A fault-based testing approach for VR applications

[£]Stevão A. Andrade, [ŧ]Fatima L. S. Nunes, [£]Márcio E. Delamaro
[£]Instituto de Ciências Matemáticas e de Computação, [ŧ]Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
stevao@icmc.usp.br, fatima.nunes@usp.br, delamaro@icmc.usp.br

Abstract: Technologies such as Virtual Reality (VR) have emerged, allowing the development of three-dimensional environments with real-time interaction. VR systems lack similarities between traditional programs, which makes it ineffective to apply traditional software testing criteria in then. Considering this motivation and the acceptance between researchers and engineers that quality is an essential factor in software development, in this paper we examine software testing practices available for the VR domain and present the possibilities for improvement to provide an automated software testing approach that can contribute to the quality assurance of VR applications.
Key words: *software testing, fault-based testing, metamorphic testing, validation, virtual reality.*

1. Introduction

Virtual Reality (VR) uses computers to create 3D environments, enabling navigation and real-time interaction. The user can navigate through the environment to explore possible features present in a 3D representation, such as training simulations.

The visual aspects are often the most related to the creation of VR environments because the vision is the main form of perception for the acquisition of information for most of the users. However, VR is not limited to what we can see. Other vital components of human perception play an essential role in shaping a virtual environment, such as sounds, and perceptions gained through touch [1].

Despite the huge benefits of adopting VR to develop applications in various areas, this poses new challenges for software quality assurance activities [2]. For example, VR developed software presents unconventional software structures, such as scene graphs, which may represent new sources of faults. These new challenges motivated the development of some approaches that aim to contribute to the quality assurance process of software in the context of VR.

Previous software testing approaches specific to VR applications fail because of a lack of generalization and the difficulty of systematizing how a testing procedure can be measured. This problem is described in the literature as the "oracle problem," and occurs in cases in which traditional approaches of measuring the execution of a test case are impractical or are not useful to judge the correctness of outputs generated from the input domain data [3, 4].

As previously discussed, it is hard to infer a test oracle for VR applications, needing, in general, the figure of a human as an oracle. This problem directly compromises the feasibility of applying automated testing techniques [4]. One possible solution to this problem is the use of metamorphic testing. Metamorphic testing aims to verify whether a given program behaves according to a set of metamorphic relations. A metamorphic relationship specifies how a particular change in a program´s input data can modify its behavior and hence its output. Therefore, it is possible to infer the correction of the program even though there is no explicit presence of a test oracle [5].

Since, in general, fault-based test approaches, which are the focus of this work, heavily rely on the figure of a test oracle, the hypothesis raised here is that:

Hypothesis – *It is possible to propose a software testing approach based on fault-based techniques, applicable to the VR domain, by combining the mutation testing criterion with the metamorphic testing technique*.

We considered that this hypothesis is feasible as various studies discuss the application of the fault-based test in several contexts [6], as well as studies that successfully present the use of metamorphic testing in multiple domains [5].

We assume that if both approaches are combined, we can propose a systematized approach to be applied to the VR domain to conduct the software testing activity, as well as to ensure a quality criterion that measures its application.

The idea is to leverage the benefits of a fault-based test approach that is well-recognized for its fault reveal ability and to use concepts extracted from metamorphic testing as a way to mitigate the oracle problem [3], which is one of the main challenges for the application of an automated testing approach in VR applications.

Therefore, the main goals of this work are:

Objective $_1$ – To define a fault taxonomy specific for the VR domain;

Objective $_2$ – To define metamorphic relationships that comply with the fault taxonomy;

Objective $_3$ – To define mutation operators that comply with the fault taxonomy;

Objective $_4$ – To propose a fault-based testing approach for VR domain.

In summary, the idea is to generate mutants that model possible faults in VR applications and use metamorphic relationships to "kill" such mutants in order

to prove that the modeled faults are not present in the original program. The goal of using mutants together with metamorphic testing is because metamorphic testing is considered a black box testing approach, which does not give much information regarding the spot location of the revealed fault; otherwise, VR applications heavily depend on visual aspects. Therefore, mutants can guide the tester more objectively to identify the point of the program that contains the revealed fault.

2. Methodology

According to the research gap characterized in the previous section, this project aims to study and propose mechanisms to enable the application of a fault-based testing approach to the context of VR applications.

Focusing on answering the presented research question, the subsections below present the main activities to be performed during the development of the work.

2.1 Definition of a VR fault taxonomy

According to the collection of norms defined by IEEE, a taxonomy is a scheme that divides a set of knowledge and defines relationships between the parts of that set. It is used to classify and understand the body of knowledge about a particular topic.

The focus of this activity is to investigate the proneness of specific types of faults during the development of VR applications, considering specific VR properties, harmful implementation mechanisms, and recurrent fault scenarios. Based on these observations, it is expected to define a defect taxonomy that should be able to categorize the variety of faults that can occur in VR software.

2.2 Definition of metamorphic relations aligned with the fault taxonomy

The effectiveness of metamorphic tests is highly related to the metamorphic relationships that are used in the process. Therefore, building effective metamorphic relationships is a critical step in the successful application of metamorphic testing. Thus, it is advisable to use a variety of metamorphic relationships to ensure that software has been evaluated adequately.

Defining good metamorphic relationships requires knowledge of the problem domain. Therefore, understanding typical defects, as well as frequent failures that occur in the development of VR applications, are fundamental steps for the success of this work step, which instantiates such information through constraints that an application must respect during its execution.

2.3 A fault-based testing approach to VR applications

Fault-based approaches have been proved in experimental evaluations as a robust test selection criterion [7, 8]. We intend to propose a fault-based approach for VR software. To do so, the VR fault taxonomy should guide the development of mutation operators that will support the application of the test criteria.

The metamorphic relations defined in the previous step should guarantee the mechanisms to evaluate the mutants that represent the faults, which are commonly made during the software development process. Those pieces of information would allow us to define an adequacy metric for the modeled faults concerning the test cases developed during the test activity.

2.4 Automation of the proposed test approach

Testing techniques heavily rely on support and automation to enable their application to real projects. Without this, testing tends to be a costly and error-prone activity [9].

This activity enables the application of the content developed during the previous steps.

3. Results

In this section, we detail the main results achieved so far. First, we present the general results obtained from an exploratory study on the application of metamorphic tests. Afterwards, we discuss the feasibility and necessity of applying automated tests for the VR domain. Thus, the remaining section responds to the previously defined research question.

3.1 Analyzing the application of metamorphic relations in similar contexts

We have observed that many works in the literature present solutions that use metamorphic testing in several domains. Therefore, in this study, we investigate the existing challenges of using existing metamorphic relationships for new problems. The aim is to verify the ability to adapt metamorphic relationships since the greatest challenge of success in metamorphic testing applications lies in the appropriate definition of metamorphic relationships. Moreover, this is a task that requires the expertise of a domain expert.

We successfully adapted, without making any major changes, metamorphic relationships to similar problems. The results of the experiments indicated good results in terms of the ability to reveal faults. These results indicate the possibility of conducting similar work for the VR domain.

The results of the experiments indicated good results in terms of the ability to reveal faults. These results show the possibility of conducting similar work for the VR domain [10].

3.2 Software testing practice in the VR domain

We discussed the main challenges of software testing practice in the VR domain [11]. Some of the critical issues related to the quality of these systems were pointed out and possible solutions were also discussed that could be used and adapted to deal with such issues.

The study was guided by 3 research questions, whose objectives were: to understand the state of the practice of software testing in the context of VR programs (*RQ_1*), to measure metrics and quality attributes in VR software (*RQ_2*), and finally to evaluate fault proneness in the software analyzed (*RQ_3*).

Results showed that most of the open source VR projects fail to present quality principles, such as the presence of test cases in their source code. This problem extends to the quality of the code produced as untested code tends to present a higher amount of code characteristic that may indicate a code smell. Similarly, we also note that untested code snippets tend to be more prone to failures than code snippets that have been properly tested.

3.3 Survey with interest groups

Software development is an extremely complex business. To be successful, software teams also need to consider aspects such as user satisfaction, as well as the functionality of the software itself. User satisfaction can only be guaranteed when the software development team implements an entire user-centric development process. [12].

Similarly, we believe that the software testing phase, which also makes up one of the development phases, should also have this kind of concern. Therefore, it is necessary to understand the point of view of researchers, developers, designers, and users in order to target software testing activity that address stakeholders' major issues, thus prioritizing verification for failures that should have a greater impact on VR applications.

To do so, we asked stakeholders´ opinions through a survey, in which the participant answered questions about what types of failures in VR applications contribute to a poor experience. The purpose of this study is to gain an insight into stakeholders (e.g., which types of failures are most critical, which are less relevant, how much each affects the quality of the end product, etc.) and to investigate the knowledge of the groups of interest regarding specific types of failures in VR applications.

4. Discussion

After analyzing the results obtained from the study described in Subsection 3.1, which sought to understand the effectiveness of applying metamorphic relationships already existing in the literature to similar problems, it was decided, based on the results and the bibliographic survey conducted during the development of this work, to perform a similar study applied to the VR domain.

The results presented by Chan et. Al. in [13] and by Donaldson et. Al. [14] are under analysis in order to verify the applicability of the proposed metamorphic relations to the VR domain. To achieve this, an experiment that will use the set of applications adopted during the development of the activity described in Subsection 3.2 is currently being prepared.

The studies cited above discuss, respectively, the application of metamorphic testing to the domain of programs that generate meshes and shader compilers. Due to the similarity of characteristics between such applications and VR applications, we believe that we can explore the results of these studies so as to propose similar metamorphic relations for the VR domain. We intend to use the results of this activity to corroborate the expected results in the development of the activities described in Subsection 2.1.

The results of the work described in Subsection 3.3 discuss the main challenges related to using software testing practice in the VR domain. A collection of 119 VR projects, available in open source projects and manually analyzed, was cataloged to understand the state of the practice concerning the application of software testing techniques. We observed that most of the projects do not maintain software testing artifacts.

The lack of concern regarding software testing implies a high incidence of code smells in the projects analyzed. The presence of code smells points out that about 12% of the analyzed VR classes are fault-prone, revealing a significant risk to the success and maintenance of the projects. The distribution of these classes was also evaluated when observed concerning the size of the projects analyzed. It was observed that the larger a project becomes, there is a higher incidence of fault-prone classes.

The preliminary evaluation of the results of the survey being conducted shows a high recognition of the need for software testing practice for VR applications (up to 80% of the respondents), but a low observation, from the point of view of the roles involved, regarding which approaches and techniques should be applied during the VR application development process. The results also indicate a low concern of the respondents regarding the need to observe quality aspects related to the network, audio and network. In general, interest groups tend to attach more importance to visual aspects such as physics and design issues. These results serve as a motivation for the development of the activity discussed in Subsection 2.1, as they show aspects considered fundamental by the roles involved in the process.

One last point that can be noted is the poor observation of the respondents regarding various quality aspects that should be guaranteed in a VR application. Based on the set of major flaws cataloged for 3D games [15], which also fall under VR applications, we observed that respondents are not aware of most of them. This

result points to a scenario that may indicate that those involved are not aware of the risk of the presence of this type of failure.

5. Conclusion

The differences between general purpose software and VR applications, such as the lack of complex data structures, conditionals, and loops makes the application of traditional software testing methods inefficient. This paper discussed the major challenges existing in order to apply software testing methods for VR applications and recognized some limitations. Furthermore, we presented a proposal that addresses such problems by combining fault-based testing and metamorphic testing; we also outline the focus of our research and present the main objectives of our work and some preliminary results.

Regarding the preliminary results, we focused on investigating the possibility of applying existing knowledge developed in similar areas that could be used in the VR domain. Furthermore, we discussed the main challenges related to using software testing practices in the VR domain by analyzing open-source VR projects. We also started collecting information regarding major issues that impacts experience in VR applications. In order to do so, we started interviewing principal roles (researchers, developers, designers, professors, students, and users) involved in the development process of VR applications. By understanding their point of view, we intend to deliver a less biased software testing approach. As future work, we are consolidating the results of Objectives 1 and 2 in order to address the major goal of the project that consists of proposing a software testing approach specific to VR domain considering the details and specificities related to it.

Acknowledgements

Stevão A. Andrade research was funded by FAPESP (São Paulo Research Foundation), process number 2017/19492-1. This study was also financed in part by FAPESP (São Paulo Research Foundation) process number 2019/06937-0. The authors are grateful to Brazilian National Council of Scientific and Technological Development (CNPq) for assistance (process 308615/2018-2), and National Institute of Science and Technology – Medicine Assisted by Scientific Computing - INCT-MACC (Process 157535/2017-7).

Bibliography

[1] LaValle, S. M. (2019) Virtual Reality. Cambridge University Press.

[2] Santos, A. C. C.; Delamaro, M. E.; Nunes, F. L. S. (2013) The relationship between requirements engineering and virtual reality systems: A systematic literature review. XV Symposium on Virtual and Augmented Reality: 53-62. IEEE. DOI: 10.1109/SVR.2013.52

[3] Rapps, S.; Weyuker, E. J. (1985) Selecting software test data using data flow information. IEEE Transactions on Software Engineering, volume (SE-11, Issue 4): 367-375. DOI: 10.1109/TSE.1985.232226

[4] Barr, E. T.; Harman, M.; McMinn, P.; Shahbaz, M.; Yoo, S. (2015) The oracle problem in software testing: A survey. IEEE Transactions on Software Engineering, volume (41, Issue 5): 507-525. DOI: 10.1109/TSE.2014.2372785.

[5] Chen, T. Y.; Kuo, F.C.; Liu, H.; Poon, P. L.; Towey, D.; Tse, T. H.; Zhou, Z. Q. (2018) Metamorphic testing: A review of challenges and opportunities. ACM Computing Surveys, volume (51, Issue 1): 4.1-4.27. DOI: 10.1145/3143561

[6] Harman, M.; McMinn, P.; Shahbaz, M.; Yoo, S. (2013) A Comprehensive Survey of Trends in Oracles for Software Testing. University of Sheffield Tech. Rep. CS-13-01: 1-32. DOI: 10.1.1.371.9004

[7] Smith, B. H.; Williams, L. (2009) Should software testers use mutation analysis to augment a test set?. Journal of Systems and Software, volume (82 Issue 11): 1819-1832. DOI: 10.1016/j.jss.2009.06.031

[8] Chekam, T. T.; Papadakis M.; Le Traon, Y.; Harman M. (2017) An Empirical Study on Mutation, Statement and Branch Coverage Fault Revelation That Avoids the Unreliable Clean Program Assumption. 2017 IEEE/ACM 39th International Conference on Software Engineering: 597-608. IEEE. DOI: 10.1109/ICSE.2017.61

[9] Lu L. (2001) Software testing techniques. Carnegie Mellon University Tech. Rep. 17-939A: 1-20. http://www.cs.cmu.edu/~luluo/Courses/17939Report.pdf Accessed on 21/11/2019.

[10] Andrade, S. A; Santos I; Junior, C. B; Júnior, M; Souza, S. R.S; Delamaro, M. E. (2019) On Applying Metamorphic Testing: An Empirical Study on Academic Search Engines. 2019 IEEE/ACM 4th International Workshop on Metamorphic Testing: 9-16 DOI: 10.1109/MET.2019.00010

[11] Andrade, S. A; Nunes, F. L. S; Delamaro, M. E. (2019) Towards the Systematic Testing of Virtual Reality Programs. XXI Symposium on Virtual and Augmented Reality: 180-189 DOI: 10.1109/SVR.2019.00044

[12] Scholtz, J.; Morse, E. (2003) Using consumer demands to bridge the gap between software engineering and usability engineering. Software Process: Improvement and Practice, volume (8, Issue 2): 89-98. DOI: 10.1002/spip.172.

[13] Chan, WK.; Ho, J. CF.; Tse, TH. (2007) Piping classification to metamorphic testing: An empirical study towards better effectiveness for the identification of failures in mesh simplification programs. 31st Annual International Computer Software and Applications Conference, volume (1): 397-404. IEEE. DOI: 10.1109/COMPSAC.2007.167

[14] Donaldson, A. F.; Evrard, H.; Lascu, A.; Thomson, P. (2017) Automated testing of graphics shader compilers. Proceedings of the ACM on Programming Languages, volume (1): 93.1-93.29. ACM. DOI: 10.1145/3133917

[15] Levy, L.; Novak, J. (2009) Game development essentials: Game QA & testing. Cengage Learning