A fault-based testing approach to VR applications

Stevão A. Andrade
Universidade de São Paulo - ICMC
São Carlos, Brasil
stevao@icmc.usp.br

Fatima L. S. Nunes
Universidade de São Paulo - EACH
São Paulo, Brasil
fatima.nunes@usp.br

Márcio E. Delamaro
Universidade de São Paulo - ICMC
São Carlos, Brasil
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 on 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.**

***Index Terms*—software testing, fault-based testing, metamorphic testing, validation, virtual reality**

I. 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 great benefits of adopting VR for the development of applications in several areas, this poses new challenges for software quality assurance activities [2]. For example, VR developed software present 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.

II. Motivation

Software testing and VR are two independent research areas, each one with its characteristics and natures. Software testing is a subarea of Software Engineering that seeks to define mechanisms and processes that aim to structure and improve the quality assurance in the software development cycle. VR enables interaction with 3D environments, allowing the creation and simulations of risk activities, to perform real tasks remotely, bringing as one of its main advantages the fact that a simulated world does not necessarily have to obey the natural laws of the real world.

Stevão Andrade research was funded by FAPESP (São Paulo Research Foundation), process number 2017/19492-1.

As a result of the increased use of VR applications in a wide range of contexts, the demand for mechanisms to ensure that the developed products conform to the specifications that they were initially designed also increased. This problem, together with the motivation previously presented, gives rise to the following research question:

RQ – How a fault-based test approach can contribute to improving the software production process in the context of RV applications?

The above research question is concerned with the definition of mechanisms that are capable of providing a systematically based software testing approach and raise the following problem: *how mutation testing technique can propose mechanisms that allow its application on VR domain?*

III. Related Work

A feature evidenced in the development of VR applications is that, in general, the software testing activity is usually performed manually and is mostly conducted only after the end of the development cycle [3]. In order to tackle this problem, two papers in the literature present possible solutions for the systematization of the testing activity in VR domain through the definition of specific software testing criteria.

Bezerra et al. [4] observed the scene graph as an abstraction of a VR application, and used it to derive test requirements. The idea is to exploit an approach similar to structural testing, which uses the control flow graph to derive the test requirements. They established a set of test criteria based on the scene graph in order to ensure that the test set created can cover specific behaviors of VR applications such as rotations, scales, transformations, and other operations represented by the scene graph.

Posteriorly Souza et al. [5] developed a *Virtual Reality-Requirements Specification and Testing* (VR-ReST) approach, which aimed to design requirements specification of VR application. To support the specification of the generated requirements, a domain-specific language, called *Behavior Language Requirement Specification* (BeLaRS) was proposed. The aim was to allow the requirements specification of VR application based on domain-case descriptions and use a requirements graph to infer test requirements and explore the generation of test data.

Despite the efforts, the works described stumbling in the challenge of measure the behavior of a test case since it is hard to measure the outputs generated from the execution of the test cases and still require the intervention of the human figure.

IV. Proposed Solution

As previously discussed, it is hard to infer a test oracle for VR applications, needing, in general, the figure of a human as a oracle. This problem directly compromises the feasibility of applying automated testing techniques [6]. 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 input data can modify its behavior and hence its output. Therefore, it is possible to infer about the correction of the program even though there is no explicit presence of a test oracle [7].

Since, in general, fault-based test approaches, that are the focus of this work, heavily relies 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 technique, applicable to the VR domain, by combining the mutation testing criterion with the metamorphic testing technique.*

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 [8], 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:

O_1 – To define a fault taxonomy specific for the VR domain;

O_2 – To define metamorphic relationships that complies with the fault taxonomy;

O_3 – To define mutation operators that complies with the fault taxonomy;

O_4 – To propose a fault-based testing approach to 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 in a more objectively way to identify the point of the program that contains the revealed fault.

V. Preliminary Results

A. Analyzing the application of metamorphic relations in similar contexts

In this study, we investigate the existing challenges of using existing metamorphic relationships for new problems. The idea is to verify the ability to adapt metamorphic relationships since the greatest challenge of success in the application of metamorphic testing lies in the appropriate definition of metamorphic relationships since this is a task that requires the expertise of a domain expert.

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.

B. Software testing practice in the VR domain

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 objective was: 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 collection of 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 since untested code tends to present a higher amount of code characteristic that may indicate a code smells. Similarly, we also note that untested code snippets tend to be more prone to failures than code snippets that have been properly tested.

VI. Conclusion

The differences between general purpose software and VR applications, such as the lack of complex data structures, conditionals, and loop makes inefficient the application of traditional software testing methods. This paper examined the existing testing methods for VR applications and recognized some limitations. Further, we presented a proposal that addresses such problems by combining fault-based testing and metamorphic testing; we also delimited to focus of our research as well as presented the major objectives of our work and some preliminary results.

References

[1] S. M. LaValle, *Virtual Reality*. Cambridge University Press, 2017.

[2] A. C. C. Santos, M. E. Delamaro, and F. L. S. Nunes, "The relationship between requirements engineering and virtual reality systems: A systematic literature review," in *2013 XV Symposium on Virtual and Augmented Reality*, 2013.

[3] M. H. V. K. J. Schlueter, H. Baiotto and G. Evans, "Best practices for cross-platform virtual reality development," in *Proceedings of SPIE Defense + Security*, 2017.

[4] A. Bezerra, M. Delamaro, and F. Nunes, "Definition of test criteria based on the scene graph for vr applications," in *Virtual Reality (SVR), 2011 XIII Symposium on*, 2011.

[5] A. C. C. Souza, F. L. Nunes, and M. E. Delamaro, "An automated functional testing approach for virtual reality applications," *Software Testing, Verification and Reliability*, 2018.

[6] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, "The oracle problem in software testing: A survey," *IEEE Transactions on Software Engineering*, 2015.

[7] T. Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, T. H. Tse, and Z. Q. Zhou, "Metamorphic testing: A review of challenges and opportunities," *ACM Comput. Surv.*, 2018.

[8] S. Rapps and E. J. Weyuker, "Selecting software test data using data flow information," *IEEE Trans. Softw. Eng.*, 1985.