Deliverable 2

Project Name: *UNO Card Game*

Team Members: *Nathaniel*

Date: *July 21, 2024*

---

1. Project Background and Description

Uno is the card game i am developing and it is where players have to 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. Every 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. In order 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).

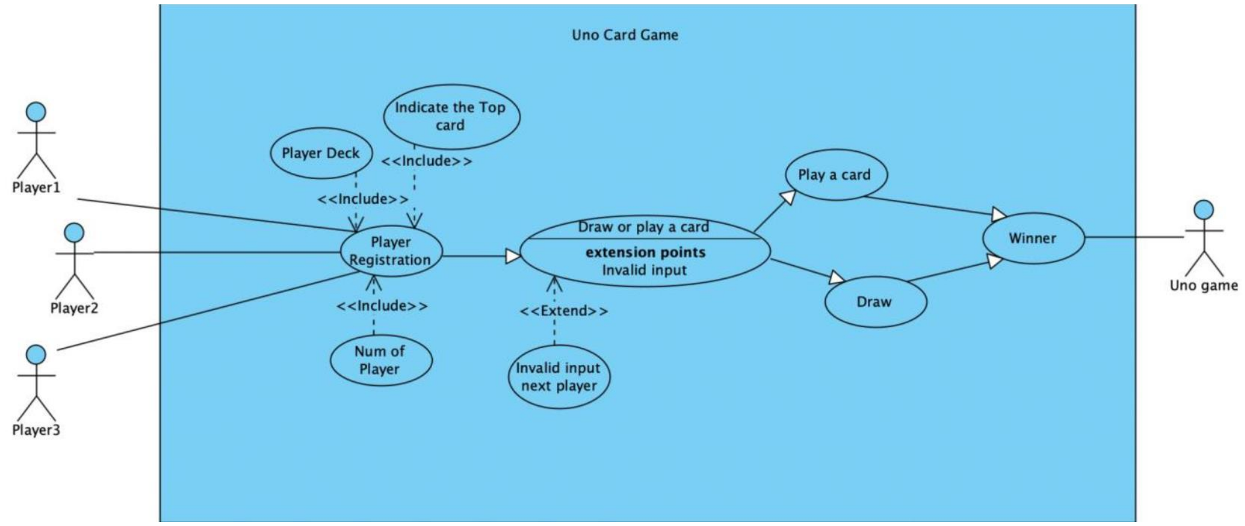
3. Finally in order to Win the Game a player must play all their cards to win the current round. To win the entire game players must gather points from those rounds and let's say 400 points, the first player to achieve 400 points wins the entire game.

2. Design Considerations

Original Design Considerations: ￼

- 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).

Use case designs:



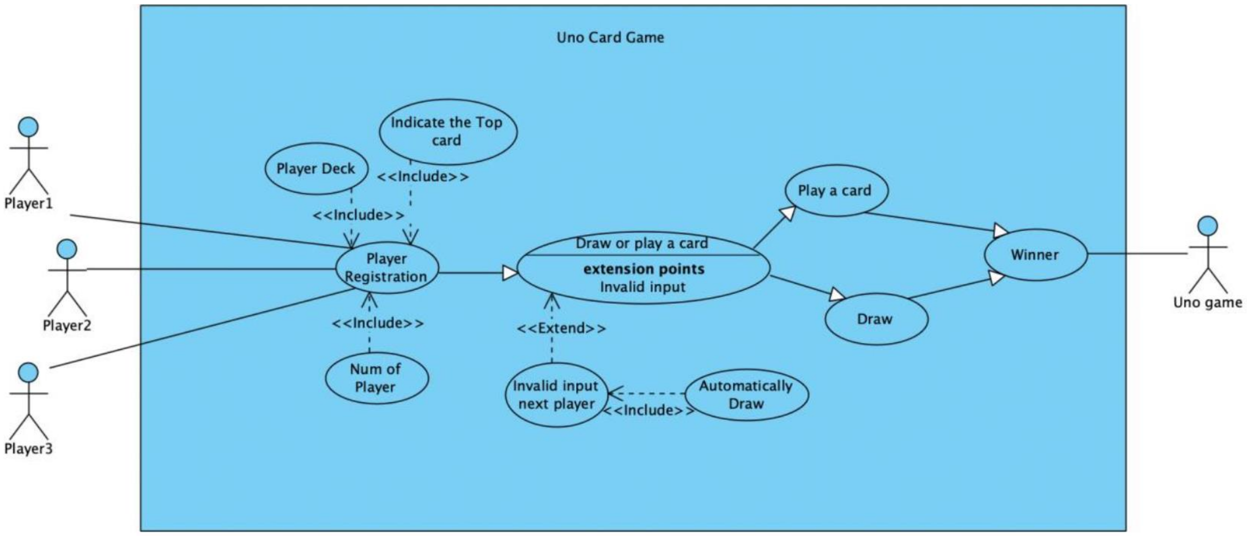
Narrative Scenario (Main path):

1. Players first register the number of players and their preferred usernames.

2. Game provides users with their deck and indicates the current top card.

3. A Player selects a card handed to him, and either plays or draws.

4. Uno Game keeps playing until a user becomes the winner.



Narrative scenario (Alternate path):

1. Players first register the number of players and their preferred usernames.

2. Game provides users with their deck and indicates the current top card.

3. A Player selects a card handed to him, and either plays or draws.

4. If a player inputs an invalid input, they automatically draw a card, and the game moves on to the next player.

5. Uno Game keeps playing until a user becomes the winner.

4. Key New Design Considerations

- **Game Rules**: Introduce and expand the Rule class to manage game rules.
- **Player Registration**: Implement player registration with a Register class.
  The code will continue to maintain:

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

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

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