CMPE 492

Senior Project 2

Test Plan Report

“VR PROJECT BLUE”

Project Supervisor

Dr. Özlem Albayrak

Project Team Members

Sancaralp Elmas

Hasan Yasin Yıldız

Uğur Evren Çamalan

Eren Çağlaroğlu

Project Jury Members

Dr. Ulaş Güleç

Dr. Elif Kurtaran Özbudak

Course Coordinator

Dr. Yücel Çimtay

Project Website

teduvrsp.github.io

Table of Content

1. Introduction

1.1 Purpose

We, the development team at “VR BLUE” present this comprehensive Test Plan Report for our groundbreaking serious game in Virtual Reality (VR). The purpose of this document is to articulate the meticulous approach, methodologies, and strategies employed in the testing phase of our immersive cyber security educational experience.

1.2 Scope

1.2.1 Holistic Testing Approach

This test plan encompasses an all-encompassing evaluation, ensuring a seamless blend of entertainment and education. It delves into the identification of features, testing methodologies, and risk analysis associated with our VR game.

1.2.2 Educational Puzzles and Gameplay Mechanics

The scope extends to the incorporation of educational puzzles within the gameplay, designed to impart fundamental knowledge of the cybersecurity field. We focus on evaluating VR interactions, such as holding and throwing objects, to ensure an engaging and educational experience for the player.

2. Defining Scope

2.1 Game Overview

"VR BLUE” is an immersive Virtual Reality (VR) serious game that combines entertainment with education, focusing on the critical field of cybersecurity. This educational experience is designed to engage players through puzzles while delivering foundational knowledge about cybersecurity principles. The game falls within the educational genre and is targeted at individuals interested in cybersecurity awareness and learning. It is developed for VR platforms, ensuring an interactive and immersive gameplay experience.

Key Features:

Educational Puzzles

VR Gameplay Mechanics (holding, throwing, etc.)

Compelling Cybersecurity Storyline

Target Audience:

Individuals interested in cybersecurity education.

VR enthusiasts seeking an engaging and educational experience.

Platforms:

Virtual Reality (VR) platforms

2.2 Understanding the Game

2.2.1 Story and Levels

The game follows a captivating storyline set in the realm of cybersecurity. Players progress through levels, each presenting unique challenges and educational puzzles. The length of the story is designed to provide a comprehensive yet engaging experience, ensuring players grasp fundamental cybersecurity concepts.

2.2.2 Characters, Weapons, and Abilities

Players encounter diverse characters, each contributing to the storyline and educational elements. While the game may not feature traditional weapons, it incorporates cybersecurity tools and abilities that players must master to progress.

2.3 Testing Needs Assessment

2.3.1 Number of Testers

Given the complexity of intertwining entertainment with education in a VR environment, we estimate that 3 or 4 skilled testers will be required. Testers with a keen understanding of VR mechanics and an interest in cybersecurity education will be particularly valuable.

2.3.2 Estimated Testing Time

The testing phase is expected to span minimum 8 weeks to ensure a thorough evaluation of all features, mechanics, and educational components. This timeframe includes iterative testing to address any identified issues.

2.3.3 Resources

Resources required for testing include VR headsets, compatible controllers, testing environments that simulate various player scenarios, and access to cybersecurity experts for content validation.

3. Risk Analysis

3.1 Prioritization of Testing Efforts

There are many elements that significantly affect the player experience. These are like user interface and gameplay mechanics. First of all, since we are developing a VR game, visuality in the game is at the forefront for us. We attach importance to the visual environment we design attracting the attention of the user as much as possible. In this regard, we attach importance to the feedback of different users in terms of the development of our game. Additionally, another testing priority is game mechanics. Since the user will perform various activities using these mechanics in the game, we made sure that the game mechanics do not overwhelm the user. In the test studies, we observed as much as possible whether these mechanics could be easily implemented by the user.

3.2 Identification of High-Risk Areas

Player Comfort and Safety: Since our game primarily aims to entertain while teaching, it is very important that the player feels comfortable and safe in the game. For example, the design of the VR headset used may make the user feel uncomfortable. Or the audio or visual objects used in the game may disturb the player in the game. Within the scope of this risk, we are careful to use visual and audio objects that will make the user feel as comfortable as possible. We also attach great importance to not using any elements in the game that could frighten or traumatize the user.

Data privacy: Data privacy has become a very important element with the development of technology. For this reason, protecting data privacy has become of great importance. We attach great importance to ensuring that the game we develop complies with the law on the protection of personal information in terms of data collection. In this context, we do not share our users' information with any 3rd party applications without their permission.

VR Mechanics and User Interface: Some mechanics used in VR games can cause physical harm to the user. In this context, we attach great importance to ensuring that the mechanics in the game do not harm the user. In addition, sometimes the visual objects used can physically tire the user. With this awareness, we attach importance to the fact that the visual objects we use do not negatively affect the user in any way.

3.3 Response to High-Risk Scenarios

Test results and information must be shared directly with the development team to prevent high risk scenarios from occurring. For this reason, providing feedback is very important during the testing phase. Test results should be reviewed regularly in team meetings and integrated into development strategies. Additionally, testing priorities must be set and determined. In this way, the development team can prioritize high-risk areas. When high-risk problems are identified, effective solution strategies must be produced and implemented. In this way, the development team has more comprehensive information about the possibility of high risk scenarios occurring.

4. Scheduling

4.1 Development Stages

The development process of VR Project Blue typically consists of four main stages:

Pre-Alpha:

Objective: To assess the core gameplay mechanics and ensure they are enjoyable and engaging for players.

Testing Activities:

Preliminary testing: Conduct small-scale playtests with a limited group of users to gather initial feedback on the game's mechanics and overall concept.
Alpha:

Objective: To refine the game's mechanics, identify and fix major bugs, and gather feedback on the overall gameplay experience.

Testing Activities:

Play testing: Conduct more extensive playtests with a larger group of users to gather detailed feedback on the mechanics, level design, user interface, and overall gameplay experience.
Bug fixing: Prioritize and fix major bugs identified during playtests.
Beta:

Objective: To polish the game, identify and fix any remaining bugs, and gather feedback on the game's overall quality and marketability.

Testing Activities:

Play testing: Conduct large-scale playtests with a diverse group of users to gather comprehensive feedback on the game's overall quality, including gameplay, bugs, performance, and user interface.
Game testing: Conduct rigorous game testing to identify and fix any remaining bugs, performance issues, or compatibility problems.
Final:

Objective: To ensure the game is ready for release and meets all quality standards.

Testing Activities:

Game testing: Conduct final rounds of game testing with a wide range of users to verify that the game is bug-free, performs well, and is compatible with different hardware and software configurations.

Play Testing vs. Game Testing

While both play testing and game testing are essential components of VR game development, they serve distinct purposes:

Play testing: Focuses on evaluating the core gameplay mechanics, level design, and overall user experience. It aims to identify areas that may hinder player enjoyment or engagement.

Game testing: Encompasses a broader range of aspects, including bug identification, performance optimization, compatibility checks, and overall game stability. It ensures the game meets technical and quality standards before release.

4.2 Test Timeline

Phase	Start Date	End Date	Milestones	Key Deliverables
Pre-Alpha	June 1, 2023	June 30, 2023	Preliminary testing results	- Initial feedback on core gameplay mechanics - Identification of potential issues or concerns
Alpha	July 1, 2023	August 31, 2023	Play testing feedback and bug fixes	- Detailed feedback on gameplay mechanics, level design, and user interface - Prioritization and resolution of major bugs
Beta	September 1, 2023	October 31, 2023	Comprehensive play testing and game testing results	- Extensive feedback on overall game quality, including gameplay, bugs, performance, and user interface - Identification and resolution of remaining bugs and compatibility issues
Final	November 1, 2023	November 30, 2023	Final game testing and bug fixes	- Verification of game stability and bug-free operation - Optimization of game performance and compatibility across hardware and software configurations

5. Test Approach

5.1 Testing Types and Methodologies

Functional Testing:

- Verify the core gameplay mechanics, such as movement, object interaction, puzzle completion, and educational elements.
- Validate that the cybersecurity concepts are accurately conveyed and integrated with the gameplay.
- Ensure the seamless functioning of VR interactions (holding, throwing, etc.).
- Test the user interface (UI) for intuitive navigation and clarity.

Smoke Testing:

- Confirm that the game can start and shut down without issues.
- Check that the main menu, settings, and early levels are accessible.
- Verify basic VR interactions and game progression.

Performance Testing:

Assess the game's performance on various hardware configurations to ensure smooth operation and optimal frame rates.

Compatibility Testing:

Validate compatibility with different VR headsets and operating systems to guarantee a seamless experience across various platforms.

Accessibility Testing:

Ensure the game is accessible to players with disabilities by incorporating alternative input methods and visual enhancements.

Playtesting:

Conduct iterative playtesting sessions with a diverse group of users to gather feedback on gameplay, engagement, educational value, and overall experience.

5.2 Test Case Creation and Management

Test Case Creation:

- Gather requirements from game design documents and specifications.
- Identify key game features and functionalities to be tested.
- Decompose features into testable scenarios and create detailed test cases.
- Define expected outcomes and pass/fail criteria for each test case.
- Prioritize test cases based on risk and importance.

Test Suite Organization:

- Group test cases logically based on features or functionalities.
- Create separate test suites for different testing types (functional, smoke, regression).
- Utilize a test case management tool for organization and tracking.

Test Case Management:

- Track test case execution and results throughout testing phases.
- Update test cases as the game evolves.
- Maintain test case documentation for future reference and regression testing.

5.3 Utilization of Testing Tools

Compatibility Testing Tools:

Leverage compatibility testing tools to validate game compatibility across various hardware and software configurations.

Playtesting Tools:

Employ playtesting tools to record and analyze player behavior, gather feedback, and identify areas for improvement.

6. Defect Management

6.1 Bug Reporting

A standardized bug reporting process is essential for effective defect management:

Bug Identification:

Testers will identify and document any encountered bugs or discrepancies during testing sessions.

Bug Reporting Form:

Testers will utilize a bug reporting form to provide detailed information about each bug, including steps to reproduce, severity level, and any relevant screenshots or videos.

Bug Repository:

Reported bugs will be stored in a centralized bug repository, such as Bugzilla for easy tracking and management.

6.2 Bug Resolution Process

A streamlined bug resolution process ensures timely and effective bug fixes:

Bug Prioritization:
Reported bugs will be prioritized based on their severity and impact on the game's functionality and overall experience.

Bug Assignment:
High-priority bugs will be assigned to developers for immediate attention and resolution.

Bug Fixing:
Developers will implement bug fixes and thoroughly test the affected areas to ensure regression issues are prevented.

Regression Testing:
Regression testing will be conducted to verify that bug fixes have not introduced any new issues.

Bug Closure:
Resolved bugs will be marked as closed in the bug repository and communicated to testers.

Communication:
Developers and testers will maintain open communication throughout the bug resolution process to ensure timely updates and issue resolution.

7. Conclusion

7.1 Summary

This Test Plan Report outlines a comprehensive approach to testing VR Project Blue, ensuring that the game meets the highest standards of quality and delivers an engaging, educational, and enjoyable experience for players. The testing strategy encompasses a variety of methodologies, tools, and processes to thoroughly evaluate all aspects of the game, from core gameplay mechanics to accessibility considerations. By adhering to this rigorous testing plan, we aim to deliver an exceptional VR game that seamlessly blends entertainment and education, promoting cybersecurity awareness and knowledge.

Appendix

A. References

- "Software Testing Fundamentals: A Comprehensive Guide for Beginners" by A.G. Stefanescu and I. Budimac
- "Testing Computer Software" by Cem Kaner, James Bach, Pettichord, and Elizabeth Hunt
- "Game Testing: A Practical Guide" by Ian Myre
- "VR Testing: A Practical Guide for Developers" by Jason Jermaine

B. Glossary of Terms

- Functional Testing: Verifies that software functions as intended and meets all requirements and specifications.
- Smoke Testing: A quick test to ensure that the basic functions of the software are working before more comprehensive testing is performed.
- Regression Testing: Verifies that changes made to the software have not introduced any new bugs.
- Performance Testing: Assesses the performance of the software under various workloads