# (temp) Requirements Checklist

Note:

- This checklist is **NOT** a replacement for the Requirements Documentation!
- This checklist may contain outdated information!
- This checklist is for **internal use** only!

**Version History**

| Version | Date | Author(s) | Summary of Changes |
|---|---|---|---|
| 0.1 | 07 Feb 2024 | Chun Ho Chan | Page created |

**Table of Contents**



## References

- Course Outline
- Project Specification v1.0

## 1) Functional Requirements ???

### 1.1) User Interface (GUI)

- [x] 2D graphics
- [x] Has mouse input for accessing menus and gameplay
- [x] Has alternative keyboard shortcuts for accessing menus and gameplay
- [x] Has a logical Tab order for UI elements
- [x] Has a visual feedback system
- [x] Has a colourblind mode
- [x] Display game state
- [x] Respond to user inputs
- [x] (optional) Has an audio feedback system
- [x] (optional) Animated Text/ Images

#### Screens

- [x] Main Menu screen
- [x] Gameplay screen
- [x] Instruction/ Tutorial screen
- [x] High Score Table screen
- [x] Progress/ Result screen
- [x] Teacher Mode screen
- [x] Debug Mode screen

- [x] (optional) Settings screen
- [x] (optional) Loading screen

### 1.2) Main Menu screen

- [x] Text: Title of the game
- [x] Image: Logo/ Banner of the game
- [x] Text: Credits: members, team ID, course term, CS2212 UWO.
- [x] Button: Create a new game
- [x] Button: Load a saved game
- [x] Button: Go to the Instruction/ Tutorial screen
- [x] Button: Go to the High Score List screen
- [x] Button: Exit the game

### 1.3) Instruction/ Tutorial screen

- [x] Can be text, image, or interactive level
- [x] Detailed instructions on how to play the game
- [x] Detailed instructions on mouse/ keyboard controls
- [x] Summary of what the game teaches, what players will learn, major features
- [x] UX: instructions must be understandable by the target age group

### 1.4) Gameplay and Mechanics

- [x] Intuitive rules and systems (for player interactions and progress)
- [x] Has a scoring system (reward player solutions)
- [x] Has a progression system (unlock new content based on score)
- [x] Has at least 3 levels of difficulty (e.g. 3 stages or 3 difficulty modes)
- [x] Has educational puzzles (with the same theme) in every stage
- [x] UX: puzzles must be solvable by the target age group
- [x] Perform in-game actions with player key presses
- [x] Give per-stage feedback on the player's solution's correctness
- [x] (optional) Timer-based challenges
- [x] (optional) Life system (i.e. remaining number of attempts)
- [x] (optional) Button: Go to the High Score List screen when a stage ends

### 1.5) Save & Load Game

- [x] Has a checkpoint Save system (independently for each player)
- [x] Save must include: player name, score, and checkpoints reached
- [x] Save multiple game files (slots/ file system)
- [x] Text: Confirmation message for saving
- [x] Has a checkpoint Load system (independently for each player)
- [x] Load multiple game files (slots/ file system)
- [x] Text: Confirmation message for loading
- [x] Has checkpoints that save the game (i.e. stage number)
- [x] (optional) Button: Save the game manually at any time
- [x] (optional) Save the game automatically using timers
- [x] (optional) Save temporary game data as well (e.g. characters in a text box)
- [x] (optional) Delete a saved game

### 1.6) High Score Table (local)

- [x] Display a list of top entries: player names (full/ initials), and scores.
- [x] Display at least 5 entries (including null entries)
- [x] Auto updates when new entries are created (e.g. auto scan all save files? or store the list in a separate file (portability issue, may overwrite another PC's list)?)
- [x] Can keep the list between restarts
- [x] Sorted in decreasing order (highest score at the top)
- [x] UX: Clear visual separation between entries (e.g. border for each entry)
- [x] (optional) UX: A way for teachers to calculate the score difference between any two entries (e.g. a third column displaying the score difference between a selected entry to all other entries, with the top score selected by default?)

### 1.7) Multiple Players (local, asynchronous)

- [x] Prompts and stores current player's name/ initials/ username/ email.
- [x] (optional) Account Login system

### 1.8) Teacher Mode

- [x] Protected by a password (hardcoded)
- [x] Displays each player's name, score, current unsolved stage, and all solved stages.
- [x] (optional) Display each player's number of attempts, time spent on each stage, aggregated statistics of all players, etc.

### 1.9) Debug Mode

- [x] Protected by a password (hardcoded)
- [x] Can access all stages and skip any puzzles
- [x] Can manipulate game data (e.g. score, lives remaining)
- [x] (optional) Can activate other debugging tools

### 1.10) Error Handling

- [x] Can exit the game cleanly (to the main menu)
- [x] Can exit the app cleanly (to desktop)
- [x] Can save data correctly on app exit
- [x] Can load data correctly on app start
- [x] Can scale window size correctly (or lock window size)
- [x] Can minimize window correctly (or disable this)
- [x] Can maximize window correctly (or disable this)
- [x] Does not crash unexpectedly when an error occurs
- [x] Prompt the player when an error is encountered (e.g. invalid input)
- [x] (optional) Other quality-of-life UX features

## 2) Non-Functional Requirements ???

- [x] Be educational, i.e. effective in teaching
- [x] Be entertaining, i.e. engaging to play (not a quiz)
- [x] All members must **contribute to all aspects** of the project
- [x] Discussed in weekly group meetings (under 20 minutes)
- [x] Discussed in weekly TA meetings
- [x] All content, documentation, design work, and comments must be in **English**
- [x] The file size of the entire project is **less than 1 GB** (gigabyte)
- [x] Has no lag at all times

- [x] Developed in Java 19 (or newer)
- [x] Use an object-oriented approach
- [x] Use and explain design patterns (Test-Driven Development?)
- [x] Has accurate, relevant, up-to-date, appropriate educational content
- [x] Educational content (i.e. instructions) must be sourced from credible sources
- [x] Has GUI with good UX practices
- [x] Cohesive educational theme in all stages
- [x] Cohesive visual language in all UI
- [x] Can store all data locally in separate files (e.g. .json, .csv, .xml, .tsv)
- [x] All third-party libraries must be free!!! (provide installation instructions for TA)
- [x] All project code and files must be stored on **BitBucket Git**, at least weekly
- [x] All design work and diagrams must be stored and developed on **Confluence**
- [x] All design work and diagrams must be tracked and updated on **Jira**
- [x] All code must be commented on using Javadoc (class + method comments) (author, purpose, etc.)
- [x] All code generated by external tools must be **cited** as in-line comments
- [x] All code must be unit-tested on **JUnit5** (except GUI elements)
- [x] All code must follow a common coding convention/ style (e.g. naming, indentation)
- [x] All members must use the same IDE and/ or tools, as in the contract
- [x] 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)
- [x] All code must be easy to maintain/ update
- [x] All audio must follow sound software engineering principles
- [x] Has gathered, refined, and documented additional non-functional requirements related to the game idea and educational objective(s)

## End of Checklist

wow, you read all these…