

Problem Statement and Goals

Software Engineering

Team #25, The Crazy Four

Ruida Chen

Ammar Sharbat

Alvin Qian

Jiaming Li

22.09.2025

Table 1: Revision History

Date	Developer(s)	Change
9.20	Jiaming Li	Goals and Stretch Goals
9.22	Jiaming Li	Separating FG and NFG
9.22	Jiaming Li	Reflection
9.22	Ammar Sharbat	Problem Statement

1 Problem Statement

1.1 Problem

The prevailing use of the decimal number base (base-10) is rooted in historical and cultural conventions rather than mathematical utility. Decimal relies on prime factors 2 and 5, which limits the number of fractions that terminate cleanly, leading to recurring decimals for simple ratios such as one-third or one-sixth. This complicates arithmetic, measurement, and teaching, especially when working with fractions and divisibility. By contrast, the dozenal number base (base-12) provides superior utility, as 12 has four small divisors (2, 3, 4, 6). These divisors allow for more efficient and intuitive representation of fractions, aligning better with practical subdivisions historically used in trade and measurement. Despite its advantages, dozenal remains marginalized, leaving students, educators, and professionals constrained by an inferior numeric system.

1.2 Inputs and Outputs

The problem can be characterized in terms of abstract inputs and outputs. The **inputs** include the numeric systems in use today (decimal as the default, and dozenal as an alternative), mathematical tasks requiring division into fractions, and cultural and educational assumptions that reinforce base-10 usage. The **outputs** of interest are representations of numbers, ease of computation, and the pedagogical clarity with which fractions and ratios can be expressed. Decimal often outputs cumbersome repeating decimals for simple divisions, whereas dozenal outputs concise terminating forms. A solution would aim to clarify these differences, provide comparative demonstrations, and yield recommendations for integrating dozenal into education and practice. Ultimately, the expected output is a stronger case for adoption or at least parallel usage of dozenal alongside decimal.

1.3 Stakeholders

Several groups are directly affected by the shortcomings of the decimal base and stand to benefit from exploring dozenal alternatives. **Mathematicians**, particularly dozenal enthusiasts such as Dr. Michael Rapoport, are invested in advancing the theoretical and practical merits of base-12. **Students**, like ourselves, are key stakeholders because clearer fraction representations can simplify learning and reduce confusion. **Educational institutions** shape curricula and determine whether alternative bases are introduced, giving them an influential role in adoption. **Computer scientists** also stand to gain, as they already work with multiple bases (binary, octal, hexadecimal) and could leverage dozenal in certain contexts. Finally, the **general public** is indirectly affected: tradespeople, engineers, and everyday users of measurement systems may find practical utility in dozenal through clearer subdivisions and easier mental arithmetic.

1.4 Environment

The environment for this problem encompasses technical, educational, cultural, and historical dimensions. Technically, decimal dominates current systems, from calculators to international standards like SI units, creating an environment resistant to alternatives. Educationally, curricula overwhelmingly emphasize decimal, leaving little room for exposure to dozenal or other bases. Culturally, societies are entrenched in base-10 due to “ten fingers” counting and legal standardization, despite persistent historical use of dozenal in commerce (dozens, grosses, hours). From a computing environment perspective, multiple numeric bases already coexist—binary and hexadecimal are routine for computer scientists—demonstrating that the mental and practical adoption of alternative bases is possible. Historically, dozenal has proven convenient in measurement systems, suggesting its potential remains untapped. The environment is therefore both a challenge and an opportunity for reconsidering numeric bases.

2 Goals

2.1 Minimum Viable Product (MVP) Goals

2.1.1 Functional Goals

Table 2: MVP Functional Goals

Goal	Explanation	Reasoning
Two-Player Core Loop	Support a one-versus-one match with turn-taking, drawing, discarding, and win checks. Includes starter card, discard pile, and reshuffling the stock pile.	Two players are the smallest playable unit; finishing this proves the core loop.
Classic Rules Engine	Implement standard Crazy Eights: match by suit or rank; an “8” is wild and lets the player declare a suit; draw if no valid move exists.	Correct classic rules establish a solid baseline before adding variations.
Dozenal (Base-12) Scoring/Display	Display scores, counters, or thresholds in base-12 notation while keeping classic rules unchanged.	Introduces dozenal in a simple, non-disruptive way that highlights novelty while retaining accessibility.

2.1.2 Non-functional Goals

Table 3: MVP Non-functional Goals

Goal	Explanation	Reasoning
Move Validation and Feedback	Provide immediate invalid-move feedback, clear suit selection UI after playing an “8,” and real-time state indicators.	Reduces errors and learning curve; improves usability.
Testability and Determinism	Support seeded shuffling and provide basic logs or replays for each session.	Enables reproducible unit/integration testing and debugging.
Stability and Performance	Ensure responsive UI $< 200\text{ ms}$, no crashes, no deadlocks, and correct reshuffling.	Reliability is the baseline for acceptance and live demonstration.
Minimal UI	Provide a desktop or web interface showing hand, discard pile, current state, and a dozenal score tracker.	Covers essential interactions while limiting complexity at the MVP stage.

Table 4: MVP Non-functional Goals

Goal	Explanation	Reasoning
Move Validation and Feedback	Provide immediate invalid-move feedback, clear suit selection UI after playing an “8,” and real-time state indicators.	Reduces errors and learning curve; improves usability.
Testability and Determinism	Support seeded shuffling and provide basic logs or replays for each session.	Enables reproducible unit/integration testing and debugging.
Stability and Performance	Ensure responsive UI, no crashes, no deadlocks, and correct reshuffling.	Reliability is the baseline for acceptance and live demonstration.
Minimal UI	Provide a desktop or web interface showing hand, discard pile, current state, and a dozenal score tracker.	Covers essential interactions while limiting complexity at the MVP stage.

2.1.3 Non-functional Goals

3 Stretch Goals

3.1 Functional Stretch Goals

Table 5: Functional Stretch Goals

Goal	Explanation	Reasoning
3–4 Player Matches	Extend gameplay to three or more players with clear turn rotation and visualization.	Increases replayability; closer to common play.
Online Multiplayer	Create/join rooms and synchronize game state; provide basic reconnection.	Demonstrates system design capability and supports real usage.
AI Opponent	Computer-controlled players with simple heuristics for card choice and suit declaration.	Enables single-player testing and showcases extensibility.
Advanced Dozenal Variants	Optional rule packs (e.g., effects for 12-related cards; base-12 scoring thresholds such as 60_{12}).	Deepens the dozenal theme while keeping the classic mode intact.
Rule Configurator	Toggles for house rules (draw-until-playable, stacking eights, scoring methods in decimal or dozenal).	Shows variability management; supports experimentation and future product-line thinking.

3.2 Non-functional Stretch Goals

Table 6: Non-functional Stretch Goals

Goal	Explanation	Reasoning
Tutorial and Visual Guidance	First-game tutorial, invalid-move highlights, and play suggestions.	Lowers the learning curve and improves usability.
Save/Replay System	Save completed games and replay event sequences.	Aids debugging, user study, and documentation.
Cross-Platform Packaging	Deploy as a web app or desktop executable (Windows/macOS).	Lowers the barrier for evaluators and external users to try the system.

4 Extras

Appendix — Reflection

1. What went well while writing this deliverable?

Our team collaborated effectively to define the scope of the project. We were able to clearly separate functional goals (gameplay features) and non-functional goals (quality attributes) after reviewing the lecture guidance. The use of LaTeX tables also helped us present goals in a structured and consistent format. Team communication through version control (Git) went smoothly, and everyone contributed ideas.

2. What pain points did you experience during this deliverable, and how did you resolve them?

One major challenge was formatting issues in LaTeX, especially with long text wrapping inside tables and ensuring that columns aligned properly. We also struggled at first with distinguishing between functional and non-functional goals, since some goals could be interpreted as both. These issues were resolved by researching LaTeX table packages, adjusting column widths, and reviewing lecture notes and examples for proper classification. Another pain point was Git conflicts during merges, which we resolved by carefully rebasing and keeping frequent communication.

3. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?

Initially, we considered including multiple advanced rule variants and a fully-featured online multiplayer system in the MVP. However, we realized this would be too ambitious for the timeline. Instead, we limited the MVP to a two-player local game with classic rules and base-12 scoring. More complex features, such as AI opponents and online multiplayer, were moved to the stretch goals. This adjustment ensures that the MVP is achievable while still leaving room for meaningful extensions to demonstrate complexity.

4. How did you handle version control and collaboration challenges?

At first, there was some confusion with branching and merging in Git. By agreeing on a workflow (feature branches merged into main through pull requests) and keeping commit messages clear, we reduced conflicts and improved traceability of changes.

5. How will this deliverable guide your next steps in the project?

The problem statement and goals document provides a shared roadmap for the team. It will serve as a reference point when making design decisions and prioritizing features, especially if scope adjustments are needed later in the term.