Software Quality Strategy.

Team: 14, Los Thunder Contribution:

- Salma: completed the introduction, communication, document sharing, task allocation, efficiency, compatibility, testing, learnability operability, accessibility as well structuring the document and checking over it.
- Alina: In the software quality strategy, I wrote about the maintainability and how we will maintain a good quality game throughout the duration of the project.
- Gugundeep: For the quality document, I contributed by stating the different risks which could arise to our game in the development process stage and the different types of security principles and practices we could use to stop them from occurring. During the final week of the deadline Salma helped me with reading what I had written and giving constructive feedback to what I could do to improve it.
- Ebrahim: Contributed to security principles and practices. I wrote about the design-first development and Heroku and lastly the potential threat of DDos attacks. All of these small areas go into detail on how they affected our game and how we use these to improve.

Table of Contents:

1. Introduction 2

2. Communication 3

3. Document sharing 3

4. Task allocation 3

5. Efficiency 4

6. Compatibility 4

7. Testing 4

8. Learnability 4

9. Operability 4

11. Security principles 5

11.1. Cross-site scripting 5
11.2. Integrity 5
11.3. Confidentiality 5
11.4. Availability 6
11.5. Server security 6

12. Maintainability 6

1. Introduction

This document will be highlighting our approach towards developing and delivering our system while certifying that it reaches a standard which is both good for the players and the developers. The document will also aid as a guideline to follow throughout the development process.

2. Communication

As a team we aim to track our progress through weekly meetings where all team members get the opportunity to voice concerns and share new ideas, as well as confirming what changes and progress has been made regarding other aspects of the project. After each meeting we keep minutes to provide written notes on what was discussed in that meeting as well as the tasks we need to complete before our next meetings, this allows all team members to be on the same page and for our understanding of the discussion to be consistent. It is important to highlight that when adding new open actions, we also allocate them during the meeting to identify the workload of each member as a means to ensure that each member is active and has an achievable workload for the week. The open actions list serves as a to do list and a point of reference for each member to help them complete the tasks logically.

The meetings that we have are audio calls as it is more effective to have real-time discussions in contrast to sending messages and receiving replies later throughout the day, this also improves our productivity as a team and means that we can conclude, or problem solve more methodologically.

3. Document sharing

Furthermore, a part of communication would be how documents are shared, each member of the team has access to all the documents which we are working on, one copy of each task (i.e., Java code for the game or HTML code for the website) is provided to avoid repetition and to allow us all to work collectively. Allowing access to all members of the task means that if there is ambiguity about a certain aspect of any of the development, they can easily reference the work to find out. For code sharing GitHub is being used, whereas for document sharing Google Drive has been the most efficient as it enables to see what changes have been made by who.

However, we have limited certain files to read-only access to avoid a member, who is not specifically working on said task, accidentally adds something and causing the code to potentially fail for instance.

4. Task allocation

For the tasks we strategically decided to split up in smaller sub-teams to work on specific parts of the software development, this was done to make certain that there was no disorientation as well as having the same members on the same task as they will know what needs to be done and how, in contrast to doing half of the task and then reassigning it to another member who hasn't worked on that specific task, as this may propose complication. It is important to note that although the members were assigned to different tasks, we're all still heavily involved in other tasks and able to provide help or advice throughout the progress, insomuch as the team always discussed minor developments in our weekly meetings allowing the team to have a high level of understanding and involvement in all the other tasks.

Moreover, when a member of the team is idle, which may happen if they complete their task or run out of tasks to do, they will be shuffled to another part of the team if they need extra support, this is to maximise the performance of a team as a whole.

5. Efficiency

In order to maintain the product quality, we recognize that the performance efficiency of the game will be of great importance, especially since it will be a multiplayer game which is run on a web server, which can therefore mean that if there are too many players on the server, it may crash or lag. However, we will not be looking into efficiency until we have a powerful basic code, and we encounter that the code runs slow at a certain time or point of the game. At this point we will look into utilizing a profiler but not before thoroughly testing the code before implementing these improvements. Once that is done, the VisualVM can be employed to allow us to identify where the memory is being used the most, the time complexity of chosen data structures and algorithms as well as the usage of particular features/instructions.

6. Compatibility

Considering that this game will only be available for desktop computers as well as laptops, there isn't much compatibility to recognize, however the software needs to be developed in a way that would allow forward as well as backward coding (newer versions can be released in a way that does not break existing producers). Moreover, there may be co-existence compatibility issues, therefore it may be favourable to have the play testing done on a personal computer, which will already have most systems and softwares that the target audience will have (security systems etc.) to make certain for the most part that there are little to no coexistence issues, although we still perceive that there may be some as we can't check all the possibilities of this kind of issue.

7. Testing

Several types of tests will be conducted to confirm the functionality and non-functionality of the overall product. To highlight, for the functionality testing we will manage unit testing, integration testing, black box and white box testing. For the non-functionality testing we will regulate load testing as this is a webserver game and avoiding crashes or lags is one of our main concerns and priorities, adding to this, we will also conduct performance testing.

Moreover, other static analysis tools may be used in this part of the process such as PMD and FindBugs when deemed necessary. However, as a team we have highlighted that testing each part or addition to the code is more than crucial as we rather deal with one error at a time in contrast to a large number of errors on one occasion even if we believe the logic behind a piece of code cannot be wrong. We will also aim to have the code tested by other members of the team to verify that the code is tested without any bias, and on the occasion where it does fail that it fails as expected. Lastly, after the game is tested for functional and non-functional perspective, it will then be tested by a gamer, preferably of the target audience selection, to see if the game is appealing in the sense of its enjoyment and engagement.

8. Learnability

For the learnability aspect of the game, there will be a help button on the launching screen which will consist of the game rules, and instructions on the controls, i.e., how to move. Seeing that the nature of the game is straightforward and that the users do not have an account to keep track of their level and can choose which level they want to play; we do not require an introduction level game, however the first level within the game is extremely is, both in terms of obstacles and questions, this means that new players will understand the goal of the game and gain an understanding as well as experience by playing the first level.

9. Operability

The game will operate with the directional keys to navigate through the map, i.e., to move the character, while the mouse will be used to click the answers when faced with a question.

10. Accessibility

To make the game accessible to visually impaired players, the already existing features of the game make it accessible to those with hearing difficulty, as for instance, the screen will have a message stating that the countdown has started, or that the player has consumed a power-up and thus has x seconds until it wears off.

11. Security principles

Moving on, in relation to the software quality, we have considered many threats which could arise within our game in relation to the security during the software development processes.

11.1. Cross-site scripting

We recognise that our software may be prone to “cross-site scripting”, this is a client-side code attack, which aims to remove scripts in a browser of the user who is being attacked by using code which is malicious on a web page. The attack occurs when the victim uses the web-side, which of course has been identified as a possible security threat since the game will be web server based. To avoid this threat, we’ve highlighted to ensure the programming of code used to prevent such attacks. In addition to this, another to prevent this type of threat is to sanitize the code. If a user would need to use a programming language, then this would not work as they would essentially break the valid tags, therefore it is important to use a specific library depending on the developing language.

11.2. Integrity

Moreover, integrity is another highly crucial aspect seeing as the number of errors should be minimal, this takes place when implementing the code for the game. Using the correct practices during the development can help achieve safety, security and help with the maintainability of the program. This can be accomplished by thoroughly following code standards guidelines, in order to make certain that the program is secure from cyber-attacks, reliable, testable, maintainable, portable during all stages of development. This will naturally be achieved by regular testing. As a team we have realized that making sure we test the code frequently will help us in relation to the overall result of the game in regard to preventing receiving future cyber-threats.

Additionally, to certify that we have fully taken into account the integrity issues during the design we have concluded a method to deploy which will aid in ensuring that issues such as integrity of the software design are not faced. The approach which was appropriate was the “design-first development” method, this is a prototype of code which can be used further down the line when advancements in the code take place and issues need to be resolved. These issues may be far back in the code meaning that you could use the prototype and make amendments to see whether the new idea of code works properly. It is important to note that the date is not tampered or used by users who are not supposed to be using it.

11.3. Confidentiality

The second part is confidentiality, similar to privacy and extremely critical in terms of our project. Confidentiality will enable us to the limit of access to information which is stopped by rules. There are measures which need to be undertaken to make sure that the confidentiality is used to prevent and stop sensitive data from reaching the wrong people, in this case that is attackers.

11.4. Availability

Another principle to consider is availability, this is making sure that during the development or even after, we are allowing the game to be available for all individuals, this will in turn mean making sure the program is tested for any bugs or faults, allowing it to become easy to play at any times after production. During the production if the code gets removed or accidentally deleted then making sure we have a back-up of code is imperative; this is especially important in terms of the production deadlines. In terms of the practices again it is crucial that data loss is prepared for by ensuring that there is a data recovery plan to be able to continue the production or development without it coming to a halt, as mentioned previously, this means keeping a backup of the latest version of the code at all times.

11.5. Server security

In addition, Heroku was used as a good security practice. To elaborate, Heroku enables the deployment and execution of Java, PHP, and other programming languages. it offers a framework for the game to be played on a live server. The main logic as to why Heroku was utilized in the development was that it defends against threats by implementing security controls at any layer, from physical to device, isolating data and applications, and deploying rapid security updates without requiring user interaction or service disruption.

11.6. DDos attacks

Moreover, another security risk that was recognised were DDoS attacks, which occurs when multiple computers are used to flood the target IP address, considering that our game is a multiplayer game, it had to be made certain that to make sure we took this into consideration, there are ways in which a user of the game will be able to find out if they are a victim and there are many solutions to this such as resetting the IP address or requesting a new IP address. There are several ways to avoid a DDoS attack. The first is to establish a DDoS attack response plan. Getting a response plan in place in case an attack happens is critical; the plan would outline how to keep the game going after the attack is successful. A network vulnerability evaluation is another way to avoid a DDoS attack. It's critical to spot a flaw in your network before a malicious user does; this assessment entails finding security vulnerabilities so we can be better prepared for future cyber-attacks.

12. Maintainability

We will maintain a good quality game throughout this project by making certain that the team is working well together, and everyone is completing their tasks on time. We will do this by maintaining good communication between all team members and by dividing tasks to spread the workload equally. The quality of the work will remain at a high standard throughout the project since everyone will participate and help each other out with any difficulties we come across. Each team member will allocate a certain amount of time to complete their work with ease and without any distractions. This will enable everyone to have enough time to finish their tasks and make sure it is up to a good standard. The work may also be checked by other members who are allocated to other tasks as a means to give insight to members who have been continuously working on the same aspect, this will allow for more errors to be caught out and in turn push for a higher quality software development.