



UNIT 13 SOFTWARE TESTING

Assignment 2

Learning Aim B & C

Carry out a range of testing methods on a software product to meet a client's needs &
Review and present the results from software tests to meet a client's needs and suggest
improvement

Oliver Collins-Cope

2102775@rutc.ac.uk

Contents

Introduction	2
Game Description	2
Using Agile	2
Activity diagram	3
Basic Test Plan.....	4
Improved Test Plan	7
User acceptance testing against requirements	7
Analysis of test results	8
Basic test plan	8
Test 1	8
Test 2	8
Test 3	9
Test 4	9
Test 5	9
Test 6	10
Test 7	10
Test 8	10
Improved test plan	11
Test 1	11
Test 2	11
Test 3	11
Evaluation	12
Testing quality	12
Test plan improvements	13

Introduction

This case study aims to depict the impact and importance of testing in different project management methodologies, and how choosing a different project management methodology can heavily influence the testing, ranging from the quality of tests, the tests used, and the final outcome of the tests. This will be documented in an orderly fashion in order to maintain a clear documentation of the work. Additional sources and projects used include a games development project, and can be found linked in the bibliography.

Game Description

The game that I had decided to make for this was a 3D puzzle game that was supposed to meet a number of requirements. These requirements were various and include:

- 8+ years old
- Collision detection
- Score
- Levels
- Lives

In order to satisfy these requirements, I found unique ways to implement them into my game, like adding in a timer that counts down until the player loses their one life and restarts. Furthermore, I aimed to add two levels, therefore meeting another requirement that was provided to me. This is an example of the different types of requirements and how I worked to implement them in the game.

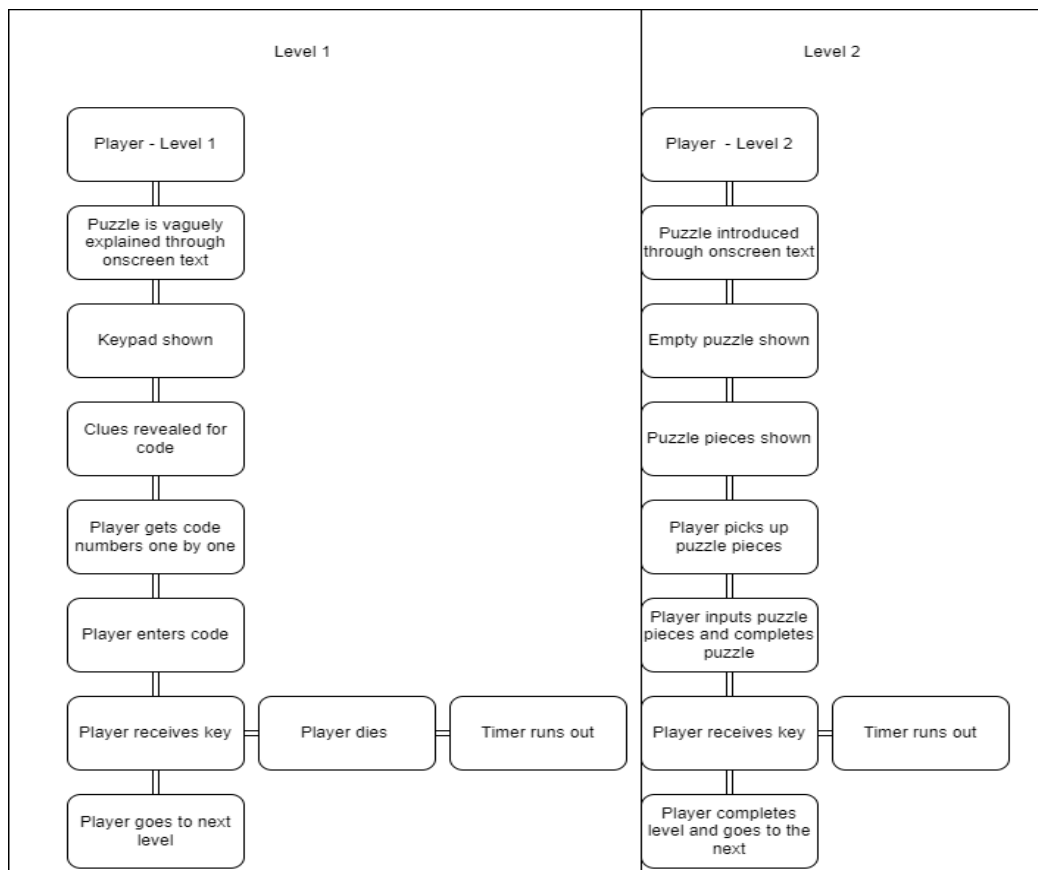
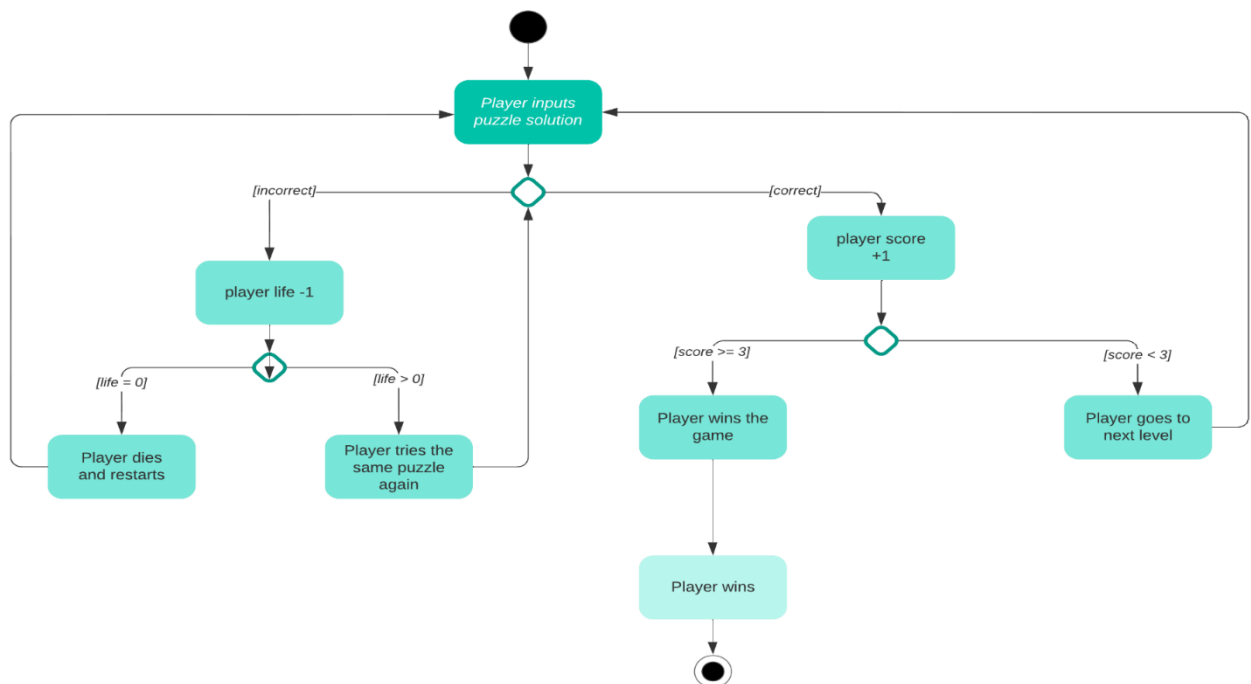
To summarise, I had a 3D puzzle game that met the above requirements, and these successfully made a completed game.

Using Agile

When using Agile within my game I found that it was essential due to its iterative and flexible approach to development. When it came to testing the actual game, I found that the principles ingrained into Agile development were essential to the development of my game to a high quality. The software testing was seamlessly integrated throughout the whole of the development process, allowing for different iterative phases of testing and development. Testing was often run in tandem with development in order to ensure that all of the introduced features were fully functional, and any developments made had instant feedback on their impact and implementation.

Furthermore, the incremental nature of Agile allowed for smooth implementation and testing of new features, ensuring that all of the features were added correctly and there were no bugs within the software. The emphasis on prioritisation of tasks led to an increase in productivity when it came to resolving issues as they were put at the top of the to do list in order to ensure that final product was a bug free as possible. Regular self-reviews, as a placeholder for meetings, also allowed me to identify better what was a priority, i.e., a new feature vs bug fixes, allowing for a high-quality final product.

Activity diagram



An activity diagram is an essential component in game planning as it allows for the developers, like me, to create a clear understanding of how I wanted the games structure and progression to exist. The emphasis on using the activity diagram here is due to the



illustrative ability it provides, displaying logical processes that must occur and that enable the player to move forward through these different points.

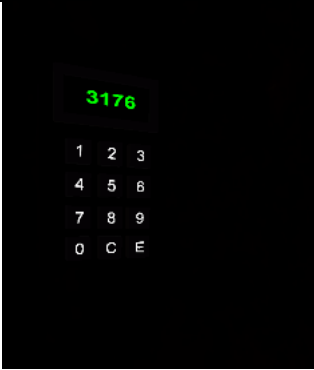

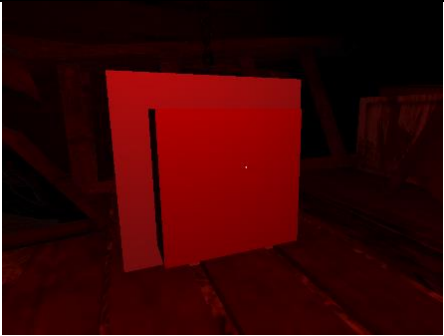
Through the visual display of these different points, like receiving a key, this offers a comprehensive overview of the players potential journey through the level, and what those objectives signify to the development/programming team, allowing for them to create and develop around those ideas and ensure a smooth playing experience where everything the player does makes sense.

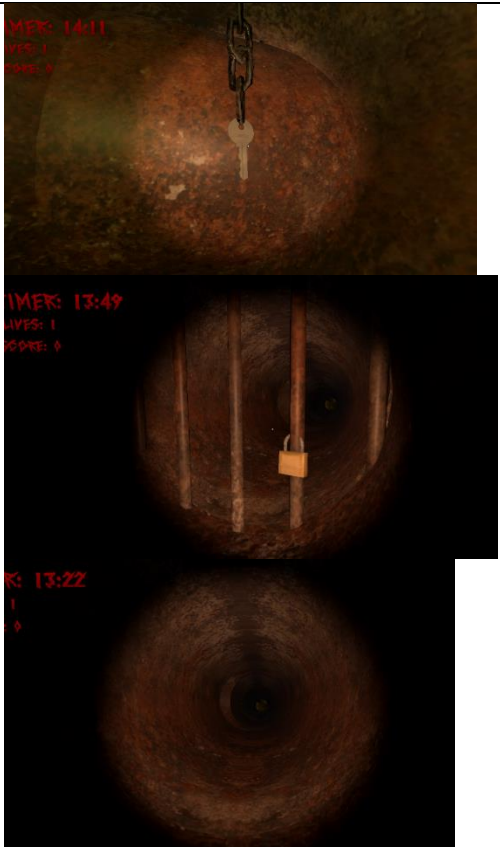
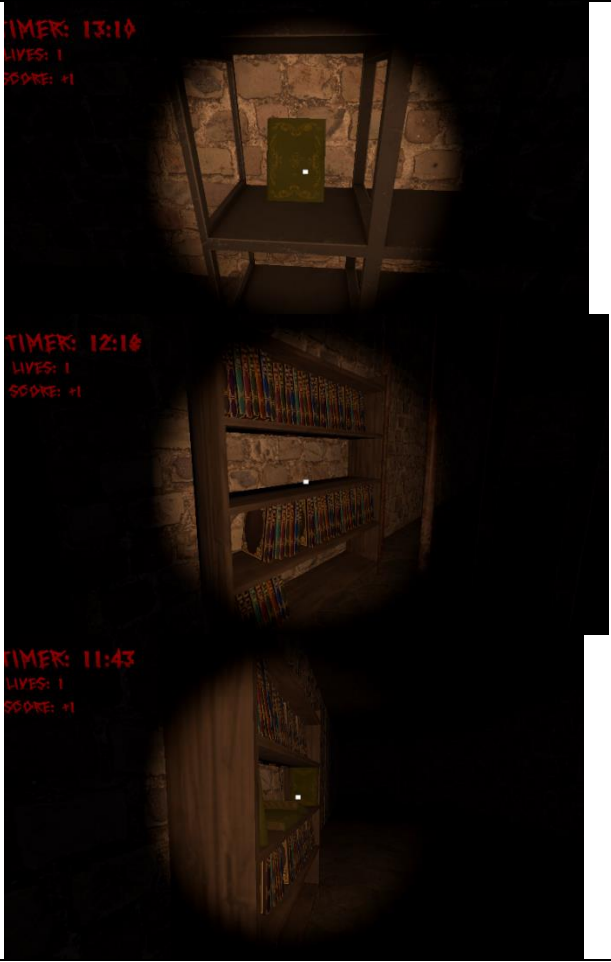
This allows a clear view at the game's progressions, while also having the additional benefit of identifying where the game can have key moments, like introducing different features or aspects at moments where they are needed to solve a problem, therefore teaching the player through doing things.

Finally, the incorporation of the activity diagrams in the final plan of the game has aided in creating a streamlined experience for the player where the game progresses linearly and in a way that provides the best experience.


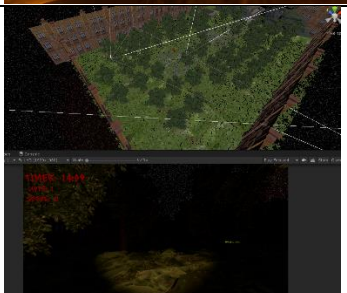
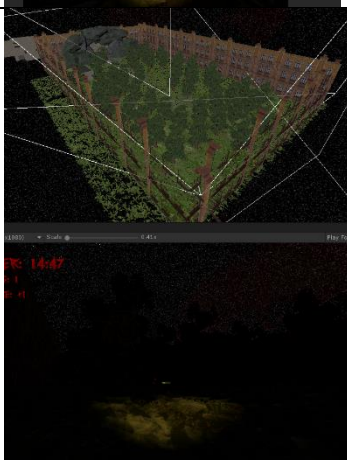
Basic Test Plan

Test No.	Test case	Type of Test	Expected outcome	Actual outcome	Result	Evidence
1.	Player will be able to pick up objects (i.e., flashlight)	Unit testing	The player will be able to pick up the object.	The object could not be picked up.	Fail	
2.	Player being able to turn a flashlight on and off by pressing a button.	Unit testing	The player will be able to turn the flashlight on and off.	The player turned the flashlight on and off.	Pass	
3.	The player is able to move around and pick	Element testing	The player will be able to move	The player was able to move and interact with the	Pass	See submitted screenshots above, and submitted video.

	up the flashlight, and then turn it off and on.		around and turn it off and on after picking it up.	flashlight, picking it up and turning it off and on.		
4.	The player will input a password into a keypad, and it will open a door next to it, both having different scripts.	Integration testing	The door will open after the password is verified.	The password flashed green and then the door opened	Pass	
5.	The game will run smoothly without discrete graphics card.	Performance testing	There game will run without issue on both levels and function well.	The second level encountered some performance issues.	Failed	 <p>Also see previously submitted document detailing this.</p>
6.	The cube, when interacted with, will load the next level, and put the player there.	Element testing	The next level will load without issue. When the player interacts with the cube.	Interacting with the cube did nothing.	Failed	 <p>Pressing "E" does nothing</p>

7.	The player will be able to pick up the key and use it to interact with the door on the first level.	Integration testing	The key can be picked up and the door is opened.	The door was opened perfectly fine after the key was picked up.	Pass	 <p>The first screenshot shows a key hanging from a chain on a dark, textured surface. The second screenshot shows a door with a keyhole and a small orange padlock. The third screenshot shows a dark, circular opening, possibly a tunnel or a well.</p>
8.	The player will be able to pick up the books and put them in the bookshelf.	Element testing	The player will be able to interact and pick up the books, and then put them in the shelf.	The player was able to pick up the books and put them in the shelf.	Pass	 <p>The first screenshot shows a bookshelf with a single green book. The second screenshot shows a bookshelf with many books. The third screenshot shows a bookshelf with a single green book.</p>

Improved Test Plan

Test No.	Test case	Type of Test	Expected outcome	Actual outcome	Result	Evidence
1.	The player will be able to pick up objects.	Unit testing	The player will be able to interact and pick up objects.	The player picked up the objects.	Pass	
2.	The game will run smoothly without a discrete graphics card.	Performance testing	The game will function well on both levels smoothly.	Both levels are running smoothly now.	Pass	
3.	The cube, when interacted with, will load the next level, and put the player there.	Integration testing	The next level will load without issue. When the player interacts with the cube.	The next level is loaded when interacting with the cube.	Pass	 <p>Loads the player to the next level.</p>

User acceptance testing against requirements

Test No.	Test case	Type of Test	Expected outcome	Actual outcome	Result	Evidence
1.	There are multiple levels in the final game.	Acceptance testing	There will be more than one level in the final game.	There are two levels in the game, not including the main menu.	Pass	See submitted game and video.

2.	There is a life system and life deduction in the game.	Acceptance testing	The player will die if the timer runs out.	The player restarts from the beginning if the timer runs out.	Pass	See submitted game and video.
3.	Collision detection will be present.	Acceptance testing	The player will not fall through the world and be able to collide.	The collision detections is there and the player cannot fall through the floor.	Pass	See submitted game and video.
4.	There will be score and adding score.	Acceptance testing	The score will increase when the level progresses.	The player gets +1 score when going to level 2.	Pass	See submitted game and video.
5.	The game is suitable for 8+.	Acceptance testing	The levels will be safe for all ages and the puzzles will not be too simple to prevent older audiences playing.	The levels are designed that although literally dark, are not creepy or gory, making it suitable for 8+. The puzzles are solvable for all ages, although some will take a little more time.	Pass	See submitted game and video.

Analysis of test results

Basic test plan

Test 1

The initial test of the basic test plans has the test case “Player will be able to pick up objects”, which refers to the ability of the player being able to interact with objects, like a flashlight, and store them internally, allowing them to perform other functions/tasks, like turning on a flashlight.

This was a unit test as it was designed to isolate the picking up aspect of the game and ensure that it worked. This meant that this was the sole focus of the test.

The result of the test showed that unfortunately the object was not picked up and therefore it had to be resolved before I could continue with the other tests as this was integral part of the testing plan. This meant that the testing continued in the order displayed, however the initial test of the improved test plan occurred before the second test of the basic test plan.

Test 2

This test case includes a specific test case focusing on the player's ability to turn a flashlight on and off by pressing a button. The objective is to ensure that the player can interact with the flashlight and control its state effectively.

The test conducted for this test case is classified as unit testing. It aims to isolate and specifically test the functionality related to turning the flashlight on and off. The focus of the test is solely on verifying the correctness functionality of this specific feature.

The result of the test indicated a pass, as the player was able to successfully turn the flashlight on and off by pressing the designated button. This outcome confirms that the feature works as intended and meets the desired functionality. The successful execution of this unit test demonstrates the correctness of the flashlight's control mechanism.

Test 3

This test case involves testing the player's ability to move around, pick up the flashlight, and subsequently turn it on and off. The objective is to ensure that the player can perform these actions smoothly and without any issues.

The test conducted for this test case is categorised as element testing. It focuses on testing the individual elements or components involved in the described functionality, including player movement, object interaction (picking up the flashlight), and flashlight control (turning it on and off).

The result of the test indicates a pass, as the player was able to move around the game environment, successfully pick up the flashlight, and effectively turn it on and off as desired. This outcome confirms that the tested elements are functioning correctly and interact seamlessly with each other.

Test 4

This test case focuses on testing the interaction between the player inputting a password into a keypad and the subsequent opening of a door located next to it. The objective is to verify that both components are functioning together correctly.

The test conducted for this scenario falls under integration testing. It aims to evaluate the integration and interaction between the keypad and door components, each having separate scripts. The test ensures that when the player inputs the password correctly, the door opens as intended.

The result of the integration test indicates a pass, as the player successfully inputs the password into the keypad, triggering the opening of the adjacent door. This outcome confirms that the integration between the keypad and door components, with their respective scripts, is functioning correctly.

Test 5

This test case focuses on evaluating how well the game will perform without a discrete graphics card, and specifically focuses on the fact that the game should perform well. This means that the game should run optimally here.

This means that due to the focus on performance, the test is naturally a performance test. This aims to evaluate the game's performance under specific conditions, such as not having a discrete graphics card, and to understand how different parts of the game perform, like the frame rate, etc.

The result of this test was a fail, as the game struggled on the second level with loading the terrain that the player explores. The absence of a dedicated graphics cards heavily impacted the games performance and therefore makes it inaccessible to certain players. This will have to be rectified in the improved test plan.

Test 6

This test case focuses on the interaction with a cube in the game. When the cube is interacted with, it should load the next level and place the player in that level.

The test conducted for this scenario is categorised as element testing. It aims to specifically test the functionality related to the interaction with the cube and the subsequent loading of the next level with the player placed correctly. The purpose is to ensure that these elements work as intended and seamlessly transition the player to the next level.

Unfortunately, the result of the element test indicates a fail. Upon interacting with the cube, it did not load the next level as expected, and the player was not placed in the correct position within the level. This outcome suggests that there is an issue with the interaction mechanism, or the level loading functionality associated with the cube. It will require further investigation and improvement in the improved test plan to rectify this failure.

Test 7

This test case focuses on the player's ability to pick up a key and use it to interact with a door on the first level of the game.

The test conducted for this scenario is integration testing. It aims to evaluate the interaction and integration between the key and the door components, specifically testing their functionality when used together. The purpose is to ensure that the player can successfully pick up the key and interact with the door, allowing progression to the next level of the game.

The result of the integration test indicates a pass, as the player was able to pick up the key and successfully use it to interact with the door on the first level. This outcome confirms that the integration between the key and door components, in terms of their functionality and interaction, is working as intended.

Test 8

This test case focuses on the player's ability to pick up books and place them in a bookshelf within the game.

The test conducted for this scenario is element testing. It aims to specifically test the functionality associated with picking up books and placing them in the bookshelf. The purpose is to ensure that these elements work as intended, allowing the player to interact with the books and store them in the designated bookshelf.

The result of the element test indicates a pass, as the player was able to successfully pick up books and place them in the bookshelf within the game. This outcome confirms that the tested elements, including book interaction and bookshelf functionality, are functioning

correctly. The positive test result ensures a satisfying gameplay experience, allowing the player to engage in the book collection mechanic and progress the game.

Improved test plan

Test 1

Following the previous failed test, the updated test plan focuses on retesting the player's ability to pick up objects in the game.

The test conducted for this scenario remains as unit testing. It aims to specifically test the functionality related to picking up objects, considering the issues identified in the previous failed test. The purpose is to verify that the updated scripts have resolved the previous issue and now allow the player to successfully pick up objects.

The result of the unit test indicates a pass, as the player was able to pick up objects without encountering any issues. The updated scripts have addressed the previous problem, ensuring the expected functionality is now working correctly. This successful test outcome demonstrates that the issues identified in the previous failed test have been resolved.

Test 2

Following the previous failed performance test, the updated test plan focuses on retesting the game's performance without a discrete graphics card.

The test conducted for this scenario remains as performance testing. It aims to evaluate the game's performance under the updated conditions, considering the modifications made to address the previous performance issues. The purpose is to assess whether the adjustments, such as reducing foliage and polygons and stretching two remaining elements to fit the outside scene, have improved the game's performance.

The result of the performance test indicates a pass, as the game now runs smoothly without a discrete graphics card. The modifications made, including the removal of most foliage and polygons and the stretching of two remaining elements, have significantly improved the game's performance. These adjustments reduced the graphical load on the system, resulting in improved frame rates. This positive test outcome demonstrates that the game can now provide an acceptable user experience even without a dedicated graphics card. The successful performance test highlights the effectiveness of the implemented changes in enhancing the game's performance on systems without a discrete graphics card.

Test 3

Following the previous failed integration test, the updated test plan focuses on retesting the interaction between the cube and the loading of the next level.

The test conducted for this scenario remains as integration testing. It aims to specifically test the integration between the cube interaction and the loading mechanism, considering the fix implemented to resolve the issue identified in the previous failed test. The purpose is to validate that the updated scripts now correctly load the next level and place the player there upon interacting with the cube.

The result of the integration test indicates a pass, as the cube successfully triggers the loading of the next level, and the player is placed in the correct position within that level. The issue that caused the failure in the previous test, where the script did not specify the destination level, has been fixed. The updated scripts now correctly identify the intended level, allowing for progression through the game.

Evaluation

Testing quality

The test plan displays several strengths in ensuring the functionality of the game. One strength is the inclusion of targeted test cases that focus on crucial aspects of the game's functionality, such as object interaction, player movement, input validation, and performance. By specifically addressing these key features, the test plan aims to identify and resolve any issues or inconsistencies that may arise during gameplay. Additionally, the test plan exhibits a diverse range of test types, including unit testing, element testing, and integration testing. This approach allows for an extensive evaluation of different aspects of the game. This ensures that all crucial functionalities are thoroughly tested. By combining various test types, the test plan aims to cover a wide range of scenarios and potential issues, providing a better assessment of the game.

Adding clear expected outcomes for each test case is another strength of the test plan. By defining specific criteria for success, the test plan provides a clear objective against which the actual outcomes can be compared. This promotes a straightforward assessment of the test results, enabling testers and developers to determine whether the game's functionalities meet the intended requirements or if any adjustments are needed.

Furthermore, the implementation of the test plan, supported by agile software testing methodologies, contributed to enhancing the overall quality of the final game. The iterative nature of agile testing allowed for continuous refinement throughout the development process, resulting in a more polished game. By adopting an agile approach, the testing process was combined into the development cycle. This approach helped identify and address issues at an early stage, reducing the likelihood bugs making their way into the final game.

However, there are a few weaknesses that could be addressed to further enhance the test plan. One aspect is the limited information provided about the underlying implementation, test environment, and specific tools or frameworks used. If there was more information available about the tests and the details surrounding them, like when the test occurred, what was the software the test was conducted on, etc, this would offer a better understanding of the test conditions and therefore the test results, improving the quality and accuracy of the testing plan.

Information about the test environment can be considered critical in order to effectively evaluate the test results, with variables including the hardware and operating systems, among others. These significantly impact the games performance and functionality, therefore making them priorities when considering the testing environment and in the

future if the test plans could include this information, that would make it much easier to replicate these tests in the future.

Finally, including details about the tools and frameworks utilised during testing is crucial as it offers valuable insights into the chosen testing approach and methodology. This information plays an important role in accurately interpreting the test results and understanding any issues introduced by the testing tools themselves. Furthermore, documenting the specific tools and frameworks used promotes collaboration among testers and developers. By providing this information, it becomes easier for others to replicate the tests and compare their own results, creating a collaborative testing environment that promotes knowledge sharing and improvement.

To summarise, the evaluation of the test plan reveals both strengths and weaknesses in the testing approach. The various testing types, including unit testing, element testing, and integration testing, demonstrates an extensive approach to ensure different aspects of the game were thoroughly tested. The successful completion of several test cases showcases the effectiveness of the testing process in identifying and resolving issues, resulting in improvements to the final game. However, the lack of specific information about the circumstances of the tests, such as underlying implementation details, test environment specifics, and testing tools used, poses a weakness that limits the overall comprehensiveness and replicability of the tests. By addressing these weaknesses and providing additional context and details in future test plans, the overall quality and effectiveness of the testing process can be further enhanced.

Test plan improvements

Based on the evaluation of the test plan and the identified strengths and weaknesses, several improvements can be implemented to enhance future test plans.

Firstly, it is essential to provide detailed test environment information. Including specific details about hardware specifications, operating systems, and dependencies will ensure consistency and enable accurate replication of tests. By documenting and sharing this information, testers and developers can better understand the context in which the tests were conducted and compare results effectively.

Another significant improvement is to specify the testing tools and frameworks employed during the testing process. Clear communication about the tools used aids in understanding any limitations or biases introduced by those tools. Future testers can replicate the tests and compare outcomes using the same set of tools. Transparency in testing approaches fosters a more cohesive and productive testing environment.

Furthermore, including test coverage metrics is recommended. Test coverage metrics includes information that tells you how much the testing has covered, such as functionality, performance, and user experience. By measuring the percentage of coverage achieved, testers can identify any gaps in testing and prioritise areas that require further attention. This helps ensure that functionalities are tested, and potential risks are avoided.

Lastly, it is important to remember the iterative approach. Including feedback from previous test cycles, addressing identified issues, and improving the test cases based on lessons learned will contribute to the overall effectiveness of the testing process.