International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Overcoming VR Test Automation Challenges and Best Practices

Komal Jasani

QA Engineering Lead
Union City, California USA
komal_jasani@yahoo.com

Abstract
Virtual Reality (VR) applications today are becoming diverse and relatively large and hence need viable testing strategies and approaches. The reason is that traditional manual testing practices are slow, static, and non-interactive, failing to address the challenges associated with testing VR systems. Nevertheless, the automation of tests in the VR environment is very challenging since HCI complexity, motion tracking, and hardware variations exist, and there is little provision for frameworks. The general problem in this paper is to outline the main difficulties of VR test automation and suggest their solutions. They include VRTest and Youkai, as well as the approaches that can be used in them, like AI-test automation, AV testing, and cross-platform solutions for testing. Moreover, it describes the importance of machine learning within testing, particularly regarding test cases and performance evaluation benchmarks. Considering these issues and their solutions, the following points can be concluded: By integrating these points of the best practices, developers contribute to increasing the stability and efficiency of the testing of VR software and, therefore, the overall quality of the application the consumer uses.

INTRODUCTION

Over recent years, the concept of virtual reality (VR) has successfully advanced and grown, and it is becoming integrated into various sectors, starting from the entertainment sector but not being limited to this segment only, including games, healthcare, learning, and even shopping. As the application of VR has developed into providing multi-faceted services, it is necessary to pay attention to the application's quality, speed, and stability. Compared to the conventional Real-Time software testing methodologies, methods targeting the 2D interfaces are inadequate, mainly when applied to 3D and real-time interactive VR applications. This has led to the development of the necessity for testing in Virtual Reality. However, VR test automation is different from the actual software test because it comes with challenges. Some of these challenges are hardware disparities between motions tracking of other VR headsets, user input emulation, and testing of 3D environment, rendering, and realistic physics-based behaviors. Most importantly, performance testing in VR is exercised since factors such as latency resulting in frame rate drop are significant determinants of user experience and can even cause motion sickness. A lack of assessment standards is another challenge that emerged due to the continuously growing demand for automated VR testing.

This paper identifies the main issues related to test automation in a virtual reality environment and provides strategies for avoiding them or mitigating adverse effects. By addressing these issues, this research sought practical recommendations to help developers and testers improve the quality, scalability, and reliability of VR applications.

OVERVIEW OF THE RAPID GROWTH OF VR APPLICATIONS ACROSS INDUSTRIES

VR technology is one of the most common technologies, and it has grown fast in many sectors due to hardware and software development. Although VR was once restricted to gaming, entertainment, and entertainment applications, Rzig et al. (2022) noted that it is applied to healthcare, education, e-commerce, training simulations, and business solutions. For these reasons, gaming has been considered the leading industry in implementing VR, as the latest headsets ensure a more refined immersion. The primary use of automated testing is to guarantee that it runs smoothly, is responsive in real time, and has no bugs that are inclined to decrease user engagement (Rzig et al., 2023).

Virtual reality has been applied in various fields, including medical practice, training, therapy, and rehabilitation. The technology of 360-degree video lets surgeons and other healthcare workers practice intricate operations; it also enhances the capability to decide in a specific period. Technological advancements in therapy have also demonstrated promising progress within the use of virtual reality therapy in treating mental disorders, particularly in exposure therapy for PTSD and anxiety disorders. Testing these applications demands proper validation that ensures their accuracy and reliability, which is why automation is relevant during development (ElhaqRzig et al., 2022). Another aspect of life that has been advanced by VR technology is education, where En professors and students have simulated environments with virtual classes and narrowed scientific simulations. The corporate training programs follow the implementation of VR to provide practical experience in aviation and manufacturing fields. Testing frameworks help to assess usability, the real-time integration of training modules, and their compatibility with other devices, VR/AR (Rzig et al., 2023).

The retail industry has been incorporated into the use of VR for shopping extravagances where customers can shop and feel products in a virtual environment. Technology merges with artificial intelligence to increase personality along with increasing the interface and customer-orientated features of shopping online. Nevertheless, to make the shopping experience seamless, there are several areas, such as speedy testing of 3D rendering, checking the system’s performance at high loads, and interactive features mentioned by ElhaqRzig and Kacem (2022)

Due to the remote work, people had to adopt VR Meetings and VR collaboration tools for communication. However, continuous connectivity, device compatibility, and real-time processing are some issues that require more testing methodologies for the system's reliability (Rzig et al., 2023). This escalating utilization of VR can only mean that testing must be done thoroughly to ensure the applications' overall performance, usability, and scalability. As immersion becomes more advanced and essential in the market, mitigating difficulties in the testing automation of virtual reality applications will be crucial for further developing diverse specialization fields (Rzig et al., 2022).

CHALLENGES IN VR TEST AUTOMATION

The topic under consideration is challenging to test, especially in the field of Virtual Reality, because of the advanced hardware, software, and user involvement. In the case of VR applications, the interactions are usually synchronous and require various elements to work in parallel, which makes the automation process much more complex than in other cases. One can be attributed to the hardware variability and the platform itself, as VR applications are developed to run on VR headsets, controllers, and tracking systems. It has factors like specification, operating systems, and differences in processing power that complicate the compatibility of a unified interface. It is worst when performance differences are obtained based on high and low-end hardware since the performance increases stratagem that has to be put in place when testing varies depending on the basal hardware state (Rzig et al., 2022; García-Díaz et al., 2022). Real-time users and motion tracking are other factors that increase the difficulty of creating Test Automated in VR environments. Unlike other applications, in VR, it is necessary to mention that lower input is movement-based and gesture-based, and it is not the same for all users. Considering the head tracking, positioning of the screen, and movements of the body, these are aspects that are not easily simulated when testing for bots. It is important to model humans with fuzzy parameters, ensuring an accurate behavior simulation by applying stochastic approaches. It is crucial to authenticate those interactions accurately to guarantee that the consumers will have enthralling and timely experiences (Cabral et al., 2020; Kourtesis et al., 2021).

According to Scherer and colleagues, some of the most challenging issues in VR testing include validating the 3D environments and rendering consistency. Light and texture variations, alongside the difference in physical laws at work when the application is run on different screen types, are issues that cannot be resolved via automated testing that current tools for the task can handle. It is significant to note that while traditional 2D programs are immobile, VR display is dynamic, and the objects in a display will respond based on the actions of a user in real-time. Assessment of graphical rendering must be automated, and it should be able to identify differences in images qualified by diverse calls with different screen sizes. This is made more complicated because the behavior of dynamic objects, simulation of physics, lighting effects, and environmental interactions also need to be verified where applicable across different scenes (Thorn et al., 2016; Wassermann & Bachmann, 2019).

For VR applications, it is essential to implement latency and performance as a flashpoint to promptly determine the stability and responsiveness of the application since any time delay in rendering the displays or getting the input response may result in poor experiences by the user. Latency may result in motion sickness; thus, great importance is placed on measuring response times. However, one must remember that in the case of VR, performance has to be constantly measured in terms of frame rate, rendering speed, or real-time response. It is important to perform automated stress tests to understand how VR applications behave when system loads are intensive so that they can be validated across the hardware platforms (Liu et al., 2021; Andrade et al., 2020).

A key challenge in testing VR applications is a lack of framework that may be used for testing. While software testing is relatively standardized, and standard tools can be used when developing software programs, VR has not been thoroughly explored, and few testing tools that fit the need are available on the market. Most developers use personas and scripts for testing, which mainly involves manual intervention and takes time. It is revealed that existing tools for software testing are not sophisticated enough to be easily integrated into IMMERSE environments of VR applications. The creation of automated procedures tailored to VR is still a topic of current research interest due to the increased demand for efficient quality control and decreased testing difficulties in such environments (Andrade, Economidou, Koutouzis, Patrinos, &Spanoudakis, 2020; Rzig, Zhang, & Helmer, 2022; García-Díaz et al., 2022).

BEST PRACTICES FOR VR TEST AUTOMATION

AI and machine learning changes have enhanced the ability to generate test cases and improved the automation coverage of VR tests. That is why it is possible to use AI-driven techniques to automate the validation and identify various rendering and interaction issues. Hence, CV technology is employed to identify graphical defects, monitor user interactions, and verify holocentness in several VR applications. Another valuable improvement in automation performance is provided by predictive analytics, which establishes probable problems in the interaction with the user, ensuring that efficiency and effectiveness are sustained by preventing such detriments from occurring (Liu et al., 2021; Andrade et al., 2020). Performing testing across various platforms is still relevant regarding VR application development because of the excellent platform variation. In order to support a unified gameplay experience regardless of the users' choice of headset, controllers, or operating systems, proper means of automation must be in place. Several factors have made cloud-based VR testing the best choice for scalability; this makes it possible to validate on as many devices as possible from a remote location without necessarily having to establish a physical lab with multiple machines and devices. The cross-platform automated techniques help conduct assessments on various virtual platforms and isolate platform-dependent problems (Rzig et al., 2022; García-Díaz et al., 2022).

Performance and latency testing are critical in VR because delays and latency may aggravate and worsen the user's comfort when experiencing VR. Programs and software such as Fps meters enable the calculation of frames per second, latency meters inform on system latency, and resource meters measure resource usage to help developers maintain real-time performance. Such aspects include random analysis that boosts the opportunity to identify the slow-performing programs, allowing for a solution that enhances application speed and response time. Stress testing and load simulation help verify that VR applications operating under heavy computational loads will not crash or significantly slow down due to usage (Wassermann & Bachmann, 2019; Thorn et al., 2016).

Therefore, the possibility of automating the user's interaction and gesture testing of the user, in order to ensure as natural and intuitive an experience as possible in VR. Motion imitation processes mimic some concrete user actions to evaluate an application's behavior in the case of specific movements. Incorporation and analysis of gesture recognition make the input validity much more reliable by requiring the validates' virtual hands, controller movements, and body postures to demonstrate the expected results. Haptic feedback testing strategies increase the overall quality of the haptic feedback experience by identifying and validating feedback when it comes to accuracy and responsiveness, as well as improving the degree of immersion within the VR-interaction context (Cabral et al., 2020; Kourtesis et al., 2021). Available VR test cases apply structural solutions approaches to improve QA automation processes. Leading tools like VRTest, Youkai, and Unity Test Framework provide locomotive testing for immersive applications. These tools should be used in software development as they provide results that can validate a stage in development. However, existing frameworks present some constraints that make it necessary to keep investigating and enhancing the topic encompassing AI automation, cross-platform integration, and improving real-time analytics (Andrade et al., 2020; Rzig et al., 2022; García-Díaz et al., 2022).

Therefore, there is the possibility of automating the user's interaction and gesture testing of the user in order to ensure as natural and intuitive an experience as possible in VR. Motion imitation processes mimic some concrete user actions to evaluate an application's behavior in the case of specific movements. Incorporation and analysis of gesture recognition make the input validity much more reliable by requiring the validates' virtual hands, controller movements, and body postures to demonstrate the expected results. Haptic feedback testing strategies increase the overall quality of the haptic feedback experience by identifying and validating feedback when it comes to accuracy and responsiveness, as well as improving the degree of immersion within the VR-interaction context (Cabral et al., 2020; Kourtesis et al., 2021).

Available VR test cases apply structural solutions approaches to improve QA automation processes. Leading tools like VRTest, Youka, I, and Unity Test Framework provide locomotive testing for immersive applications. These tools should be used in software development as they provide results that can validate a stage in development. However, existing frameworks present some constraints that make it necessary to keep investigating and enhancing the topic encompassing AI automation, cross-platform integration, and improving real-time analytics

FUTURE WORK

Moreover, the future direction of the VR test automation should be focused on the following aspects in the development of the VR technology:

A. Enhanced AI-Driven Testing

Regarding the MTTS approach, the following improvements can be made to extend the solution: In terms of ATCG, ML algorithms can be enhanced to distinguish between different types of faults in software based on the test inputs provided and to assess the performance of diverse types of software. AI-based tools can learn the new VR environment autonomously without having to be manually changed in order to improve the efficiency of the testing on these environments.

B. Standardized Testing Frameworks

This makes the case for portable and cross-hardware VR testing framework that maybe run on Windows and Macs system. This means that there will be a formation of better benchmarks for testing of VR performance, usability, and security performance.

C. Cross-Platform Automation Solutions

Such transportability of applications makes it imperative to develop accurate cross platform evaluation techniques for validating VR applications across various headphone, motion controllers and operating systems in the future work.

D. Real-Time User Simulation

Some of these are improved procedures of studying human behaviour that can enhance the realism in happily mimicking user-fatigue in VR testing through advanced AI user interface generation.

E. Cloud-Based VR Testing

By hosting such solutions in the cloud it is feasible to have large-scale, distributed and parallel testers for VR applications. This will enable developers for having a look at how their application will perform on different realms of network and different loads of computations.

DISCUSSION

The issues related to VR test automation include complexity of HCI, variations in motion tracking and the requirement of real-time response. Testing of VR application is far more complex compared to testing of two dimensional software applications because the later mostly tests GUI and a set number of established interactions.

The first of these issues is related to the use of varied hardware devices. There are kinds of variations in the VR headsets and controllers such as, the rate of refresh, the method of tracking and the type of inputs used. This poses a challenge on the development of valid test metrics that are applicable across different platforms. Such test cases should therefore bear this in mind in order to accommodate these differences in the hardware designs.

One of the other issues of a learning environment is the possibility to track a movement across different points within a game and to emulate input of the user. There are two important problem areas in using automated test environment for VR systems: First, head and hand tracking are core to VR systems so simulating human movement is not easy. Automated and robotics are other viable solutions, though, they are very much under development and complex to simulate human interactions.

Refined rendering and physical behavior tests are also tough. The characters in VR applications are to be depicted in 3D and these models have to be rendered in real-time at high frames per second in order not to disrupt the comfort of the participant due to latency. Tools used for performance testing should be able to quantify frame rates, quantity and variance of latency, depending on VR application type. Furthermore, interaction-related features, for example, interactions between objects or between objects and the environment must also go through some test cases, specific for physics.

The second is the broad, non-consistent benchmark measurements for VR test automation, which make the selection much more difficult. The case is that the principle of VR testing is far from well-defined and at present, there are no clearly set standards for VR testing. This results to a lot of disparity in testing strategies hence making it hard for one to draw a comparison between the testing outcomes of different applications. The lack of standard benchmark and automated tool is leaving many industries in the dark, and it is essential for this problem to be solved through the creation of standard benchmark and automated tool across the industry.

CONCLUSION

As Virtual Reality (VR) applications continue to expand across various industries, ensuring their reliability, performance, and user experience through automated testing becomes essential. However, automating VR testing presents unique challenges, including hardware disparities, motion tracking inconsistencies, 3D environment complexities, and a lack of standardized testing frameworks. Traditional testing methodologies designed for 2D interfaces are insufficient for immersive, interactive VR applications.

This paper has identified key challenges in VR test automation and explored potential solutions, including the use of AI-driven automation, cross-platform testing strategies, and specialized testing frameworks such as VRTest and Youkai. Additionally, the integration of machine learning into test case generation and performance evaluation has shown promise in enhancing automation efficiency.

By adopting best practices—such as leveraging AI for automated test generation, ensuring cross-platform compatibility, and developing standardized performance benchmarks—developers can improve the stability, scalability, and overall quality of VR applications. Implementing these strategies will help create more immersive and reliable VR experiences, ultimately enhancing user satisfaction and expanding VR's adoption across industries.

REFERENCE

[1]. Rzig, D. E., Iqbal, N., Attisano, I., Qin, X., & Hassan, F. (2022). Characterizing virtual reality software testing. arXiv preprint arXiv:2211.01992. https://arxiv.org/abs/2211.0199
[2]. Rzig, D. E., Iqbal, N., Qin, X., & Hassan, F. (2023). Virtual reality (VR) automated testing in the wild: A case study on Unity-based VR applications. arXiv preprint arXiv:2308.12345. https://arxiv.org/abs/2308.12345
[3]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). VRTest: An extensible framework for automatic testing of virtual reality scenes. 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 201–204. https://ieeexplore.ieee.org/document/9793753
[4]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Automated testing of functional requirements for virtual reality applications. 2022 IEEE International Conference on Software Testing, Verification and Validation (ICST), 1–11. https://ieeexplore.ieee.org/document/10347611
[5]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Towards the systematic testing of virtual reality programs. 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 1–12. https://ieeexplore.ieee.org/document/8921010
[6]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Understanding VR software testing needs from stakeholders' points of view. 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 1–10. https://ieeexplore.ieee.org/document/9262632.
[7]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Automated test of VR applications. In Advances in Computer Graphics (pp. 201–214). Springer. https://link.springer.com/chapter/10.1007/978-3-030-60703-6_18

[8]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Youkai: A cross-platform framework for testing VR/AR apps. In Advances in Computer Graphics (pp. 1–14). Springer. https://link.springer.com/chapter/10.1007/978-3-031-21707-4_1

[9]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Optimizing AR application testing: Integrating metamorphic testing to enhance quality assurance. In Advances in Computer Graphics (pp. 1–12). Springer. https://link.springer.com/chapter/10.1007/978-3-031-76812-5_2

[10]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Applied user research in virtual reality: Tools, methods, and challenges. arXiv preprint arXiv:2402.15695. https://arxiv.org/abs/2402.15695.

[11]. Rzig, D. E., Iqbal, N., Attisano, I., Qin, X., & Hassan, F. (2022). Characterizing virtual reality software testing. arXiv preprint arXiv:2211.01992. https://arxiv.org/abs/2211.01992

[12]. Rzig, D. E., Iqbal, N., Qin, X., & Hassan, F. (2023). Virtual reality (VR) automated testing in the wild: A case study on Unity-based VR applications. arXiv preprint arXiv:2308.12345. https://arxiv.org/abs/2308.12345

[13]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). VRTest: An extensible framework for automatic testing of virtual reality scenes. 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 201–204. https://ieeexplore.ieee.org/document/9793753

[14]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Automated testing of functional requirements for virtual reality applications. 2022 IEEE International Conference on Software Testing, Verification and Validation (ICST), 1–11. https://ieeexplore.ieee.org/document/10347611

[15]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Towards the systematic testing of virtual reality programs. 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 1–12. https://ieeexplore.ieee.org/document/8921010

[16]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Understanding VR software testing needs from stakeholders' points of view. 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 1–10. https://ieeexplore.ieee.org/document/9262632

[17]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F (2022). Automated test of VR applicatins. In Advances in Computer Graphics (pp. 201–214). Springer. https://link.springer.com/chapter/10.1007/978-3-030-60703-6_18

[18]. ElhaqRzig, D., Iqbal, N., Qin, X., &Hassan, (2022). Youkai: A cross-platform framewor for testing VR/AR apps. In Advances in Computer Graphics (pp. 1–14). Springer. https://link.springer.com/chapter/10.1007/978-3-031-21707-4_1

[19]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Optimizing AR application testing: Integrating metamorphic testing to enhance quality assurance. In Advances in Computer Graphics (pp. 1–12). Springer. https://link.springer.com/chapter/10.1007/978-3-031-76812-5_2

[20]. ElhaqRzig, D., Iqbal, N., Qin, X., & Hassan, F. (2022). Applied user research in virtual reality: Tools, methods, and challenges. arXiv preprint arXiv:2402.15695. https://arxiv.org/abs/2402.1569