

SYST 17796

DELIVERABLE 1

Mario Martinez
Oluwatoyosi Kuponiyi
Nusrath Syed

SYST 17796 TEAM PROJECT



Table of Contents

TABLE OF CONTENTS 1

TEAM CONTRACT 2

UML DESIGN 8

DESIGN DOCUMENT OVERVIEW 9

IMPLEMENTATION PLAN 10

DESIGN CONSIDERATIONS 10

REFERENCES 12

Team Contract

SYST 17796 TEAM PROJECT


Team Name: WAR Team

Please negotiate, sign, scan and include as the first section in your Deliverable 1.

Please note that if cheating is discovered in a group assignment each member will be charged with a cheating offense regardless of their involvement in the offense. Each member will receive the appropriate sanction based on their individual academic honesty history.

Please ensure that you understand the importance of academic honesty. Each member of the group is responsible to ensure the academic integrity of all of the submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Every member will be expected to act as project leader. Vote will be taken to resolve disagreements.		
Mario Martinez	<i>Mario Martinez</i>	991 362 634
Oluwatoyosi Kuponiya	Oluwatoyosi Kuponiya	991 332 886

Nusrath Syed		991 627 519

For further information read **Academic Honesty Policy on AccessSheridan**.



By signing this contract, we acknowledge having read the Sheridan Academic Honesty Policy as per the link below.




<https://policy.sheridanc.on.ca/dotNet/documents/?docid=917&mode=view>




Responsibilities of the Project Leader include:



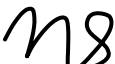
- Assigning tasks to other team members, including self, in a fair and equitable manner.
- Ensuring work is completed with accuracy, completeness and timeliness.
- Planning for task completion to ensure timelines are met
- Any other duties as deemed necessary for project completion

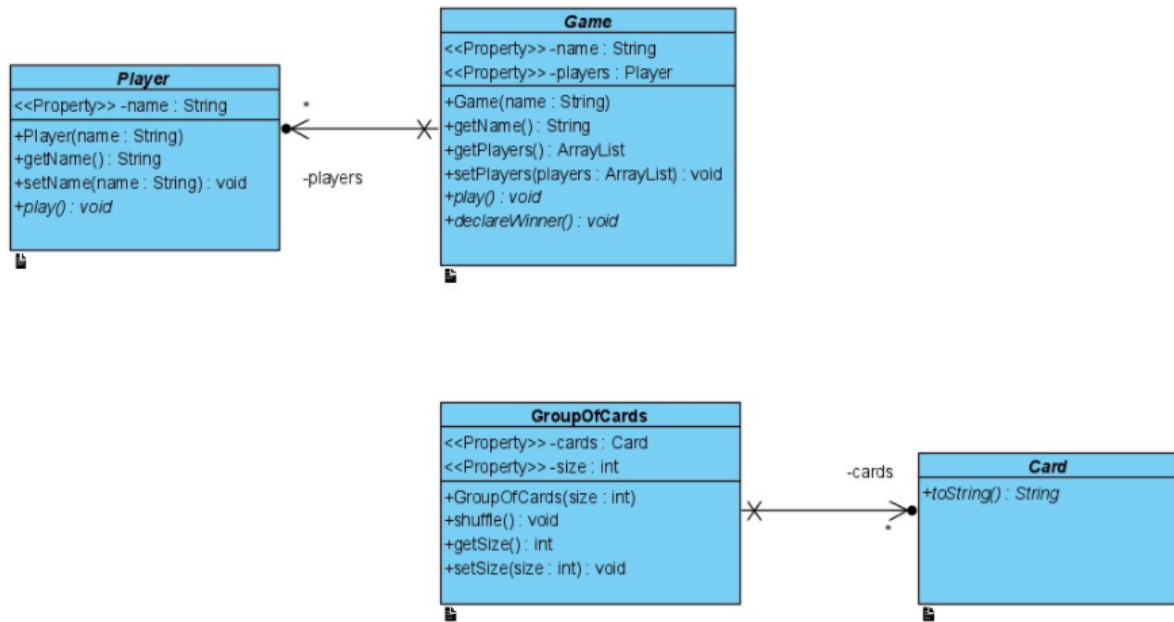
What we will do if . . .

Scenario	Accepted initials	We agree to do the following
Team member does not deliver component on time due to severe illness or extreme personal problem	M.M. O.K 	Team will absorb the workload temporarily. Team member who was unable to deliver their component will be expected to make up for it by taking on more during the next deliverable.
Team member cannot deliver component on time due to lack of ability	M.M. O.K 	Team member will be expected to try to learn how to complete the component. If they are unable to do so and assisting them involves too much work for other members, they will

Scenario	Accepted initials	We agree to do the following
		<p>swap components with another team member.</p> <p>Team member will be expected to learn the missing information for next deliverable.</p>
Team member does not deliver component on time due to lack of effort	<p>M.M. O.K </p>	<p>Team member will need to explain what went wrong.</p> <p>Team will absorb the workload and be honest about the lack of contribution from team member.</p> <p>If this behaviour becomes a pattern, team will take a vote on how to deal with the situation.</p>
Team member does not attend team meeting	<p>M.M. O.K </p>	<p>Team proceeds without him/her and will assign work to the absent member.</p> <p>Reason for absence must be provided.</p> <p>Missing member will be expected to catchup on what they missed.</p>
An unforeseen constraint occurs after the deliverable has been allocated and scheduled (a surprise test or assignment)	<p>M.M. O.K </p>	<p>Team meets and reschedules deliverable.</p> <p>Action plan will be voted on.</p>

Scenario	Accepted initials	We agree to do the following
Team cannot achieve consensus leaving one member feeling "railroaded", "ignored", or "frustrated" with a decision which affects all parties	M.M. O.K. 	Team is led by shared leadership so majority vote decides all. 3 members means there will always be a majority vote. Team will hear out team member that feels ignored if they consistently lose the vote.
Team members do not share expectations for grade desired	M.M. O.K. 	Team votes on each submission's quality. Team will ask for individual marking and will identify sections by author.
Team member behaves in an unprofessional manner by being rude or uncooperative	M.M. O.K. 	Team requests meeting with professor to problem-solve. Team agrees to avoid use of all vocabulary inappropriate to the business setting.

Scenario	Accepted initials	We agree to do the following
Team member assumes or requests that his/her name be signed to a submission but has not participated in production of the deliverable	M.M. O.K 	Team agrees that this is cheating and is unethical.
There is a dominant team member who is content to make all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	M.M. O.K 	Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote.
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	M.M. O.K 	Team forces decision sharing by routinely voting on all issues.



UML DESIGN

Design Document Overview

Project Background and Description

Describe the project goals and final vision. Include a brief description of how to play the game you have chosen and a reference to the rules of the game you have chosen. Also describe the current starting base code. Use technical terms to describe the code including what language it is written in, any patterns you can see and any coding conventions used.

The project goal is to create the card game called War from a base code that is provided. The goal of War is to win all the cards. Ranking of the cards goes from highest to lowest is A K Q J 10 9 8 7 6 5 4 3 2. Nothing can beat an Ace and a 2 cannot beat anything. The deck is split equally among the players, if 2 players then 26 cards, if 3 players then 17 cards and if 4 players then 13 cards.

How to Play: Place your stack of cards face-down on the table, players are not allowed to look at their cards. Count to 3 and then flip the top card of your own stack at the same time as your opponent. Compare each card that is flipped to see which is higher ranked. The player with the higher ranked card wins and collects both cards to add to their hand. It is declared 'War' when you and your opponent have the same value of card. To go to war: each player places 3 more cards face-down on the table. Flip over a fourth card as you would flip a card when not at War, whoever has the higher card now is the person who takes all 10 cards from that round. If a player does not have enough cards, they must place their last card face-up, this will be the card used to play war. Play until 1 person wins all the cards in the deck (wikiHow, 2021).

Starting base code is provided in java language. There are 4 classes provided, the class called GroupOfCards includes an arraylist that will get the group of cards. Another abstract class is called Player which models each Player in the game. Players have a unique identifier. Lastly there are two other abstract classes one for Game and the other for the Card. The game class has the array list as well, it includes the players parameter, declaring the winner method and a getter and setter for the players. The card class only has a toString method written to return a String representation of a card.

Project Scope

Describe the names and roles of each team member. Describe the technical scope of the project by talking about the interface and how you will know when the project is complete.

Mario Martinez, Oluwatoyosi and Nusrath will all act as the project leader and delegate tasks to each other as needed.

Mario will be providing support with the declaring overall project objectives.

Nusrath and Oluwatoyosi and Mario will be contributing by providing individual information required for the deliverables and providing insight on coding and any other required tasks in the project goals.

The interface will be the use of a main public class called WarGame this will instantiate the 2 decks. Includes a tracker variable which tracks user wins and computer wins and draws. We will know when the project is complete when the program runs smoothly and has no issues and a person outside of our group can play the game and understand it.

High-Level Requirements

Describe the high level requirements:

The new system must include the following:

- **Ability for each player to register with the game**

Using the Player class and using a string and an integer for entering their uniqueName and age, respectively. This will implement each player in the class using variables like `this.uniqueName = uniqueName;` and `this.age = age;` Once the player is registered they will be able to start the game using the Game class.

- **Ability for the game to communicate a win or loss**

The method `getWinner` or `getLoser` will be able to communicate for winning or losing. The Card class will show the value of the card, the suit does not matter in this instance and we will use switch and case statements to distinguish the cards. Next the `getWinner` or `getLoser` method will be followed by if and else if statements. A `toString` will be provided to show the result.

- **Ability for players to always know their status (score)**

The main program class will have instantiation of 2 decks and will include 4 hands and method that shuffles the deck.

There will be variables written for the 4 hands referred to by integers, the integers will be `int count`, `int wins` and one `int draw`, and they are all set to 0. This will help to keep know your score there will be a `count++` to indicate and a print statement to show the result.

Implementation Plan

Include your Git repository URL here and a brief description of the expected use (i.e. each developer checks in code at the end of each day/week). Text files are stored under a separate directory, code, UML diagrams have their own folders etc.

Include information on coding standards you intend to follow and tools you expect to use (VP, NetBeans, eclipse, Junit...)

<https://github.com/mariomartz/WarCardGame>

Our Git repository is public so viewing access is available to anyone with the link. Team members are set as collaborators so they will have access to edit the repository as well.

Our repository will hold all our project files. For this reason, it has been split up into different folders for different types of files. (Example: docs folder for documents)

Team members will be required to check in at least once per week. As well as notify the team when changes are made or being made.

Our team will be working using NetBeans and following OOP coding standards.

Design Considerations

Talk about how the current code is structured as it relates to the following OO principles. Each principle should have 2 or 3 specific examples from the base code or your intended additional code (i.e. potential for improvement).

- Encapsulation
 - Data will be encapsulated in several ways.
 - Deck cards will be encapsulated into their own object, with the deck holding every card and limiting the total number of cards plus instances of each card type
 - Individual cards will be encapsulated into a separate object to hold the values each individual card
 - Player's cards will be encapsulated into their own object as well, this object will hold the cards that each player has available to them and be drawn from the deck object

- Player's will be encapsulated into objects of their own with information on each individual player, for example their identity
- Lastly, game rules will be encapsulated into their own class/object and hold all the rules as well as determine the winner and keep track of the score
- Delegation
 - Delegation will work as mentioned above
 - The task of holding individual card values will be delegated to the Card class
 - The task of holding all cards available for distribution will be delegated to the Deck class
 - The task of holding all cards available to each player will be delegated to the PlayerHand class
 - The task of identifying each player will be delegated to the Player class
 - The task of maintaining the rules of the game and deciding a winner will be delegated to the GameManager class
- Flexibility/Maintainability
 - Flexibility/Maintainability will come from Encapsulation and Delegation of game systems
 - Since our code is split up into separate objects and tasks are delegated, each system will be independent of the others in some way
 - Each class has its own task so changes made to individual classes will not greatly effect others
 - Example: Game Rules can be changed in the GameManager class. Card values can be changed in the Card class. Total cards available can be changed in the Deck class.
 - This object oriented approach creates flexibility and maintainability of the code.

References

wikiHow. (2021, May 13). *How to Play War (Card Game)*. [https://www.wikihow.com/Play-War-\(Card-Game\)#:~:text=Play%2052%2Dcard%20war.,wins%20all%20of%20the%20cards.](https://www.wikihow.com/Play-War-(Card-Game)#:~:text=Play%2052%2Dcard%20war.,wins%20all%20of%20the%20cards.)