SYSC 3110 Project – Uno Flip card game!

The Objective of our team project is to create a version of the Uno Flip card game. For those unfamiliar with the game, you can experience the classical Uno card game online. The rules are available on Wikipedia, and you can also find the Uno Flip rules through this link. Additionally, we've included a helpful video tutorial on how to play Uno Flip.

Our project is divided into four distinct iterations, each ending with a milestone corresponding to deliverables that will be evaluated. Constructive feedback from the Teaching Assistants (TAs) in one iteration will be applied to enhance the subsequent iteration's progress.

Milestone 1:

We aim to develop a text-based, playable version of the Uno card game, allowing players to engage through the console using keyboard input. Within this version, players will have the capability to:

1. View their drawn cards.
2. Place cards using the official notation as detailed in the Wikipedia link.
3. Draw one card.
4. Execute actions associated with special cards, including Reverse, Skip, Wild, and Wild Draw Two cards.
5. Observe the resultant state of the cards, presented in text format.

The primary challenge in implementing this game lies in ensuring the correctness of card placement, meaning that all cards formed after placement must be validated and added to the score. While this milestone may involve incomplete implementation, we will ensure to document the remaining tasks to be addressed. It's important to note that support for the Uno Flip variation and its corresponding rules will be addressed in Milestone 3. (Hint: you are encouraged to use Enum to define colors or card types. This can avoid messy use of string representations everywhere.)

Also required: the UML modeling of the problem domain (class diagrams with complete variable and method signatures, and sequence diagrams for important scenarios), detailed description of the choice of data structures and relevant operations: you are providing an initial design and implementation for the Model part of the MVC. Do not worry about any GUI yet.

1. **Deliverables**: readme file (see explanation below) + code (source + executable in a jar file) + UML diagrams + documentation, all in one zip file.
2. **Deadline**: Sunday Oct 22.

SYSC 3110 Project – Uno Flip card game!

Milestone 2:

We are progressing to develop a GUI-based iteration of the game, which will include the addition of a View and Controller. This version of the game will feature a graphical display within a JFrame, and user interaction will be facilitated through mouse input. While we have flexibility in making other GUI-related decisions, we will ensure a user-friendly and visually appealing interface. (Note that, for GUI, do not move to next player right away, use a button click to do so. This will make the marking easier for TAs to see what happened before the GUI update.)

Furthermore, our development will encompass the creation of unit tests for the Model component. Specifically, we will design tests to cover any areas within the game logic that are either missing or incomplete, particularly focusing on the card placement, scoring, and penalty aspects. These unit tests will be structured to initially fail, providing a robust framework for validating the game's functionality as it evolves.

1. **Deliverables**: readme file + design + corresponding tests + code + documentation, all in one zip file. In particular, document the changes you made to your UML and data structures from Milestone 1 and explain why.
2. **Deadline**: Sunday November 12th.

Milestone 3:

At this stage, our project requires that all tests pass, ensuring the accurate functionality of card placement and scoring. In addition to these core functionalities, we will introduce the following enhancements:

- **Uno Flip Integration**: We will incorporate Uno Flip cards into the game, implementing their specific rules and scoring mechanisms.
- **AI Player Capability**: The game will now accommodate an arbitrary number of "AI" players. The behavior of these AI players will be designed for flexibility, allowing for different strategies. One potential strategy is to generate all possible legal moves (or a subset if processing time is a concern) and select the highest-scoring move to simulate intelligent gameplay. Alternatively, you can simply use rules such as “The first valid card will be placed”.

SYSC 3110 Project – Uno Flip card game!

1. **Deliverables**: readme file + code + corresponding tests + refined design + documentation. The program must work robustly, and the code must be "smell-free" (we will be hunting for smells!). Make sure that you document the changes since the last iteration, and the reason for those changes.
2. **Deadline**: Thursday November 23ed.

Milestone 4:

In addition to the existing features, we will be incorporating two crucial functionalities into our project:

- **Redo capabilities:** Players will now have the ability to redo/ undo their moves within the game.
- **Reply Capability**: Players will now have the ability to replay the game, allowing them to restart and enjoy multiple rounds.
- **Save/Load Features**: We will implement a save and load system.

You may use Java Serialization. This will enable players to save their current game progress and later resume it. Additionally, customization options can be defined and stored in XML or JSON formats, providing a flexible and user-friendly way to personalize the game experience to their preferences.

1. **Deliverables**: readme file + code + tests + documentation. Your project should be well packaged, and the program(s) should be easy to install and run.
2. **Deadline**: Friday December 8th.

General Requirements:

Milestones must contain all necessary **files and documentation**, even those items that are unchanged from previous milestones. Missing files cannot be submitted after the deadline, no exceptions. Verify that your submission contains all necessary files (in particular, don't forget to include your source code!) before submitting on Brightspace.

The "**readme**" file, listed as a deliverable for each iteration, is typically a short text file that lists and describes: the rest of the deliverables, the authors, the changes that were made since the previous

deliverable, the known issues (known issues are graded less severely than undocumented ones!) and the roadmap ahead.

"**Documentation**" includes up-to-date UML diagrams, detailed descriptions of design decisions made, complete user manuals, and javadoc documentation.

Note that nobody is stopping you from working ahead of schedule! In fact, iteration i+1 will very often give you good insight into iteration i.

This is a **team** project. Each team should have 4 or exceptionally 3 members. A project's success obviously depends on contributions from each member! So divide the work and cooperate. **Each contribution (source code, documentation, etc.) must contain the name of its author**: we will use this to determine whether there is any significant difference in the quality and quantity of the contributions of the team members. If any such difference is detected, the individual grades will be adjusted accordingly.

You are expected to use **Github** to manage your project (version control, issue-tracking, etc...), but the deliverables for each milestone should also be submitted for marking on Brightspace. Please use a private Github team repository; Github provides this for free for student accounts. The TA assigned to your project should be invited as a member of the Github team, so that they can track your use of the tool, verify that all group members are contributing, and their feedback will be given by opening new Github Issues.

The marking scheme for each iteration will be comprised of completeness of the deliverables, the quality of the deliverables, and, for iterations 2 to 4, we also look at whether you took into account the feedback you received from the TA in the previous iteration.

Copyright matters!

Note that Uno card game is a commercial product, and there's copyright attached to it. It is one thing to use copyright material under "fair use" for educational purposes, it would be quite another to profit from someone else's work by offering a copycat version. The law is complex on that front, so please exercise due diligence if you want to make your Github repo public once the course is over.