SFWRENG 3XA3: Software Requirements Specification
Rummy For Dummies

Lab 2 Group 7, Rummy For Dummies
Joy Xiao, xiaoz18
Benson Hall, hallb8
Smita Singh, sings59

February 12, 2021

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Feb. 12, 2021	1.0	Revision 0 of SRS document.

This document describes the requirements for Rummy for Dummies. This document will be used as a reference point for all future documentation and development. This requirement specification document follows the Volere template [1].

1 Project Drivers

1.1 The Purpose of the Project

The purpose of this project will be to recreate the open-source version of Gin Rummy. Gin Rummy is a popular, multi-player game, liked by many people. This implementation is targeted towards individuals seeking to entertain themselves while remaining in their homes during lockdown. The project will convert a conventionally multiplayer game, so that it can be played alone with a computer opponent. The game will be easy to use and play, and be more visual than the open-source version.

1.2 The Stakeholders

1.2.1 The Client

The clients of the project will be the teaching assistants of SFWRENG 3XA3 and Dr. Bokhari. They will provide insight and assistance throughout the development of this project.

1.2.2 The Customers

The customers of the project will be card game enthusiasts, and individuals of all ages looking to play a game of Gin Rummy or variants of the game.

1.2.3 Other Stakeholders

The other stakeholders include the developers, who will be working to implement a improved version of the open-source project.

1.3 Mandated Constraints

1.3.1 Solution Constraints

Description: The project shall be written in Java and unit tested using JUnit.
Rationale: The developers are already familiar with using Java and JUnit, making this an ideal choice.
Fit Criterion: The source code will be written entirely in Java.

Description: The game shall be played through the CLI.
Rationale: It ensures that the game is simple for users to play and has a minimalist interface, which uses less resources.
Fit Criterion: The project will have no GUI, and will use ASCII characters to represent cards to create the visual experience for users on the CLI.

Description: The program will be used on operating systems such as Windows, MacOS and Linux.
Rationale: These operating systems can run the project without any additional installations.
Fit Criterion: The project will be made to compile and execute on the above stated operating systems.

1.3.2 Implementation Environment of the Current System

1.3.3 Partner or Collaborative Applications

This implementation does not have any partner or collaborative applications. It relies solely on the operating system to run a Java application.

1.3.4 Off-the-Shelf Software

There is no current off-the-shelf software that the developers will use within the implementation.

1.3.5 Anticipated Workplace Environment

The anticipated workplace environment is the user's home. However, the product can run on most desktop or laptop computers anywhere. There is

no need for an internet connection, and the game can be played locally on the user's machine.

1.3.6 Schedule Constraints

The project must be completed by the week of April 5th, 2021. This is when the final demonstration will take place.

1.3.7 Budget Constraints

The budget is $0 because there is no monetary funding. Creating this project should have no monetary cost, because there will be no costly resources used.

1.3.8 Enterprise Constraints

The game is free to download and play on the CLI for any user.

1.4 Naming Conventions and Terminology

Table 2: Naming Conventions and Terminology

Term	Definition
CLI	Command Line Interface
JRE	Java Runtime Environment
GUI	Graphical User Interface
Stock Pile	Remaining cards in the deck after dealing
Discard Pile	Cards that were discarded by the players
Deadwood Cards	Cards that are not part of a meld
Meld	Cards that form a valid sequence or have a common value

1.5 Relevant Facts and Assumptions

The project will implement a specific variant of Gin-Rummy. [2]

2 Functional Requirements

2.1 The Scope of the Work and the Product

Currently, launching the original source project and play a game of Gin Rummy requires the code to be compiled and built before operation. As the code is built in Go, it requires an installation of Go to compile and execute. [3]

As a command line and JRE generally come with a majority of desktop computers, the goal is to re-create Gin Rummy in Java for general use.

2.1.1 The Context of the Work

This project is to be developed for the SFWRENG 3XA3 course. It is developed for the professor of the course, Dr. Asghar Bokhari, and the teaching assistant assigned to our team, Mehdi Jafarizadeh.

Prior to development of the implementation, several documents will be created to efficiently establish the logistics of the project. These documents include:

- Test Plan

- Development Plan

- Module Guide

- Module Interface Specifications

The plan is to re-implement Gin Rummy in Java and to make it more accessible to the general public. The project will be developed such that it can be easily compiled and run in the command line.

2.1.2 Work Partitioning

Table 3: Work Partitioning Events

Event Number	Event Name	Input	Output
1	Starting the Game	Keyboard	New Game
2	Reading the Rules	Keyboard	Rules
3	Draw A Card from the Stock Pile	Keyboard	Request to Discard Card
4	Discard A Card to Discard Pile	Keyboard	Manipulated Hand
5	Draw A Card from Discard Pile	Keyboard	Request to Discard Card
6	Check All Melds in Hand	Keyboard	Current Melds in Player's Hand
7	Check the Current Score of Hand	Keyboard	Hand Score
8	Player knocks	Keyboard	Calculated Score & New Round

Table 4: Work Partitioning Summaries

| Event Number	Summary
1	The user, through the keyboard input, decides to start a new game of Gin Rummy.
2	The user, through keyboard input, chooses to read the rules of Gin Rummy.
3	If the user chooses to draw a card from the stock pile, they must discard a card from their hand after the card is added to their hand.
4	The user selects a card to add to the discard pile through keyboard input.
5	If the user chooses to draw a card from the discard pile, they must discard a card from their hand after adding the card to their hand.
6	If the user wishes to see all their melds, a list of all possible melds are displayed.
7	If the user wishes to see their current score, the score will be calculated from the deadwood cards currently in their hand.
8	If the user wishes to knock and has a score less than or equal to 10, the round ends and points are calculated based on deadwood score of both sides.

2.1.3 Individual Product Use Cases
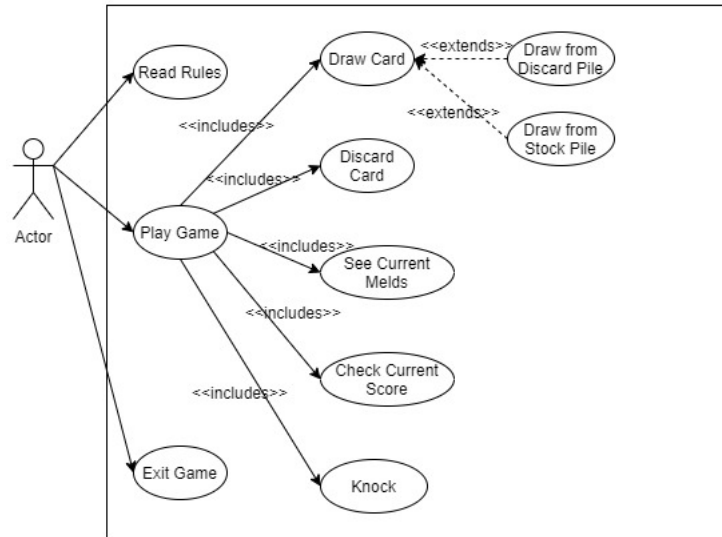
Figure 1: Use Case Diagram

2.2 Functional Requirements

Business Event 1: The user wants to start a new game.

- FR1. The system shall ask the user to input their name.

- FR2. The system shall display a random set of cards as the users card hand, and it shall display the top card on the discard pile.

- FR3. The system shall display the set of options the user can take.

Business Event 2: The user wants to draw a card from the stock pile.

- FR4. The system shall add the top card of the stock pile to the user's hand.

- FR5. The system shall display the new card to the user through the CLI.

- FR6. The system shall prompt the user to discard one of the cards in their hand.

Business Event 3: The user wants to pick up a card from the discard pile.

- FR7. The system shall add the top card on the discard pile to the user's hand.

- FR8. The system shall display the new card to the user through the CLI.

- FR9. The system shall prompt the user to discard one of the cards in their hand.

Business Event 4: The user wants to check for melds.

- FR10. The system shall show the user any melds that can be created from the user's hands.

Business Event 5: The user wants to check total points of the hand.

- FR11. The system shall calculate and display the user's points.

Business Event 6: The user wants to discard a card.

- FR12. The system shall prompt the user to input which card the user wants to discard.

- FR13. The system shall discard the chosen card if valid and display the new card hand.

- FR14. The system shall place the card on top of the discard pile faced up.

- FR15. The system shall make a move on behalf of the computer agent.

Business Event 7: The user wants to knock.

- FR16. The system checks if the user has less than 10 deadwood points.

- FR17. If the user has 10 or fewer deadwood points, then the system shall end the round.

- FR18. The system will calculate the difference in deadwood points between the user and AI and determine the winner.

- FR19. The system will award the points to the winner.

- FR20. The system will prompt the player if they want to play a new round.

Business Event 8: The user wants to exit the game.

- FR21. The system shall terminate the game and game data will not be saved.

3 Non-Functional Requirements

3.1 Look and Feel Requirements

LF.1 The system shall have a simplistic look.

3.2 Usability and Humanity Requirements

UH.1 The system shall be easy to play for users of any age.

UH.2 The system shall be easy to run for users of any age.

3.3 Performance Requirements

P.1 The system shall respond to valid user interactions within 0.5 seconds.

P.2 The system shall adhere to rules of the game Rummy.

3.4 Operational and Environmental Requirements

OE.1 The system shall be able to be operated on Windows, Linux, and MacOS operating systems.

3.5 Maintainability and Support Requirements

MS.1 The system shall be clearly documented and commented for ease of maintainability.

MS.2 The system shall be as modularized as possible.

3.6   Security Requirements

N/A

3.7   Cultural Requirements

C.1 The system shall not have any offensive images or text.

C.2 The language of the system shall be in English.

3.8   Legal Requirements

L.1 This system shall adhere to the same copyright properties as the original
project: MIT License.

3.9   Health and Safety Requirements

There are no health and safety concerns for this project. It does not store
user data and the game will not post a health concern to the user.

4   Project Issues

4.1   Open Issues

The following issues may be encountered during development of the project:

- Implementation of the computer opponent.

- Graphically representing playing cards in the CLI.

4.2   Off-the-Shelf Solutions

There are existing products on the app store that implement the game of
Gin-Rummy.

4.3   New Problems

One of the limitations of attempting to trace source code implementation for the computer opponent is that Go is not an object-oriented language, so it will be difficult to imitate some of the source code in Java.

Our project is played through the CLI and the user will download the files to play the game. This is a problem, as people may find it difficult to run even with sufficient documentation.

4.4   Tasks

The Gantt Chart for the project will be followed for task milestones. There will be two development phases: Revision 0 and Revision 1 of the project. Revision 1 will go through documents in Revision 0 and make updates to be consistent with the final product.

Table 5: Task due dates

Task	Due Date
Proof of Concept Demonstration	February 22, 2021
Test Plan Revision 0	March 5, 2021
Design and Document Revision 0	March 18, 2021
Revision 0 Demonstration	March 22, 2021
Final Demonstration	April 5, 2021
Final Documentation	April 12, 2021

4.5   Migration to the New Product

Not applicable.

4.6   Risks

There are several risks associated with the development of this project:

- Project size is too large

- Difficulty testing the computer opponent

- Bad programming practices (i.e., modules have unclear purposes)

4.7 Costs

There will be no costs for this project. All resources used are free. The amount of effort spent on this product is around 6 hours a week per team member.

4.8 User Documentation and Training

The README.md file in the Gitlab repository will clearly document the setup of the game and its instructions.

4.9 Waiting Room

Not applicable.

4.10 Ideas for Solutions

A solution to the issue is to trace the original project and see how it is implemented there. We may need further investigation through online resources.

A solution to representing playing cards is to use ASCII art.

References

[1] Atlantic Systems Guild Ltd. Volere requirements specification template, 2019.

[2] Wikipedia. Gin rummy, 2021.

[3] vlmlee. Gin-rummy, 2017.

5   Appendix

5.1   Symbolic Parameters

- TOTAL_CARDS = 52

- RANK = {A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K}

- SUIT = {HEARTS, CLUBS, DIAMONDS, SPADES}