Capture and Replay Testing Tool for XR Applications

Seyed Nami Modarressi[1, *], Frank Maurer[1] and Nanjia Wang[1]

[1] University of Calgary, Calgary, Alberta, Canada

Abstract

In the world of software, it's crucial to make sure applications work well. Testing is a big part of this, but as software gets more complex, especially in XR apps, testing becomes time-consuming and can miss important things. This research suggests a new testing tool, inspired by a successful method in web development, to make testing XR apps faster and more efficient. The goal is to help developers save time and money, filling a gap in the XR domain where such tools are lacking. This could make a significant difference in improving the quality and reliability of applications.

Keywords

Software Testing, XR Application, Capture and Replay, Testing Tool, Test Automation

1. Introduction

The widespread use of software in various domains underscores the crucial need to ensure the quality and reliability of applications. In contemporary development, testing is pivotal for ensuring efficacy and dependability. However, the growing complexity and size of software have made testing more time-consuming, risking the oversight of crucial components and introducing potential financial and non-financial risks in projects.

Addressing the challenges of extensive software testing, especially in the realm of XR applications, is imperative. Advanced hardware and a growing awareness of problem- solving potential within the XR domain highlight the need for robust testing methodologies tailored to this context. The absence of dedicated software testing tools for XR applications has forced developers to manually create testing protocols, contributing to a tendency to neglect testing altogether.

Drawing inspiration from the success of the Record and Play testing methodology in streamlining testing processes, particularly in web development using Selenium, this research endeavors to explore the applicability of such an approach in the testing of XR applications. The proposed research aims to design, implement, and evaluate the feasibility and effectiveness of a Record and Play testing tool specifically tailored for XR applications.

This research aims to improve XR development testing by introducing an efficient, time-saving methodology that mitigates financial and temporal burdens associated with extensive

RealXR: Prototyping and Developing Real-World Applications for Extended Reality, June 4, 2024, Arenzano (Genoa), Italy

[*] Corresponding Author

✉ seyednami.modarressi@ucalgary.ca (S. N. Modarressi); fmaurer@ucalgary.ca (F. Maurer); nanjia.wang1@ucalgary.ca (N. Wang)

🌍 https://seriousxr.ca (F. Maurer)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

software testing. The absence of similar solutions in the XR domain underscores the significance of this research, offering a valuable contribution to the toolkit of developers aiming to enhance the quality and reliability of their applications.

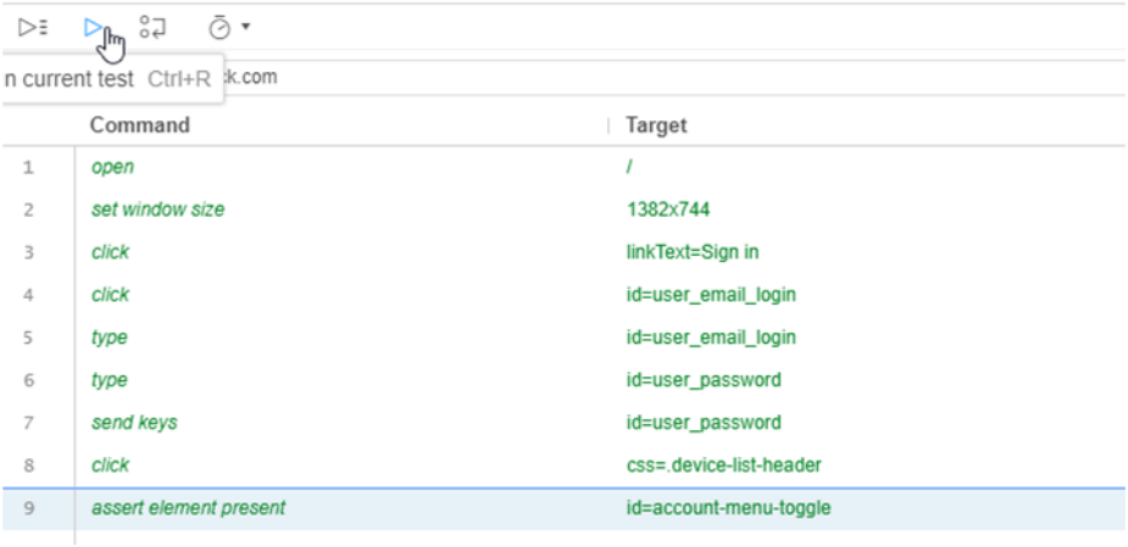
2. Background

The escalating prevalence of software applications in diverse domains, such as social networks, artificial intelligence, system control, and automation, accentuates the critical imperative to ensure their quality and reliability [1]. In the context of modern software development, testing emerges as a pivotal mechanism for guaranteeing the effectiveness and dependability of these applications. However, the growing complexity of software architecture has led to a concomitant increase in the time-consuming nature of testing, potentially resulting in the oversight of crucial components [1]. This trend becomes particularly alarming when considering the real-world consequences, as exemplified by the 2014 Heartbleed Bug incident, where a lack of rigorous testing and code review allowed a serious security vulnerability in OpenSSL to persist for over two years, compromising sensitive data on numerous web servers [1].

Addressing the formidable challenges posed by extensive software testing, especially in the realm of Extended Reality (XR) applications, becomes imperative [2]. The heightened hardware sophistication and increased problem-solving awareness in the XR domain necessitate tailored testing methodologies. Unfortunately, the absence of dedicated testing tools for XR has led developers to resort to manual creation of testing protocols, often resulting in testing neglect [2]. One of the key challenges in XR testing is UI testing, further complicated by the 3D dimension, making GUI automation more challenging due to overlapping objects and the need for a coordination system to test object positions [2].

In the pursuit of effective testing methodologies, the literature highlights the method of record and play as an intelligent approach [3]. This method involves recording all the actions required in a specific part of the software, whether it be a scene in an XR application or a page on a website. After recording these actions, developers or testers can play back the recorded test to observe the result, eliminating the need for manual testing or coding [3, 4, 5]. This not only streamlines the testing process but also facilitates comprehensive testing without the need for extensive manual efforts.

The challenges associated with testing graphical user interfaces (GUIs) have been a longstanding concern, exacerbated by the adoption of Agile development methodologies [3]. These methodologies emphasize comprehensive testing to showcase software functionality from both developers' and users' perspectives [3]. In this context, test-driven development (TDD) emerges as a significant area where automated GUI tests can seamlessly integrate into an agile process [3]. There are instances when scripting automated tests manually for an existing system before modifications may not be practical. In such cases, the use of "record & playback" testing stands out as a viable alternative, offering a pragmatic approach to test creation without the need for exhaustive manual efforts [3, 5]. The documented benefits of record and playback testing further hint at its potential to simplify end-to-end testing for XR applications, making it a promising avenue for ensuring the reliability and functionality of these complex software systems [3].



	Command	Target
1	open	/
2	set window size	1382x744
3	click	linkText=Sign in
4	click	id=user_email_login
5	type	id=user_email_login
6	type	id=user_password
7	send keys	id=user_password
8	click	css=.device-list-header
9	assert element present	id=account-menu-toggle

Figure 1: Selenium Record and Play Testing Tool for Web Development. You can see list of the recorded action. Developer can run the same list of actions just by clicking the run test button.

3. Methodology

To create a tool for testing in Unity, it's crucial to understand how Unity stores objects. Each object has a unique ID, and there are various associated properties for each ID. It's essential that after every operation, at least one property undergoes a change. Our goal is to capture these changes and enable developers to replicate and test them. In this case, developers just need to run their app and start interacting with their environment and objects. Everything is going to be recorded, and afterward, developers or testers can simply repeat each recorded test after making any changes to their objects or environment.

Arium[6, 7], a framework designed for Unity automation testing, is helpful in accessing object information and facilitating automation testing. After creating tests in the Arium framework, they can be automatically executed, producing clear results (see Figure 2). However, a drawback of Arium is the requirement for manual test writing. In other words, developers or testers need to write tests for their programs and then utilize Arium for test automation.

In response, our framework, an extension of Arium for Unity's automation testing, provides an innovative solution. Our tool can automatically record tests, eliminating the need for manual script creation. This extension simplifies the testing process, making it easy to record and execute tests within the Arium environment. This advancement streamlines the testing workflow, making it more efficient for Unity developers.

To sum up, developers can define their scenes and objects. Then, they can run everything and make their own changes in the playmode of Unity. After that, everything will be recorded. With the help of Arium, they can automatically run the same test. This means they can conduct automation testing without the need for coding. It also implies that they can test their code much faster and with less effort.

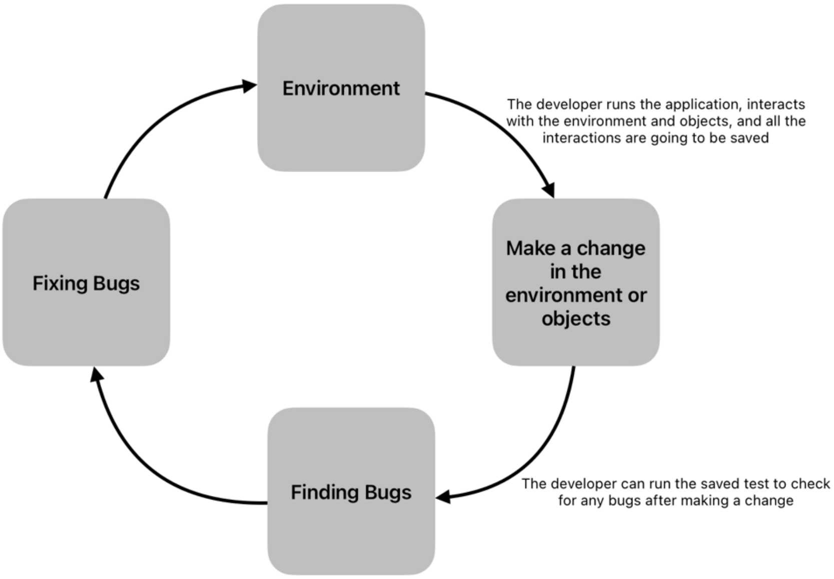

Environment
The developer runs the application, interacts with the environment and objects, and all the interactions are going to be saved
Make a change in the environment or objects
The developer can run the saved test to check for any bugs after making a change
Finding Bugs
Fixing Bugs

Figure 2: The Process of Using Record and Play Testing Method for Testing XR Applications

4. Evaluation

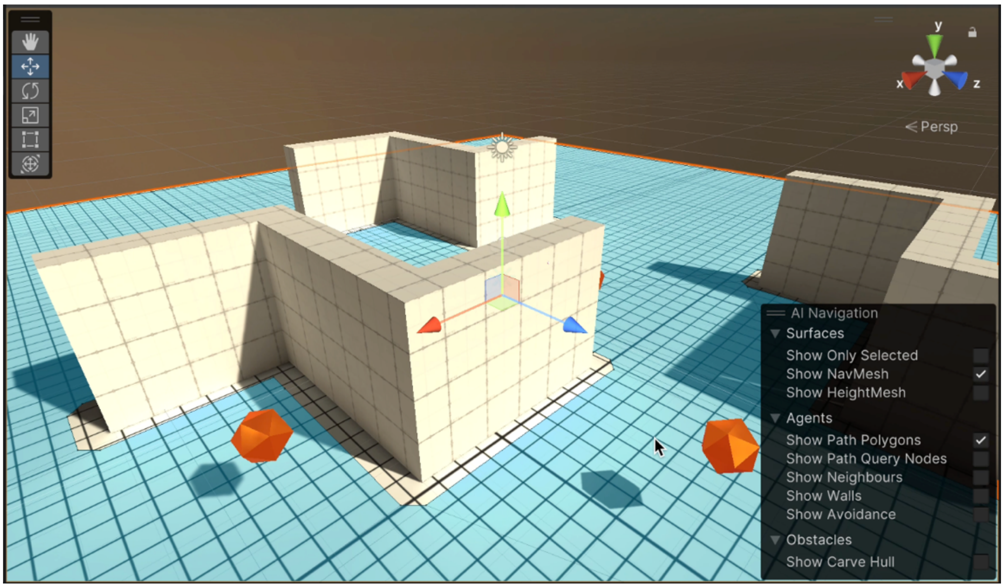
To evaluate the testing tool, we need to conduct a study. In this case, we can invite developers. There are two phases. In the first phase, we give developers a project. In this, there is a scene with different objects and there is a bug that they need to find and fix. At first, we ask them to read the documentation of the project and try to write code to find that bug. In the second step, we give them the same project but with a different bug of the same difficulty level. However, this time we provide them with our testing tool. Based on the comparing results from these two scenarios, we can determine if the testing tool is effective or not.

Additionally, we can use questionnaires to gather feedback on their experience with the testing tool and compare it to their experience without it. It's important to note that while there are alternative methods like autoethnography, evaluating this tool in a real-world environment with actual XR developers is crucial for its practical application.

As shown in Figure 3, developers can be tasked with testing a project that comprises one environment and contains various objects. When users collect these objects, they accumulate points. Subsequently, users can navigate within the environment to gather all the point objects. In this scenario, developers can swiftly build this environment and proceed to test it. Initially, they need to conduct manual testing, which involves writing code for the process. Following the initial manual testing, we can then instruct them to utilize our testing tool, specifically employing the record-and-play method. Subsequently, we can analyze the relevant parameters and administer a questionnaire to assess the effectiveness of our testing tool.

5. Expected Outcomes

Creating a tool to record and play tests for XR (Extended Reality) apps is a big step in the world of immersive technologies. This idea is inspired by a common practice in web development called capture and replay, where recording and replaying interactions helps with debugging, testing, and improving the user experience. In the XR world, including virtual reality (VR), augmented reality (AR), and mixed reality (MR), interactions are complex, and three-dimensional spaces are dynamic, requiring new testing solutions.



AI Navigation
Surfaces
Show Only Selected ☐
Show NavMesh ☑
Show HeightMesh ☐
Agents
Show Path Polygons ☑
Show Path Query Nodes ☐
Show Neighbours ☐
Show Walls ☐
Show Avoidance ☐
Obstacles
Show Carve Hull ☐

Figure 3: Sample Project

The proposed tool aims to make it easier for developers to create XR apps by letting them record user interactions in immersive environments and replay them for testing and optimization. This approach, successful in web development, is expected to bring significant benefits to XR app development.

The goals of this project are diverse. Firstly, the tool should make XR app development more efficient by letting developers record and replay user interactions, making troubleshooting and refining the user experience more straightforward. Additionally, it should contribute to improving the overall quality of XR apps by making testing and refinement more accessible.

This tool's impact could go beyond individual developers, influencing entire XR development teams and the industry at large. Introducing a specialized record and play testing tool for XR apps may set a new standard in immersive technology development. It has the potential to change how XR apps are tested and refined, creating a more robust and efficient development environment.

To sum up, the effort to make a record and play testing tool for XR apps is based on the success of similar methods in web development. Adapting and refining these concepts to fit the challenges of extended reality environments is expected to benefit individual developers and potentially transform how XR development is done. As immersive technologies evolve, creating innovative tools like this one shows a commitment to advancing the field and ensuring the smooth creation of exciting XR experiences.

6. Conclusion

In conclusion, the advent of the record and play testing tool for XR applications marks a significant leap forward in the immersive technology landscape. This innovative tool, inspired by successful practices in web development, not only addresses the unique challenges posed by extended reality environments but also introduces a transformative potential for the entire XR industry. By streamlining the testing and refinement of user experiences in virtual, augmented, and mixed reality spaces, the tool enhances development efficiency and elevates the overall quality of XR applications.

The diverse advantages offered by this tool, from efficiency gains to the prospect of industry-wide transformation, underscore its pivotal role as a driver of positive change. Its seamless integration into XR development workflows is a testament to the industry's commitment to adaptability and innovation, signaling a lasting impact on the crafting of captivating and seamless XR experiences for a global audience.

In essence, the record and play testing tool signifies a paradigm shift in XR application development, going beyond mere efficiency gains to become an integral catalyst for positive change in the immersive technology landscape. As it becomes a standard practice within the industry, this tool is poised to leave a lasting impression, emphasizing the dedication of the XR community to adapt, innovate, and contribute to the creation of immersive and compelling experiences for users worldwide. The continued evolution of such tools ensures the sustained progress of immersive technologies and sets the stage for a dynamic and exciting future in XR development.

References

[1] M. Carvalho, J. DeMott, R. Ford, and D. A. Wheeler, Heartbleed 101, IEEE Secure. Priv., vol. 12, no. 4, pp. 63:67, Jul. 2014, doi: 10.1109/MSP.2014.66.

[2] T. Hellmann, Automated GUI Testing for Agile Development Environments, 2015, doi: 10.11575/PRISM/25075.

[3] G. Meszaros, Agile regression testing using record & playback, in Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, Anaheim CA USA: ACM, Oct. 2003, pp. 353:360. doi: 10.1145/949344.949442.

[4] G. J. Myers, T. Badgett, and C. Sandler, Eds., The Art of Software Testing, 1st ed. Wiley, 2012. doi: 10.1002/9781119202486.

[5] A. Bruns, A. Kornstadt, and D. Wichmann, Web Application Tests with Selenium, IEEE Softw., vol. 26, no. 5, pp. 88:91, Sep. 2009, doi: 10.1109/MS.2009.144.

[6] Thoughtworks. (n.d.). Thoughtworks/arium: Automation testing framework for xr(unity) applications. GitHub. https://github.com/thoughtworks/Arium

[7] Neelarghya, “Arium - an automation framework for unity/XR,”Medium, https://medium.com/xrpractices/arium- an-automation-framework-for-unity-xr-d51ed608e8b0 (accessed Mar. 26, 2024).