Non-Functional Requirements

Table of Contents

List of Non-Functional Requirements

1. **Graphical User Interface**
    a. The application must have an easy-to-navigate Graphical User Interface (GUI).
    b. The GUI will use bright colours and soft edges to appeal to the younger target audience.
    c. The GUI will be well explained, and purposely made to be navigatable by students of the age group.
    d. The GUI is very user-friendly to make it easy for kids in that age range to use.
    e. The content will be aimed at middle school students students.
    f. The UX design patterns must be consistent throughout the project.
    g. This interface must be user-friendly and align with best UX practices to ensure intuitiveness and ease of navigation.
    h. The application will run efficiently and not use excessive computer resources.
    i. The interface will be responsive and not freeze during the game.
2. **Data Storage**
    a. The application will store all data locally, including analytical information used in the creation of the levels.
    b. All data are stored locally in separate files (e.g. .json, .csv, .xml, .tsv)
    c. The file size of the entire project is less than 1 GB (gigabyte).
3. **Development**
    a. The Educational Game will be developed in Java 19 (or newer) using the Eclipse development environment with an object-oriented approach.
    b. The code will be well-tested to ensure the program is robust and can't be broken by the user.
    c. The coding style is consistent to ensure the code is easy to read and debug.
    d. The code is thoroughly documented, up to the standards presented in the CS2212B class.
    e. The code will be well structured, allowing for easy updates and maintenance.
    f. The application will be executable on a Windows 10 system and will be executable on any system that has Eclipse downloaded.
    g. All project code and files must be stored on BitBucket Git, at least weekly.
    h. All design work and diagrams must be stored and developed on Confluence.
    i. All design work and diagrams must be tracked and updated on Jira.
    j. Every user action will present a visible response by the application.
    k. The application will give an error message if the action is not something that it recognizes.
    l. All code must be commented on using Javadoc (author, purpose, class description, method description, etc.).
    m. All code generated by external tools must be cited as in-line comments.
    n. All code must be unit-tested on JUnit5 (except GUI elements).
    o. All code must follow a common coding convention/ style (naming, indentation, etc.).
    p. All members must use the same IDE and tools, as in the team contract.
    q. All files created by the app must be contained within the app's directory (i.e. the app does not modify any parent directories or subdirectories).
    r. The game's content, including code, documentation, files, etc., will be written in English entirely.
    s. All code must be easy to maintain/ update.
    t. All third-party libraries must be free.
4. **Game Design**
    a. The game will teach middle school students students the fundamentals of coding. Loops, variables, logic, and the basics of path-solving algorithms will be taught.
    b. The game will use code "blocks" to simplify the coding part for the students, as the blocks will act as pseudocode to allow the students to understand it better.
    c. The instructions must be understandable by the target age group of middle school students.
    d. The game has intuitive rules and systems for gameplay interactions and progression tracking.
    e. The puzzles must be solvable by the target age group.
    f. The game shall be educational, i.e. effective in teaching and entertaining, not quiz-like.
    g. The game has accurate, relevant, up-to-date, and appropriate educational content.
    h. The game's educational content (i.e. instructions) is sourced from credible sources.
    i. The game has a cohesive educational theme in all stages.
    j. The game should provide feedback to students on their progress and performance, helping them identify areas for improvement and reinforce their learning.