

# 1. Project Background and Description ([Andreijustion Agetano](#))

## Brief Description of how to play the war game

The object of this two-player game is to collect all the cards set up shuffle a standard deck of playing cards with jokers and deal out all the cards each player picks up their deck and simultaneously flips the top card face-up in front of themselves compare the two cards and the player with the highest ranking card collects both cards in a personal face-up discard pile each player then flips again ace is the highest rank followed by king all the way down to two if you run out of cards in your deck turn your collected cards face-down shuffle and continue playing if you both flip over identical cards of the same rank this is called a war each player lays three cards face-down staggered on top of the face-up card then each player flips one card face-up compare the new face-up cards played and the player with the higher rank wins all the cards in the war including the face-down ones if however the cards tie again then you have another war place down another three cards and continue the process until there is a winner if a player doesn't have three cards to place down because they ran out of cards in their deck and discard then they lose the war the first player to collect all the cards in the deck wins.

## War Rules

1. Each player gets dealt half the deck, 26 cards, and the cards are put face-down in a stack in front of the players.
2. Both players turn their top cards face up at the same time. The person with the highest card wins that draw and takes both the cards. They are put to the side to form a new stack, which the player can use when he finishes his current stack.
3. If both players draw a card of the same rank, e.g., they both draw 8s, then there is a war. The face-up cards are left on the table and each player puts three cards face-down on the table and then puts one card face up. The face-up card determines who wins the war and gets all 10 cards that are on the table at this point. If the face-up card is again the same rank, then the war goes on, three more face-down, one face-up, etc.
4. The first player to finish all their cards loses the game.
5. If a player finishes their cards during a war without having enough cards to finish the war, then he loses immediately.

## **Describe the project goals and final vision**

- Everyone can play this game
- It is quite simple to play
- It is a source of fun and excitement.
- Furthermore, it is a way of entertainment and refreshes your mind.
- It is a great method to stay connected with family and friends (play with each other).
- The game ends when one player has won all the cards.
- The objective of the game is to win all the cards.

## **Describe the current starting base code**

- Understand how to test your program as it is being developed.
- Understand how to deal with errors.
- Know how to use pseudocode.
- Understand how to use array methods.

## **2. Project Scope (Brittany Hanford) (content writes by Jingxin Jiang at the last minute)**

Our war game will include Card, CardGenerator, UserConsole, CardCompare, and WinCards class. All group members set up and update those classes in their branch, adjust the function of each class, and discuss building a perfect function in order to meet the war card game requirement. When one player wins all 52 cards through the code, the project will be done.

## **3. High-Level Requirements (Jingxin Jiang)**

Our War card game can achieve the following requirements:

- Our game allows two players to register with the game at the same time and define the account names player1 and player2. Each player can check their status through the user console.
- Each player turns up a card at the same time, and the player with the higher card takes both cards and puts them on the new stack. Get one point. The user console will show the score and who wins the cards.

- If the cards are the same rank, it is War. Each player turns up one card face down and one card faces up. The player with the higher cards takes both piles (six cards). If the turned-up cards are again the same rank, each player places another card face down and turns another card face up. The player with the higher card takes all 10 cards, and so on. The winner gets one point. If they still are the same rank and the cards used up. They will put their cards back to their own new stack.
- The first round ends when 26 cards are used up. Whoever wins the card's number larger, will be won another 10 points. Score higher will win this round. The game will show their score, the winner, and how many cards they have won.
- Then, the player will be asked whether they would like to continue the game. If those two players both select yes, the next round will start with cards which they have won.
- Finally, The game ends when one player wins 52 cards. The game will show their total scores and who wins the game.

## 4. Implementation Plan (**Mohammedsalik Mohammedsalik, modified by Jingxin Jiang**)

Our Git Repository URL is: <https://github.com/JingxinJiang/GroupProject1.git> (**modified by Jingxin Jiang**)

This repository URL can be accessed by anyone in the group from their own local tree branch. Anyone can make changes in codes and use commands such as the push or fetch & merge to change the remote repository. We create different folders for text, UML, and codes respectively so the group members can sort and edit or make changes accordingly. We'll be using NetBeans for this project coding, and we'll be using java coding conventions and standards to make War Game work virtually. All the group members will be working on different sections of the project to make all the ends meet at the end and to make the project run successfully. Below is the attached graph of folders for diagrams and the template in GitHub under master branch /GroupProject1/src/ as shown in Fig 1.

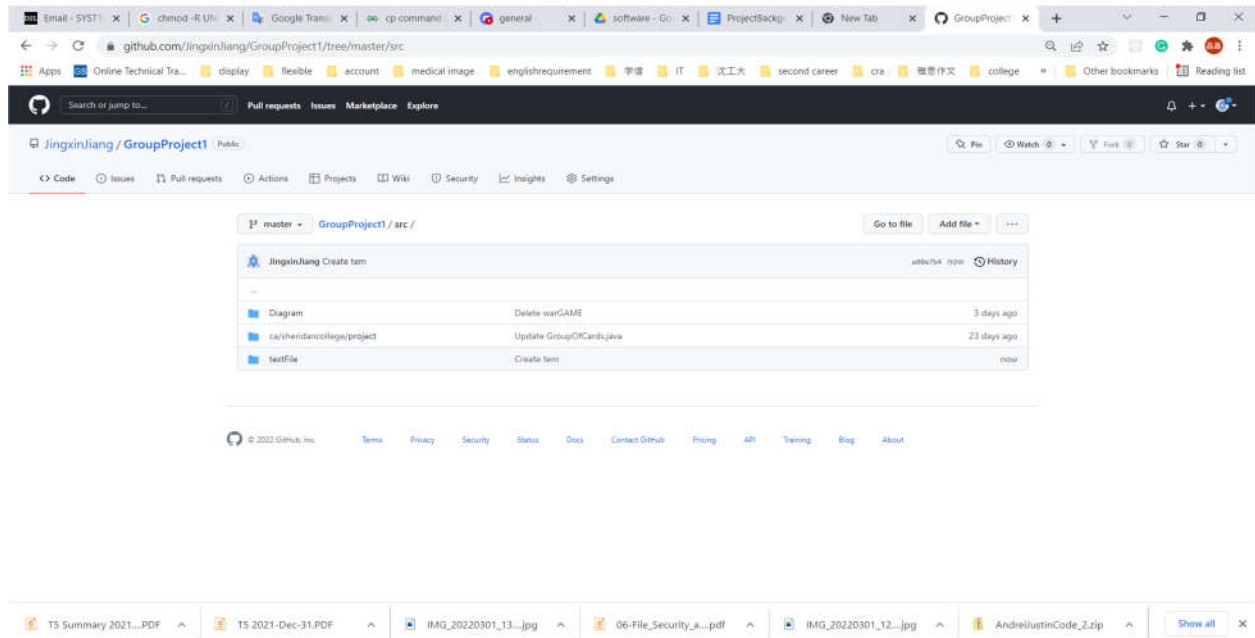


Fig 1. Folder for our template, diagram, and code (**Figure added by Jingxin Jiang**)

## 5. Design Considerations (**Everyone writes at least one example**)

For our war card game, first, in order to have high cohesion, we plan to divide some classes according to their function:

- Card class includes the suit and value of the cards.
- CardGenerator was used to generate 26 random cards for each player.
- UserConsole class just is the interface of users.
- CardCompare class used to compare which card is big.
- WinCards class used to store the card which the player wins.
- War class used to run the war program.

Second, we plan to use delegation in our program in order to achieve better communication between cooperators in our group. Based on the basic class with basic functions, each member in our group can change the function in their own branch, and code is also easily changed in the future. (**Above content writes by Jingxin Jiang**)