***SYST 17796 TEAM PROJECT***

***Group 3***

***Deliverable 1***

***Table of Content***

* ***Team Contract***
* ***UML Diagram***
* ***Design Document Template***

|  |  |
| --- | --- |
| SYST 17796 TEAM PROJECT  Team Name: Group 3 |  |

*Please negotiate, sign, scan and include as the first page in your Deliverable 1.*

Team Contract

**Please note that if cheating is discovered in a group assignment each member will be charged with a cheating offense regardless of their involvement in the offense. Each member will receive the appropriate sanction based on their individual academic integrity history.**

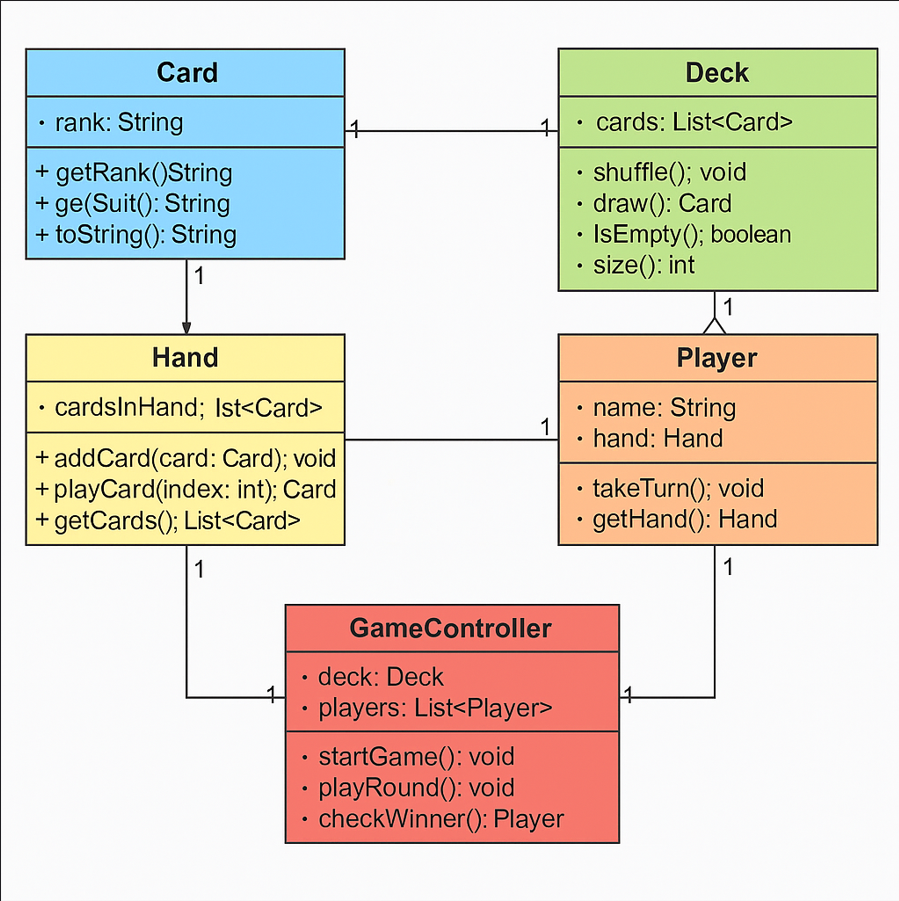
**Please ensure that you understand the importance of academic honesty. Each member of the group is responsible to ensure the academic integrity of all of the submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.**

|  |  |  |
| --- | --- | --- |
| Team Member Names (Please Print) | Signatures | Student ID |
| Project Leader: Prabhjot Singh Aulakh | PrabhjotSinghAulakh | 991802100 |
| Harjot Kaur | HarjotKaur | 991801308 |
|  |  |  |
|  |  |  |

**For further information, read Academic Integrity Policy here :** [**https://caps.sheridancollege.ca/student-guide/academic-policies-and-procedures.aspx**](https://caps.sheridancollege.ca/student-guide/academic-policies-and-procedures.aspx)

By signing this contract, we acknowledge having read the Sheridan Academic Integrity Policy

* ***UML Diagram***



* ***Design Document Template***

1. **Project Background and Description**

The card game Uno serves as the inspiration for our idea. Players in the turn-based card game Uno match cards according to color or number and use special action cards to alter the rules of the game. Being the first to use all of your cards is the aim.  
  
We are use the Java basic code that has been provided, which has classes like:

* Card: represents an individual playing card.
* Deck: a set of cards, dealing logic, and shuffle.
* Hand is their collection of cards.
* Player: holds the name and hand of the player.
* GameController: controls the course of the game.

In order to incorporate Uno-specific elements like Wild cards, Reverse, Draw Two, and Skip logic, we will restructure and expand this code. We will make sure that the fundamental gameplay rules are upheld by using Java object-oriented programming.

1. **Project Scope**

**Team Members:**

* Prabhjot Singh Aulakh – 991802100 – Git setup, player logic, testing
* Harjot Kaur – 991801308 – UML diagram, GameController logic, documentation

**Scope**:

Our goal is to use object-oriented Java to create a text-based, playable version of Uno. The project will manage turns, apply card rules, accommodate two or more players, and select a winner. The game has no graphical user interface and is solely console-based.

Measures of project completion will include:

* An Uno game that is totally playable
* Enforcing the rules accurately
* The ability to sign up players
* When a hand is empty, a winner is declared.

1. **High-Level Requirements**

The new system needs to:

* Permit each participant to sign up using their name.
* Deal and shuffle a deck of cards.
* Permit players to see and use legitimate cards.
* Enforce Uno-specific regulations (special cards, color/number matching).
* Turns can be properly skipped or reversed.
* When a player has no more cards, declare them the winner.
* Display each player's standing and score throughout the match.

1. Implementation Plan

Git Repository: <https://github.com/Aulaprab/SYST17796--GROUP-PROJECT.git>

**Tools and Standards**:

* IDE: NetBeans
* GitHub for version control
* UML: Draw.io or Visual Paradigm

1. **Design Considerations**

**Encapsulation**

Private fields with public accessors are used by every class. Encapsulation is enforced by the Card and Hand classes to safeguard internal data.  
  
**Delegation**

Actions are delegated to Player and Deck by GameController. For instance, instead of controlling the hand directly, it asks the player to take a turn and play a legitimate card.  
  
**Flexibility/Maintainability**

The game can be readily expanded to accommodate additional players or future variations by separating the logic of the Card, Deck, and Player. We'll make sure that every class is loosely connected and adheres to single responsibility.