Team Contract

SYST 17796 TEAM PROJECT

Team Name: Battle Deck

Please negotiate, sign, scan and include as the first page 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 integrity 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.

For further information, read Academic Integrity Policy here :
https://caps.sheridancollege.ca/student-guide/academic-policies-and-procedures.aspx

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Yashraj Solanki	Yashraj S	991752068
Ayushi Goswami	Ayushi G	991752568
Diya Patel	Diya P	991752600

By signing this contract, we acknowledge having read the Sheridan Academic Integrity Policy

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.
- Notifying the professor of any issues in a timely manner so that corrective measures can be taken.
- Any other duties as deemed necessary for project completion.

What we will do if . . .

Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**
Team member does not regularly attend team meetings and/or does not respond to communications in a timely manner.	a	Project leader emails the student citing the concerns and cc's the professor so they are aware of the situation at the very onset _X_ (**Mandatory**). a) __ In addition to above, the leader/team will (add your own content here):
Team member does not deliver component on time due to severe illness or extreme personal problem.	B, C	a) Team absorbs workload temporarily __ b) Team seeks advice from professor _X_ c) Team shifts target date if possible _X_ d) __ Other (specify):
Team member has difficulty delivering component on time due to lack of understanding or ability.	B, C	a) Team reassigns component __ b) Team helps member _X_ c) Team member must ask professor for help _X_ d) __ Other (specify):

Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**
Team member does not deliver component on time due to lack of effort.	C	a) Team absorbs workload __ b) Team member(s) ask professor to request a Participation Form from <u>all</u> team members. This *may* result in individualized grades being awarded for a deliverable __ c) Both a. and b. above _X_ d) __ Other (specify):
Team cannot achieve consensus leaving one or more member(s) feeling that their voice(s) is/are not being heard in a decision which affects everyone.	A, B	a) Team agrees to abide by majority vote _X_ b) Team seeks advice from the professor _X_ c) __ Other (specify):
Team members do not share expectations for the quality of work on a particular deliverable.	A, C	a) Team members will draw on each other's strengths to help bring the quality of the deliverable to a minimal acceptable level _X_ b) Team votes on each submission's quality __ c) Team member(s) ask professor to request a Participation Form from all team members, which may result in individualized grades being awarded for a deliverable _X_ d) __ Other (specify):
Team member behaves in an unprofessional manner, e.g. being rude, uncooperative and/or making one or more	B, C	a) Team agrees to avoid use of all vocabulary inappropriate to a business/college setting __

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
member(s) feel uncomfortable.		b) Team attempts to resolve the issue by airing the problem at a team meeting _X_ c) Team requests a meeting with the professor to discuss further _X_ d) __ Other (specify):
There is a dominant team member who insists on making all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	A, B	a) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote _X_ b) Team will express subordination feelings and attempt to resolve issue _X_ c) Team seeks advice from the professor __ d) __ Other (specify):
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	A, C	a) Team forces decision sharing by routinely voting on all issues _X_ b) Team routinely checks with each other about perceived roles __ c) Team discusses the matter at team meeting _X_

SYST 17796 DELIVERABLE 1
DESIGN DOCUMENT
Group 9

OVERVIEW

1. Project Background and Description

The War card game has been implemented using Java in this project. The goal is to develop a basic, text-based version of the game that follows the rules of War, a two-person card game in which the player with the higher card wins both cards. There is a "war" and more cards are played if the cards are equal. The game goes on until either a maximum number of rounds is achieved or one person has all the cards. The project is built using object-oriented programming (OOP) principles in Java and includes classes for Card, Deck, Player, and WarGame. The provided base code contains basic classes that represent players, cards, and game flow. This design document describes how the code will be extended and structured to finish the game.

2. Project Scope

Team Members: Ayushi Goswami, Diya Patel, Yashraj Solanki

The goal of this project is to use Java to design and create a text-based War card game. A console-based interface and object-oriented programming approaches will be used in the implementation. The game will have basic features including card distribution, deck creation, and gaming techniques. The project will be considered finished when the code complies with OOP principles, players may participate in an entire game session, and the game accurately selects a winner.

3. High-Level Requirements

The game needs to include:

- Creating the Deck: A 52-card deck is shuffled and initialized.
- Half of the deck is distributed to each player.
- The gameplay loop involves players picking and comparing cards every round.
- Conditions for Winning or Losing: The player who has every card wins.
- Console Output: The game gives players productive results.

4. Implementation Plan

Git repository URL: https://github.com/ihsuya1/Gameproject.git

Git repository is a public repository and can be accessed by clicking the hyperlink given. All the files would be properly organized by creating different folders, like code in one folder, UML diagrams in separate folder.

5. Design Considerations

- Encapsulation:

Encapsulation ensures data security and integrity by ensuring that class attributes are private and only available through methods. Internal state is managed by the Deck and Player classes without direct disclosure of details. To maintain consistency, the Card class use encapsulation to limit direct changes to its properties. This avoids unwanted changes, makes debugging easier, and improves maintainability.

- Delegation:

To maintain separation of concerns, the WarGame class assigns duties such as card distribution to Deck and gameplay handling to Player. WarGame is made simpler by the player controlling its own hand of cards. The basic game logic is kept simple by the Deck class, which distributes and shuffles cards.

- Flexibility/Maintainability:

Because the project was created with modularity in mind, it is simple to expand or change. It is easier to debug and test when concerns are separated. New features may be added to existing code without affecting it through Java interfaces and inheritance. An organized project encourages better teamwork and long-term maintainability.

6. UML Class Diagram: