Team Contract

SYST 17796 TEAM PROJECT

Team Name:
__CodeCrafters_____

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.

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Varunjot Singh	Varunjot Singh	991719098
Varleen Kaur	Varleen Kaur	991715937
Aman Aman	Aman Aman	991721004
Anchalpreet Kaur	Anchalpreet kaur	991717997

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

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 . . .

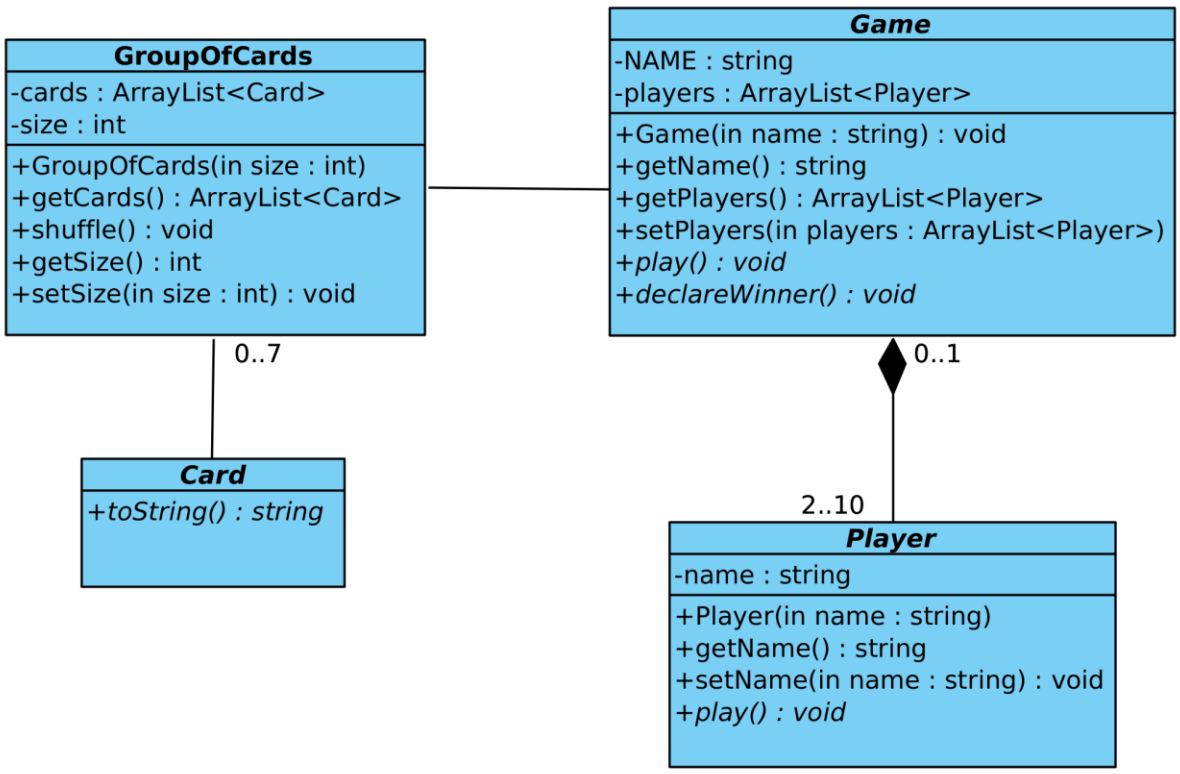
Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**
Team member does not regularly attend team meetings and/or does not respond to communications in a timely manner.		Project leader emails the student citing the concerns and cc's the professor 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): In addition to above, the leader or team will fix a meeting with the professor so that their grades would not be affected by a single person and seek help from the professor.
Team member does not deliver component on time due to severe illness or extreme personal problem.		a) Team absorbs workload temporarily __ b) Team seeks advice from professor _X_

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
		c) Team shifts target date if possible __ d) __ Other (specify):
Team member has difficulty delivering component on time due to lack of understanding or ability.		a) Team reassigns component __ b) Team helps member _X_ c) Team member must ask professor for help __ d) __ Other (specify):
Team member does not deliver component on time due to lack of effort.		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) Team agrees to abide by majority vote _A_ b) Team seeks advice from the professor __ c) __ Other (specify):

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
Team members do not share expectations for the quality of work on a particular deliverable.		a) Team members will draw on each other's strengths to help bring the quality of the deliverable to a minimal acceptable level __ 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 _C_ d) __ Other (specify):
Team member behaves in an unprofessional manner, e.g. being rude, uncooperative and/or making one or more member(s) feel uncomfortable.		a) Team agrees to avoid use of all vocabulary inappropriate to a business/college setting __ b) Team attempts to resolve the issue by airing the problem at a team meeting _B_ c) Team requests a meeting with the professor to discuss further __ 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) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote __ b) Team will express subordination feelings and attempt to resolve issue _B_ c) Team seeks advice from the professor __ d) __ Other (specify):

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted		a) Team forces decision sharing by routinely voting on all issues __ b) Team routinely checks with each other about perceived roles _B_ c) Team discusses the matter at team meeting __

UML CLASS DIAGRAM:



SYST 17796 DELIVERABLE 1
DESIGN DOCUMENT TEMPLATE

OVERVIEW

1. Project Background and Description

The card game selected for implementation is UNO, which will be developed using Java programming language. The provided base code constitutes the fundamental structure for creating a card game. Various classes have been defined to represent essential components of the game, including players, hands of cards, and other relevant entities. The code is well documented with comments and hints and it follows the camel case convention for the name of methods.

A brief introduction to the game rules: The objective of UNO is for players to be the first to empty their hand of cards by strategically matching them with the top card of the discard pile based on either color, number, or symbol. The deck consists of four color-coded suits: red, green, blue, and yellow, each containing cards numbered from 0 to 9, along with special action cards (draw two, reverse, stop, wild card, etc.). The special cards are used to stop other players from winning for e.g. the player who uses wild card will choose the color of the card the next player has to use. The winner is decided when one of the players have no card in hand i.e. they have used all their cards. Players take turns playing cards from their hand onto the discard pile, ensuring that the card they play matches either the color, number, or symbol of the card on top of the pile. If a player cannot play a card, they must draw a card from the deck. The game continues until one player successfully empties their hand, at which point they are declared the winner.

All the players receives 7 cards in the beginning. A random card is drawn out of the remaining deck of cards and is placed, this is the first card of the discard pile and this marks as the starting of the game.

2. Project Scope

The Team members are appointed particular task in order to work efficiently. The roles of each member is depicted below:

- Varun : Team Lead
- Varleen Kaur : UI Designer
- Aman : Code Organiser and Developer
- Anchalpreet Kaur : Lead Developer

In order to make this project playable, the most important aspect will be to create a UI (User-Interface) so that the players can interact with the game. The UI will be a text-based command system. The player will enter the command in text and it will be used to perform actions in-game. This also include the cards-in-hand of the current player from where they'll able to choose the suitable card to play. It will also show the number of cards of the remaining players and it can also label the next player and the previous player. As mentioned there is no GUI required so everything will be displayed on a terminal.

3. High-Level Requirements

The new system must include the following:

- Game Initialization:
    - The game should initialize with a standard UNO deck consisting of 108 cards
    - The deck should be shuffled before the game starts.
- Player Setup:
    - Each player should be dealt an initial hand of 7 cards from the shuffled deck.
    - Players should be able to see their own hand of cards but not the hands of other players.
- Winning Conditions:
    - The game should continue until one player has emptied their hand of cards by playing their last card.
    - When a player plays their last card, they win the round, and the game should display a winning message with their name.
- User Interface (UI):
    - The UI should provide commands for players to select and play cards, as well as commands for drawing cards.
- Game Rules Enforcement:
    - The game should enforce the rules of UNO, preventing players from making illegal moves (e.g., playing a card that doesn't match the color or number of the top card on the discard pile).
    - The game should handle special action cards (Skip, Reverse, Draw Two) and Wild cards according to UNO rules.
- Scalability:
    - The game should be scalable to support different numbers of players and customizable game settings (e.g., number of cards dealt, scoring rules)

4. Implementation Plan

URL –

Creating a Git repository will be a good practice as this will remove the possibility of creating multiple copies of the project which will add confusion among the members. Having a version control system in place will help in keeping the record of the progress of the project and members will also have many feature of Git at their disposal, which might be useful (such as going back to the last changed state if trying something experimental that might break the already stable system).

The work for creating different methods and functionality in the code is divided among the members. This will reduce the work load on a single person. Among all of them <Aman> will be responsible for integrating the implemented systems in the project and ensuring different individual systems communicate with each other properly so that the whole project can function as intended. <Varleen Kaur> will be responsible for creating the command line based UI. All the members are aware of the basic coding standards of Java (such as camelCase for the method names, naming the constants / final variables in capitals) and will make sure to implement them in a rightful manner.

5. Design Considerations

The code base in its current state is just the bare bones and is a basic representation of the project. It exhibits some of the following Object Oriented principles:

- Encapsulation: It is the bundling of data and methods that operate on the data into a single unit (class), and restricting access to some of the object's components. (e.g. encapsulation can be seen in all the classes as the attributes in a class are not accessible by other classes without the use of setter / getter methods.)
- Composition: This is a relationship where a class contains an instance of another class as a member. It implies that the contained object cannot exist without the container object. (e.g. composition is depicted in the class Game where an ArrayList of Player objects is instantiated and the lifecycle of these objects is managed by Game class)