Rummy (Text-based game)

Requirements Specification

Team Members: Neil West, Gurdeep Singh, Maruf Tonmoy

Contents

1. The Purpose of the Project

This project’s goal is to represent an accurate recreation of the card game rummy that can be played as a single player console game for linux. Consequently, the user should be able to use text commands to carry out the same operations that they would be able to carry out in a physical card game of rummy.

2. The Stakeholders:

Subject matter experience: Summarizes the users' knowledge of the subject matter/business. Rate as novice, journeyman, or master.

2.1 Players

Players are the most important stakeholders because they have the final say on acceptance of the project and they will be playing the game in the end. Players can be of any age over 8 years and may have no past experience in card games and/or Rummy. Player should be asked whether or not to play a tutorial depending on the players' experience level. Listed below are some categories of the players along with their attributes.

2.1.1 Children

The players aged between 8 and 15 fall under this category.

1. **Subject matter experience (Rummy):** Novice
2. **Technological Experience:** Novice
3. **Other user characteristics:** More attraction towards visuals
4. **Physical abilities/ disabilities:** no disabilities
5. **Intellectual abilities/ disabilities:** no disabilities
6. **Attitude towards job** : NA
7. **Attitude toward technology**: welcome new technologies
8. **Physical location:** NA
9. **Education:** preliminary education
10. **Linguistic skills:** limited
11. **Age group:** 8 - 15 years
12. **Gender:** NA
13. **Ethnic group/s:** NA

2.1.2 Adults

The players aged between 16 and 65 fall under this category.

1. **Subject matter experience (Rummy):** Master
2. **Technological Experience:** Master
3. **Other user characteristics:** No
4. **Physical abilities/ disabilities:** might have disabilities (e.g. vision)
5. **Intellectual abilities/ disabilities:** might have disabilities
6. **Attitude towards job** : NA
7. **Attitude toward technology**: welcome new technologies
8. **Physical location:** NA
9. **Education:** high school or more
10. **Linguistic skills:** excellent
11. **Age group:** 16-65 years
12. **Gender:** NA

13. **Ethnic group/s:** NA

2.1.3 Seniors

The players aged 65 fall under this category.

14. **Subject matter experience (Rummy):** Master
15. **Technological Experience:** Journeyman
16. **Other user characteristics:** No
17. **Physical abilities/ disabilities:** impaired vision
18. **Intellectual abilities/ disabilities:** might have disabilities
19. **Attitude towards job** : NA
20. **Attitude toward technology**: reluctance towards new technology
21. **Physical location:** NA
22. **Education:** high **school** or more
23. **Linguistic skills:** excellent
24. **Age group:** 65 +
25. **Gender:** NA
26. **Ethnic group/s:** NA

2.2 Development Team (or other stakeholders)

2.2.1 Developers

Team of around 3 people will be required to complete the project. Developers should be efficient in designing, developing and testing the game as per the requirements of the course. C++ must be used for developing the software and design document must be followed to avoid discrepancies in the expected and obtained product.

2.2.2 Client (Dr. Anvik)

Dr. Anvik is the source for further clarifications of requirements.

2.2.3 Testers

Testers are responsible for testing the game by applying the testing techniques discussed in the class.

2.3 User Priority:

1. Key users:

   Adults (2.1.2) are the key users. There is more chance of the user being familiar with the linux and command line experience.

2. Secondary users:

Children and Seniors (2.1.1 & 2.1.3) are the secondary users. Limited to no command line experience should be expected. Seniors are expected to have experience with the card games though.

3. Constraints

Note: All the constraints, functional, non-functional and other requirements have an ID associated with them. For e.g. #1 is for Operating System and #6 is for first functional requirement

3.1 Solution Constraints:

1. Operating system

Priority	Highest priority
Requirement type	Constraint
Use cases	1, 2, 3, 4
Rationale	The client (project markers) will be using the software on the lab computers, so above all else the software *must* run on their system or the project as a whole will be a failure.
Fit Criterion	Before delivery, the software should be tested on the university gitlab servers and on the lab computers themselves.
Description	Software must run on the university of lethbridge computer science lab computers.

2. Program type

Priority	Medium priority
Requirement type	Constraint
Use cases	1, 2, 3, 4
Rationale	GUI’s are far too expensive to work with in the limited time-scope of the project, so a text application fits the time and skill budget

Fit Criterion	A user will be able to run the entirety of the application for a linux terminal.
Description	Software must be run as a text application from the terminal.

3.2 Budget Constraints:

3. Time

Priority	Medium priority
Requirement type	Constraint
Use cases	1, 2, 3, 4
Rationale	The time schedule for development is set to be 2 weeks, and the developers can only be expected to work up to 1-2 hours per day.
Fit Criterion	The story points allocated for this project should be between 45-90 hours of work.
Description	Application must be made by a group of 3 developers working 15-30 hours each.

4. Personnel

Priority	Medium priority
Requirement type	Constraint
Use cases	1, 2, 3, 4
Rationale	Teams will be formed with the assumption that there will be 3 members in each team.
Fit Criterion	Team does not outsource the project to individuals outside of their team, and they are allocated the proper number of students by Dr. Anvik
Description	Project must be completed by 3 CPSC 3720 students

5. Skill level

Priority	Medium priority
Requirement type	Constraint
Use cases	1, 2, 3, 4
Rationale	The minimum skill level for students working on this project is at the 3000 level, so we can’t expect them to do tasks of a higher skill level
Fit Criterion	Project does not use any more advanced programming techniques, such as neural networks.
Description	Project must be designed with the skill level of 3000 level computer science students in mind

4. Naming Conventions and Terminology

User: The person who plays the game or interacts with the software.

Player: Someone who plays the game. Can be the user or one of the computer controlled players.

Rummy: A playing cards based game in which a player tries to make sets or sequences of cards.

Card: A playing card from a standard 52 card deck.

Hand: A set of cards that a player is dealt. The player can only see the cards in his/ her own hand.

Meld: To create a group of cards, either a set or a run.

Lay off: The action of a player adding cards from their hand to an existing meld in play.

Rank: The number of a card, including aces, jacks, queens, and kings.

Run: Three or more sequential cards in the same suit.

Set: Three or four cards of the same rank.

Suit: One of four categories a card can fall in, including hearts, diamonds, clubs, and spades.

Discard pile: This is a stack of cards from which the player can draw the card from top and drop a card face up from his hand every turn. (a card drawn from the discard pile cannot be dropped back in the same turn)

Draw pile: This is a stack of cards from where the user can draw a card every turn. (cards are not visible in this pile)

Draw card: Adding card to a players’ hand from the top of either Discard pile or Draw pile

Going rummy: When a player sheds all of their cards in a single turn

5. Assumptions

This project is being designed with the following assumptions in mind:

1. The game will be run on the specified hardware without changes (Lab computers).
2. A GUI (Graphical user interface) is not required for the implementation of this game.
3. This text application will not require any console backtracking or carriage returns to update information, and all information will be displayed via new lines. Assumed since it's a very platform specific problem to deal with, and would be difficult to implement.
4. The standard c++ 11 libraries will be available for use
5. Although an in game tutorial is provided, It is expected that the user has at least some experience in playing any standard 52 card game.
6. This game comes with its own design and specifications, there might be other variants of the same game, but the user is expected to follow the rules for this game.

6. The Scope of the Project

This project aims at making a command line (text) based playing card game called “Rummy”. Each player tries to empty his/ her hand by melding the cards in form of runs, sets or individual cards.

Users can range from 8 year old children to eldery people. Users can also be visually or intellectually impaired. An Intuitive tutorial is to be provided for the first time players as well as an optional advanced tutorial for improving in the game.

The optimal computer system for running the game is a linux based operating system. Basic command line knowledge will be required by the user for playing the game. Users can play with upto 4 imaginary players that are controlled by the computer.

6.1 Use Case Diagrams

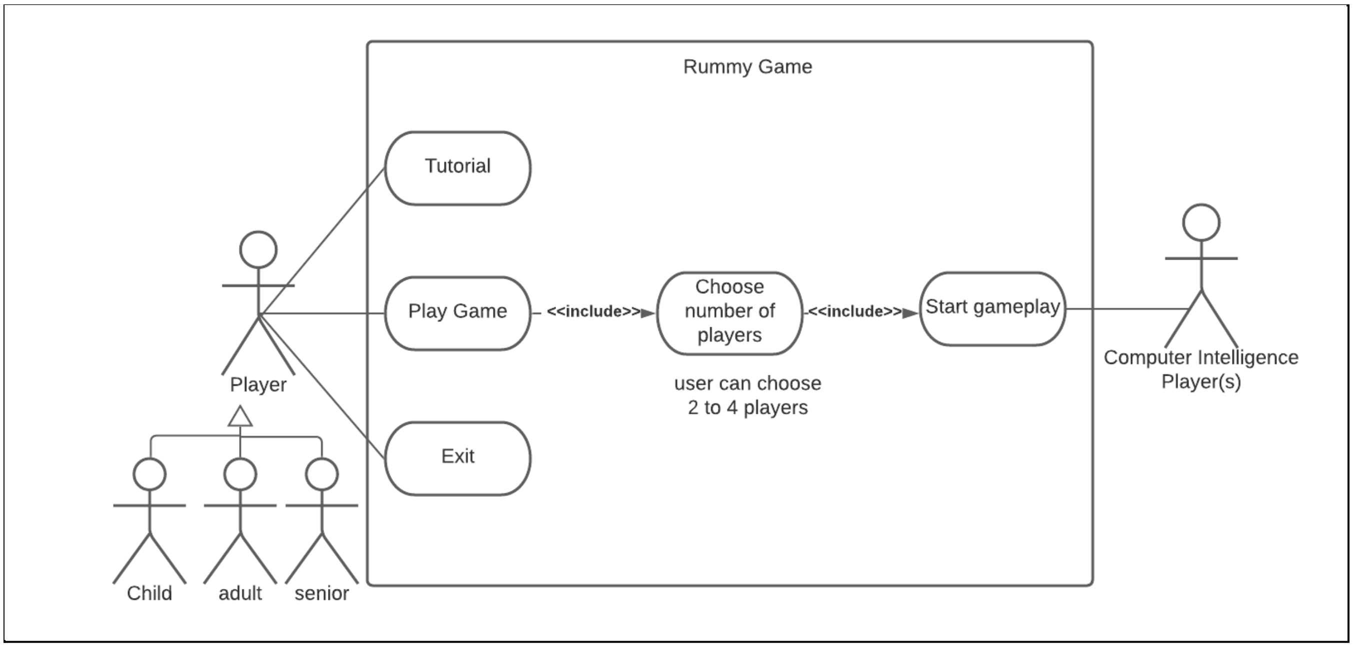

Fig. Rummy use case diagram

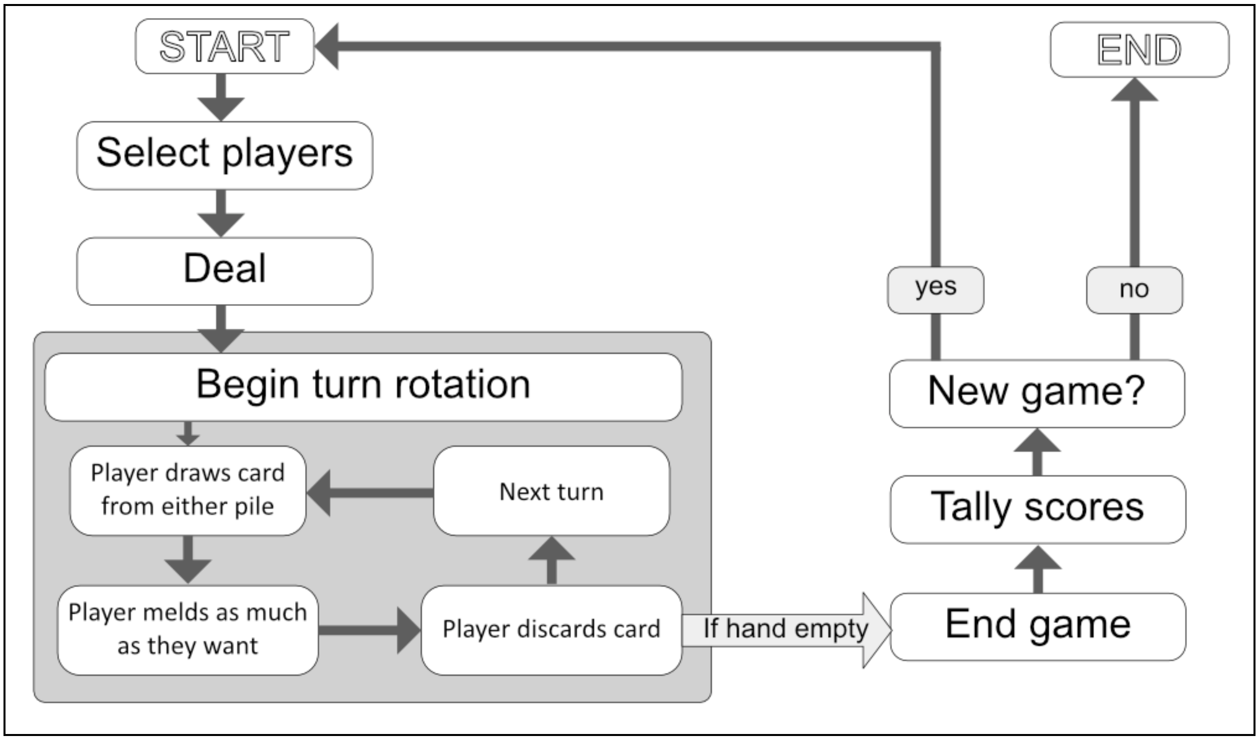

Fig: Rummy gameplay

6.2 Use Cases

6.2.1 Tutorial

Rationale:- If the user is not familiar with the game or the command line interface.

Fit criteria:- The game provides a tutorial which explains how to play the game and use the command line interface.

2 Play Game

Rationale:- To play the game.

Fit criteria:- The game plays well without any crashes and exhibiting expected behaviour.

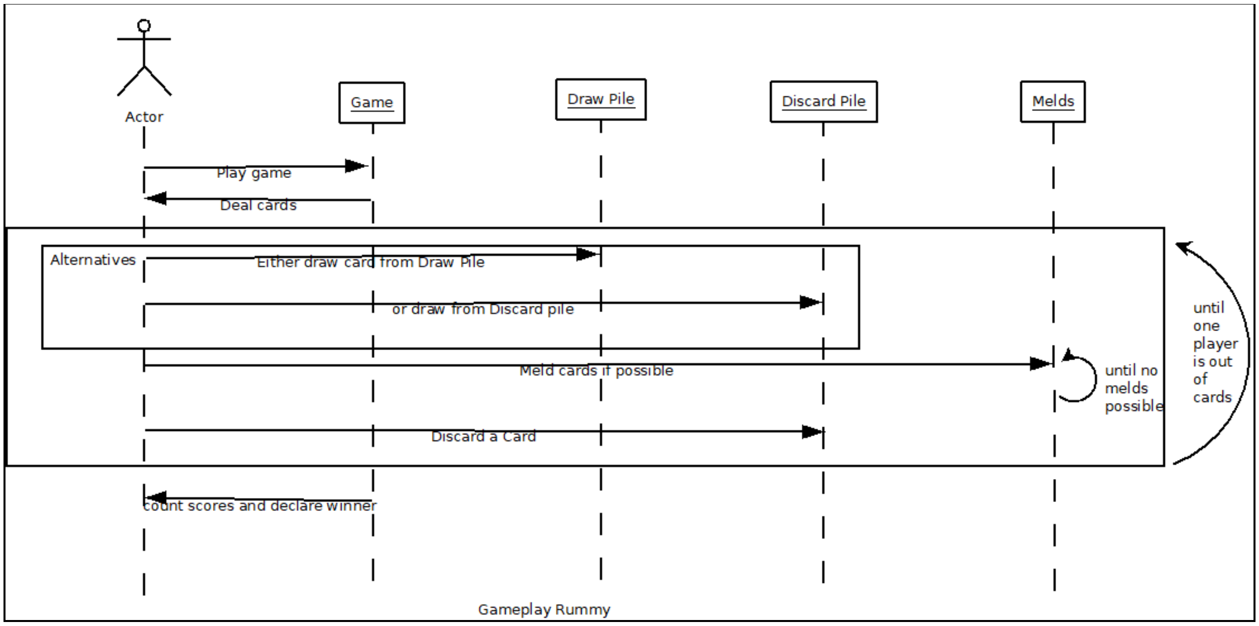

Fig: Rummy gameplay sequence diagram

3 Exit

Rationale:- If the user wants to exit the game.
Fit criteria:- The player is able to exit when he wants.

4 Choose number of players

Rationale:- To set the opponents and user profiles.
Fit criteria:- required number of player profiles are created.

6.3 Text scenarios

1. Game startup (4 players)

```
--------------------------------------------------------------------
                          Welcome to RUMMY!
 Before we being, be sure to familiarize yourself with the controls
 and the rules of the game. If you want to go through the tutorial,
 use command 'TUTORIAL'. If you want to get straight to the game,
 use command 'START'. At any point within the game you feel lost,
 use the 'HELP' command.
--------------------------------------------------------------------
> START
--------------------------------------------------------------------
How many players would you like to have? (2-4)
--------------------------------------------------------------------
> 4
--------------------------------------------------------------------
What is your name?
--------------------------------------------------------------------
> Bob
--------------------------------------------------------------------
 You are playing at a table with Alice, Joe, and Tristian seated in
 that order clockwise from your left. The dealer this round is
 Tristian.

 Tristian deals the following 7 cards to you:
  [Four of Hearts]
  [Queen of Hearts]
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]
  [Six of Clubs]
  [Seven of Clubs]

 The draw pile has a [King of Hearts]. Your turn.
--------------------------------------------------------------------
```

2. Basic player turns
   a. Player draws a card from the draw pile before discarding a card

```
...
 The draw pile has a [King of Hearts]. Your turn.
---------------------------------------------------------------------
> draw from draw pile
---------------------------------------------------------------------
 You have drawn a [Three of Diamonds].

 Your current hand:
  [Four of Hearts]
  [Queen of Hearts]
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]
  [Three of Diamonds]
  [Six of Clubs]
  [Seven of Clubs]
---------------------------------------------------------------------
> discard [Four of Hearts]
---------------------------------------------------------------------
```

   b. Player draws a card from the discard pile and tries to discard it before trying other card

```
...
 The draw pile has a [King of Hearts]. Your turn.
---------------------------------------------------------------------
> draw from discard pile
---------------------------------------------------------------------
 You have drawn a [King of Hearts].

 Your current hand:
  [Four of Hearts]
  [Queen of Hearts]
  [King of Hearts]
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]
  [Six of Clubs]
  [Seven of Clubs]
---------------------------------------------------------------------
> discard [King of Hearts]
---------------------------------------------------------------------
 You cannot discard the card you just drew from the discard pile.
 Pick a different card.
---------------------------------------------------------------------
> discard [Four of Hearts]
```

```
---------------------------------------------------------------------
```

c. Player melds a run

```
...
 The draw pile has a [Four of Spades]. Your turn.
---------------------------------------------------------------------
> draw from draw pile
---------------------------------------------------------------------
 You have drawn a [Ten of Hearts].

 Your current hand:
  [Ten of Hearts]
  [Queen of Hearts]
  [King of Hearts]
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]
  [Six of Clubs]
  [Seven of Clubs]
---------------------------------------------------------------------
> meld [Ten of Hearts] [Queen of Hearts] [King of Hearts]
---------------------------------------------------------------------
 You have successfully created a run.

 Current melds on table:
  [Ten of Hearts]
  [Queen of Hearts]
  [King of Hearts]

 Your current hand:
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]
  [Six of Clubs]
  [Seven of Clubs]

---------------------------------------------------------------------
> discard [King of Hearts]
---------------------------------------------------------------------
```

d. Player melds a set

```
...
 The draw pile has a [Four of Spades]. Your turn.
---------------------------------------------------------------------
> draw from draw pile
---------------------------------------------------------------------
 You have drawn a [Eight of Clubs].

 Your current hand:
  [Four of Hearts]
  [Queen of Hearts]
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]
  [Six of Clubs]
  [Seven of Clubs]
  [Eight of Clubs]

---------------------------------------------------------------------
> meld [Six of Clubs] [Seven of Clubs] [Eight of Clubs]
---------------------------------------------------------------------
 You have successfully created a set.
 Current melds on table:
  [Six of Clubs]
  [Seven of Clubs]
  [Eight of Clubs]

 Your current hand:
  [Four of Hearts]
  [Queen of Hearts]
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]

----------------------------------------------------------------------
> discard [King of Hearts]
----------------------------------------------------------------------
```

e. Player lays off a card

```
...
 The draw pile has a [Four of Spades]. Your turn.
----------------------------------------------------------------------
> draw from draw pile
----------------------------------------------------------------------
 You have drawn a [Nine of Clubs].

 Current melds on table:
 1.
  [Six of Clubs]
  [Seven of Clubs]
  [Eight of Clubs]

 Your current hand:
  [Four of Hearts]
  [Queen of Hearts]
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]
  [Nine of Clubs]

----------------------------------------------------------------------
> lay off [Nine of Clubs] meld 1
----------------------------------------------------------------------
 You laid off your [Nine of Clubs].

 Current melds on table:
 1.
  [Six of Clubs]
  [Seven of Clubs]
  [Eight of Clubs]
  [Nine of Clubs]

 Your current hand:
  [Four of Hearts]
  [Queen of Hearts]
  [Two of Spades]
  [Six of Spades]
  [Two of Diamonds]

----------------------------------------------------------------------
> discard [King of Hearts]
----------------------------------------------------------------------
```

3. CPU player turns

a. CPU player melds a run

```
---------------------------------------------------------------------
 Alice's turn.

 Alice draws from the draw pile.

 Alice melds a run of:
  [Six of Hearts]
  [Seven of Hearts]
  [Eight of Hearts]

 Alice discards an [Ace of Spades].
 Alice's turn ends with 4 cards in hand.
---------------------------------------------------------------------
```

b. CPU player melds a set and lays off a card

```
---------------------------------------------------------------------
 Joe’s turn.

 Joe draws from the discard pile.

 Joe melds a set of:
  [Ace of Spades]
  [Ace of Clubs]
  [Ace of Hearts]

 Joe lays off a [Nine of Hearts]:
  [Six of Hearts]
  [Seven of Hearts]
  [Eight of Hearts]
  [Nine of Hearts]

 Joe’s turn ends with 3 cards in hand.
---------------------------------------------------------------------
```

4. Game end

a. Player wins round

```
...
 Tristian discards an [Ace of Spades].
 Tristian's turn ends with 2 cards in hand. Your turn.
---------------------------------------------------------------------
> draw from discard pile
---------------------------------------------------------------------
 You have drawn a [Ace of Spades].

 Current melds on table:
 1.
  [Six of Clubs]
  [Seven of Clubs]
  [Eight of Clubs]
  [Nine of Clubs]
  [Ten of Clubs]

 2.
  [Four of Hearts]
  [Four of Spades]
  [Four of Clubs]

 Your current hand:
  [Ace of Spades]
  [Two of Spades]
  [Three of Spades]

---------------------------------------------------------------------
> meld [Ace of Spades] [Two of Spades] [Three of Spades]
---------------------------------------------------------------------
 You have successfully created a set. You are now out of cards.

 Congratulations! You have won this round!

 Alice's hand: 62 points
  [Ace of Hearts]
  [Six of Hearts]
  [Eight of Spades]
  [Ten of Spades]
  [Eight of Diamonds]
  [Jack of Diamonds]
  [Nine of Clubs]

 Joe's hand: 46 points
  [Ten of Spades]
  [Jack of Spades]
  [Four of Diamonds]
  [Nine of Diamonds]
  [Three of Clubs]
  [Jack of Clubs]

 Tristian's hand: 23 points
  [Jack of Hearts]
  [Two of Spades]
  [Six of Diamonds]
  [Five of Spades]

---------------------------------------------------------------------
  Score:
 Round | Bob   | Alice  | Joe  | Tristian |
 1     | 0     | 0      | 0    | 78       |
 2     | 0     | 0      | 45   | 0        |
 3     | 131   | 0      | 0    | 0        |
 ------------------------------------------
 Total | 131   | 0      | 45   | 78       |
---------------------------------------------------------------------
Starting next round...
---------------------------------------------------------------------
```

b. CPU wins round

```
...
---------------------------------------------------------------------
 Alice's turn.

 Alice draws from the draw pile.

 Alice melds a run of:
  [Six of Hearts]
  [Seven of Hearts]
  [Eight of Hearts]

 Alice discards an [Ace of Spades].
 Alice's turn ends with 0 cards in hand.
---------------------------------------------------------------------
 Alice is all out of cards.

 Too bad, Alice won this time...
 Your hand: 62 points
  [Ace of Hearts]
  [Six of Hearts]
  [Eight of Spades]
  [Ten of Spades]
  [Eight of Diamonds]
  [Jack of Diamonds]
  [Nine of Clubs]

 Joe's hand: 46 points
  [Ten of Spades]
  [Jack of Spades]
  [Four of Diamonds]
  [Nine of Diamonds]
  [Three of Clubs]
  [Jack of Clubs]

 Tristian's hand: 23 points
  [Jack of Hearts]
  [Two of Spades]
  [Six of Diamonds]
  [Five of Spades]

---------------------------------------------------------------------
  Score:
 Round | Bob   | Alice  | Joe  | Tristian |
 1     |       | 131    | 0    | 0        |
 ------------------------------------------
 Total | 131   | 0      | 0    | 0        |
---------------------------------------------------------------------
Starting next round...
---------------------------------------------------------------------
```

5. Help command

```
---------------------------------------------------------------------
                      _  _   ___   _      ____
                     ) () ( ) __( ) |    )  _)\
                     | -- | | _)  | (__  | '__/
                     )_()_( )___( )____( )_(

---------------------------------------------------------------------
 Basic rules:
...

 Command reference:
...
```

7. Functional Requirements

Note: All the constraints, functional, non-functional and other requirements have an ID associated with them. For e.g. #1 is for Operating System and #6 is for first functional requirement

6. A card drawn from the discard pile cannot be discarded on the same turn it was drawn.

Priority or implementation order	Medium
Use cases	2
Rationale:	To follow the game rules.
Fit Criterion	User is unable to discard the same card drawn from the discard pile on the same turn.
Story Points	2

7. Up to 4 multiple players should be created as requested by the user.

Priority or implementation order	High
Use cases	4, 2
Rationale:	To allow upto 4 multiple players in the game.
Fit Criterion	Game can be played with 2 to 4 players
Story Points	5

8. Correct number of cards should be dealt depending on the number of players in the game.

Priority or implementation order	High
Use cases	2
Rationale:	To deal the correct number of cards depending upon the number of players in the game.
Fit Criterion	Following number of cards are dealt depending upon the number of players.

	2 players = 10 cards each 3-4 players = 7 cards each
Story Points	1

9. User should be allowed to make valid melds only

Priority or implementation order	High
Use cases	2
Rationale:	To make sure only right cards are used for laying melds
Fit Criterion	Melded cards fit either of the following criteria: **Run**: Three or more sequential cards in the same suit are used. **Set**: Three or four cards of the same rank are used.
Story Points	3

10. Players should be able to lay off cards from their hands onto melds on the table.

Priority or implementation order	High
Use cases	2
Rationale:	To let the players add new cards into the already laid sets or runs.
Fit Criterion	Players are able to add cards to the existing melds
Story Points	1

11. Going rummy

Priority or implementation order	Medium
Use cases	2

Rationale:	Going rummy is a feature of the game rummy, which although is accesses infrequently, is still and important rule in the game
Fit Criterion	Player points for a round are doubled if the player melds/lays off their whole hand in the first round
Story Points	1

12. A new draw pile is created from the discard pile in case the old draw pile is empty.

Priority or implementation order	Low
Use cases	2
Rationale:	To facilitate continuous gameplay if the draw pile is emptied.
Fit Criterion	A new draw pile is created automatically from the discard pile when it goes empty.
Story Points	2

13. Tally the points from all the cards remaining in all the other players hands when someone wins.

Priority or implementation order	Medium
Use cases	2
Rationale:	The scoring system in rummy calculates points for a round based on the cards still in the other player's hands when a round ends based on the rank (ace = 1, 10 = 10) except for face cards which are all worth 10 points.
Fit Criterion	The number of points is correctly calculated based on their expected values.
Story Points	2

14. A player should be able to sort their hand

Priority or implementation order	Low
Use cases	2
Rationale:	It is fairly standard in card games for players to sort their cards in a way that makes it easier for them to plan their game.
Fit Criterion	Offer the player multiple methods for sorting their hand.
Story Points	3

15. Restart the game

Priority or implementation order	Low
Use cases	1,2,3,4
Rationale:	Professional games don't usually expect a player to turn the game off and turn it back on every time they want to restart it.
Fit Criterion	A restart command can bring the game back to its initial state
Story Points	1

16. Users should be able to save their scores and names, and display them in future sessions

Priority or implementation order	Low
Use cases	4
Rationale:	Short arcade games often have scoreboards with player initials and scores to let players compare themselves with other players
Fit Criterion	A save file is created and updated after each game with scoring and name information, and can be displayed on the title screen

Story Points	5

17. Players should be able to input a name for themselves

Priority or implementation order	Low
Use cases	1
Rationale:	Being called by name can add a sense of customization to the game, and for use in scoreboarding.
Fit Criterion	Game asks players for a name before a game starts.
Story Points	1

18. CPU player run by simple algorithm

Priority or implementation order	Low
Use cases	1
Rationale	Can’t make the game multiplayer, so other players will have to be run by the program
Fit Criterion	Other players can use simple strategies to play the game
Story Points	5

8. Non-functional Requirements

19. After someone wins the game, the score is calculated and carried over to the next deal

Priority or implementation order	Low
Use cases	2
Rationale:	To facilitate multiple deal games.
Fit Criterion	The winning score is calculated and carried over to the next game.
Story Points	2

20. A tutorial is provided with instructions about how to use the terminal.

Priority or implementation order	Low
Use cases	1
Rationale:	To facilitate learning to new users to the command line interface.
Fit Criterion	A command line interface tutorial is provided.
Story Points	3

21. A game rules help is provided

Priority or implementation order	Medium
Use cases	2, 1
Rationale:	To provide help to the user when he needs to know the game rules before starting the game and within the game.
Fit Criterion	Game rules and tutorials are provided within and outside the game.
Story Points	1

22. A feature to **let the user set** how many deals he wants to play or how much someone should score within multiple deals to win the game.

Priority or implementation order	Low
Use cases	2, 4
Rationale:	To let the user decide the winning criteria.
Fit Criterion	User is asked how many deals should be played or what is the required score to win the whole series.
Story Points	2

23. User should be able to input anything into the textbox without causing any issues.

Priority or implementation order	Low
Use cases	2
Rationale:	The user should be able to focus on playing the game rather than making typos.
Fit Criterion	These inputs don't break the game:- $ What the heck! $ 234987234 $ ::)) $
Story Points	3

24. User should have multiple options for how they refer to which cards they are using.

Priority or implementation order	Low
Use cases	2
Rationale:	Some users prefer fast inputs whereas some users prefer easy to understand inputs
Fit Criterion	The parsing system can understand multiple inputs of varying complexity to mean the same thing
Story Points	3

9. Look and Feel Requirements

25. A welcome message and main menu should be provided

Priority or implementation order	Medium

Use cases	1, 2, 3, 4
Rationale:	To provide a minimalistic main menu
Fit Criterion	A nice looking main menu is provided
Story Points	1

26. User should be able to control the game with inputs that look similar to normal written language.

Priority or implementation order	Low
Use cases	1, 2, 3, 4
Rationale:	Users don’t want to spend time memorizing commands and would rather have controls that are simple to pick up
Fit Criterion	A parsing system can convert regular english seeming sentences into in-game commands
Story Points	13

10. Risks

These are the risks we are taking into account for this project.

1. Requirements/Design/Estimation
    a. The planned project is too large to implement. This can be dealt with by cutting out less essential requirements from the project. E.g. spend less time on computer players.
    b. The planned project is too small to take up the expected amount of work. *In this case, the developers can spend more time refining existing requirements so they integrate with the system better.*
    c. Certain requirements are more challenging to implement than expected, and eat too much of the development team's time. *Same solution as implemented in a*.
    d. Faulty design leads to major restructuring midway through the development lifecycle. *This will inevitably cause setbacks in the project, so the project would have to shrink in scale,*

   e. Story point estimation is faulty, making it difficult to estimate how much of the project has been completed. *Go through and re-estimate the story points for each feature in response.*
   f. Requirement priorities are faulty, causing the developers to spend time on requirements that aren't essential before critical requirements. *If the developers notice that requirements are incorrectly prioritized, they should modify the priority themselves before more issues arise.*

2. People
   a. A team member could join midway, slowing down development for time to get them up to speed with the project. *This can be dealt with by giving the new member a chance to read through the design document on their own, and pick features that haven't been worked on yet for their own portion of development. Probably the portions which are not dependencies for critical parts.*
   b. A team member falls sick (e.g. catches COVID). *The workload will be equally divided into other team members until the team member is back. Low priority requirements will be discarded and the client (Dr. Anvik) will be informed about the situation.*
   c. A team member could leave midway, leaving the development team understaffed. *The response to this would be similar to the project being too large, having the development team leave out lower priority features from the project.*
   d. A team member is not carrying their weight, and not contributing to development. *This can be dealt with in the report at the end of the sprint.*

3. Learning & Tools
   a. Inexperience with new tools. *If a team member is unfamiliar with C++, gitlab, and gtest, then they can take a supporting role for that sprint where they are not left in charge of critical portions of development so they have a chance to learn the tools first with non critical components.*
   b. Learning curve for tools is steeper than expected. *If this happens, then requirements involving these tools can be given more story points, and further broken down into smaller tasks.*
   c. Tools don't work together in an integrated way. *In this case, there's either the option of building something to connect the tools, or building a new tool that is better designed to work together*