Requirements and Design Document

Notes

- User stories = functional requirements
    - Recall: functional requirements describe what the system should do

Project Sprints

Sprint	Dates	Goals
Sprint 1	Oct. 20 - Nov. 3	Core functionality (movability, basic UI, navigation between sections)
Sprint 2	Nov. 3 - Nov. 17	Data management and initial implementation of mini-games
Sprint 3	Nov. 17 - Dec. 1	Polishing features (help button, missed questions, etc.), game fine-tuning

User Stories

Game Functionality

As a user, I want to be able to move around so that I am able to explore the map. *(6 hours)*

- Design an alien graphic to function as the main character sprite *(2 hours)*
- Design a character class with attributes (character type, movement speed, etc.) and event listeners to implement a WASD movement system *(2 hours)*
- Write tests *(1 hour)*
- Pass tests *(1 hour)*

As a user, I want to be able to get help to learn the topics I don't already understand or need to work on. *(11 hours)*

- Design a "Need Help" button graphic *(1 hour)*
- Design a popup system for help button *(2 hours)*
- Create videos or graphics for each class of problem *(5–8 hours)*
- Write tests *(1 hour)*
- Pass tests *(1 hour)*

As a user, I want to be clearly informed when certain games or activities (like rapid-fire math) cannot be redone, so that I can understand when incorrect inputs have consequences. *(4 hours)*

- Create a popup graphic for non-repeatable activities *(2 hours)*
  Add the popup graphic to the game where needed *(2 hours)*

As a user, I want to be prompted when mini-games happen and what they are so that I can be prepared when it comes. *(6 hours)*

- Create a popup graphic for incoming games or activities *(2 hours)*
- Add the popup graphic to the game where needed *(2 hours)*
- Write tests *(1 hour)*
- Pass tests *(1 hour)*

As a user, I want to be able to pause, resume, go back to the main menu, and see my current progress so I can control my gameplay. *(6 hours)*

- Implement an event listener for a pause/menu popup *(1 hour)*
- Design a menu popup graphic *(2 hours)*
  Write and pass tests *(2 hours total)*

Game State

Prerequisite: Data management is set up

Review mistakes mode

As a user, I want to review the questions I get wrong with correct answers shown so I can understand my mistakes.

- Store all questions and answers presented to the player *(1–2 hours)*

As a user, I want to see brief explanations for why an answer is correct so I can learn from my mistakes.

- Store explanations for each question *(3–4 hours)*
- Implement on.click() or on.hover() handlers to display explanations *(1–3 hours)*

As a user, I want a clear summary screen after each planet showing which questions I answered incorrectly.

- Design a summary splash screen

Retry Missed Questions mode

As a user, I want to retry the questions I missed so I can improve my score.

- Implement a retry button *(1–2 hours)*
- Fetch missed questions and answers, allow users to try again *(2–3 hours)*

As a user, I want the game to track whether I answered retried questions correctly so I can see my improvement.

- Integrate with data management to fetch and compare answers *(1–3 hours)*

Graphics

As a user, I want to see on-screen indicators showing which buttons to press (e.g., Space to continue, Enter to go back) so I always know how to progress through the game. *(5-7 hours)*

- Hard-code indicator types (moving, answering, etc.) *(3–4 hours)*
- Allow other classes to access and display them *(1–2 hours)*
- Display Testing *(1 hour)*

As a user, I want to see a main display so I can see my progress. *(3–4 hours)*

- Design a map including planets and a character. Map includes: an astronaut, 4 planets, menu bar *(2–3 hours)*
- Display Testing *(1 hour)*

As a user, I want to see my main gameplay display so I can begin to answer questions. *(3–4 hours)*

- Design a mini-map showing user's checkpoints on the planet. Mini-map includes: back indicator, checkpoints button, planet in detail graphics, character on the checkpoint. *(2–3 hours)*
- Display Testing *(1 hour)*

As a user, I want to see a question and answer display so I can see the question and solution after answering the question. *(3–6 hours)*

- Design an interactive question pop up when users click into the level checkpoint on the planet. *(1–2 hours)*
- Design an answer display prompting whether users are correct or incorrect then giving out the correct answer below. *(1–2 hours)*

- Main game logic and display testing *(1–2 hours)*

UI & UX

As a user, I want all texts, buttons, and panels to follow the same scheme and design so that the game looks cohesive and professional. *(2–3 hours)*

- Create a global theme manager for dynamic updates
- Create UI factory functions for reusable components
- Add a centralized configuration file (fonts, colors, etc.)

As a user, I want to be able to interact with objects so that I have a more engaging game experience. *(4-6 hours)*

- Use event handlers (e.g., on.click()) and animations for interaction. *(3–4 hours)*
- Interactive Components Testing *(1–2 hours)*

Audio

As a user, I want to hear game audio (music, action sounds) so that I have a more interactive experience. *(4–5 hours)*

- Create a folder of audio clips for various actions
- Allow event handlers to trigger corresponding sounds

Data Management

Progress Tracking

As a user, I want to see which planets I've completed and which are locked so I can track progress. *(4–6 hours)*

- Create progress.json structure
- Display planet icons with lock/unlock visuals
- Add condition for 80% accuracy unlock

As a user, I want the game to save my highest accuracy per planet so I can beat my personal record. *(3–4 hours)*

- Add bestAccuracy field to localStorage
- Compare current vs. saved scores
- Display best score on planet select screen

As a user, I want a summary at the end of each planet showing my accuracy, correct answers, time taken, and a recap if I perform poorly. *(6–8 hours)*

- Create SummaryScreen UI in Konva
- Implement timer tracking
- Add recap text if <80% accuracy

Game State

As a user, I want my scores to save automatically so I can return where I left off. *(3–5 hours)*

- Implement autoSave() on state change
- Store data in localStorage with error handling

As a user, I want to continue my journey from the same planet when reopening the game. *(3–4 hours)*

- Read saved progress and load last planet

As a user, I want to enter my username at the start of the game so my progress is personalized. *(3–5 hours)*

- Create username input screen
  Save username to localStorage

As a user, I want my username to be remembered between sessions so I don't have to re-enter it every time. *(2–3 hours)*

- Auto-load last used username
- Add "switch user" button

Question and Content Management

As a user, I want questions randomized each session so gameplay feels fresh. *(2–3 hours)*

- Implement shuffle function for question array
- Add unique seed per playthrough

As a user, I want the game to track which questions I missed so I can review them later in a Retry mode. *(3–4 hours)*

- Add missedQuestions[] array per planet
- Log question IDs of incorrect answers

Scoring and Accuracy

As a user, I want to see which questions I got wrong so that I know what to work on. *(4–5 hours)*

- Add “Review Mistakes” overlay
  Store wrong question data temporarily

As a user, I want to see my overall score and accuracy after each planet and mini-game so I can track improvement. *(3–4 hours)*

- Implement scoring formula (correct / total * 100)
- Show score at summary screens

Analytics & Feedback

As a user, I want to view a performance dashboard summarizing accuracy by topic so I can see what I’ve learned. *(5–7 hours)*

- Create dashboard canvas (bar chart using Konva)
  Fetch average accuracy per topic

As a user, I want a ‘Review Mistakes’ option showing incorrect answers with explanations so I can learn from errors. *(6–8 hours)*

- Load missedQuestions[] data
  Display question, user’s answer, correct answer, and hint

Game Logic

Mercury – Add, subtract, multiply, divide small integers *(6–9 hours)*

- **As a user, I want to practice small-number arithmetic to improve basic math skills.**

Venus – Same as Mercury, larger integers *(3–5 hours)*

- **As a user, I want to practice operations with larger numbers to strengthen my arithmetic foundation.**

Earth – Add/subtract time *(6–8 hours)*

- **As a user, I want to practice time arithmetic to apply math in real-world contexts.**

Mars – Prime numbers and factors *(6–8 hours)*

- **As a user, I want to identify primes and factors to build number theory understanding.**

Jupiter – Patterns and sequences *(4–6 hours)*

- **As a user, I want to find the next number in a sequence to recognize arithmetic/geometric patterns.**

Saturn – Metric conversions *(5–7 hours)*

- **As a user, I want to convert between metric units to practice measurement skills.**

Uranus – Fractions with like denominators *(6–8 hours)*

- **As a user, I want to add/subtract fractions with like denominators to master foundational fraction skills.**

Neptune – Fractions with unlike denominators *(7–9 hours)*

- **As a user, I want to combine fractions with unlike denominators to strengthen my fraction problem-solving.**

Mini Game Logic (Melanie)

Asteroid Factor Field *(11–14 hours)*

- **As a user, I want to control a spaceship that blasts asteroids with correct factors/multiples so I can apply what I learned on Mars.**

Fuel Fractions *(10–13 hours)*

- **As a user, I want to solve fraction problems quickly to refuel my ship before time runs out.**
    - Add streak counter and reward animations

Rapid-Fire Math Questions (Finale)

- **As a user, I want to answer timed math questions to test my accuracy under pressure.**