Department of Computer Science,
National University of Computer and Emerging Sciences,
Islamabad

Automated Games Review Analyzer

ALI SHAHBAZ
20I-2020

DR. KHUBAIB AMJAD

Some Hits/Guidelines for MS Thesis title:

1. **The title of the thesis should be possibly specific while still describing the full range of the work.**
2. **The title should reflect the aim and approach of the work, not the results. Rarely very little about the results and conclusions are hinted at in the title.**
3. **Depending on the relevant field of study, some of the backgrounds may have to be included. If the thesis work had been based on simulation (or some other approach), would generally be included in the title.**
4. **The overall goal of a title should be clear and informative and needs to answer the following questions:**
    1. **Does the title of the thesis, seen in isolation, give a full yet concise and specific indication of the work reported?**
    2. **Would someone interested in the exact topic of your thesis work, reading this title, be inclined to read the abstract?**

Version: Proposal_Dec_2020
Prepared by: MSRC Computing

Revision History

Date	Version	Description	Supervisor Signatures
3/08/21	1.0	MS thesis proposal	Dr. Khubaib Amjad

[Note: The template is provided for use as MS Thesis Proposal for Computer Science Students of National University of Computer and Emerging Sciences, Islamabad. ***Text enclosed in square brackets and displayed in blue color is included to guide the author and should be deleted before submitting the document.****]*

Formatting guidelines:

This proposal document should be prepared using "Times New Roman" font. The font size should be as follows: chapter headings 14 bold, section headings 12 bold, and the body text should be the 12-size font. Figures and Tables caption should be in 10 font size (Times New Roman). The document is to be prepared with a line spacing of 1 and a Justified text option (to evenly distribute the text between left/right margin).

Table of Contents

MS Thesis Proposal

1. Introduction

1.1 Introduction and Background

The game industry has become one of the most profitable markets in the entertainment industry. Knowing what customers and users think and how they feel about a game is a central piece to drive the decision-making process, of any game developer or game studio, towards the user satisfaction. The steadily increasing popularity of computer games has led to the rise of a multi-billion-dollar industry.

Due to the scale of the computer game industry, developing a successful game is challenging. In addition, prior studies show that gamers are extremely hard to please, making the quality of games an important issue. Most online game stores allow users to review a game that they bought. Such reviews can make or break a game, as other potential buyers often base their purchasing decisions on the reviews of a game.

Studying gamer reviews help developers better understand the concerns and further improve the user-perceived quality. Bugs that persist into releases of video games can have negative impacts on both developers and users, but particular aspects of testing in game development can lead to difficulties in effectively catching these missed bugs. It has become common practice for developers to apply updates to games in order to fix missed bugs.

In order to cope up with these problems we are generating a bug report from user reviews on steam platform which help the developers and team to understand which are the areas which need to be fixed.

1.2 Motivation

Other people's opinions are the most important piece of information for most of us. Social networking sites such as Facebook allow users to post their comments or opinions about any issues and topics. Here, we will see the importance of reviews shared by the peoples. Many researched had been done by which we can identify the negative and positive reviews of the users. But very little research is done on this domain by which we can identify the bugs which are mentioned by the user in the reviews. So our motivation is to train a model based on the user reviews and then from negative reviews on steam platform we will generate bug report.

2. Literature Review

Some bugs reappear even after an update attempts to fix them. An approach is developed a taxonomy for bug types in games that is based on prior work. They examine 12,122 bug fixes from 723 updates for 30 popular games on the Steam platform. Then label the bug fixes included in these updates to identify the frequency of these different bug types, the rate at which bug types recur over multiple updates, and which bug types are treated as more severe. They find that challenges in testing, code quality, and bug reproduction have a close association with bug persistence. These findings help developers identify which aspects of game development could benefit from greater attention in order to prevent bugs. [1]

Video games are complex, emergent systems that are difficult to design and test. This difficulty invariably leads to failures being present in the game, negatively impacting the play experience of some. A taxonomy of possible failures, divided into temporal and non-temporal failures is presented. The taxonomy can guide the thinking of designers and testers alike, helping them expose bugs in the game. This will lead to games being better tested and designed, with fewer failures when released.[2] Most online game stores allow users to review a game that they bought. Such reviews can make or break a game as other potential buyers often base their purchasing decisions on the reviews of a game. Empirical study of the reviews of 6224 games on the Steam platform, one of the most popular digital game delivery platforms, to better understand if game reviews share similar characteristics with mobile app reviews, and thereby understand whether the conclusions and tools from mobile app review studies can be leveraged by game developers.[3]. Unity Linter, a static analysis tool that supports Unity video game developers to detect seven types of bad smells they have identified as relevant in video game development. Results of our empirical investigation indicate that developers well received performance- and behavior-related issues, while some maintainability issues are more controversial. Unity Linter is, I general, accurate enough in detecting smells (86%-100% precision and 50%-100% recall), and our study shows that the studied smell types occur in 39%-97% of the analyzed projects. [4]

3. Problem Statement

The massive increase of user-generated content on the web and other platforms that provides opinions of people on different subjects. Sentiment analysis is the computational study of analyzing people's feelings and opinions for an entity. The field of sentiment analysis has been the topic of extensive research in the past decades. And user reviews are the best way to get to know about what customer thinks. By this technique we were able to identify the negative and positive reviews but very few research is done on bugs mentioned by the users. We are training a model using deep learning technique which will first find out the negative reviews by the user then from them it generates the bug report which will fall in specific category. And this model will be trained on different gaming categories like action, adventure, puzzle etc.

4. References

[1] A. Truelove, E. Santana de Almeida, and I. Ahmed, “We’ll Fix It in Post: What Do Bug Fixes in Video Game Update Notes Tell Us?,” pp. 736–747, 2021, doi: 10.1109/icse43902.2021.00073.

[2] C. Lewis, J. Whitehead, and N. Wardrip-Fruin, “What went wrong: A taxonomy of video game bugs,” *FDG 2010 - Proc. 5th Int. Conf. Found. Digit. Games*, pp. 108–115, 2010, doi: 10.1145/1822348.1822363.

[3] D. Lin, C. P. Bezemer, Y. Zou, and A. E. Hassan, *An empirical study of game reviews on the Steam platform*, vol. 24, no. 1. Empirical Software Engineering, 2019.

[4] A. Borrelli, V. Nardone, G. A. Di Lucca, G. Canfora, and M. Di Penta, “Detecting Video Game-Specific Bad Smells in Unity Projects,” *Proc. - 2020 IEEE/ACM 17th Int. Conf. Min. Softw. Repos. MSR 2020*, pp. 198–208, 2020, doi: 10.1145/3379597.3387454.