best practices. The post-mortem teaching was organized in three parts. First, students were taught about the structure and importance of post-mortems, as well as testing aspects in game development. Second, students were asked in an assignment to choose and analyze three published post-mortems. Third, students explained in class their findings from the analyzed post-mortems.

The post-mortem analysis should be an integral part of game development courses. The analyzed post-mortems done in the course provided valuable information on what went right and what went wrong in past game development projects from the gaming industry. The post-mortem work as a radiography, allowing students and practitioners to see through game development projects and visualize gaming development issues and best practices, including testing aspects that happened in real-life game design and development. This allows students to read about past hands-on game testing experiences. Fortunately, many game post-mortems can be easily retrieved from printed and online sources. Some of the online post-mortems include extra information such as comments and suggestions written by readers, which can also be beneficial by students in their game development learning process. The post-mortem analysis can positively complement the student writing of student post-mortems to avoid previous mistakes done by gaming companies and to follow best practices inferred from already-published post-mortems.

Further work includes devising other pedagogical activities such as practical in-class game testing exercises following testing best practices from post-mortems, and student knowledge assessment about the best practices obtained from the post-mortems. In addition, students will be asked to write post-mortems about their own course projects. These educational activities will be based on Kolb's Experiential Learning theory (Kolb & Kolb, 2013), promoting practical and hands-on game development activities by taking into account previous knowledge, experiences and best practices explained in published post-mortems, as well as students reflecting on their own projects while they are writing their post-mortems.

References

- Dingsoyr, T. (2005). Postmortem reviews: Purpose and approaches in software engineering. *Information and Software Technology*, 47(5):293-303.
- Fullerton, T. (2018). *Game development workshop*, 4th Edition, Boca Raton, FLA: CRC Press.

- Grossman, A. (ed.) (2013). *Postmortems from game developer: Insights from the developers of Unreal Tournament, Black & White, Age of Empire, and other top-selling games*. Hoboken, NJ: Focal Press.
- Kolb, A.Y., & D.A. Kolb (2013). *The Kolb learning style inventory*. Boston, MA: Hay Resources Direct/Experience Based Learning Systems, Inc.
- Maxim, B. R., Grosky, W. I., & Baugh, J. P. (2007, October). Work in progress-introducing information technology through game design. In *2007 37th Annual Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports* (pp. T1B-1). IEEE.
- McAvoy, J. (2006). Evaluating the evaluations: Preconceptions of project post-mortems. *Electronic Journal of Information Systems Evaluation*, 9(2).
- Miles, M.B., & Huberman, A.M. (1994). *Qualitative data analysis: An expanded sourcebook*. London, UK:Sage.
- Myllyaho, M., Salo, O, Kaariainen, J., Hyysalo, J., & Koskela, J. (2004). A review of small and large post-mortem analysis methods. In *Proceedings of ICSSEA*.
- Poulsen, M. (2011). Learning by producing. In Poulsen, M. and Kober, E. (eds.). *The Game IT Handbook*. Oslo, Norway” European Commission.
- Redavid, C., & Farid, A. (2011). An Overview of Game Testing Techniques. *Technical Report*. Vasteras: Malardalen University, Sweden.
- Scacchi, W. (2015, June). Case Studies and Practices in Local Game Jam Software Development Organization: A Software Engineering Perspective. *Presented at the Foundations of Digital Games 2015 Workshop on Game Jams, Hackathons and Game Creation Events, Pacific Grove, CA*.
- Schultz, C.P., & Bryant, R.D. (2017). *Game testing: All in one*. Dulles, VA: Mercury Learning and Information.
- Tschang, F. T., (2005). Videogames as interactive experiential products and their manner of development. *International Journal of Innovation Management*, 9, 01, 2005, 103–131.
- Volk, D. (2008, November). How to embed a game engineering course into a computer science curriculum. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share* (pp. 192-195).
- Wang, A. I. (2009, August). Post-mortem analysis of student game projects in a software architecture course. In *2009 International IEEE Consumer Electronics Society's Games Innovations Conference* (pp. 78-91). IEEE.
- Washburn, M., Sathiyarayanan, P., Nagappan, M., Zimmermann, T., & Bird, C. (2016). What went right and what went wrong: an analysis of 155 postmortems from game development. In *Proceedings of the 38th International Conference on Software Engineering Companion* (pp. 280-289). ACM.
- Wawro, A. (2015). 10 seminal game postmortems every developer should read. [online]. Available: https://www.gamasutra.com/view/news/238773/10_seminal_game_postmortems_every_developer_should_read.php