Jinghu Lei

CS591 D1

Blackjack Object Specification

*Game Dynamics*

1. Game

Important Methods and Variables

private Player player;

private Dealer dealer;

private int round;

public static Game start() {} //entry-point

private void play(Scanner sc) {}

public Game(boolean ai, Scanner sc){} // constructor

Player is the only player for an instance of the game. It is initialized on Game.start() which creates a new instance.

Summary

The main component and access point to the game. The game controls the rounds and entire blackjack logic. It starts the game and asks for the number of players. Then it decides whether to use an AI or not for the dealer. Then it asks the player for input until the player is done and switches to having the dealer do their actions. At the end of the round, it compares the hands of the dealer and player (with the exception of a bust, which will stop it early) and selects a winner.

*Players*

1. CardPlayer

The abstract class and parent of Player and Dealer.

This class contains methods for checking bust, stand, and restarting the round. It also has a main constructor for setting the name and new hand for the player. It also has a helper toString function for returning the hands of the players as a string. These are shared between player and dealer, but each implement it differently.

1. Player

The actual player class, which controls actions after the input. It inherits from CardPlayer and uses its constructor and toString methods. As a CardPlayer it also uses the stand and bust methods to check if any more actions can be taken by the player. It has a money attribute which keeps track of if the player can still bet or split/double etc. All of the player actions are implemented as instance methods and use an array of Hand elements. It uses an array because once it splits a hand, it is played as two hands so a sets of logic is needed for each hand.

1. Dealer

The dealer class, which distributes the cards and plays against the player. It also inherits from CardPlayer. The dealer class has a deck for which it refreshes and shuffles after each round. It also uses the stand and bust methods provided by CardPlayer to decide if the dealer can keep going. It has additional hit, dealCard, and dealSelf methods for distributing cards to the correct player and hand. For a live dealer, it has a turn() method which starts by flipping over the hidden card. Then in the Game instance, input is taken and actions are performed by the dealer.

1. Computer

Extends dealer and performs actions automatically. The only method it uses is the turn method which was in Dealer. It gives input automatically to hit until it is at value 17 or over.

*Deck*

1. Card

The card class is just a model of a playing card. It has a suit and value and has a toString method to return a view of such. Ex. 2 of Hearts is [2H]. Card also has a visibility option for the dealer to have his facedown card and if it tries to return a string of a face down card just gets [??].

1. Hand

Hand contains an ArrayList of Cards. It starts empty and is filled through the dealer. The hand has the logic for determining if the hand is bust or has a blackjack. It has all the logic to compute a value, even with an Ace. If it has an Ace and the value is over 10 without the ace, then the Ace value is 1, else 11. The Player and Dealer utilize Hand.done() to see if anymore actions can be performed on their hand/hands.

1. Deck

The Deck is also an ArrayList of cards. It is initialized with a full deck (all the suits and values). Every round, the dealer gets another full deck and shuffles the deck.