**Developer: Kris Nickeson Project: Blackjack / Game of “21”**

1.0Background information / current status:

A well-known card game throughout the world, I will be implementing a single-player version of Blackjack (21) where the player is opposed by the dealer (PC). This version will be text-based and follow the 'Bicycle' standard rules. It must be played from a Terminal Window with a starting chip balance of $25 and max accrued winnings of $500 ($500 + Player's original $25 = $525 MAX). Play continues while the player still has a balance in his/her account, or until they choose to exit the game at the prompt or manually exit the game (Ctrl-C).

2.0 Goals and Objectives (Requirements):

**Goal 1: Class and Object Development**

Objective 1.1: Player (Parent Class) Object Development

Objective 1.1.1: Implement Fields (Chips Total, Cards, Cards Total, Player Name)

Objective 1.1.2: Implement Player Methods (see Game Logic)

Objective 1.1.3: Implement Inheritance Schema - max flexibility for future development

Objective 1.1.4: Implement Access Level Controls - max flex for future development

Objective 1.2: Dealer (Sub-Class) Object Development

Objective 1.2.1: Implement Fields (Chips Total, Cards, Cards Total, Player Name)

Objective 1.2.2: Implement Dealer Methods (see Game Logic)

Objective 1.2.3: Implement Inheritance Schema - max flexibility for future development

Objective 1.2.4: Implement Access Level Controls - max flex for future development

Objective 2.1: Shoe of Cards (Parent Class) Object Development

Objective 2.1.1: Use collection - ArrayList of Cards - Number of Decks: 6)

Objective 2.1.2: Re-shuffle Marker – (randomNum generates value between 65 and 90 cards from end of deck – marker reached triggers shuffle of all 6 decks

Objective 2.2: Deck (Sub-Class) Object Development

Objective 2.2.1: Four Suits – Hearts, Diamonds, Spades, Clubs

Objective 2.2.2: Card Values: Ace through King (no Jokers). All face cards = 10pts

Objective 2.2.3: Aces – One or 11? (Both – adjust counting logic accordingly, 'Soft 17')

Objective 3.1: Chips Object Development

Objective 3.1.1: Implement Fields – Values ($1, $5, $10, OverMAX, OUT)

Objective 3.1.2: Implement Chips Methods (Count, Bet, Gain/Loss, OverMAX, OUT)

Objective 3.1.3: Player's Chips Total Value Storage (Memory Resident)

**Goal 2: Game Logic Development**

Objective 1.1: Game Initialization Methods

Objective 1.1.1: Default player to $25 in chips

Objective 1.1.2: Shuffle 6 decks and set or re-set the 're-shuffle marker' w/randomNum Objective 1.1.3: Deal Cards + Print opening Player / Dealer card values

Objective 2.1: Player's Available Methods

Objective 2.1.1: Acquire / Parse Bet Input

Objective 2.1.2: Acquire / Parse Player's Move (Hit=H / Stand=S)

Objective 2.1.3: Check Card Count (overloaded method)

Objective 2.1.4: Check Chips Count (overloaded method)

Objective 3.1: Dealer's Available Methods

Objective 3.1.1: Deal Cards (During Game Initialization)

Objective 3.1.1: Dealer action (hit / stand)

Objective 3.1.2: Message to Player re: Dealer's action/choices

Objective 3.1.3: Check Card Count (overloaded method)

Objective 3.1.4: Check Chips Count (overloaded method)

Objective 4.1: Console Messaging Methods

Objective 4.1.1: Intro script: Welcome, Chip Balance notice, List min/max bet values

Objective 4.1.2: Bet script: Force values to $1 / $5 / $10 only

Objective 4.1.3: Play Script: Request Player Action (Hit / Stand)

Objective 4.1.4: Outro script1: Win/Loss + Chips Total Message

Objective 4.1.5: Outro script2: (Player > $0) → Play Again? → Parse → <LOOP or END>

Objective 4.1.6: Outro script3: Player >= $525 → Special message <END>

Objective 5.1: Misc Methods

Objective 5.1.1: Bet Settlement ('Natural' or 'BlackJack' = 1.5x original bet payout)

Objective 5.1.2: Shuffle once marker is reached, assuming game is still in play

Objective 5.1.3: Exchange Chips for other values (automatic / assumed → no coding)

**Goal 3: Testing and User Feedback**

Objective 1.1: Testing

Objective 1.1.1: Verify game play is accurate under all play conditions / options

Objective 1.1.2: Gain user feedback on game play for future game enhancement

Objective 1.1.3: Brainstorm additional features / development options

3.0 Stakeholders:

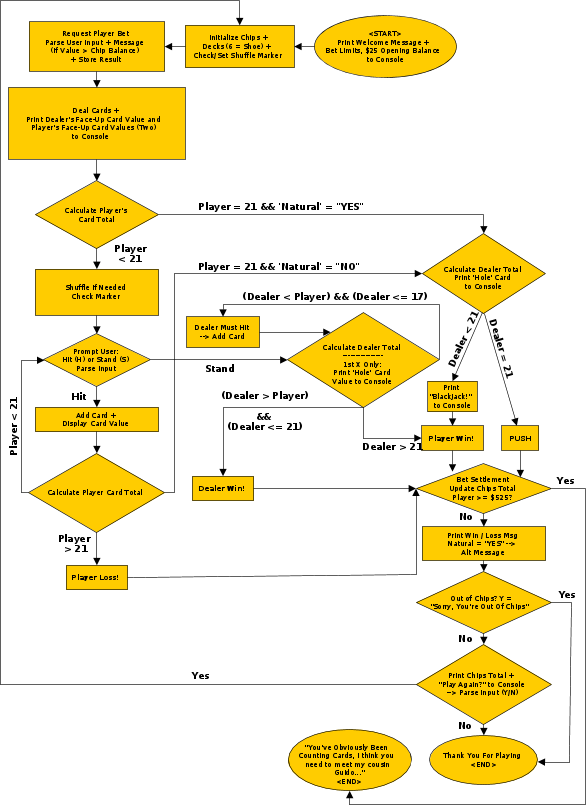
|  |  |
| --- | --- |
| **Name** | **Feedback** |
| Kris Nickeson | 1st S&R and PFD Complete |
| James Camire | Insert Feedback Here |
| Mike Downard | Insert Feedback Here |

4.0 Project Scope (Work to be performed):

* Develop all Objects / Classes
* Develop Game Logic
* Verify / Debug / Troubleshoot
* Get User feedback
* Ideas for Upgrades / Future Development

5.0 Dependencies

* Requires Java8SE / JVM Installed
* Console-based, requires terminal window access
* Assumes single-player game vs. computer (PC acting as dealer).
* Assumes $25 starting chip balance for player
* Assumes 6 decks of cards (Shoe): Ace through King, no Jokers, 13 cards per suit
* Assumes dealer has access to $500 in chips (additional to Player's opening $25 balance)
* Assumes player knows standard rules to BlackJack (this version follows 'Bicycle' standard rules)
* Assumes dealer must hit on 'Soft 17'
* Assumes $1 minimum bet, $10 maximum bet
* Assumes no 'Side Rules' are implemented (Double Down, Insurance, Splitting Pairs, Surrender, Swedish Pub)

**BlackJack Process Flow Diagram**