Class Descriptions Blackjack

- **Attributes**:
    - deck: An instance of the Deck class, managing the deck of cards.
    - player: An instance of the Player class, representing the game player.
    - dealer: An instance of the Dealer class, representing the dealer.
    - bet_amount: The current wager amount for the round.
    - winner: Tracks the winner of the round.
- **Methods**:
    - __init__(): Initializes game state, including player input.
    - start(): Begins the game, prompting for bets and managing the game flow.
    - deal(): Deals initial cards to the player and dealer.
    - player_turn(): Manages the player's actions (hit, stand, double).
    - dealer_turn(): Manages the dealer's actions.
    - determine_winner(): Compares hands and determines the winner.

Function Descriptions

- **start()**
    - **Description**: Initiates the game by collecting bets and managing the game flow.
    - **Parameters**: None
    - **Returns**: None
- **deal()**
    - **Description**: Deals two cards each to the player and dealer, and updates their hand values.
    - **Parameters**: None
    - **Returns**: None
- **player_turn()**

    - **Description**: Handles player input and decisions during their turn.
    - **Parameters**: None
    - **Returns**: None

- **dealer_turn()**
    - **Description**: Manages the dealer’s actions based on their hand value.
    - **Parameters**: None
    - **Returns**: None
- **determine_winner()**
    - **Description**: Evaluates the final hands of the player and dealer to determine the round's winner.
    - **Parameters**: None
    - **Returns**: Player or Dealer object representing the winner, or None for a push.

Class Descriptions Card

- **Attributes**:
    - suit: A string representing the suit of the card (e.g., "Hearts", "Diamonds").
    - value: A string representing the value of the card (e.g., "Ace", "10").
- **Methods**:
    - __init__(suit, value): Initializes a new instance of the Card class with specified suit and value.
    - __str__(): Returns a string representation of the card.

Function Descriptions

- **init(suit, value)**
    - **Description**: Constructor that initializes a Card object with a suit and value.
    - **Parameters**:
        - suit (str): The suit of the card.
        - value (str): The value of the card.

    - **Returns**: None (initializes the object).
  - **str()**
    - **Description**: Provides a string representation of the Card instance, formatted as "value of suit".
    - **Parameters**: None
    - **Returns**: A string representing the card (e.g., "Ace of Spades")

Class Descriptions

- **Dealer**
  - **Attributes**:
    - hand: An instance of the Hand class, which manages the dealer's cards.
    - balance: An integer representing the dealer's (casino's) balance.
  - **Methods**:
    - __init__(): Initializes a new dealer with a hand and a predefined balance.
    - __str__(): Provides a string representation of the dealer's hand and its value.
    - play(deck): Manages the dealer's turn by drawing cards from the deck according to specific rules.

Function Descriptions

- **init()**
  - **Description**: Constructor that initializes a Dealer object with an empty hand and a balance.
  - **Parameters**: None
  - **Returns**: None (initializes the object).
- **str()**
  - **Description**: Returns a string representation of the dealer's hand and its total value.
  - **Parameters**: None

    - **Returns**: A string formatted as "Dealer's Hand: [hand] ([hand value])".
- **play(deck)**
    - **Description**: Handles the logic for the dealer's turn. The dealer draws cards based on their hand value until they reach 17 or higher.
    - **Parameters**:
        - deck (Deck): An instance of the Deck class used to draw cards.
    - **Returns**: The dealer's current hand (an instance of the Hand class

Class Descriptions

- **Deck**
    - **Attributes**:
        - deck: A list that holds the cards in the current deck, initialized as an empty list.
    - **Methods**:
        - __init__(): Initializes a new deck as an empty list.
        - __str__(): Provides a string representation of the cards in the deck.
        - create_deck(): Generates a standard 312-card deck (6 decks of 52 cards).
        - shuffle_deck(): Randomizes the order of cards in the deck.
        - hit(): Draws and returns the top card from the deck.

Function Descriptions

- **init()**
    - **Description**: Constructor that initializes a Deck object with an empty card list.
    - **Parameters**: None
    - **Returns**: None (initializes the object).
- **str()**
    - **Description**: Returns a string representation of the deck, showing all cards.
    - **Parameters**: None

    - **Returns**: A string formatted as a list of cards.

- **create_deck()**
    - **Description**: Generates six decks of cards, creating a total of 312 cards and adding them to the deck.
    - **Parameters**: None
    - **Returns**: None (modifies the deck attribute).
- **shuffle_deck()**
    - **Description**: Randomizes the order of the cards in the deck using the built-in random.shuffle() method.
    - **Parameters**: None
    - **Returns**: None (modifies the deck attribute).
- **hit()**
    - **Description**: Draws the top card from the deck and removes it from the list.
    - **Parameters**: None
    - **Returns**: The top Card object from the deck, or None if the deck is empty

Class Descriptions

- **Button**
    - **Attributes**:
        - image: The image displayed on the button.
        - rect: The rectangle defining the button's position and size.
        - clicked: A boolean indicating whether the button has been clicked.
    - **Methods**:
        - __init__(x, y, image, scale): Initializes a button with its position, image, and scale.
        - draw(): Draws the button on the screen and handles click events.

Function Descriptions

- **init(x, y, image, scale)**

    - **Description**: Constructor that initializes a Button object with specified position, image, and scale.
    - **Parameters**:
        - x (int): The x-coordinate for the button's position.
        - y (int): The y-coordinate for the button's position.
        - image (Surface): The Pygame surface representing the button image.
        - scale (float): The scaling factor for resizing the button.
    - **Returns**: None (initializes the object).
- **draw()**
    - **Description**: Draws the button on the screen and checks for mouse interactions.
    - **Parameters**: None
    - **Returns**:
        - action (bool): Returns True if the button was clicked, otherwise False.

Game Loop

Main Game Loop

- The main game loop initializes the Pygame window and listens for events while rendering buttons.
- The background is filled with a specified color, and button actions are printed based on user interaction.

Event Handling

- Handles quitting the game when the window is closed.

Class Descriptions

- **Hand**
    - **Attributes**:
        - hand: A list holding the current cards in the hand.

    - hand_value: An integer tracking the total value of the hand.
    - allowed_to_hit: A boolean indicating whether the player can still hit.
  - **Methods**:
    - __init__(): Initializes a new hand with no cards and a value of zero.
    - __str__(): Provides a string representation of the cards in the hand.
    - calc_hand_value(): Calculates and updates the total value of the hand.

Function Descriptions

- **init()**
  - **Description**: Constructor that initializes a Hand object with an empty hand and zero value.
  - **Parameters**: None
  - **Returns**: None (initializes the object).
- **str()**
  - **Description**: Returns a string representation of the hand, showing all cards.
  - **Parameters**: None
  - **Returns**: A string formatted as a list of cards.
- **calc_hand_value()**
  - **Description**: Calculates the total value of the hand and updates hand_value. Considers Aces specially to account for their dual values.
  - **Parameters**: None
  - **Returns**: The total value of the hand after calculation.
- **Player**
  - **Attributes**:
    - name: A string representing the player's name.
    - hand: An instance of the Hand class containing the player's current cards.
    - balance: An integer tracking the player's balance.

        - already_hit: A boolean indicating if the player has already hit this turn.
    - **Methods**:
        - __init__(): Initializes a player with a name and balance.
        - __str__(): Returns a string representation of the player's hand.
        - place_bets(): Placeholder method for placing bets (not yet implemented).
        - hit(): Handles the action of hitting (drawing a card) for the player.
        - stand(): Updates the player’s status to indicate they have stood.
        - double(): Manages the doubling down action, allowing the player to draw one more card.
        - check_bust(): Checks if the player has exceeded 21, indicating a bust.

Function Descriptions

- **init(name, balance)**
    - **Description**: Constructor that initializes a Player object with a name and balance.
    - **Parameters**:
        - name: The player's name.
        - balance: The starting balance for the player.
    - **Returns**: None (initializes the object).
- **str()**
    - **Description**: Returns a string representation of the player's hand, including their name and hand value.
    - **Parameters**: None
    - **Returns**: A formatted string of the player's hand.
- **place_bets(amount)**
    - **Description**: Placeholder for a method to handle betting logic (not yet implemented).
    - **Parameters**:

        - amount: The amount to bet.
    - **Returns**: None.
- **hit(deck)**
    - **Description**: Handles the player's action to hit (draw a card). Checks the current hand value and updates the hand.
    - **Parameters**:
        - deck: The current deck from which to draw a card.
    - **Returns**: The updated hand after hitting.
- **stand()**
    - **Description**: Updates the player's state to indicate they have chosen to stand, preventing further hits.
    - **Parameters**: None
    - **Returns**: The player's hand.
- **double(deck)**
    - **Description**: Manages the action of doubling down, allowing the player to draw one additional card.
    - **Parameters**:
        - deck: The current deck from which to draw a card.
    - **Returns**: The updated hand after doubling.
- **check_bust()**
    - **Description**: Checks if the player's hand value exceeds 21, indicating a bust.
    - **Parameters**: None
    - **Returns**: A boolean indicating if the player has busted.