Blackjack Game Test Cases

1. Test Menu Options

Test ID: TC_Menu_01
Description: Verify that the game menu displays correctly and handles all input options.

Steps and Expected Result:

1. Launch the game.
2. Enter option 1 to start a new game: **Starts a new game**.
3. Enter option 2 to load a saved game: **Loads a saved game**.
4. Enter option 3 to exit the game: **Exits the game**.
5. Enter an invalid option (e.g., 0, 5, -1, or letters): **Displays the error message** - `"Invalid option! Please make another choice."`

---

2. Test Player Creation

Test ID: TC_Player_01
Description: Verify that a new player is created correctly and saved to a file.

Steps:

1. Start a new game (Option 1 from the menu).
2. Enter a username when prompted (e.g., "TestUser").
3. Verify that the player's balance is initialized to $1500.
4. Verify that a file named `TestUserplayerInfo.txt` is created (the username can contain letters, symbols, and numbers).

Expected Result:

- Player data is initialized correctly with the balance set to $1500.
- The file `TestUserplayerInfo.txt` should contain:

```
Username: TestUser
Balance: $1500
```

---

3. Test Player Save/Load

Test ID: TC_Player_02
Description: Verify that player data is saved and loaded correctly from a file.

Steps:

1. Start a new game, create a player, and place a bet.
2. Quit the game.
3. Restart the game and load the previously created player.

Expected Result:

- Player data is loaded from the saved file.
- The balance should reflect any wins/losses from the previous session.

---

4. Test Betting Validation

Test ID: TC_Betting_01
Description: Verify that the betting system works correctly with valid and invalid inputs.

Steps:

1. Start a new game or load an existing game with a balance > $0.
2. Enter a valid bet (e.g., $100).
3. Enter an invalid bet (e.g., $2000 when balance is $1500, $0, or negative numbers).

Expected Result:

- **Valid bet** is accepted, and the game proceeds.
- **Invalid bets** display the message: `"You can't bet more than your available balance or $0! Please choose another amount."`

5. Test Deck Functionality

Test ID: TC_Deck_01
Description: Test if the Deck class loads the deck from a file and correctly deals cards.

Steps:

1. Use the menu option 3 to test the deck.
2. Observe the two dealt cards.

Expected Result:

- Two cards are dealt randomly from the deck file.
- The dealt cards' values should match valid card values (2-10, J, Q, K, A).

Test ID: TC_Deck_02
Description: Test if the card values are interpreted correctly.

Steps:

1. Use menu option 3.
2. Observe the values of the dealt cards.

Expected Result:

- Face cards (J, Q, K) return a value of 10.
- Aces (A) return a value of 11.
- Numerical cards return their respective values (2-10).

---

6. Test Black Jack Conditions

Test ID: TC_Game_01
Description: Verify that Black Jack is handled correctly.

Steps:

1. Play the game until either the player or dealer hits exactly 21.
2. If the player has exactly 21 at the start, observe if Black Jack is declared.
3. If the dealer also has 21, observe if a tie is declared.

Expected Result:

- Player wins with a Black Jack.
- A tie occurs if both the player and dealer have 21.

---

7. Test Bust Conditions

Test ID: TC_Game_02
Description: Verify that busts are handled correctly.

Steps:

1. Keep hitting cards until the player's hand exceeds 21.
2. Observe the result.

Expected Result:

- Player busts, and the message `"You bust! The dealer wins! You've lost $X!"` is displayed.
- The player's balance should decrease by the bet amount.

---

8. Test Dealer Logic

Test ID: TC_Game_03
Description: Verify that the dealer plays correctly (dealer must draw until reaching 17 or higher).

Steps:

1. Stop drawing cards and let the dealer take their turn.
2. Observe the dealer’s actions.

Expected Result:

- The dealer hits until they reach 17 or higher.
- The dealer's final hand and total value should be displayed.

---

9. Test Balance Update

Test ID: TC_Balance_01
Description: Verify that the player's balance is updated correctly after winning or losing.

Steps:

1. Start a new game or load a game with sufficient balance.
2. Place a bet and play a full round.
3. Observe the balance after winning or losing.

Expected Result:

- **Winning** increases the balance by the bet amount.
- **Losing** decreases the balance by the bet amount.
- If **tied**, the balance remains unchanged.

---

10. Test Edge Cases

Test ID: TC_Edge_01
Description: Test edge cases, including invalid inputs and extreme conditions.

Steps:

1. Enter invalid menu options (e.g., letters, negative numbers).
2. Enter invalid bets (e.g., letters, negative numbers, amounts greater than the balance).
3. Play the game until the balance reaches $0.

Expected Result:

- Invalid inputs are handled gracefully, and error messages are displayed.

- When the balance reaches $0, the game displays: `"You have no more money to bet! Please start a new game!"`
