

CS 251
Project 3
Unus (Uno with a twist)
Due 4-14-23
Joseph Haugh

Project Description

You are tasked with creating an Uno like game for the command line which I am calling *Unus*, one in latin. *Unus* is a card game where the goal is to empty your hand of cards. Each turn you can play one card in the play area. The first player can play any card to start off with. Every player after that must match the top card in the play area. A match is found when either the color or symbol on the card matches. There are special cards which change the game in various ways. If your hand does not contain a playable card then you must draw a card from the deck. If you still do not have a playable card to play then your turn is skipped. The first player to empty their hand is the winner!

This project emphasizes the following concepts:

- Exceptions
- Inheritance
- Generics
- Collections

Project Specification

Your program must interact with the user using the `interact` function within the *Game* class. If an error message is being printed then use *System.out.println*. You **must** use the starter files and are not allowed to change any existing type signatures, function names or class names. You can add classes and functions if you deem it necessary. Every task that must be completed is marked with *TODO*: in the starter code. In *IntelliJ* you can click the "TODO" tab at the bottom to track these. A specification for each function is given in the starter code, please refer to it for specifics of implementation. You may also notice that the starter code is not up to the coding standards. You must bring all of the code, including functions/classes with no TODOs, up to the coding standards. In so doing I hope you gain a deeper understanding of the code and get practice reading code instead of just writing code.

Deck

The standard Unus deck contains 108 cards as follows:

- 19 red cards
 - 1 zero
 - 2 of every number
- 19 green cards
 - 1 zero
 - 2 of every number
- 19 yellow cards
 - 1 zero
 - 2 of every number
- 19 blue cards
 - 1 zero
 - 2 of every number
- 8 skip cards - two of each color
- 8 reverse cards - two of each color
- 8 draw cards 2 - two of each color
- 4 wild cards
- 4 wild draw 4 cards

Cards

- Number
 - Simply have a color and a number
- Skip
 - Can be used to skip any player other than yourself
 - Skips can be stacked on a specific user to continually skip their turn
- Reverse
 - Used to reverse the order of play
- Draw 2
 - The next player draws 2 cards before starting their turn
- Wild
 - Any color can be played on a wild card
 - Any symbol can be played on a wild card
- Wild Draw 4
 - Same matching rules as a wild
 - The next player draws 4 cards before starting their turn

Differences From Uno

- Skip can be used to skip anyone
- Wild card's color is not chosen by who plays it but rather by the next person to play
- Wild draw 4 can be played anytime
- If you have a play you must play it, you cannot choose to draw a card

Project Starter

I have uploaded a project starter alongside this PDF on canvas. You **cannot** change any type signatures, function names or class names! That means any function that throws an exception must remain that way. Do not listen to IntelliJ if it tells you it doesn't throw an exception that is only because you have implemented it yet. You must complete all the TODO:s as well bringing the code up to the coding standards to complete the assignment.

Submitting Your Project

You must submit a zip file containing only your java files. There should be no subfolders and no packages. If any other file is included you will be docked points.