SYST 17796 PROJECT

Team name: SANB

Team members: Khang Bui, Tin Nguyen

1. *Project Background and Description*

   Goals and Vision:

   The goal of the project is to develop a fully functional Blackjack game application. The game allows player to play against a dealer, following the standard rules of Blackjack. The vision is to create an enjoyable gaming experience with user-friendly interface for player interaction.

   Gameplay Description:

   Blackjack is a popular card game played between a player and a dealer. The objective is to beat the dealer by having a hand value closer to 21 without exceeding it. Players can draw additional cards ("hit") to improve their hand or choose to keep their current hand ("stand"). The game ends when either the player or dealer busts (exceeds 21) or when both have finished drawing cards, and the hand values are compared.

   Base Code Description:

   The base code is written in Java and follows Object-Oriented Programming (OOP) principles. It consists of classes representing different components of the game:

   - BlackJack: The main class to start the game and manage game flow.
   - Card: Represents a playing card with a suit and value.
   - Deck: Manages a deck of cards, allowing drawing and shuffling.
   - Game: Manages the game state, including player, dealer, deck, and rounds.
   - Player: Represents a player in the game, keeping track of their hand, score, and chips.
   - Dealer: Similar to Player, but specific to the dealer.

2. *Project Scope*

   Team Members:

   - Khang: Responsible for coding and implementing the game logic.

- Tin: Responsible for quality checking, UML diagram creation, and documentation.

Technical Scope:
The project aims to implement a functional Blackjack game with the following features:

- Player registration and management.
- Game flow management, including rounds and betting.
- Communication of game outcomes (win/loss).
- Real-time status updates for players (score, chips).

The project will be considered complete when the implemented game meets the specified requirements and functions without errors.

3. *High-Level Requirements*

Requirements:

- Player registration system.
- Game outcome communication.
- Player status will be updated after each round.

4. *Implementation Plan*

Repository URL: https://github.com/JesseBui/BlackJack.git
Expected Use: We are going to push the code to Git Hub, then we will start the quality control procedure to see if there are any quality-of-life changes that we can add or features that were not intended. We will then push the new code to Git Hub.
Coding Standards and Tools:
Coding standards will follow industry best practices and Java conventions.
Tools: VSCode for coding, Visual Paradigm for creating UML diagram.

5. *Design Considerations*

Encapsulation:

- The "Player" class encapsulates player data and behavior, including hand management and chip handling.

- The “Deck” class encapsulates card management, providing methods for drawing and shuffling cards.

Delegation:

- The Game class delegates responsibilities for game management to other classes, such as “Player”, “Dealer”, and “Deck”, for specific tasks like starting rounds, managing hands, and shuffling cards.