|  |  |
| --- | --- |
| SYST 17796 TEAM PROJECT  Team Name: Tech Titans |  |

*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: Dharm Patel |  | 991742403 |
| Krishna Gediya |  | 991741064 |
|  |  |  |
|  |  |  |

**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

**Project Background and Description**

* **Project Goal:** The goal of this project is to extend the starter code for a basic card game framework in Java to fully implement the popular game UNO. This project involves applying object-oriented principles and following a structured development process.
* **Game Description:** UNO is a card game where each player tries to be the first to discard all their cards. Players can play cards that match the number or color of the card on the top of the central pile. Action cards such as "Skip," "Reverse," and "Draw Two" add strategic elements to the game.
* **Base Code Overview:** The base code includes classes for Card, Player, Game, and GroupOfCards. These classes provide the skeleton of a card game. We have extended these classes by implementing an UNOCard, UNOPlayer, and UNOGame class to incorporate UNO-specific rules and mechanics.

**Project Scope**

* **Dharm Patel - Lead Developer:**  
  Responsible for designing and implementing the core functionality of the UNO game. This includes extending the base code, implementing game rules, and ensuring that the object-oriented design principles are followed.
* **Krishna Gediya - Tester & Git Manager:**  
  Handles writing and managing the test cases for the project using JUnit. They are also responsible for managing the Git repository, ensuring proper version control, and coordinating merges and branch management.

**Technical Scope:**  
The project will include a text-based console interface where players can interact with the game by drawing cards, playing cards, and viewing the game status. The project will be considered complete when the game can be played in full, with multiple players and the correct enforcement of UNO rules.

**High-Level Requirements**

* Players must be able to register for the game.
* The game should correctly display the current card in play and the cards in each player's hand.
* Action cards (e.g., Skip, Draw Two, Reverse, Wild) must work according to the rules of UNO.
* The game must communicate when a player wins or is forced to draw cards.

**Implementation Plan**

* **Git Repository:**The project repository is managed on GitHub( <https://github.com/DharmPatel693/Uno-Project.git> ), with each team member checking in code at the end of the day or after completing a feature. We follow the feature-branch workflow to avoid conflicts.
* **Development Tools:**
  + Java as the primary language.
  + Eclipse for code development.
* **Coding Standards:**

We follow standard Java coding conventions, using camelCase for variables, PascalCase for class names, and appropriate commenting practices. The project structure is as follows:

* **/src:** Contains the source code files.
* **/test:** Contains unit tests.
* **/doc:** Contains UML diagrams and design documents.

**Design Considerations**

1. **Performance**

* **Shuffling the Deck:**The deck shuffling method is efficient for a typical UNO game, though future optimizations could enhance performance for larger card sets.
* **Game Loop Efficiency:**The turn-based game loop is simple and scalable. Adding a time limit per turn could improve performance in extended games.

1. **User Interaction**

* **Text-based Interface:**The current console interface is functional but could benefit from a graphical interface for improved user experience.
* **Input Error Handling:**The system needs robust input validation to handle incorrect actions (e.g., playing invalid cards) without breaking the game.

1. **Modularity**

* **Separation of Responsibilities:**The design assigns clear roles to classes (e.g., UNOCard, UNOPlayer), ensuring changes in one module don’t affect others.
* **Extensibility:**The game logic is easily extendable, allowing for the addition of new rules or card types with minimal changes to the existing structure.

1. **Error Handling**

* **Card Validity Checks:**The game enforces rules to ensure that only valid cards can be played, maintaining game integrity.
* **Game Flow Integrity:**The game structure ensures smooth progression from turn to turn, preventing issues like skipped or out-of-order play.