2.5 Non-Functional Requirements

This sub-page will list and number the non-functional requirements of the software project (in essence, how you and the software will go about doing things). Note that much of this information can be paraphrased from the original project specification provided to you. Again, at least some rewording will be necessary in doing this; as discussed above.

1. **Programming Language and Environment:**
    - The application must be developed in Java 19 or newer.
    - The application must be executable on a Windows 10 system with a standard Java installation.
    - Each team member must be able to compile and run the application from their development environment.
2. **Design and Architecture:**
    - The game must utilize an object-oriented approach and incorporate design patterns.
    - The codebase should adhere to coding conventions and styles agreed upon by the team.
    - Code should be structured for maintainability and reusability.
3. **User Interface:**
    - The application must have a Graphical User Interface (GUI) that is user-friendly and aligns with best User Experience (UX) practices.
    - Accessibility considerations should be made for users with disabilities, including support for keyboard and mouse interaction, logical tab order, and colorblind-friendly UI design.
4. **Data Management:**
    - All data, including save states and high score tables, must be stored locally.
    - Data storage formats such as JSON, XML, CSV, or TSV are recommended.
    - If using any third-party libraries for data storage, it needs to be freely available and easy to obtain.
5. **Development Tools and Processes:**
    - The NetBeans IDE should be used and the project should be packaged as a Maven Project.
    - Bitbucket Git repository must be actively used for version control.
    - Confluence should be utilized for design work and diagrams.
    - Jira must be used for task and issue tracking.
    - Javadoc must be used for code documentation.
    - Unit testing should be performed using JUnit 5 or equivalent for the majority of the code.
6. **Performance and Efficiency:**
    - The application should run efficiently and utilize computing resources effectively.
    - It should be responsive and not freeze or lag during gameplay.
7. **System Integrity and Security:**
    - The application must be well self-contained and not modify files outside of its directory.
    - Error handling should be implemented to provide clear and informative messages for erroneous user actions.
8. **Size and Resource Limitations:**
    - The file size of the project as a whole should be under 1 gigabyte.
9. **Language and Communication:**
    - All content within the application, documentation, and communication among team members should be in English.
10. **Software Engineering Principles:**
    - The application must be designed with sound software engineering principles in mind, as discussed in the course.