

# **Software Verification & Validation Specification Document**

## **Title: Memory Card Game**

Carolina Garza, Luca Gismera, Henny Sánchez

## **1. Project Overview**

### **1.1 Description**

A command-line implementation of the classic Memory card matching game where players flip pairs of cards to find matches. The game supports 1-2 players with three difficulty levels.

### **1.2 Game Rules**

- Cards are laid face-down in a grid
  - Players take turns selecting two cards by coordinates
  - If cards match, they remain revealed and the player scores a point
  - If cards don't match, they flip back face-down
  - Game ends when all pairs are matched
  - Player with most matches wins (or total turns in single-player mode)
- 

## **2. Functional Requirements**

### **2.1 Core Features**

- **FR-001:** Display a grid-based game board with face-down cards
- **FR-002:** Support 1-2 player modes
- **FR-003:** Three difficulty levels (Easy, Medium, Hard)

- **FR-004:** Accept player name input at game start
- **FR-005:** Accept coordinate-based card selection (e.g., A1, B3)
- **FR-006:** Validate all user inputs
- **FR-007:** Track and display score for each player
- **FR-008:** Detect and announce game completion
- **FR-009:** Declare winner or display final statistics

## 2.2 Difficulty Levels

### Difficulty Grid Size Unique Pairs Total Cards

Easy	4x4	8	16
Medium	6x6	18	36
Hard	8x8	32	64

## 2.3 Command Line Interface

- **FR-010:** -h or --help flag displays usage information
- **FR-011:** -d or --difficulty [easy|medium|hard] sets difficulty
- **FR-012:** -p or --players [1|2] sets number of players
- **FR-013:** --version displays version information
- **FR-014:** Interactive mode for game setup if no flags provided

## 2.4 Display Requirements

- **FR-015:** Clear screen between turns for clean display
- **FR-016:** Show column headers (A, B, C...) and row numbers (1, 2, 3...)
- **FR-017:** Display face-down cards with consistent symbol (e.g., █ or [?])
- **FR-018:** Display matched cards with their symbol/value
- **FR-019:** Show current player's turn
- **FR-020:** Display current scores
- **FR-021:** Show selected cards briefly before hiding mismatches

## 2.5 Card Symbols

Use ASCII-friendly symbols for cards, like:

- ♠ ♥ ♦ ♣ (suits)
- ★ ☆ ♫ ♪ (special)
- @ # \$ % & + = (symbols)
- A-Z (letters for higher difficulties)

## 3. Non-Functional Requirements

### 3.1 Usability

- **NFR-001:** Game should be playable in any standard terminal (80x24 minimum)
- **NFR-002:** Input validation with clear error messages
- **NFR-003:** Instructions accessible via help command

### 3.2 Compatibility

- **NFR-004:** Cross-platform (Windows, macOS, Linux)
  - **NFR-005:** Works with standard terminal emulators
- 

## 4. Use Cases

### 4.1 Use Case 1: Start New Single-Player Game

**Actor:** Player

**Preconditions:** Game executable is available

**Main Flow:**

1. Player runs the game executable
2. System prompts for number of players
3. Player enters "1"
4. System prompts for difficulty level
5. Player selects "easy"
6. System prompts for player name
7. Player enters "Alice"
8. System initializes 4x4 game board
9. System displays game board with all cards face-down
10. Game begins

**Postconditions:** Game is ready for first turn

**Alternative Flows:**

- 3a. Player uses `-p 1 -d easy` flags to skip prompts
- 

### 4.2 Use Case 2: Make a Move

**Actor:** Current Player

**Preconditions:** Game is in progress, player's turn

**Main Flow:**

1. System displays board and prompts for first card
2. Player enters coordinate (e.g., "A1")
3. System validates coordinate
4. System reveals card at A1
5. System prompts for second card
6. Player enters coordinate (e.g., "C3")
7. System validates coordinate
8. System reveals card at C3
9. System compares cards
10. Cards match: System awards point, leaves cards revealed
11. System updates score display
12. Turn passes to next player (or same player if match)

**Postconditions:** Board state updated, score recorded

**Alternative Flows:**

- 3a. Invalid coordinate: System displays error, re-prompts
  - 6a. Duplicate coordinate: System displays error, re-prompts
  - 9a. Cards don't match: System shows cards for 2 seconds, flips back
- 

### 4.3 Use Case 3: Complete Game

**Actor:** Players

**Preconditions:** Game in progress, only one pair remains

**Main Flow:**

1. Player matches final pair
2. System awards final point
3. System detects all cards matched
4. System displays final board
5. System displays final scores
6. System announces winner
7. System prompts for new game or exit

**Postconditions:** Game session ended

**Alternative Flows:**

- 6a. Tie game: System announces tie
- 

### 4.4 Use Case 4: View Help

**Actor:** User

**Preconditions:** None

**Main Flow:**

1. User runs game with -h flag
2. System displays help information including:
  - o Game description
  - o Command-line options
  - o How to play
  - o Examples
3. System exits

**Postconditions:** User has information, game not started

---

## 5. User Stories

### Epic: Game Setup

**US-001:** Player Configuration

*As a player*

*I want to enter my name at the start*

*So that the game addresses me personally and tracks my score*

#### Acceptance Criteria:

- System prompts for player name
  - Names can be 1-20 characters
  - Special characters are allowed
  - Empty names default to "Player 1/2"
- 

**US-002:** Difficulty Selection

*As a player*

*I want to choose from three difficulty levels*

*So that I can adjust the challenge to my skill level*

#### Acceptance Criteria:

- Three options: Easy (4x4), Medium (6x6), Hard (8x8)
  - Selectable via prompt or command flag
  - Invalid selection shows error and re-prompts
-

### **US-003: Player Mode Selection**

*As a player*

*I want to choose between single and two-player modes*

*So that I can play alone or with a friend*

#### **Acceptance Criteria:**

- Option for 1 or 2 players
  - Single player tracks total turns
  - Two players alternate turns
- 

### **Epic: Gameplay**

#### **US-004: Visual Board Display**

*As a player*

*I want to see a clear grid representation of the game board*

*So that I can easily identify card positions*

#### **Acceptance Criteria:**

- Grid has labeled columns (A, B, C...) and rows (1, 2, 3...)
  - Face-down cards show consistent symbol
  - Matched cards show their actual symbol
  - Current player and scores displayed above board
- 

### **US-005: Card Selection**

*As a player*

*I want to select cards using simple coordinate input*

*So that I can quickly make my choices*

#### **Acceptance Criteria:**

- Accept format: "A1", "a1", "B3", etc.
  - Validate coordinates are on the board
  - Prevent selecting already-matched cards
  - Prevent selecting same card twice in one turn
- 

### **US-006: Match Feedback**

*As a player*

*I want to see immediate feedback when I make a match  
So that I know my attempt was successful*

#### **Acceptance Criteria:**

- Matching cards remain revealed
  - Score increments immediately
  - Encouraging message displayed
  - Non-matches shown for 2-3 seconds before hiding
- 

**US-007: Turn Management**  
*As a two-player game participant  
I want turns to alternate automatically  
So that both players get equal opportunities*

#### **Acceptance Criteria:**

- Player 1 starts first
  - Turn switches after each attempt (except on match, optionally)
  - Current player clearly indicated
  - Score tracked separately per player
- 

**US-008: Game Completion**  
*As a player  
I want to see final results when the game ends  
So that I know who won and by how much*

#### **Acceptance Criteria:**

- All cards revealed at end
  - Final scores displayed
  - Winner announced (or tie declared)
  - Option to play again or exit
- 

## **Epic: User Experience**

**US-009: Command-Line Help**  
*As a new user  
I want to access help documentation  
So that I can learn how to play without external resources*

### **Acceptance Criteria:**

- `-h` flag displays help
  - Help includes: rules, controls, examples, options
  - Clear and concise formatting
- 

### **US-010: Input Validation**

*As a player*

*I want* clear error messages for invalid inputs

*So that* I understand what went wrong and can correct it

### **Acceptance Criteria:**

- Invalid coordinates show specific error
  - Invalid menu choices re-prompt with valid options
  - Error messages are helpful, not technical
- 

### **US-011: Game State Visibility**

*As a player*

*I want to* always see my current score and turn number

*So that* I can track my progress

### **Acceptance Criteria:**

- Score displayed on every screen
  - Turn count visible (single player)
  - Remaining pairs indicator
- 

## **6. Technical Specifications**

### **6.1 Data Structures**

GameBoard:

- grid: 2D array of Card objects
- size: tuple (rows, cols)
- matched\_pairs: set of matched card IDs

Card:

- id: unique identifier
- symbol: display character
- is\_revealed: boolean
- position: tuple (row, col)

Player:

- name: string
- score: integer
- is\_current: boolean

GameState:

- board: GameBoard
- players: list of Player
- difficulty: enum (EASY, MEDIUM, HARD)
- current\_player\_index: integer
- total\_turns: integer
- is\_complete: boolean

## 6.2 Key Functions

```
initialize_game(difficulty, num_players)
setup_board(size) -> GameBoard
shuffle_cards(cards) -> list
display_board(board, show_all=False)
get_player_input(prompt, valid_range) -> coordinate
validate_coordinate(input) -> tuple or Error
reveal_card(board, coordinate)
check_match(card1, card2) -> boolean
update_score(player)
switch_turn()
check_game_complete(board) -> boolean
display_results(players)
```

## 6.3 Input Format

### Coordinate Input:

- Format: [Column] [Row] (e.g., A1, B3, H8)
- Case-insensitive

- Column: A-Z (as needed for grid size)
- Row: 1-99 (as needed for grid size)

#### **Menu Input:**

- Numbered choices (1, 2, 3)
  - Text options (easy, medium, hard)
  - Yes/No prompts (y/n)
- 

## **7. User Interface Mockups**

### **7.1 Main Menu**

```
=====
MEMORY CARD GAME
=====
```

1. New Game
2. Help
3. Exit

Select option: \_

### **7.2 Game Setup**

Number of players (1-2): 2

Player 1 name: Alice

Player 2 name: Bob

Select difficulty:

1. Easy (4x4)
2. Medium (6x6)
3. Hard (8x8)

Choice: 1

Starting game...

### 7.3 Game Board (4x4 Easy)

=====

Player 1: Alice [Score: 2]

Player 2: Bob [Score: 1] ← CURRENT TURN

Remaining pairs: 5

=====

	A	B	C	D	
1		♥		████████	★
	-----	-----	-----	-----	-----
2		████████	♦	████████	
	-----	-----	-----	-----	-----
3		████████		████████	♣
	-----	-----	-----	-----	-----
4		████████		████████	★
	-----	-----	-----	-----	-----

Bob's turn!

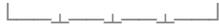
Select first card (e.g., A1): \_

### 7.4 Card Selection

Bob's turn!

Select first card: B2

	A	B	C	D	
1		♥		████████	★
	-----	-----	-----	-----	-----
2		████████	♦	████████	
	-----	-----	-----	-----	-----
3		████████		████████	♣
	-----	-----	-----	-----	-----
4		████████		████████	★
	-----	-----	-----	-----	-----



Select second card: \_

## 7.5 Game End

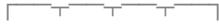


GAME COMPLETE!



Final Board:

A B C D



1		♥		♣		♦		★	
		-		-		-		-	
2		♣		♦		♪		♣	
		-		-		-		-	
3		@		♪		#		♣	
		-		-		-		-	
4		@		#		★		♥	
		-		-		-		-	



Final Scores:

Alice: 5 pairs

Bob: 3 pairs

Alice wins!

Play again? (y/n): \_

---

## 8. Command-Line Arguments

### 8.1 Synopsis

memogame [OPTIONS]

### 8.2 Options

<b>Flag</b>	<b>Long Form</b>	<b>Arguments</b>	<b>Description</b>
-h	--help	None	Display help and exit
-v	--version	None	Display version and exit
-p	--players	1 or 2	Number of players (default: 1)
-d	--difficulty	easy medium hard	Set difficulty (default: easy)
-n	--name	"string"	Player name (repeat for player 2)

## 8.3 Examples

bash

```
# Start with defaults (interactive setup)
./memogame
```

```
# Quick start: 2-player easy game
./memogame -p 2 -d easy
```

```
# Single player hard mode with name
./memogame -p 1 -d hard -n "Alice"
```

```
# Two players with names and medium difficulty
./memogame -p 2 -d medium -n "Alice" -n "Bob"
```

```
# Show help
./memogame -h
```

---

## 9. Acceptance Criteria

### 9.1 Minimum Viable Product (MVP)

- Single player mode works
- Two player mode works
- All three difficulty levels functional
- Card matching logic correct
- Score tracking accurate
- Winner determination correct
- Help command available
- Basic input validation

### 9.2 Quality Gates

- All use cases successfully executed
- No crashes on invalid input
- Game completable in all difficulty modes
- Clear visual feedback for all actions
- Playable on Windows, macOS, and Linux

### 9.3 Testing Checklist

- Start game with various flag combinations
- Complete game in all difficulty modes
- Test all invalid inputs
- Verify score calculation
- Test both 1 and 2 player modes
- Verify winner determination
- Test edge cases (same card twice, invalid coordinates, etc)
- Verify cross-platform compatibility

## 10. Error Messages Specification

This section defines the error messages that will be displayed to the user when invalid actions or incorrect inputs occur. Each message includes the **situation that triggers it** and the **exact text** that must be shown in the command-line interface.

Error ID	Situation	Error Message (shown to user)
E001	The player enters a coordinate that does not exist on the board (e.g., “Z9” in a 4×4 game).	Invalid coordinates. Please enter a valid position such as A1 or C3.
E002	The player selects the <b>same card twice</b> during their turn.	You cannot select the same card twice. Choose two different positions.
E003	The selected card has already been matched and removed from play.	This card has already been matched. Please select another hidden card.
E004	The player enters a coordinate in an <b>incorrect format</b> (e.g., “1A”, “AA”, “C”, empty input).	Invalid input format. Enter the column letter followed by the row number (e.g., A1).
E005	The player inputs a value that is not recognized when asked for game options (difficulty, number of players, etc.).	Invalid option. Please enter one of the available options shown on screen.
E006	The player name provided is empty or contains only spaces.	Player name cannot be empty. Please enter a valid name.
E007	The player attempts to enter more names than allowed (e.g., two names for a single-player game).	Too many player names entered. Only one name is required for single-player mode.
E008	The difficulty argument provided does not match the accepted values (“easy”, “medium”, “hard”).	Invalid difficulty level. Please choose: easy, medium, or hard.
E009	The number of players flag (-p) is not 1 or 2.	Invalid number of players. Please choose 1 or 2.
E010	An <b>unknown flag or command-line argument</b> is used.	Unknown option. Use '--help' to see all available commands.
E011	The terminal window is too small to properly display the game board.	Terminal size too small. Please enlarge your window to at least 80×24 characters.
E012	The player tries to quit or interrupt the game using Ctrl+C or q.	Game interrupted. Do you want to quit the game? (Y/N)
E013	A failure occurs while clearing the screen or refreshing the terminal.	Could not refresh the display. Continuing without screen clear.

**E014**

The game detects that Unicode symbols are not supported by the terminal.

Unicode characters not supported.

Switching to ASCII symbols.

**E015**

The player exceeds the maximum number of invalid attempts for input.

Too many invalid attempts. Please review the input format and try again.