Deliverable 1

Name: Nathaniel

---

1. Project Background and Description

I have selected Uno as my card game where players match cards by color or number and must try to be the first to play all their cards and gather a certain number of points to win the entire game.

1. Each player is dealt 7 cards, and the rest of the cards are flipped on their backside as a draw pile where you draw cards.

2. To play the game Uno, players must try to match the top card in play by color or number, there are special cards that allow players to put down no matter what card is on top, these are considered, special cards, they include; Skip/Block(a player is skipped or block on their turn), Reverse(the order of players turn is now going the reverse), Draw Two(a player has to pick up to cards from the draw pile), wild(allows player to change color of the top card in play), Wild Draw Four(allows player to the change the color and make the next player pick up 4 cards).

3. To Win the Game a player must play all their cards to win the current round. To win the entire game players must gather points from those rounds and let's say 400 points, the first player to achieve 400 points wins the entire game.

Describe starting base code

The starting base code includes abstract and concrete classes within the packages, Key classes are Player (abstract, models players with unique IDs), GroupOfCards (concrete, represents card collections), Game (abstract, models the game), and Card (abstract, base class for cards). The design uses abstract classes and template methods, following Java naming conventions and encapsulation principles, with Javadoc for documentation.

2. Project Scope

Nathaniel: Currently have no group, so I will be responsible for the scope of the project and managing all roles such as; writing the code, implementation considerations, designing UML diagrams, creating and organizing directories on GitHub, testing, running, and verifying all criteria and roles are completed and met for the following progress of the project.

3. High-Level Requirements

- Ability for each player to register with the game.
- Ability to implement Uno rules and real-time updates.
- Ability to keep track of real-time score and card count display.
- Ability to notify players of round and game results win or loss.
- Ability for players to always know their status (score).

4. Implementation Plan

Use: Each developer checks in code at the end of each day or week, ensuring frequent updates and integration are up to date. Will also include folders with the respected content that will be uploaded into the GitHub repository. Will look something like below;

- **/uno_card_game**: Contains all Java source code files.
- **/uml**: Contains UML diagrams and related documentation.
- **/docs**: Documentation for the project.

Git repository URL: *https://github.com/Nathanabra/Uno-Card-Game.git*

Tools: NetBeans, Visual Paradigm, GitHub.

Coding standards:

- Will be using Patterns such as; Abstract Class Pattern and Template Method Pattern
- Coding Conventions such as: Indentations, Comments for documentation and Encapsulation with private fields and public getters/setters.

5. Design Considerations

Key OO principles of the current code;

- **Flexibility and Maintainability:** The started code maintains Flexibility and maintainability by applying abstract classes such as; Player, Game, and Card classes, allowing easy extension and modification, and a flexible design that separates game components, making the codebase easy to manage and extend.

- Encapsulation: The code applies encapsulation by applying private fields with public getters and setters such as; cards in GroupOfCards and playerID in Player.

- Delegation: The code also uses delegation methods such as shuffle in GroupOfCards, which relies on Collections.shuffle, and the play method to be implemented by Player subclasses.