**UNO Game**

Names:

Kaleel Evans

Student ID: 991638770

Edwin Johnson

Student ID: 991667321

Liliana Mantilla Guevara

Student ID: 991633617

College: Sheridan College

Professor name: Sivagama Srinivasan

Course name: SYST 17796 Fundamentals of Software Design and Development

Date: 10/06/2022

Table of Contents

[**UNO Game** 3](#_Toc105796843)

[1. Project Background and Description 3](#_Toc105796844)

[1.1 Project Goals: 3](#_Toc105796845)

[1.2 Vision final: 3](#_Toc105796846)

[1.3 Description and rules of UNO game: 3](#_Toc105796847)

[1.4 Description of the code: 4](#_Toc105796848)

[2. Project Scope 4](#_Toc105796849)

[3. High-Level Requirements 5](#_Toc105796850)

[4. Implementation Plan 5](#_Toc105796851)

[5. Design Considerations 6](#_Toc105796852)

[**Appendix** 9](#_Toc105796853)

[1.1 Team Contract 9](#_Toc105796854)

# **UNO Game**

# Project Background and Description

# Project Goals:

* To design and develop a code that permits playing a UNO game.
* To create a code that is simple, functional and accomplished with secure programming rules.

# Vision final:

It has the finality to create a functional UNO game, which will develop with the force of the group team.

# Description and rules of UNO game:

UNO is a multi-player card game which has as the goal of the game is to empty a player's deck of cards. The game is formed by regular number cards 1 to 9, which can be the colour red, green, blue or yellow. The player can only play the card when the current card number or colour is the same if the player does not have a card that matches in their hand. The player needs to draw a card and add their hand. The player with only a card in his hand must yell "UNO."

Rules:

* Every player starts their hands with 7 cards.
* The player can only play the card when the current card number, colour, or symbol/action is the same.
* If the player does not have a card to match needs draw a car.
* The player with only a card in his hand must yell "UNO".
* Reverse cards can only be played on other reverse cards or same colour cards, and they change the order of the game.
* Skip cards; when a player plays this card, the next player loses the turn.
* Wild This card represents all colours. It can be placed on any card.
* Draw two; when a player plays this card, the next player must pick up two cards and forfeit the turn.
* Draw four; when a player plays this card, the next player must pick up two cards and forfeit the turn.

# Description of the code:

The current code is developed using Java’s Object-Oriented Programming, which works with abstractions like objects, classes and variables that represent code and processes more complex and deletes the duplicities in the code. Also, the code uses the encapsulation method to preserve its integrity.

# Project Scope

The Team Members:

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Roles** |
| 1 | Liliana Mantilla Guevara | Leader  Programmer |
| 2 | Kaleel Evans | Programmer |
| 3 | Edwin Johnson | Programmer |

Uno game will be developed using the Java language, which works on a local machine with NetBeans Editor and is stored in a GitHub repository. The game interface was though with a simple interface that permits the players to interact simply with the application.

It will be considered that the project is a finished success when it accomplishes the below parameters:

1. The programmers considered that the code was finished and the application testing super a sample considerable play successfully.
2. The overall code is appropriately documented.
3. The project documentation is totally made.
4. All the information is submitted on GitHub.

# High-Level Requirements

The application must permit:

* The user registry their name, and the program shows the user's name any time the program generates a message or task for the user.
* The program shows a message announcing if the player wins or loses.
* The application shows a constant count of the number of cards the player.

# Implementation Plan

Git repository URL: <https://github.com/K4NZEON/GameDEvs-Uno-Deliverable>

All the team members will use the Git Hub repository to maintain and update the project as required. We are trying to follow the best coding standards by making our code flexible to use and easy to manage and implementing proper methods for getting the best output possible. The tools we are currently using are NetBeans for coding and Visual Paradigm for making a visual representation of the codes in the form of diagrams for a better understanding of the structure of the code.

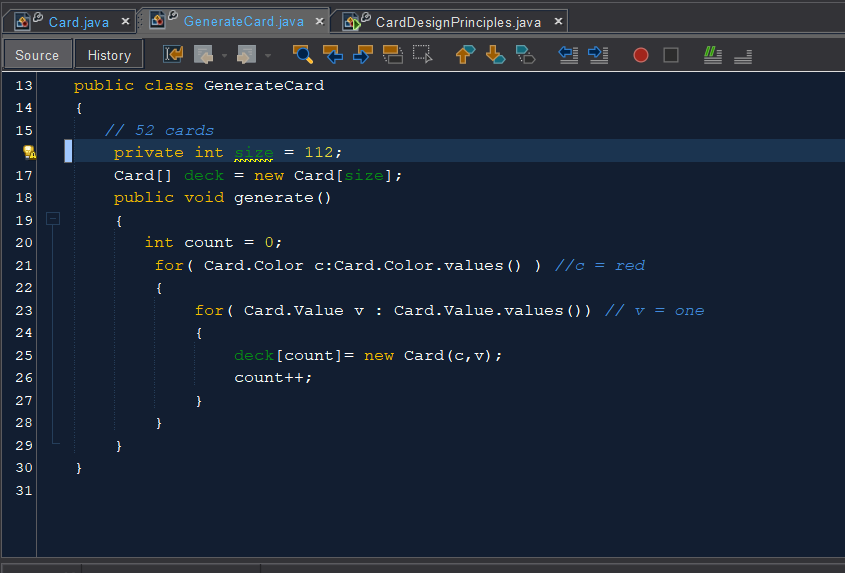
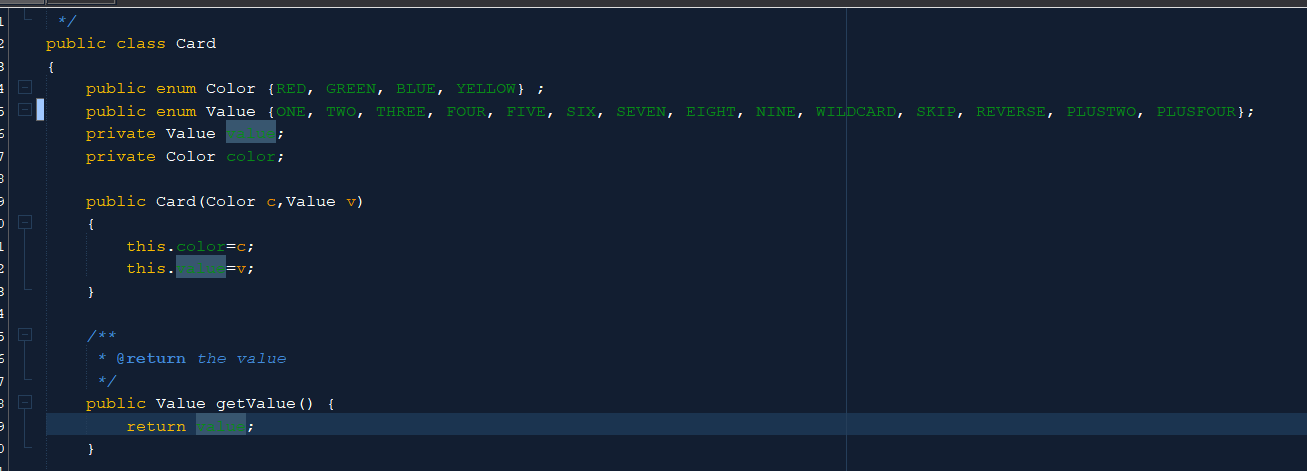
UML about basic game code:

Diagram

Description automatically generated

# Design Considerations

The current code is structured in a way that allows it to be easily updated based on the requirements. The code has implemented OO principles like Encapsulation in order to avoid loose coupling by using enum function instead of String, as shown in the following example:



**Text

Description automatically generated**The code is also made flexible by using different methods, and it can easily change or fix any issues when needed. Additionally, the rules of cohesion have also been followed. The code has been split into the classes in which they are doing what they are supposed to do.

Below is a possible appreciated example of the cohesion principle that before has been mentioned. As you can see, the primary objective of this class is just to generate the cards.

# **Appendix**

# Table Description automatically generated1.1 Team Contract