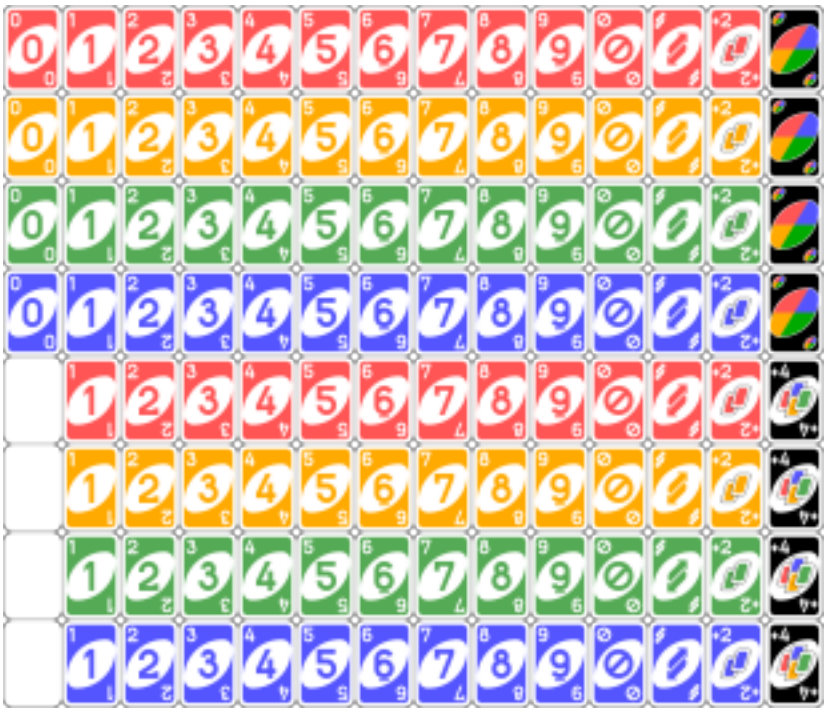
Uno Game Engine

In this assignment, you are required to build an Uno game engine to be used by other **developers** (not players).
First, let us go over some background about the game. Uno is a card game that is typically played by at 2-10 players, and contains 108 cards divided as follows:

- Numbered cards (0-9)
- Action cards (Reverse, Skip, Draw Two)
- Wild cards (Wild, Wild Draw Four)

This is a picture that shows the details of the 108 cards.



Each player is initially dealt 7 cards. Please read the official rules of the game on this Wikihow page, which we will consider to be the basic Uno game. Also, you can find online simulators of Uno that you may try to familiarize yourself further with the game.

Uno game can come with multiple variations from the basic version. For example, some games may have additional action or wild cards with new rules. Another variation is the players are maybe dealt more than 7 cards when they start the game. Also, some Uno games may have some additional penalty rules or different rules on how many cards should be drawn from the draw pile card if a player does not have a legal card to play.

What you need to do?

• First, familiarize yourself with the Uno game and some of its variations. • Next, use Java and OOP to build an Uno game engine, in which developers can build their own variation of an Uno game ☺.

- In your code, there must be an abstract class, called Game. In order to create a new game variation, ideally, a developer will only need to create a new class that extends your abstract Game class, and then do necessary implementation of abstract methods in order to create an Uno Game
- Because we want to minimize programming efforts for creating new games, your code should include predefined set of game rules that developers can choose from when creating their own game variations.
- Inside the Game class, there must be a method called play, which simulates the game. Surely, you can add other methods as well.
- Because you cannot possibly cover all game rules, think of a way that you can allow developers to extend your code and add more game rules to it (or any other features such as new cards or new card dealing mechanisms) with minimal effort.
- There should be a class, called GameDriver, that contains the main method. In the main method, you should only write two lines of code. The first line is to instantiate a game object. The second line is to invoke the play method. Optionally, you may also add additional code lines to use a scanner to get players' names if you want.
- Include in your code at least one real example of an Uno game variation.

What you need to submit:

1. Your source code, i.e., all of your *.java files
2. A report that:
    a. Explains object-oriented design in your code
    b. Explain design patterns used in your code
    c. Defend your code against clean code principles (Uncle Bob)
    d. Defending your code against "Effective Java" Items (Jushua Bloch)
    e. Defend your code against SOLID principles
3. A video that shows a demo, i.e., create an actual game and run it. Also, in the video, explain your design to me and how you succeeded in making your code organized, extensible, and reusable. Do not forget to introduce yourself at the beginning of the video, and briefly talk about your education and experience. Finally, I prefer if you use a camera in the video. Video time should not exceed 10 minutes and I prefer if you keep it under 7 minutes.