Group Project: Uno - Deliverable 1

Lian Asher Caraang, Mohamed Hamed, Cuong Luong, & Ryleigh Smith

Sheridan College

SYST 17796: Fundamentals of Software Design & Development

Section: 1255-1404

Ricardo Paharsingh

June 10th, 2025

Table of Contents

Team Contract

Team Name: Uno Third

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 submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.

Team Member Name	Signature / Initials	Student ID
Project Leader: Lian Asher Caraang	L.C	991602519
Mohamed Hamed	M.H	991661523
Cuong Luong	C.L	991344404
Ryleigh Smith	R.S	991751180

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.

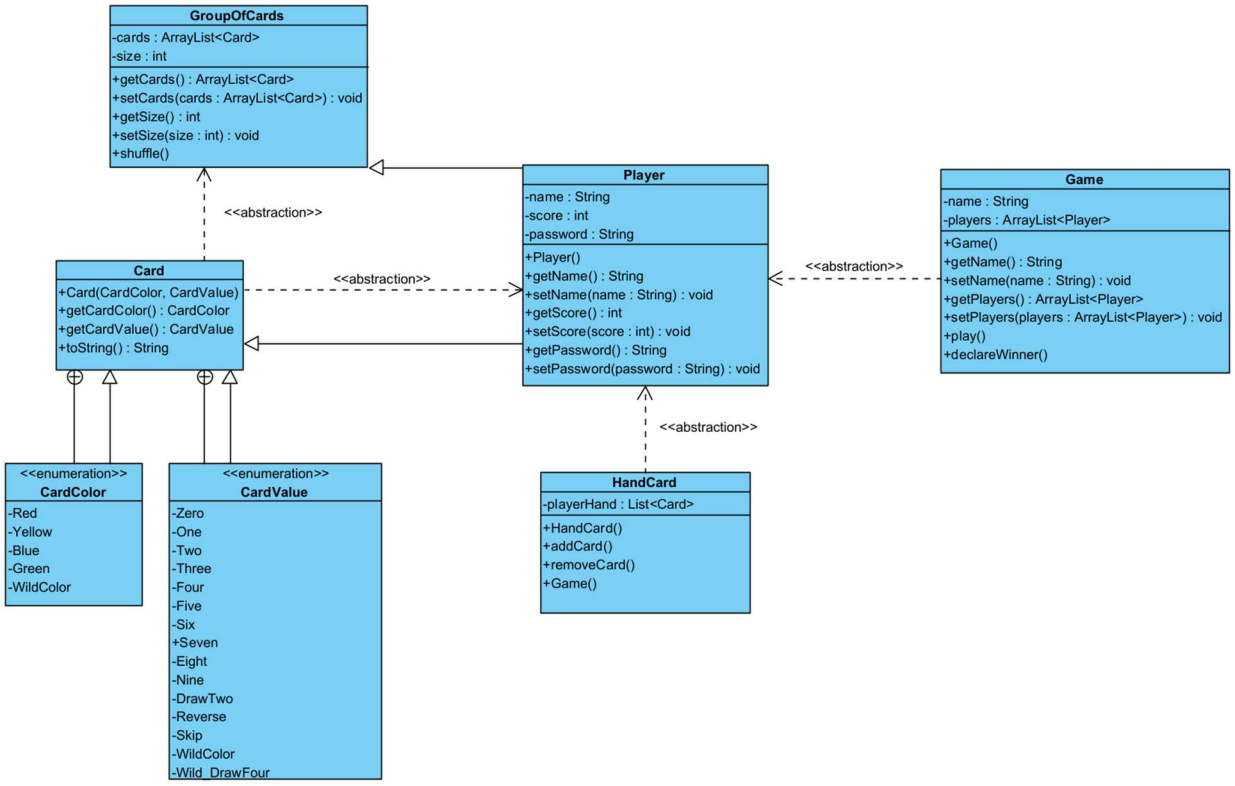
What we will do if:

Scenario	Accepted Initials	We agree to do the following:
Team member does not regularly attend team meetings and/or does not respond to communications in a timely manner.	L.C C.L R.S M.H	Project leader emails the student citing the concerns and cc's the professor, so they are aware of the situation at the very onset.
Team member does not deliver component on time due to severe illness or extreme personal problem.	L.C C.L R.S M.H	a) Team absorbs workload temporarily b) Team seeks advice from professor 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.	L.C C.L R.S M.H	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.	L.C C.L R.S M.H	a) Team absorbs workload (X) b) Team member(s) ask professor to request a Participation Form from all team members. This may result in individualized grades being awarded for a deliverable c) Both a. and b. above 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.	L.C C.L R.S M.H	a) Team agrees to abide by majority vote 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.	L.C C.L R.S M.H	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 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.	L.C C.L R.S M.H	a) Team agrees to avoid use of all vocabulary inappropriate to a business/college setting (X) b) Team attempts to resolve the issue by airing the problem at a team meeting 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	L.C C.L R.S M.H	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 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	L.C C.L R.S M.H	a) Team forces decision sharing by routinely voting on all issues b) Team routinely checks with each other about perceived roles (X) c) Team discusses the matter at team meeting

UML Diagram



Design Document Template

Project Background & Description

Game Selected: Uno

Card Details:

- 108 cards, 25 of each colour; red, yellow, green, blue & 8 wildcards.
- Each colour consists of; 1 zero card, 2 cards of each number (1-9), 2 skip cards, 2 reverse cards, & 2 draw cards.
- As for wild cards, there are 4 “pick colour” cards, and 4 draw four “pick colour” cards.

How to Play: 2 players aim to be the first person to clear their hand (get rid of all their cards). Players take turns adding / removing cards from their hand based on the card played before them.

Rules Chosen: Draw one card when otherwise unable to play.

Number of Players: 2 players, each starting with 7 cards in their hand.

Score: Calculated by how many cards a player has in their hand. Each player has a starting score of 7, goal is to reach 0.

Base Code Explanation

Language: All existing classes, (Card.java, Game.java, GroupOfCards.java, and Player.java) are written in java.

Coding Conventions:

- CamelCase is used.
- All methods, code blocks, and variables are well commented and explained.

Patterns:

- SOLID principals are used, more specifically the “Single Responsibility Principal”. Each class, such as “Card.java” or “Player.java” for example, only focuses on their one task or “responsibility”.
- Encapsulation is used, variables are private and only accessed or modified through the use of getters or setters. This allows for easy error handling and data validation.

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.

- Lian Asher Caraang → Project Leader & Coding
- Mohamed Hamed → Coding & Testing
- Cuong Luong → Coding & UML Diagrams
- Ryleigh Smith → Coding & Report Formatting

The game will have up to 2 players registered and playing at the same time. They will be able to play a full game of UNO with the players' score starting at 7 and lowering based on their hand. The game will display a congratulations message and then stop all operations.

High-Level Requirements

The new system must include the following:

- Ability for both players to register in the game with a username and a password
- All 108 cards will be generated and users provided with a randomized hand of 7 cards
- All the action cards (SKIP, DRAW, REVERSE) must affect the gameplay
- Ability to both draw and remove cards from hand
- Wild cards must function and not cause any errors
- Game must be able to communicate either a win or loss
- Players must have the ability to always know their status (score)

Implementation Plan

Git Repository URL: https://github.com/caraangl/UnoThird.git

Expected Use: Each Developer will check the existing code **WEDNESDAY** and submit their work on **THURSDAY OR FRIDAY** each week and then proceed to check it over **SATURDAY** & **SUNDAY** should changes need to be made in accordance with other developers.

File Locations: Text Files are Stored under Text-Files Branch, UML Diagrams are Stored under UML-Diagrams.

Tools Used: Tools that are expected to be used are **NetBeans** running Java for this project, as well as **Visual Paradigm** used for the UML Diagrams and **GitHub** for all official contributions to the project.

Coding Standards:

- All code should be well commented and be easily understandable for all developers viewing the project.
- Code blocks should be neatly managed, and files should be given meaningful names.
- Error handling should be produced in a way that displays issues clearly to both players and developers.
- Naming style should be CamelCase, example <myVariableName>.

Design Considerations

Encapsulation

- In Card.java, the details of each card value and color are kept private. We use methods like getValue() or setColor() to safely access or change this info.

- In Player.java, a player's ID and their hand of cards are hidden from other classes. We access them using getters and setters, so we don't mess with the data directly.
- In GroupOfCards.java, the list of cards is protected. You can only get or shuffle cards using specific methods like shuffle() or getCards().

Delegation

- In Game.java, cards or players are not managed directly. Instead, we let Player and GroupOfCards handle those parts.
- Player.java focuses on the player, and delegates card storage and logic to GroupOfCards.

Flexibility & Maintainability

- The way the Card class is built makes it simple to add more card types later (like our wild cards, or other special cards).
- Each class is separate and has its own role. This means if we want to change something, like how the game starts or how scoring works, we only need to update one part.
- Classes have single purpose and are cleanly organized. For example, player logic is in Player.java, card logic is in Card.java, etc.