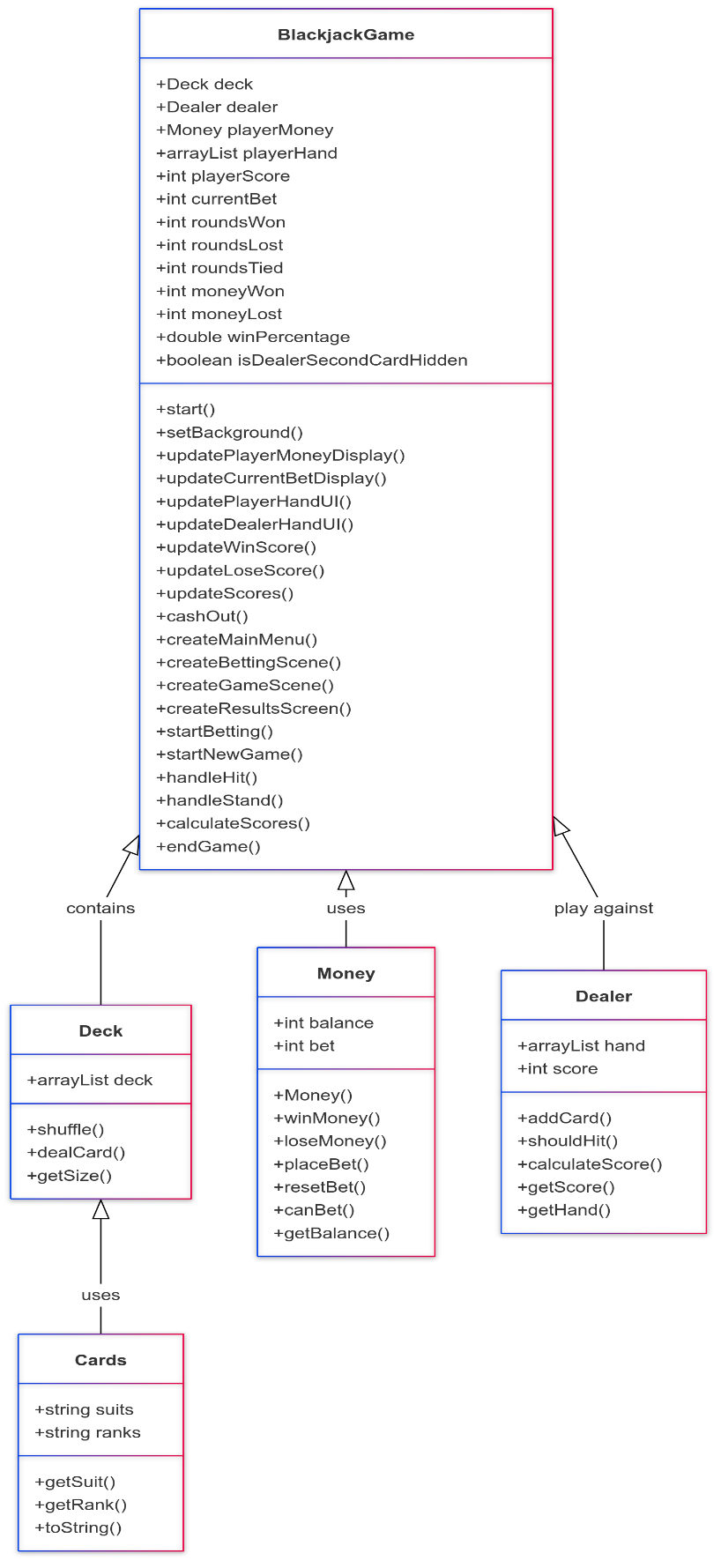
Implementation Manual

Cole Smolinski



* I started by creating the Cards class, which handles the creation of a card given a String suit and a String rank, these are implemented into the Deck class later to fully populate the deck of 52 playing cards
* Next was the Deck class, which creates a List of cards based on the typical playing card suits and ranks. So it creates a full list of Spades, Clubs, Diamonds, and Hearts, all with ranks Ace through King. It then also handles the shuffling of the deck, which just randomizes the list. Additionally, I added a used card list which keeps track of all the cards that have been played, then a replenish method to add those used cards back into the deck.
* Next is the money class, which keeps track of the player’s balance and the bet that the player made for that round. It also handles the winning and losing of money, as well as checking if the player is able to bet the amount that they input.
* Next is the dealer class, which handles the Dealer’s AI. This keeps track of the dealer’s hand and their score. It also then allows the adding of cards to the dealer’s hand, calculating the dealer’s score, and accessors for these.
* Finally is the BlackjackGameApp, which is the application itself that handles much of the game logic and the GUI. Using the other classes, it creates a new deck, new dealer, and a new amount of money. It also keeps track of the player’s hand and their score as well as the number of rounds won, lost, and tied, the player’s score, the total amount of money won and lost, the win percentage, and if the dealer’s card should be hidden. It also then creates many of the scenes using JavaFX as well as the displays and UI for the information to be displayed to the player. It handles many of the functions of the buttons and the results of the rounds by comparing the player’s score to the dealer’s score.