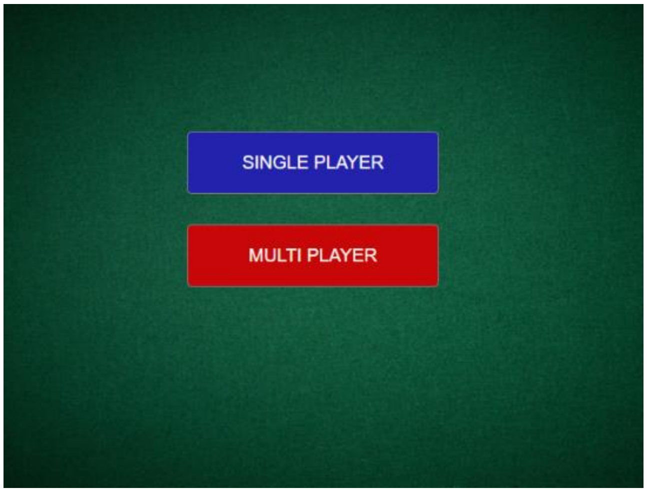
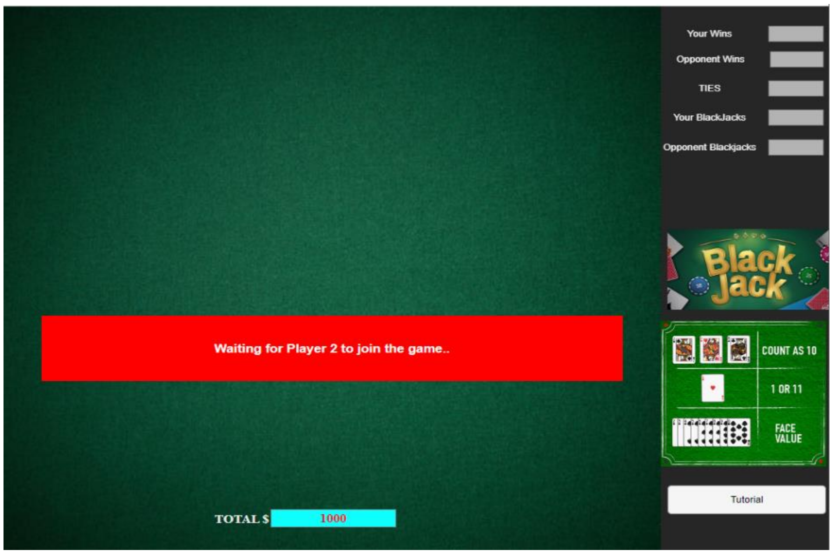
Introduction

Blackjack is a popular casino card game. The objective of the game is to have a hand with a higher total value than the other player/dealer, without going over 21. Each player is dealt two cards, and can then choose to "hit" to receive additional cards or "stand" to keep their current total. If playing against the dealer (ie singlePlayer mode), then the dealer also receives two cards, with one facing up and one facing down. The dealer must hit until their hand has a value of at least 17. A player wins if their hand is closer to 21 than the dealer's, or if the dealer goes over 21, known as "busting." Each card is worth its face value from 2 to 10. Ace’s are worth either 1 or 11, whatever is favorable for the player’s win condition. Face cards (king,queen,jack) are worth 10. An automatic win (Black Jack) occurs when a player or dealer draws an ace and face card during the start

Game Play



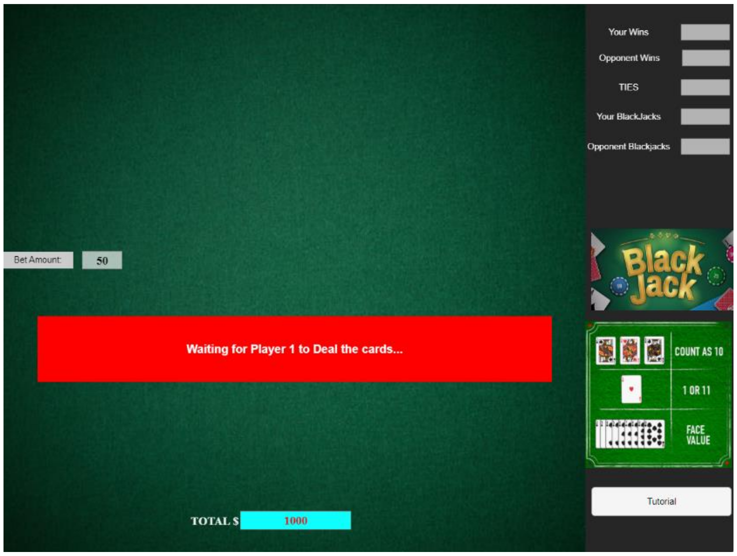
Upon launching the application, there will be home screen music playing with two buttons in the middle of the screen one for single player and the other for multiplayer.



Waiting for Player 2 to join the game..

TOTAL $ 1000

After clicking on multiplayer, the player will be greeted by a start button. Whoever clicks the start button first will be considered player one (i.e. the host) and a “Waiting for Player 2 to join game…” message will stay until player 2 clicks the start button from their computer.

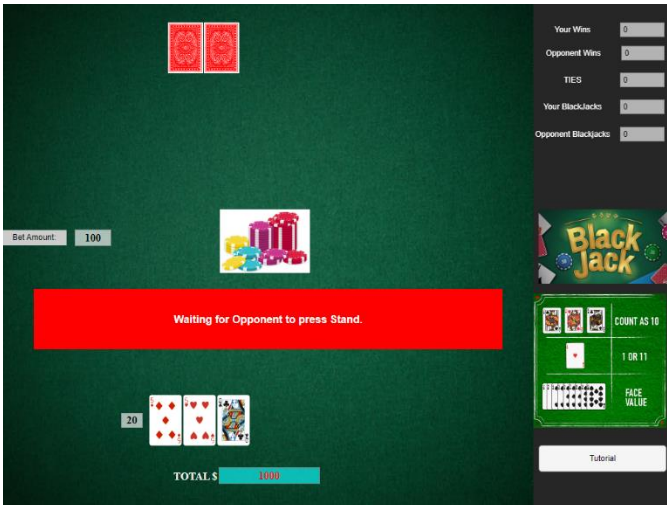


Bet Amount: 50

Waiting for Player 1 to Deal the cards...

TOTAL $ 1000

Player 2 joining the game will result in a screen where player 1 gets to decide the amount of bet for both players with the “Change Bet” button or they can choose to immediately deal. Meanwhile, player 2 has the message “Waiting fo player one to deal the cards” displayed on their screen.



Your Wins 0
Opponent Wins 0
TIES 0
Your BlackJacks 0
Opponent Blackjacks 0

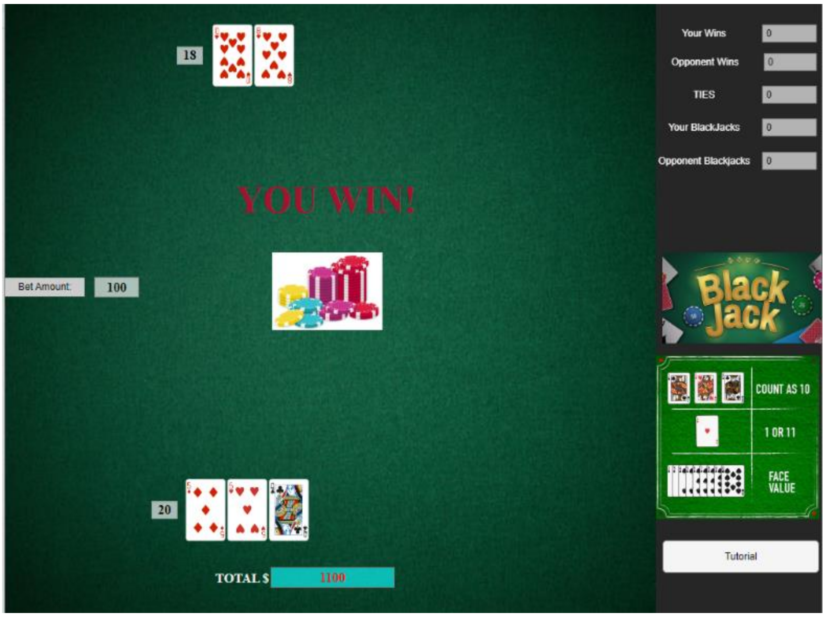
Bet Amount: 100

Waiting for Opponent to press Stand.

20

TOTAL $ 1000

Once the deal button is clicked, both players will be dealt two cards that are hidden from the other player. Players will have the option to click “Hit” adding an additional card to their hand or click “Stand” to end their turn. Players can have as many as 5 cards or if total value passes 21, which is considered a bust. Acquiring a bust will automatically end one’s turn.
The value of a player’s hand is displayed in the “totalCountField” next to each players first card.



18

Your Wins 0
Opponent Wins 0
TIES 0
Your BlackJacks 0
Opponent Blackjacks 0

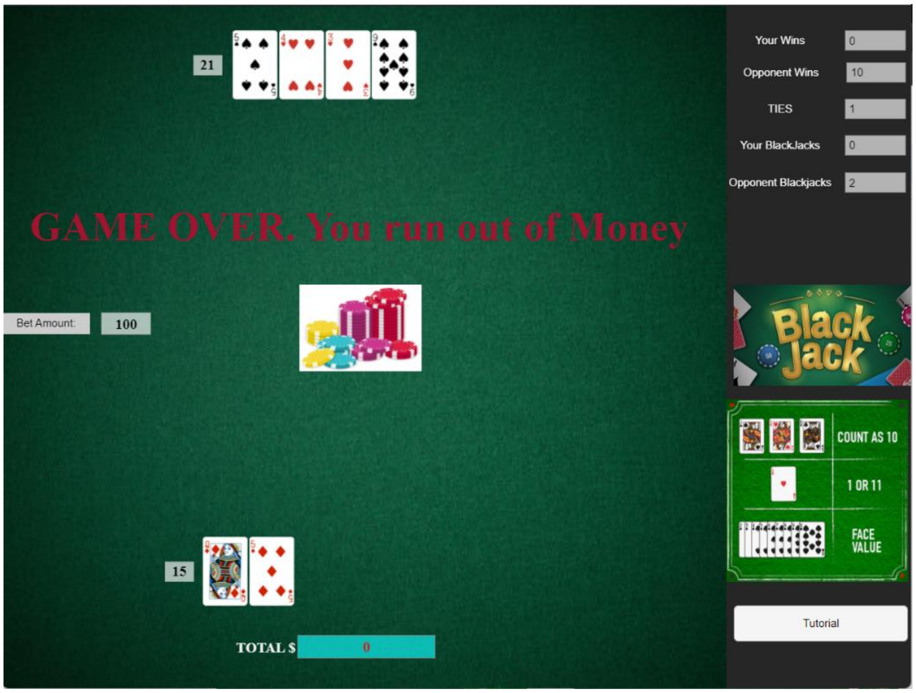
YOU WIN!

Bet Amount: 100

20

TOTAL $ 1100

Whoever’s turn ends first whether that be by bust or manually clicking stand will see the message “Waiting for other player..” until the other player ends their turn. Afterwards, both player’s hand will be revealed and total count value will be compared. A player that has a higher value while under 21 will win that round, while the other loses. If both players have the same value or both bust then that will be a tie. A player will be given a post round screen with Music that correlates to their round status, such as “YOU WIN” if the player won. After the post round screen, the players’ stats bar and total money will be updated accordingly based on round status.



21

GAME OVER. You run out of Money

Bet Amount: 100

15

TOTAL $ 0

Your Wins 0

Opponent Wins 10

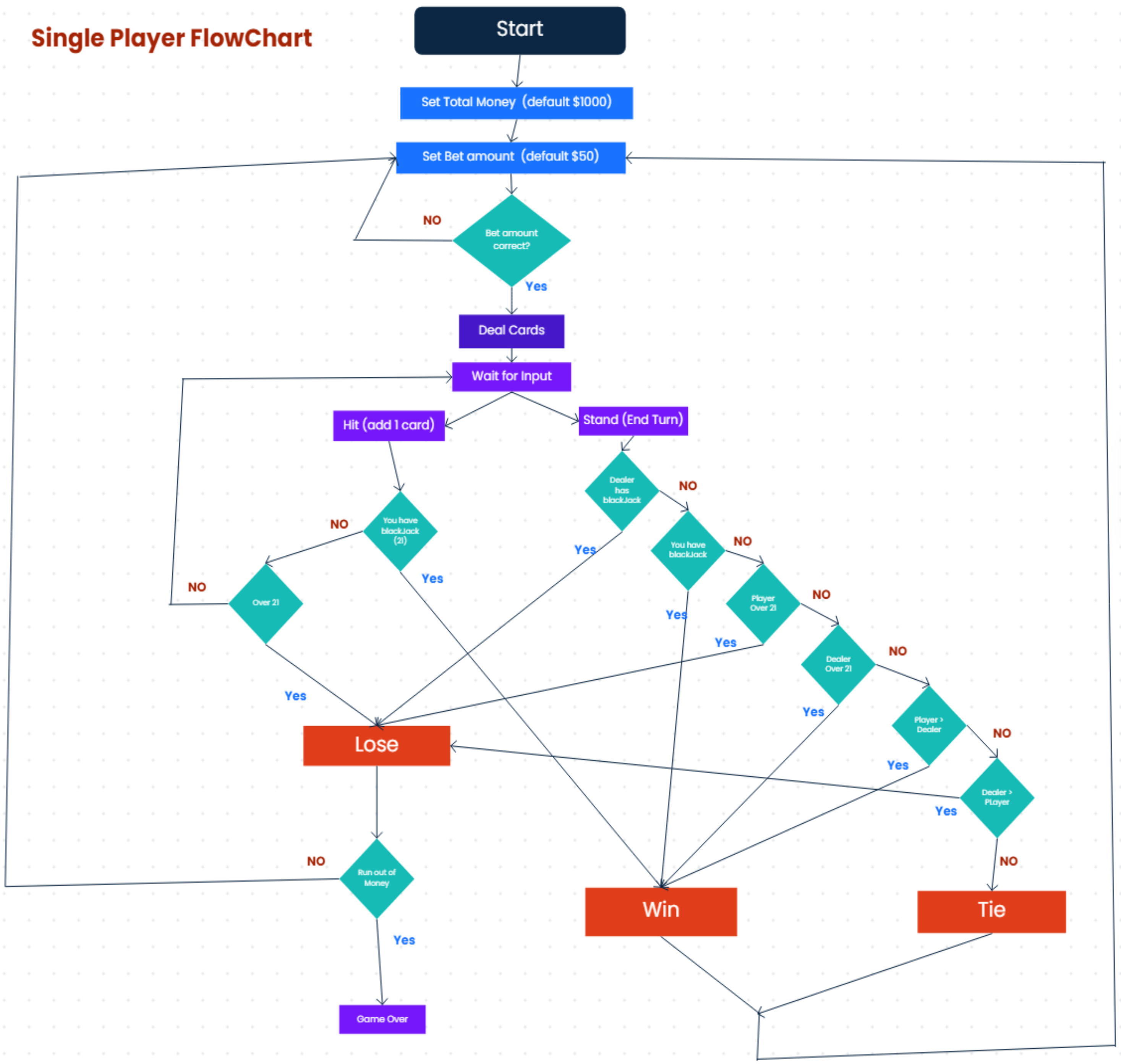
TIES 1

Your BlackJacks 0

Opponent Blackjacks 2

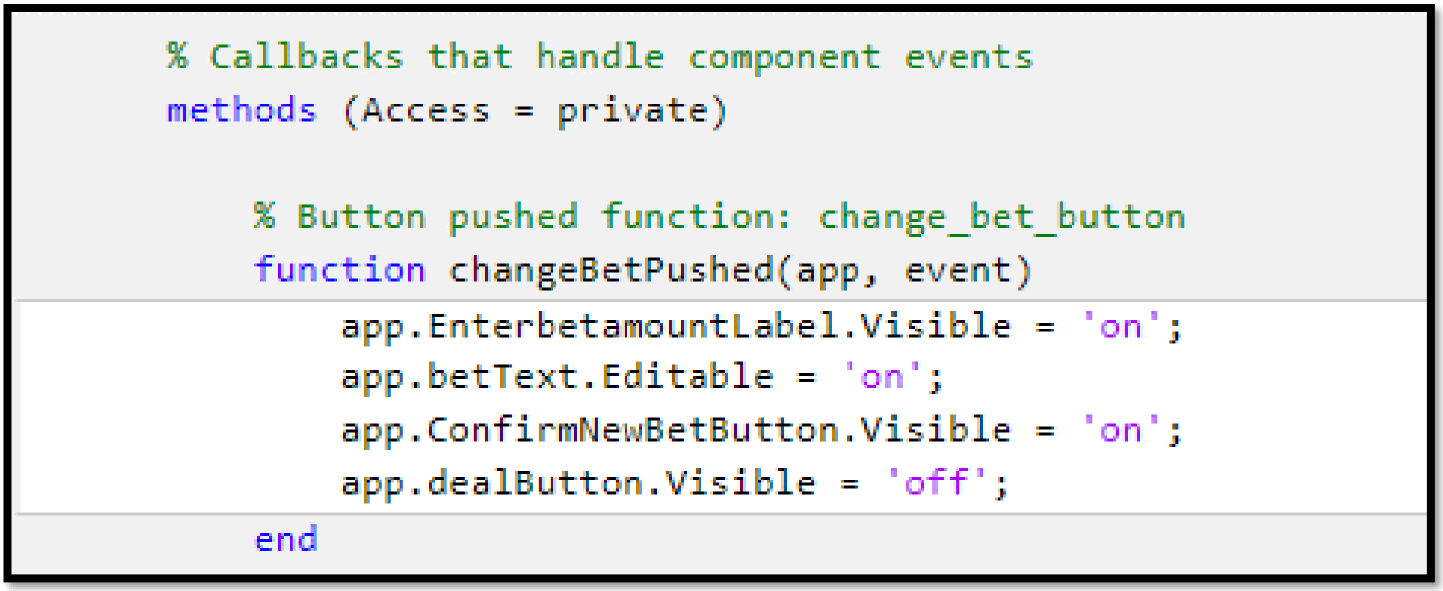
When a player loses and reaches $0 total, a game over screen will be displayed. To play again both players must close the application and select multiplayer mode.

Single Player code



- FlowChart explaining The singlePlayer game flow

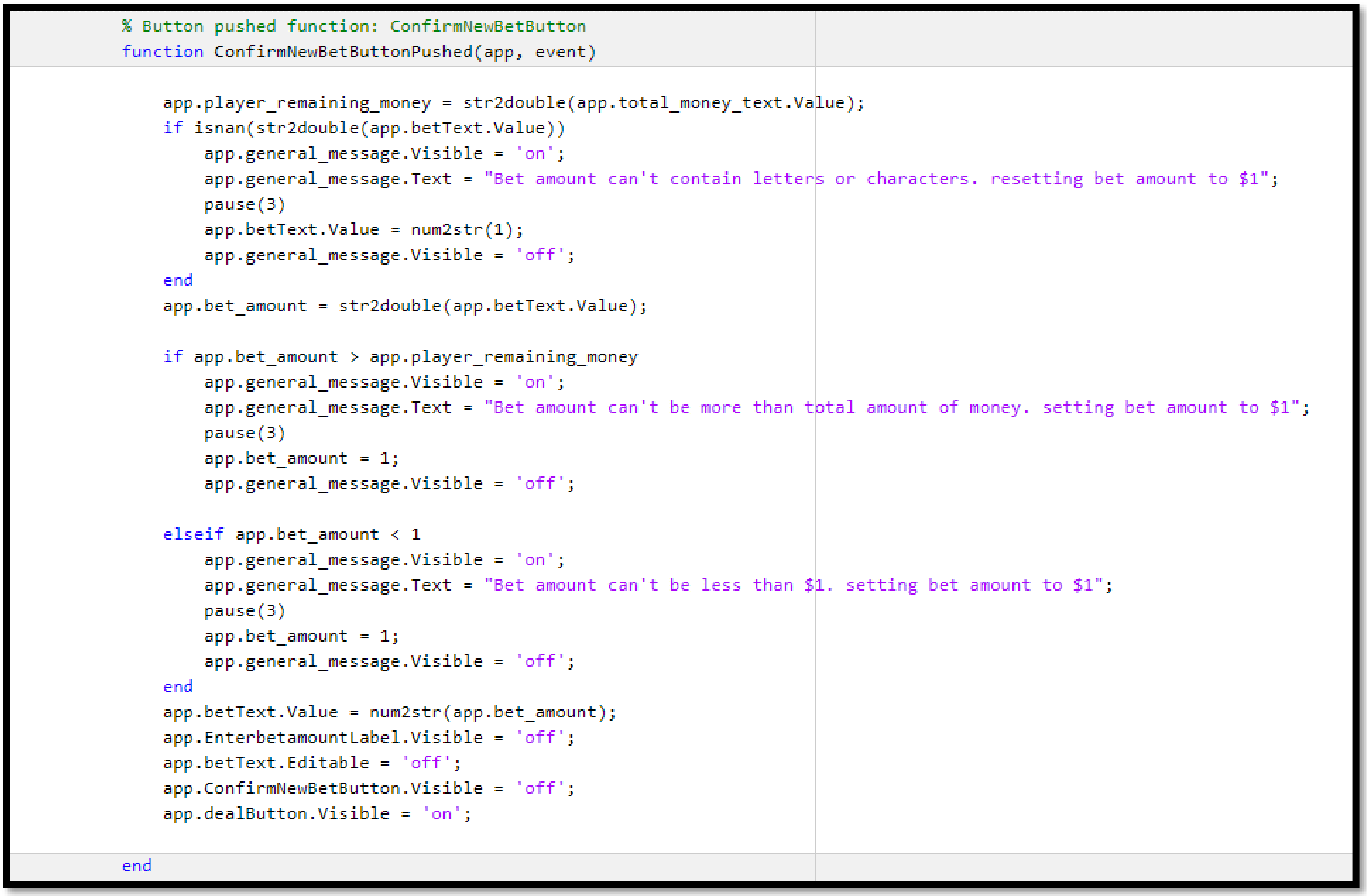
The functions:



```
    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: change_bet_button
        function changeBetPushed(app, event)
            app.EnterbetamountLabel.Visible = 'on';
            app.betText.Editable = 'on';
            app.ConfirmNewBetButton.Visible = 'on';
            app.dealButton.Visible = 'off';
        end
```

changBetPushed: This is callback function, called whenever the player presses “change BET” button.

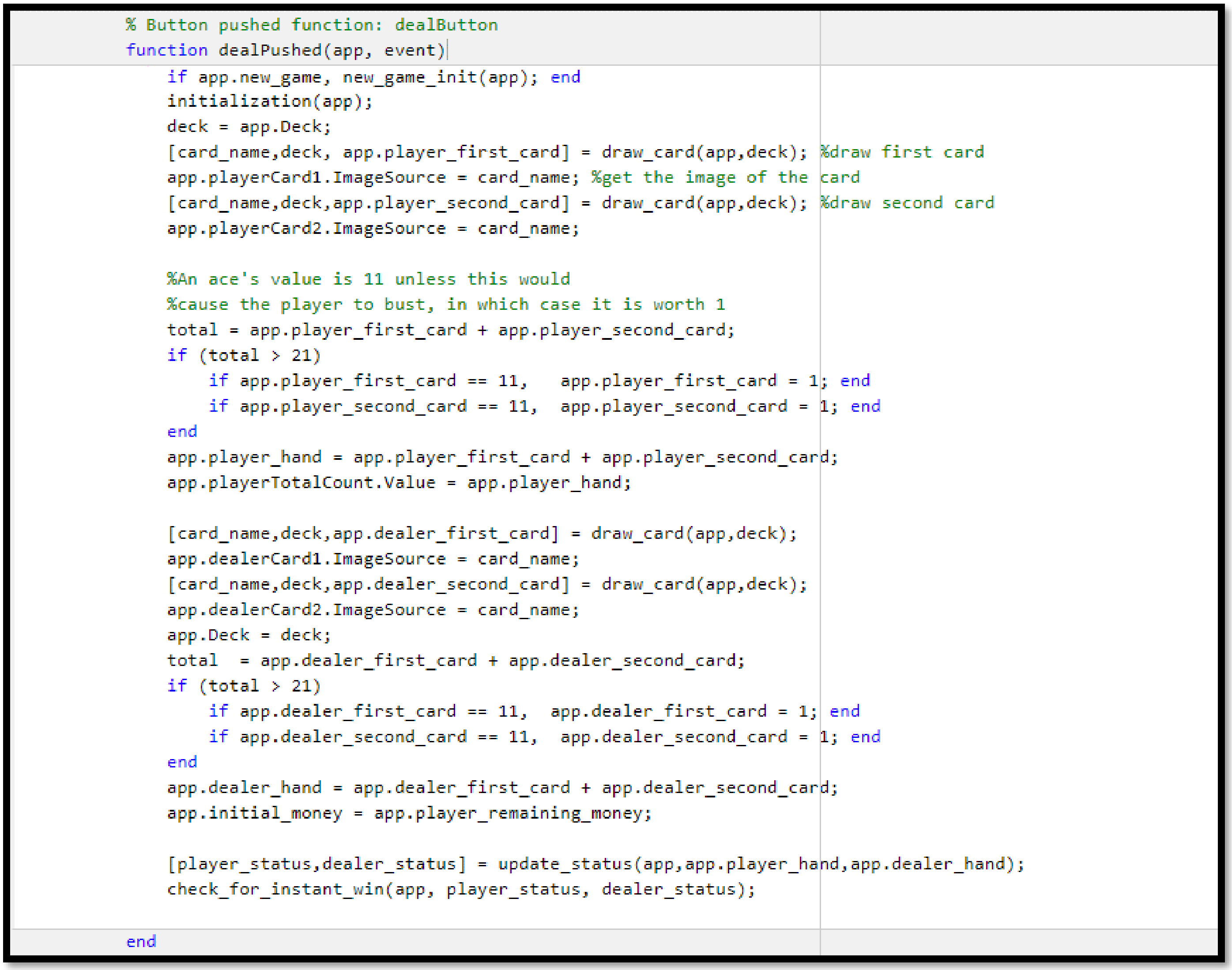
- It makes bet text field editable so that the player can adjust the bet amount, and also reveals the “confirm New Bet” button. Is disables the “deal” button since we can’t deal unless we choose a proper bet amount



```
        % Button pushed function: ConfirmNewBetButton
        function ConfirmNewBetButtonPushed(app, event)

            app.player_remaining_money = str2double(app.total_money_text.Value);
            if isnan(str2double(app.betText.Value))
                app.general_message.Visible = 'on';
                app.general_message.Text = "Bet amount can't contain letters or characters. resetting bet amount to $1";
                pause(3)
                app.betText.Value = num2str(1);
                app.general_message.Visible = 'off';
            end
            app.bet_amount = str2double(app.betText.Value);

            if app.bet_amount > app.player_remaining_money
                app.general_message.Visible = 'on';
                app.general_message.Text = "Bet amount can't be more than total amount of money. setting bet amount to $1";
                pause(3)
                app.bet_amount = 1;
                app.general_message.Visible = 'off';

            elseif app.bet_amount < 1
                app.general_message.Visible = 'on';
                app.general_message.Text = "Bet amount can't be less than $1. setting bet amount to $1";
                pause(3)
                app.bet_amount = 1;
                app.general_message.Visible = 'off';
            end
            app.betText.Value = num2str(app.bet_amount);
            app.EnterbetamountLabel.Visible = 'off';
            app.betText.Editable = 'off';
            app.ConfirmNewBetButton.Visible = 'off';
            app.dealButton.Visible = 'on';

        end
```

ConfirmNewBetButtonPushed: this callback function, called whenever the player sets the bet amount and presses “Confirm the bet amount” button

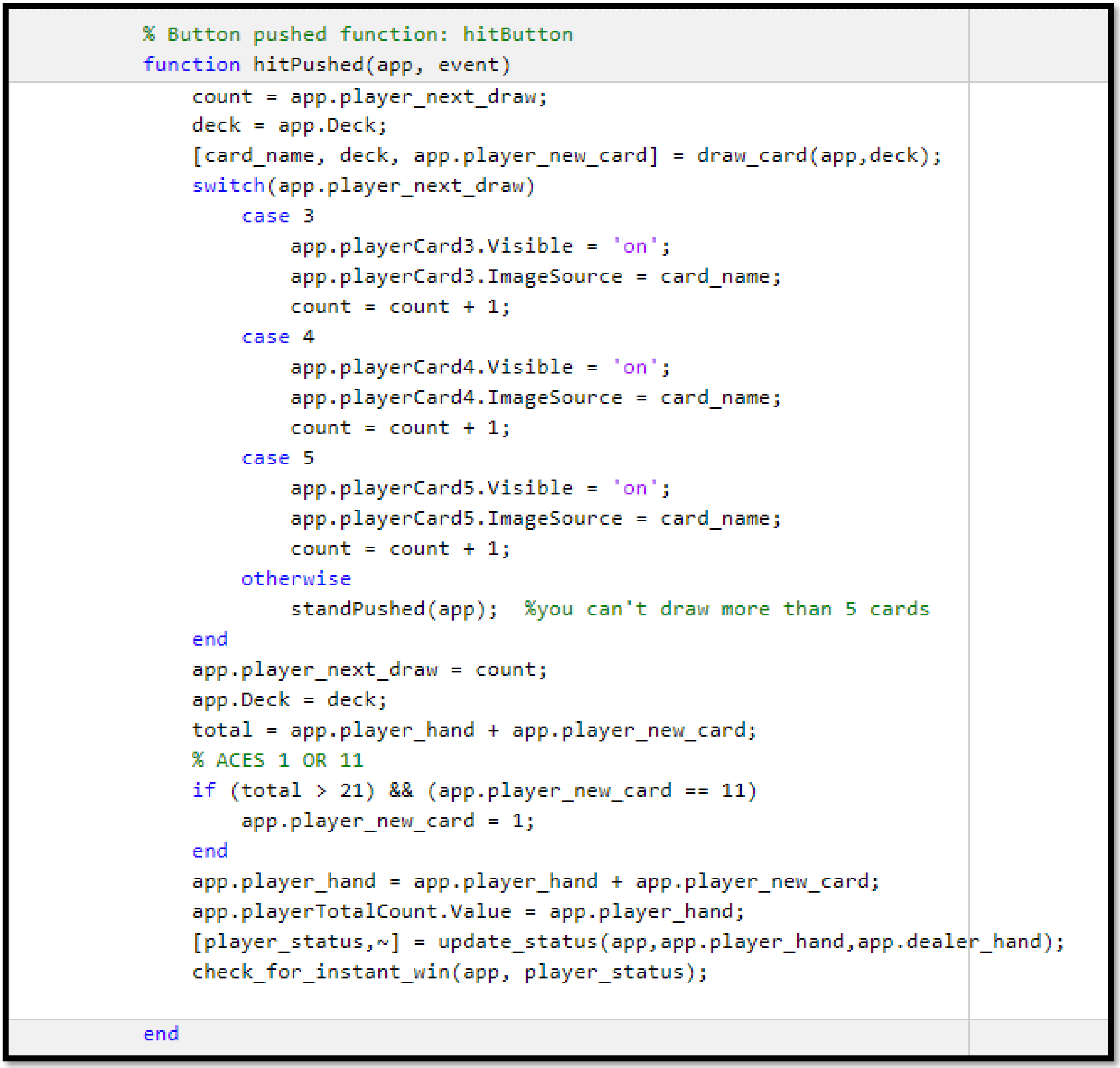
- It makes sure that the player entered numerical bet amount and didn’t enter just some random letters or characters
- If bet amount is numerical, then it makes sure that the bet is no more than the total money left. It also prevents entering negative bet amounts.
- In case player enters non-accpetable bet amount, then the bet amount will be reset to $1. Then the player either enters an appropriate bet amount or leaves the bet as $1 and continues.



```
% Button pushed function: dealButton
function dealPushed(app, event)
    if app.new_game, new_game_init(app); end
    initialization(app);
    deck = app.Deck;
    [card_name,deck, app.player_first_card] = draw_card(app,deck); %draw first card
    app.playerCard1.ImageSource = card_name; %get the image of the card
    [card_name,deck,app.player_second_card] = draw_card(app,deck); %draw second card
    app.playerCard2.ImageSource = card_name;

    %An ace's value is 11 unless this would
    %cause the player to bust, in which case it is worth 1
    total = app.player_first_card + app.player_second_card;
    if (total > 21)
        if app.player_first_card == 11,   app.player_first_card = 1; end
        if app.player_second_card == 11,  app.player_second_card = 1; end
    end
    app.player_hand = app.player_first_card + app.player_second_card;
    app.playerTotalCount.Value = app.player_hand;

    [card_name,deck,app.dealer_first_card] = draw_card(app,deck);
    app.dealerCard1.ImageSource = card_name;
    [card_name,deck,app.dealer_second_card] = draw_card(app,deck);
    app.dealerCard2.ImageSource = card_name;
    app.Deck = deck;
    total  = app.dealer_first_card + app.dealer_second_card;
    if (total > 21)
        if app.dealer_first_card == 11,  app.dealer_first_card = 1; end
        if app.dealer_second_card == 11,  app.dealer_second_card = 1; end
    end
    app.dealer_hand = app.dealer_first_card + app.dealer_second_card;
    app.initial_money = app.player_remaining_money;

    [player_status,dealer_status] = update_status(app,app.player_hand,app.dealer_hand);
    check_for_instant_win(app, player_status, dealer_status);

end
```

dealPushed: This is a callback function that is called whenever the player presses “DEAL Button” at the start of each game.

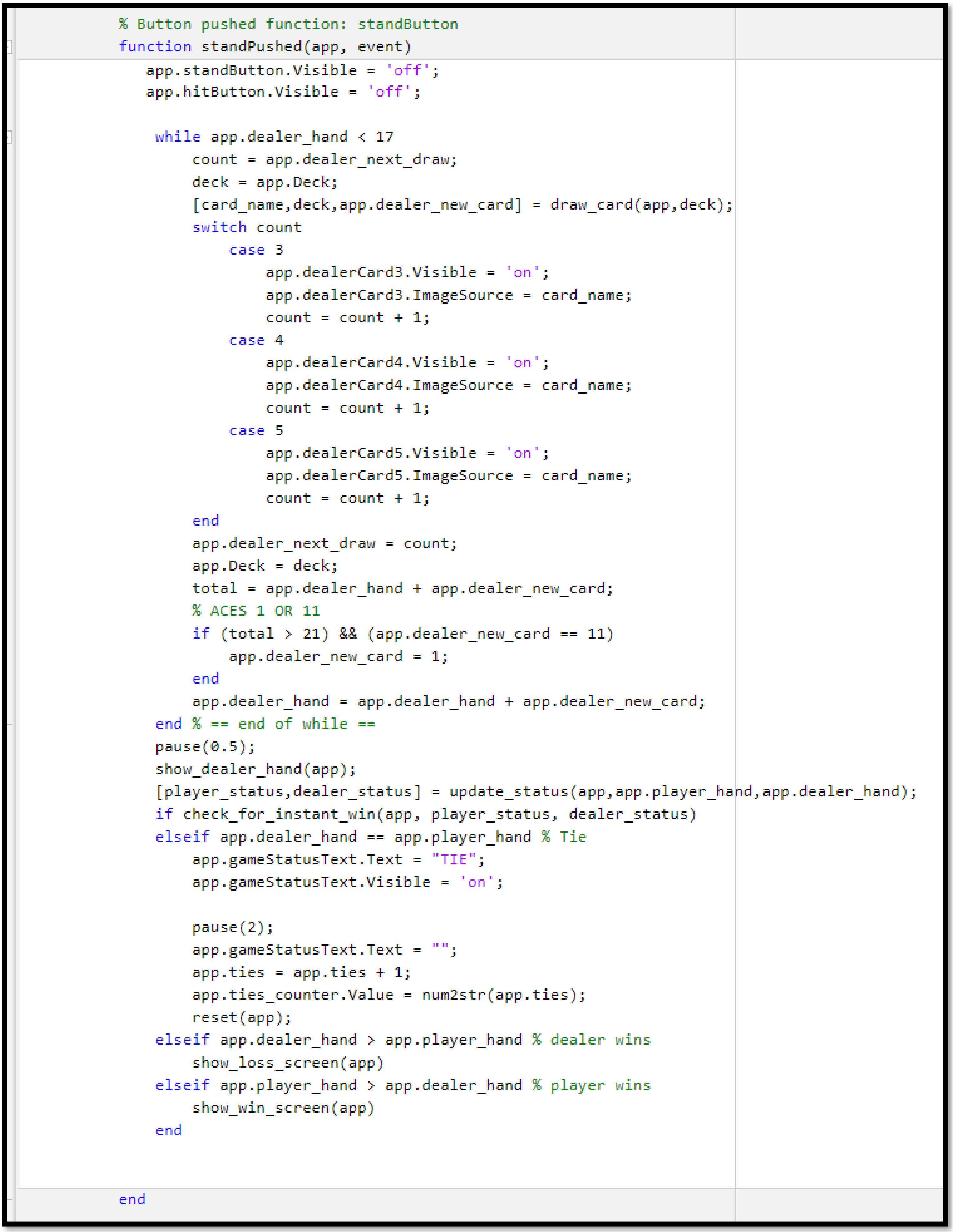
- It draws 2 cards for the player/dealer by calling “draw_card” function.
- Assigns the correct image for each card visualization
- If player/dealer has 2 Ace’s in their hand (hence they get busted), then it will make Ace worth 1 instead 11 therefore the player/dealer won’t be busted.
    - ACE either worths 1 or 11, depending on which is in your favor



```
        % Button pushed function: hitButton
        function hitPushed(app, event)
            count = app.player_next_draw;
            deck = app.Deck;
            [card_name, deck, app.player_new_card] = draw_card(app,deck);
            switch(app.player_next_draw)
                case 3
                    app.playerCard3.Visible = 'on';
                    app.playerCard3.ImageSource = card_name;
                    count = count + 1;
                case 4
                    app.playerCard4.Visible = 'on';
                    app.playerCard4.ImageSource = card_name;
                    count = count + 1;
                case 5
                    app.playerCard5.Visible = 'on';
                    app.playerCard5.ImageSource = card_name;
                    count = count + 1;
                otherwise
                    standPushed(app);  %you can't draw more than 5 cards
            end
            app.player_next_draw = count;
            app.Deck = deck;
            total = app.player_hand + app.player_new_card;
            % ACES 1 OR 11
            if (total > 21) && (app.player_new_card == 11)
                app.player_new_card = 1;
            end
            app.player_hand = app.player_hand + app.player_new_card;
            app.playerTotalCount.Value = app.player_hand;
            [player_status,~] = update_status(app,app.player_hand,app.dealer_hand);
            check_for_instant_win(app, player_status);

        end
```

hitPushed: This is a callback function that is called whenever the player presses “Hit” button.

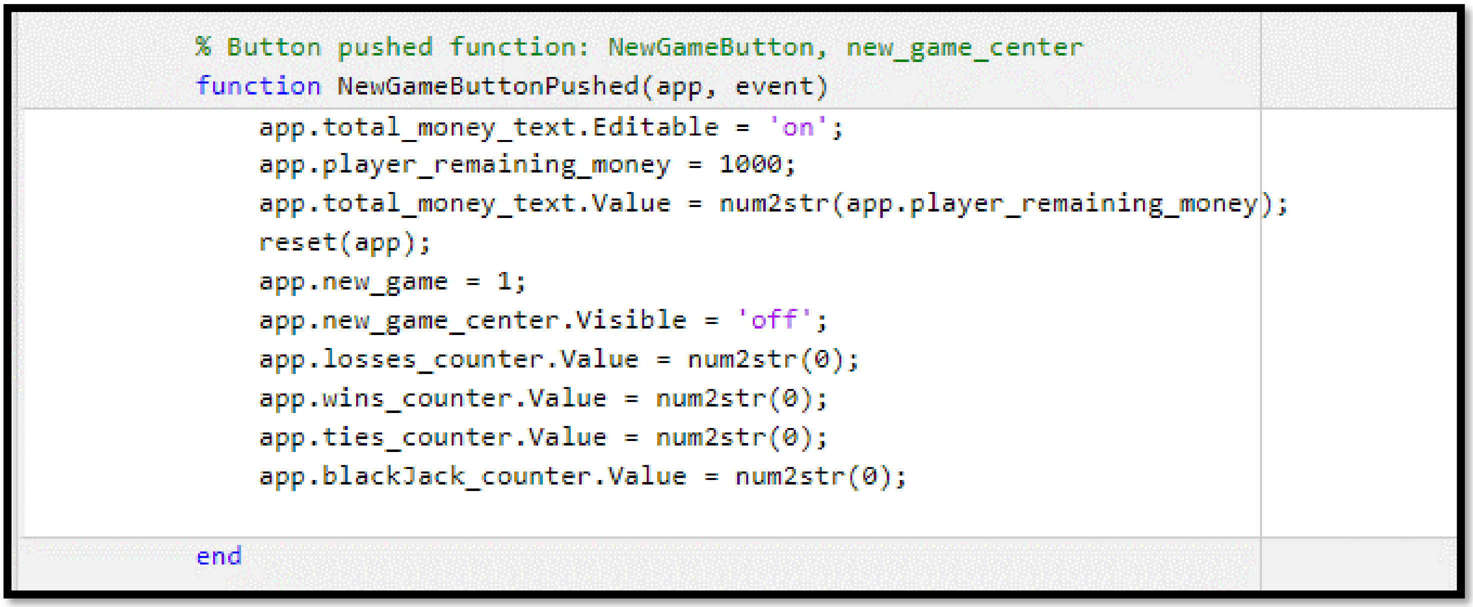
- It draws a card by calling “draw_card” function, then assigns the proper image to the card visualization.
- The game is built such that you can’t draw more than 5 cards, however that is unlikely to happen since you will probably will be busted before having 5 cards. If you drew 5 cards and you are not busted, then your turn will end.
- If you drew Ace and it’s make go busted, then the ace will worth 1 instead of 11 and you won’t be busted.



```matlab
        % Button pushed function: standButton
        function standPushed(app, event)
           app.standButton.Visible = 'off';
           app.hitButton.Visible = 'off';

            while app.dealer_hand < 17
                count = app.dealer_next_draw;
                deck = app.Deck;
                [card_name,deck,app.dealer_new_card] = draw_card(app,deck);
                switch count
                    case 3
                        app.dealerCard3.Visible = 'on';
                        app.dealerCard3.ImageSource = card_name;
                        count = count + 1;
                    case 4
                        app.dealerCard4.Visible = 'on';
                        app.dealerCard4.ImageSource = card_name;
                        count = count + 1;
                    case 5
                        app.dealerCard5.Visible = 'on';
                        app.dealerCard5.ImageSource = card_name;
                        count = count + 1;
                end
                app.dealer_next_draw = count;
                app.Deck = deck;
                total = app.dealer_hand + app.dealer_new_card;
                % ACES 1 OR 11
                if (total > 21) && (app.dealer_new_card == 11)
                    app.dealer_new_card = 1;
                end
                app.dealer_hand = app.dealer_hand + app.dealer_new_card;
            end % == end of while ==
            pause(0.5);
            show_dealer_hand(app);
            [player_status,dealer_status] = update_status(app,app.player_hand,app.dealer_hand);
            if check_for_instant_win(app, player_status, dealer_status)
            elseif app.dealer_hand == app.player_hand % Tie
                app.gameStatusText.Text = "TIE";
                app.gameStatusText.Visible = 'on';

                pause(2);
                app.gameStatusText.Text = "";
                app.ties = app.ties + 1;
                app.ties_counter.Value = num2str(app.ties);
                reset(app);
            elseif app.dealer_hand > app.player_hand % dealer wins
                show_loss_screen(app)
            elseif app.player_hand > app.dealer_hand % player wins
                show_win_screen(app)
            end


        end
```

standPushed: The is is callback function, called whenever the player presses “Stand” button.

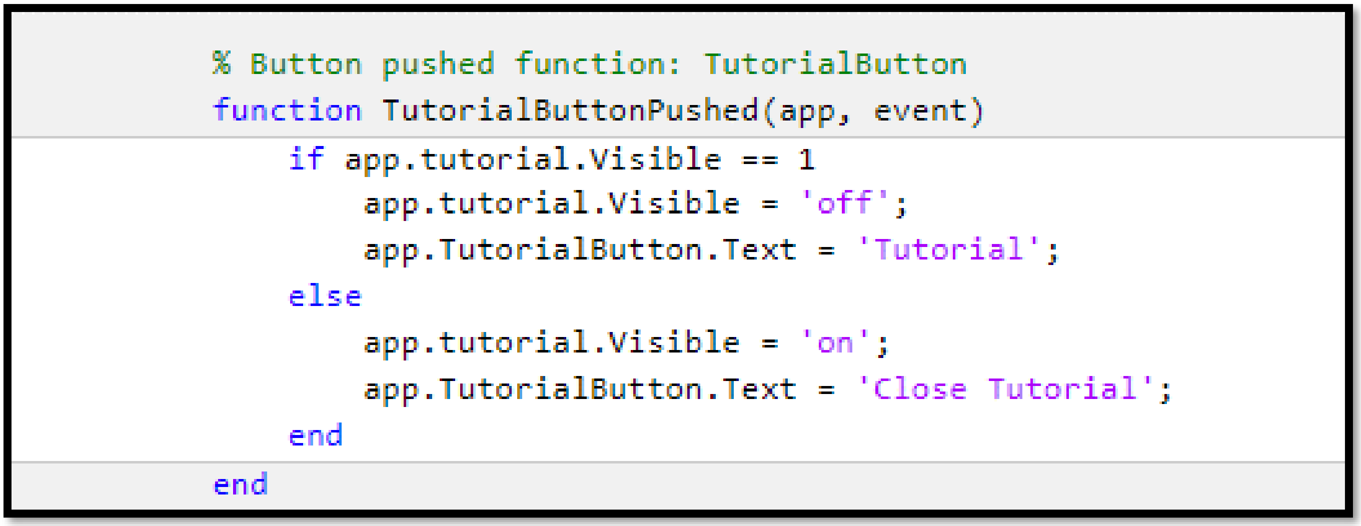
- Disables the stand and hit button, reveals the dealer hand then checks whether the player or the dealer won



```
% Button pushed function: NewGameButton, new_game_center
function NewGameButtonPushed(app, event)
    app.total_money_text.Editable = 'on';
    app.player_remaining_money = 1000;
    app.total_money_text.Value = num2str(app.player_remaining_money);
    reset(app);
    app.new_game = 1;
    app.new_game_center.Visible = 'off';
    app.losses_counter.Value = num2str(0);
    app.wins_counter.Value = num2str(0);
    app.ties_counter.Value = num2str(0);
    app.blackJack_counter.Value = num2str(0);

end
```

NewGameButtonPushed: This is callback function, called whenever the player presses “New Game” button.

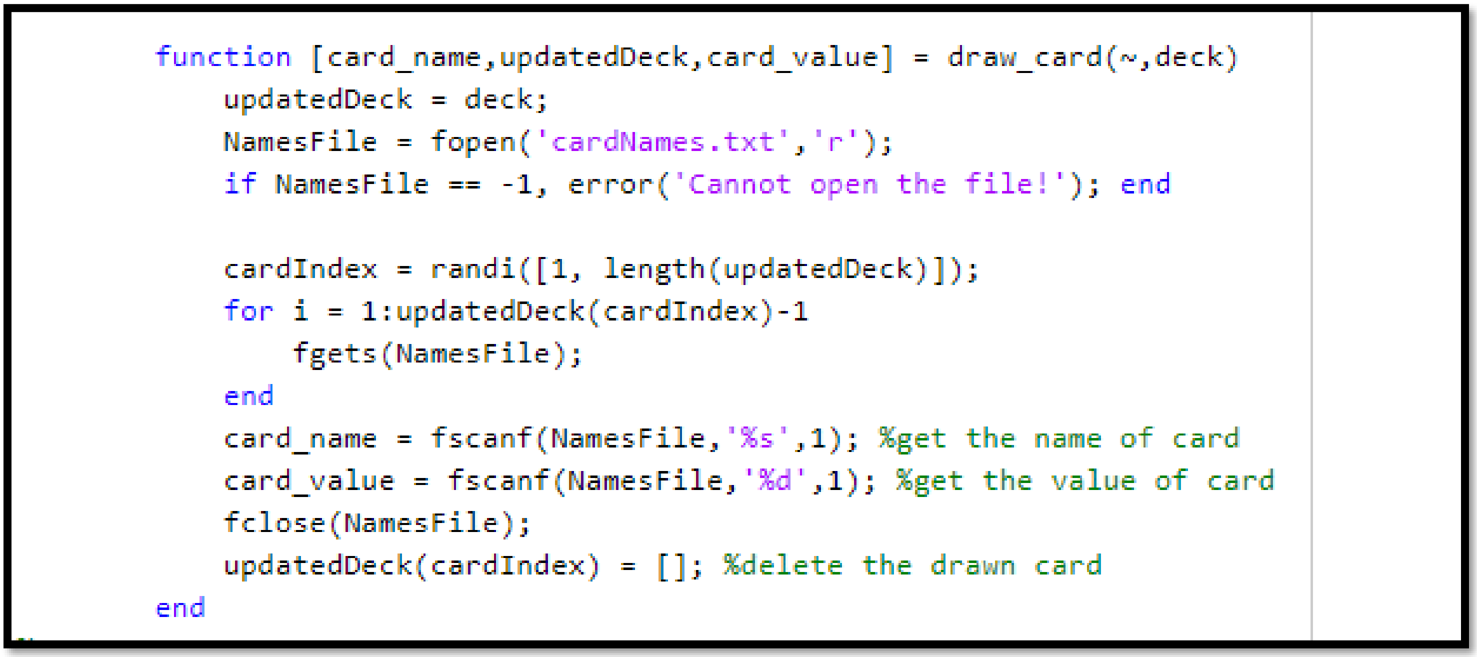
- it finishes the current session and starts a new session. Resets all the stats counter (win, loss, tie,etc..)
- make the total money field adjustable, so that the player can change total money they have for the new session



```
% Button pushed function: TutorialButton
function TutorialButtonPushed(app, event)
    if app.tutorial.Visible == 1
        app.tutorial.Visible = 'off';
        app.TutorialButton.Text = 'Tutorial';
    else
        app.tutorial.Visible = 'on';
        app.TutorialButton.Text = 'Close Tutorial';
    end
end
```

TutorialButtonPushed: callback function, called whenever the player presses “Tutorial button”

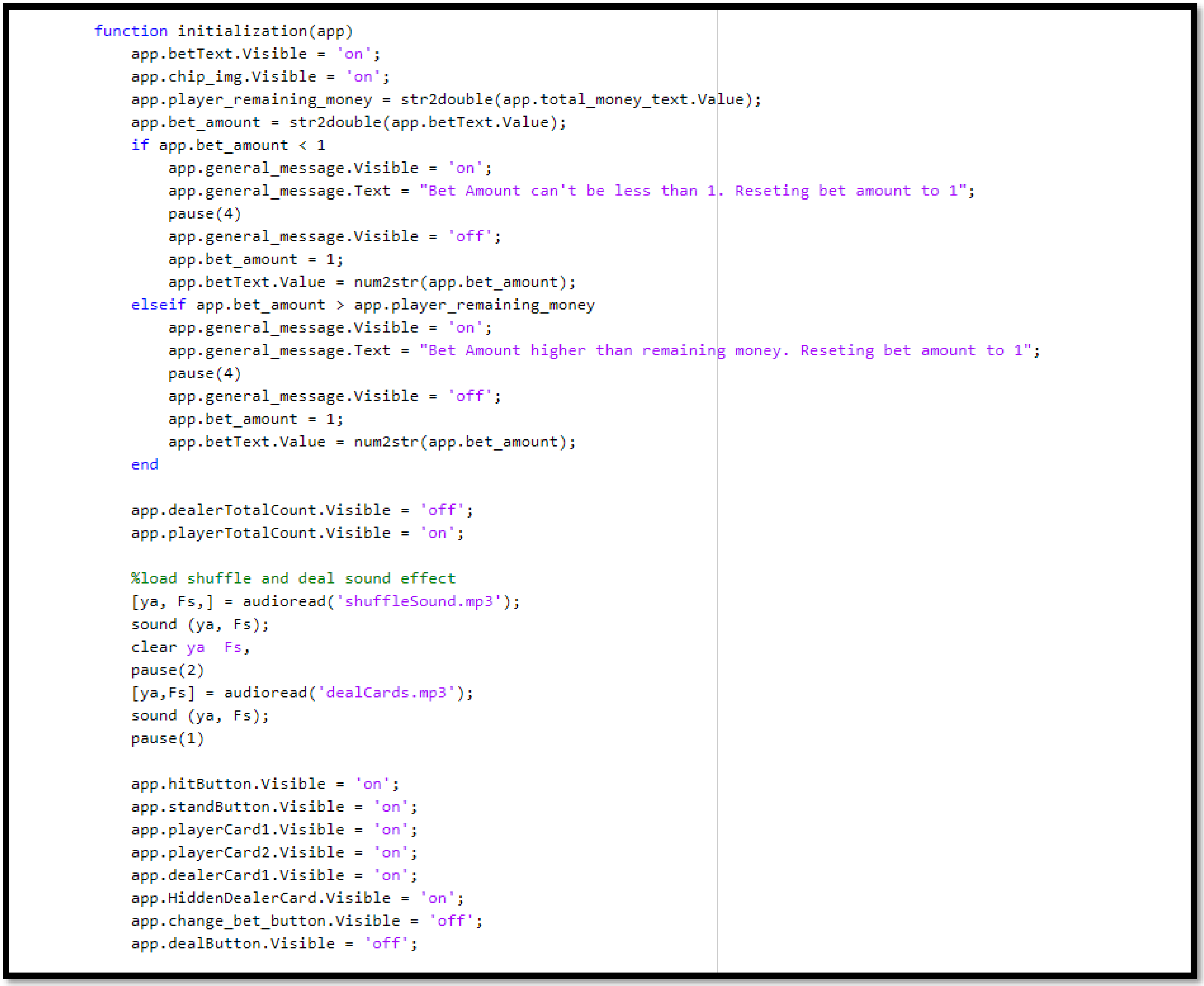
- If tutorial is not visible, then it makes the tutorial visible to the player, and changes the “tutorial” button to “close tutorial button”
- If tutorial is already visible (ie player want to close tutorial), then makes the tutorial invisible and changes the text back to “Tutorial”



```
function [card_name,updatedDeck,card_value] = draw_card(~,deck)
    updatedDeck = deck;
    NamesFile = fopen('cardNames.txt','r');
    if NamesFile == -1, error('Cannot open the file!'); end

    cardIndex = randi([1, length(updatedDeck)]);
    for i = 1:updatedDeck(cardIndex)-1
        fgets(NamesFile);
    end
    card_name = fscanf(NamesFile,'%s',1); %get the name of card
    card_value = fscanf(NamesFile,'%d',1); %get the value of card
    fclose(NamesFile);
    updatedDeck(cardIndex) = []; %delete the drawn card
end
```

Draw_card: this function draws a new card for the player or dealer

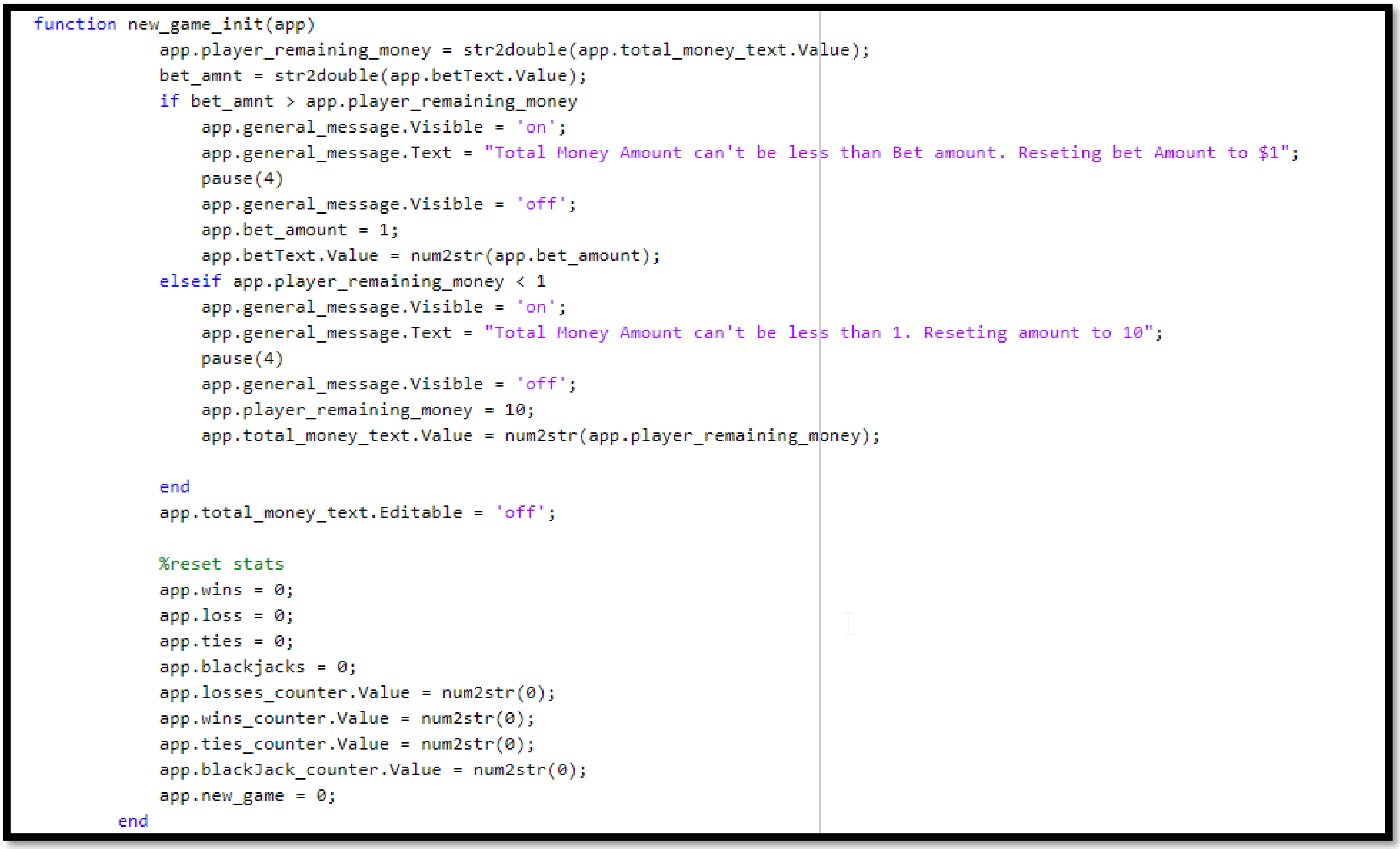
- Choose random card from the “cardsName” file, and extracts the name and the value of the card
- Deletes the drawn card from deck



```
        function initialization(app)
            app.betText.Visible = 'on';
            app.chip_img.Visible = 'on';
            app.player_remaining_money = str2double(app.total_money_text.Value);
            app.bet_amount = str2double(app.betText.Value);
            if app.bet_amount < 1
                app.general_message.Visible = 'on';
                app.general_message.Text = "Bet Amount can't be less than 1. Reseting bet amount to 1";
                pause(4)
                app.general_message.Visible = 'off';
                app.bet_amount = 1;
                app.betText.Value = num2str(app.bet_amount);
            elseif app.bet_amount > app.player_remaining_money
                app.general_message.Visible = 'on';
                app.general_message.Text = "Bet Amount higher than remaining money. Reseting bet amount to 1";
                pause(4)
                app.general_message.Visible = 'off';
                app.bet_amount = 1;
                app.betText.Value = num2str(app.bet_amount);
            end

            app.dealerTotalCount.Visible = 'off';
            app.playerTotalCount.Visible = 'on';

            %load shuffle and deal sound effect
            [ya, Fs,] = audioread('shuffleSound.mp3');
            sound (ya, Fs);
            clear ya  Fs,
            pause(2)
            [ya,Fs] = audioread('dealCards.mp3');
            sound (ya, Fs);
            pause(1)

            app.hitButton.Visible = 'on';
            app.standButton.Visible = 'on';
            app.playerCard1.Visible = 'on';
            app.playerCard2.Visible = 'on';
            app.dealerCard1.Visible = 'on';
            app.HiddenDealerCard.Visible = 'on';
            app.change_bet_button.Visible = 'off';
            app.dealButton.Visible = 'off';
```

Initializes: this function is called everytime we start a new game in the given session.

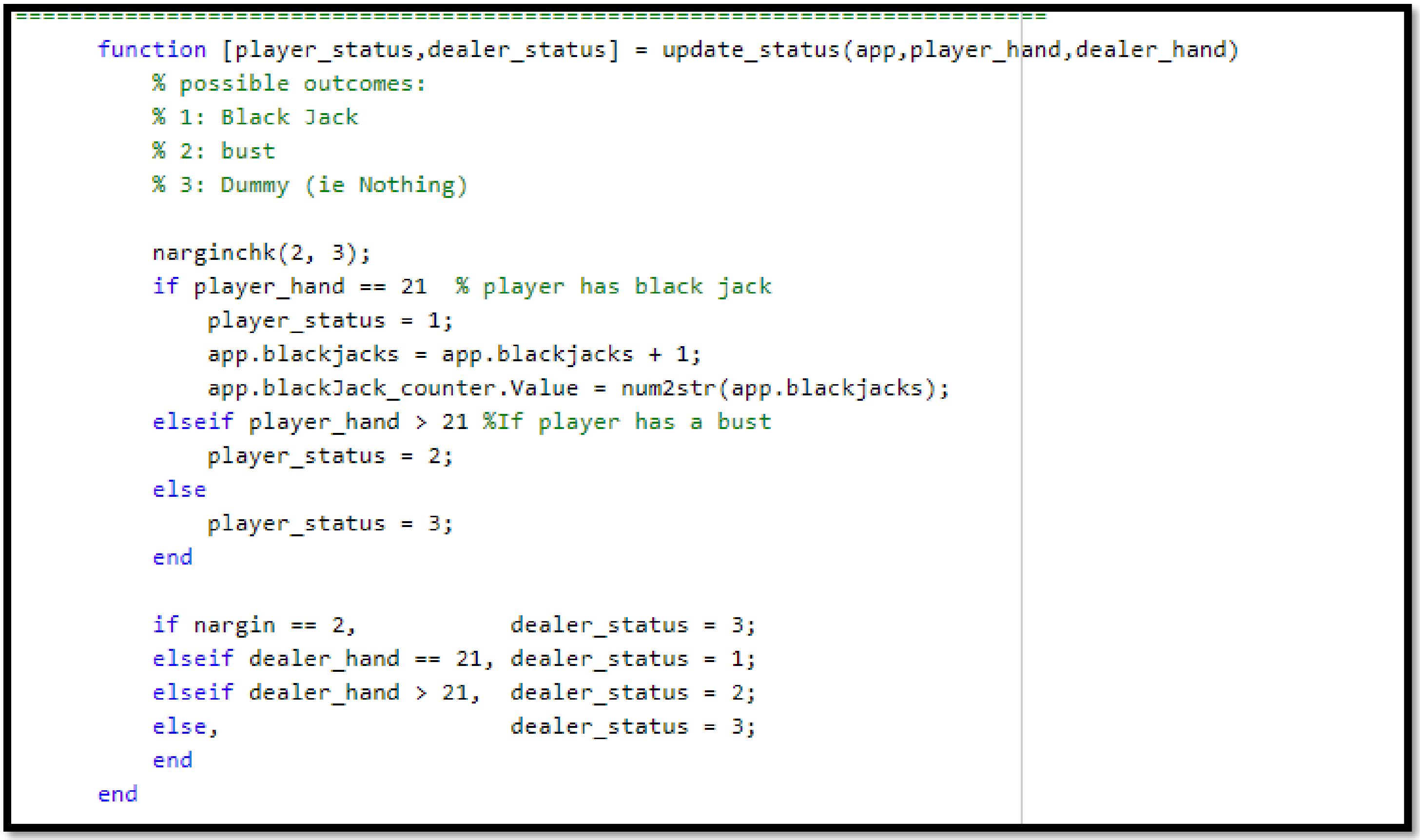
- Makes the chip image and bet text visible.
- Plays the shuffle and deal sound
- Makes the following entities visible: stand button, hit button, player1 cards, one of the dealer cards while the other card’s back will be shown



```
function new_game_init(app)
            app.player_remaining_money = str2double(app.total_money_text.Value);
            bet_amnt = str2double(app.betText.Value);
            if bet_amnt > app.player_remaining_money
                app.general_message.Visible = 'on';
                app.general_message.Text = "Total Money Amount can't be less than Bet amount. Reseting bet Amount to $1";
                pause(4)
                app.general_message.Visible = 'off';
                app.bet_amount = 1;
                app.betText.Value = num2str(app.bet_amount);
            elseif app.player_remaining_money < 1
                app.general_message.Visible = 'on';
                app.general_message.Text = "Total Money Amount can't be less than 1. Reseting amount to 10";
                pause(4)
                app.general_message.Visible = 'off';
                app.player_remaining_money = 10;
                app.total_money_text.Value = num2str(app.player_remaining_money);

            end
            app.total_money_text.Editable = 'off';

            %reset stats
            app.wins = 0;
            app.loss = 0;
            app.ties = 0;
            app.blackjacks = 0;
            app.losses_counter.Value = num2str(0);
            app.wins_counter.Value = num2str(0);
            app.ties_counter.Value = num2str(0);
            app.blackJack_counter.Value = num2str(0);
            app.new_game = 0;
        end
```

New_game_init: this function is only called when a new game starts.

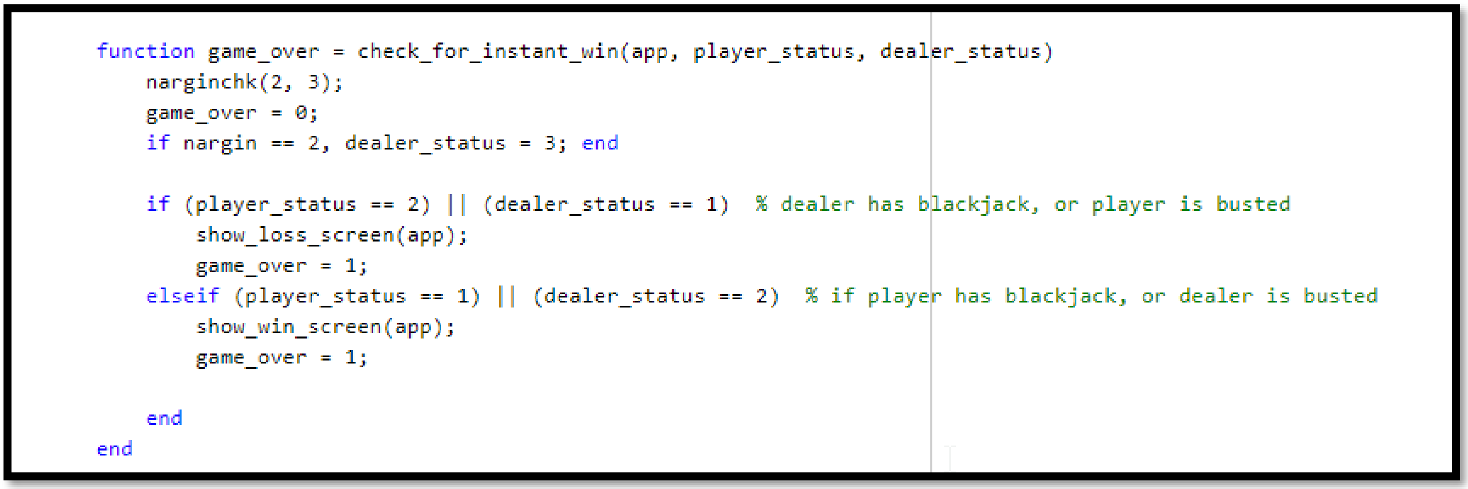
- Makes sure player entered correct bet and total money, and Resets the stats (win,loss,etc..) to zero



```
============================================================================
      function [player_status,dealer_status] = update_status(app,player_hand,dealer_hand)
          % possible outcomes:
          % 1: Black Jack
          % 2: bust
          % 3: Dummy (ie Nothing)

          narginchk(2, 3);
          if player_hand == 21  % player has black jack
              player_status = 1;
              app.blackjacks = app.blackjacks + 1;
              app.blackJack_counter.Value = num2str(app.blackjacks);
          elseif player_hand > 21 %If player has a bust
              player_status = 2;
          else
              player_status = 3;
          end

          if nargin == 2,           dealer_status = 3;
          elseif dealer_hand == 21, dealer_status = 1;
          elseif dealer_hand > 21,  dealer_status = 2;
          else,                     dealer_status = 3;
          end
      end
```

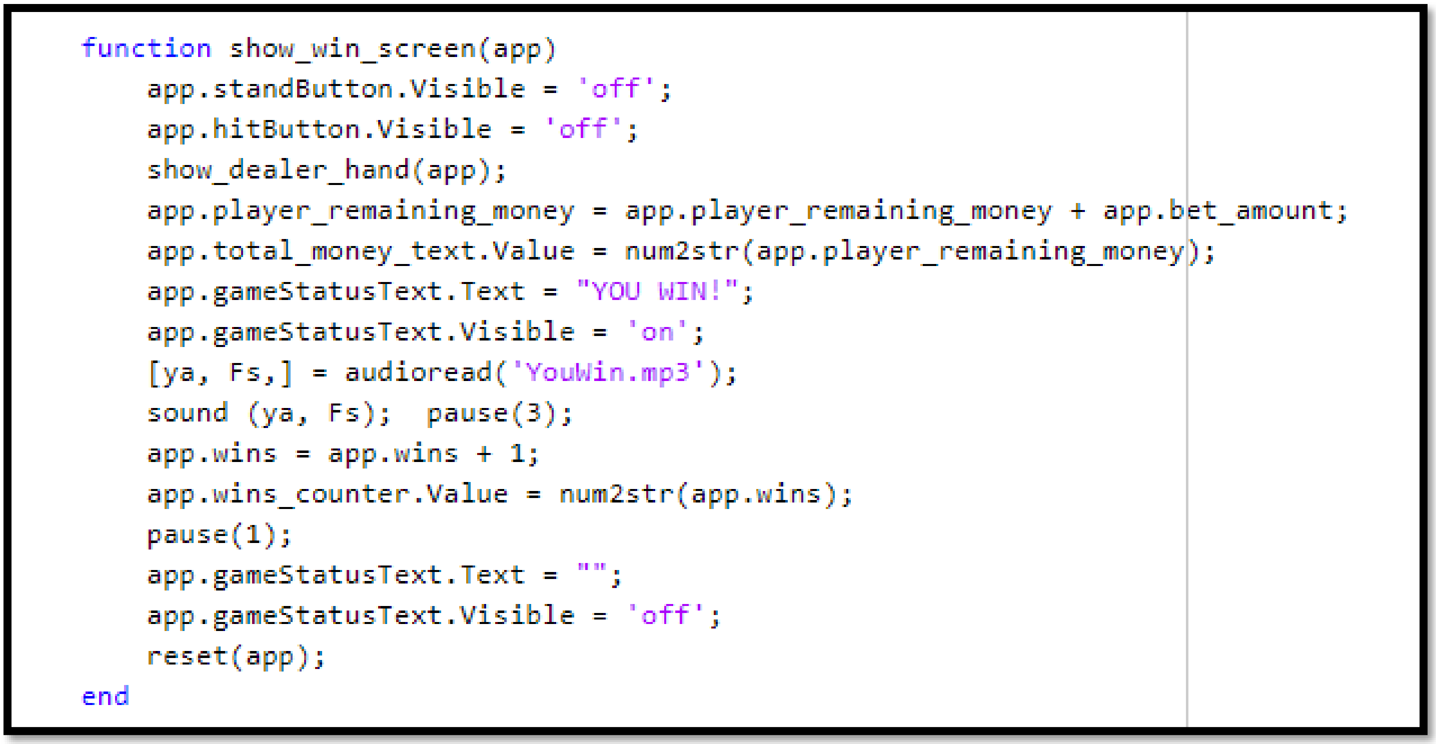
Update_status: this function updates the status of player and dealer hand

- Possible outcomes: if either the player or dealer has blackjack, then their status will be 1.
- If player/dealer is busted, their status will be 2
- Otherwise we assign dummy state 3



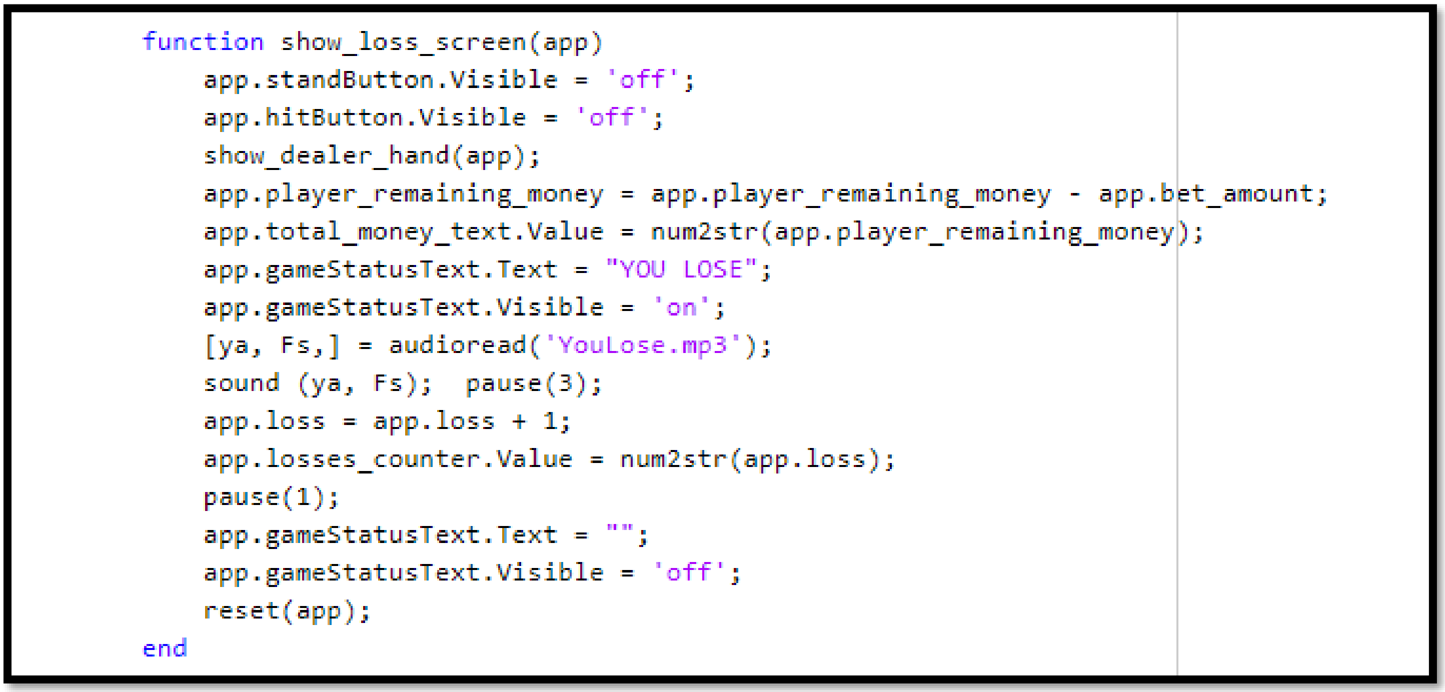
```
        function game_over = check_for_instant_win(app, player_status, dealer_status)
            narginchk(2, 3);
            game_over = 0;
            if nargin == 2, dealer_status = 3; end

            if (player_status == 2) || (dealer_status == 1)  % dealer has blackjack, or player is busted
                show_loss_screen(app);
                game_over = 1;
            elseif (player_status == 1) || (dealer_status == 2)  % if player has blackjack, or dealer is busted
                show_win_screen(app);
                game_over = 1;

            end
        end
```

Check_for_instant_win: if dealer has blackjack or player is busted, then dealer won and loss screen is shown. Else if player has black jack or dealer is busted, then player won and win screen is shown



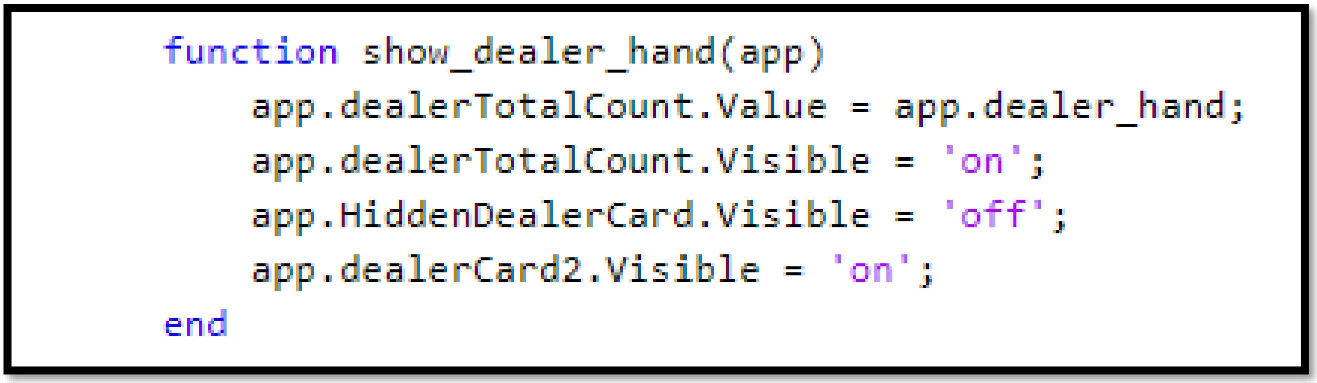
```
    function show_win_screen(app)
        app.standButton.Visible = 'off';
        app.hitButton.Visible = 'off';
        show_dealer_hand(app);
        app.player_remaining_money = app.player_remaining_money + app.bet_amount;
        app.total_money_text.Value = num2str(app.player_remaining_money);
        app.gameStatusText.Text = "YOU WIN!";
        app.gameStatusText.Visible = 'on';
        [ya, Fs,] = audioread('YouWin.mp3');
        sound (ya, Fs);  pause(3);
        app.wins = app.wins + 1;
        app.wins_counter.Value = num2str(app.wins);
        pause(1);
        app.gameStatusText.Text = "";
        app.gameStatusText.Visible = 'off';
        reset(app);
    end
```

Show_win_screen: called when the player won. It shows Win message and plays “YouWin” sound effect.



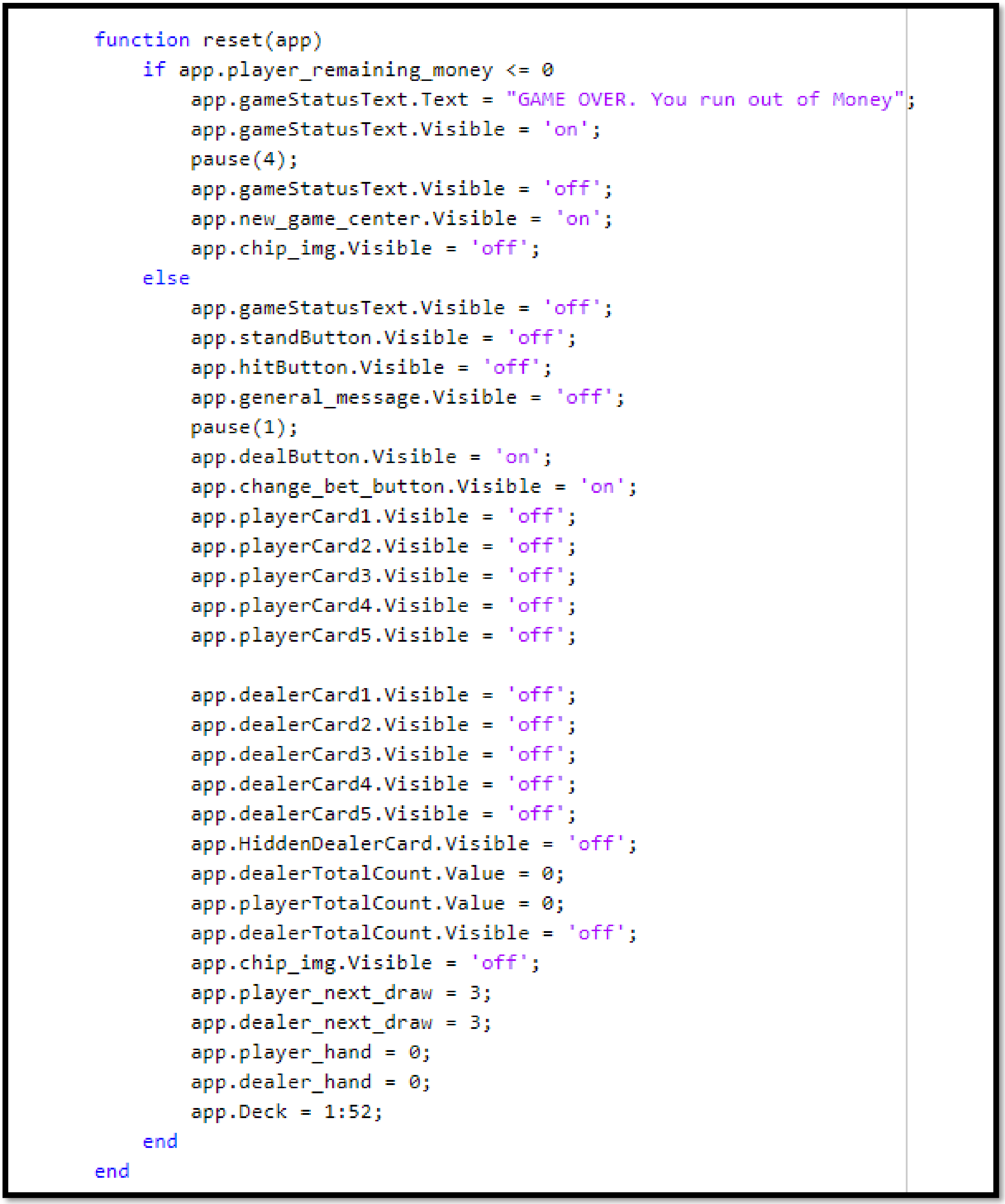
```
        function show_loss_screen(app)
            app.standButton.Visible = 'off';
            app.hitButton.Visible = 'off';
            show_dealer_hand(app);
            app.player_remaining_money = app.player_remaining_money - app.bet_amount;
            app.total_money_text.Value = num2str(app.player_remaining_money);
            app.gameStatusText.Text = "YOU LOSE";
            app.gameStatusText.Visible = 'on';
            [ya, Fs,] = audioread('YouLose.mp3');
            sound (ya, Fs);  pause(3);
            app.loss = app.loss + 1;
            app.losses_counter.Value = num2str(app.loss);
            pause(1);
            app.gameStatusText.Text = "";
            app.gameStatusText.Visible = 'off';
            reset(app);
        end
```

Show_loss_screen: called when the player losses. It shows Loss message and plays “YouLose” sound effect.



```
        function show_dealer_hand(app)
            app.dealerTotalCount.Value = app.dealer_hand;
            app.dealerTotalCount.Visible = 'on';
            app.HiddenDealerCard.Visible = 'off';
            app.dealerCard2.Visible = 'on';
        end
```

Show_dealer_hand: shows the dealer hand.

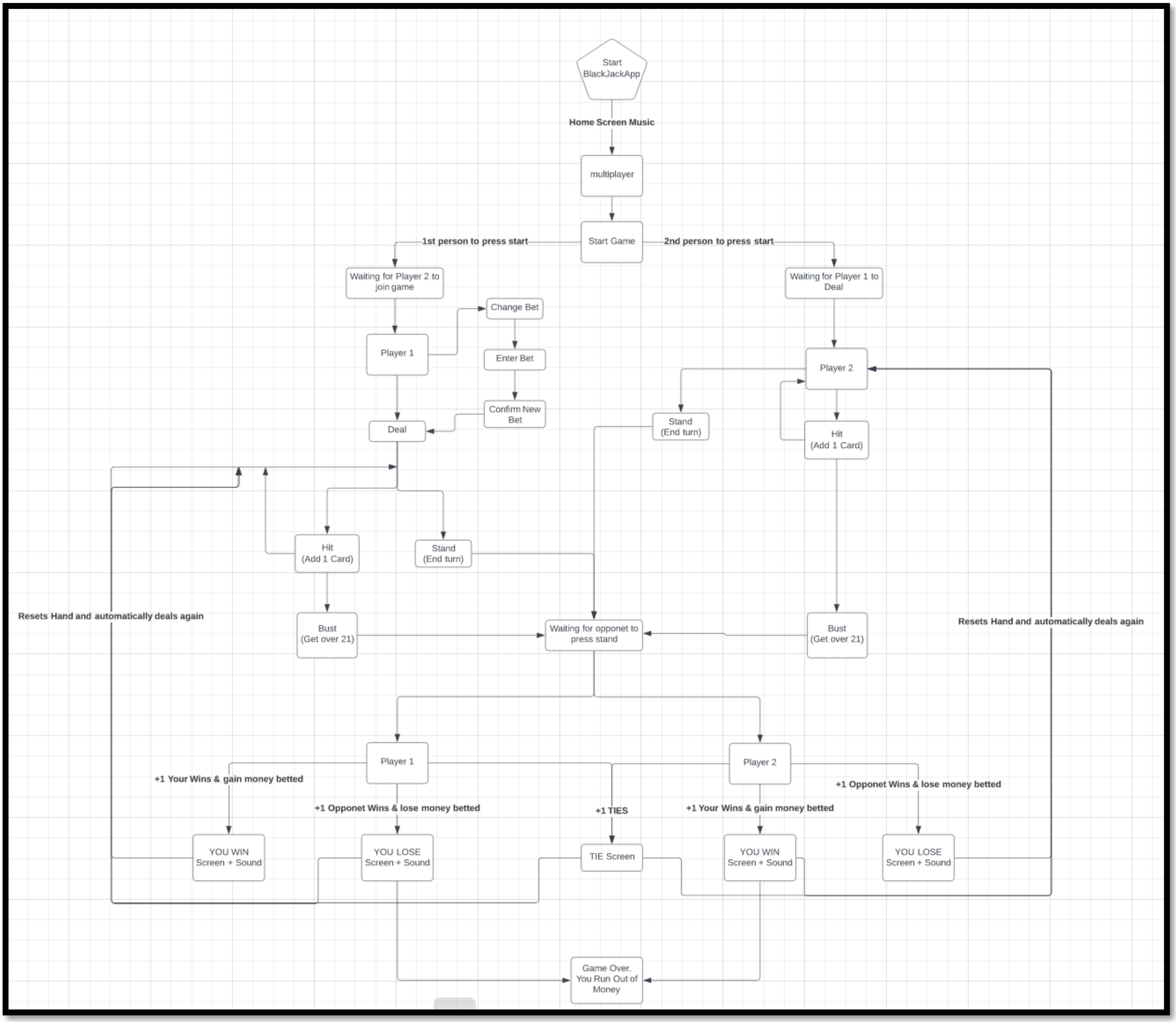


```
        function reset(app)
            if app.player_remaining_money <= 0
                app.gameStatusText.Text = "GAME OVER. You run out of Money";
                app.gameStatusText.Visible = 'on';
                pause(4);
                app.gameStatusText.Visible = 'off';
                app.new_game_center.Visible = 'on';
                app.chip_img.Visible = 'off';
            else
                app.gameStatusText.Visible = 'off';
                app.standButton.Visible = 'off';
                app.hitButton.Visible = 'off';
                app.general_message.Visible = 'off';
                pause(1);
                app.dealButton.Visible = 'on';
                app.change_bet_button.Visible = 'on';
                app.playerCard1.Visible = 'off';
                app.playerCard2.Visible = 'off';
                app.playerCard3.Visible = 'off';
                app.playerCard4.Visible = 'off';
                app.playerCard5.Visible = 'off';

                app.dealerCard1.Visible = 'off';
                app.dealerCard2.Visible = 'off';
                app.dealerCard3.Visible = 'off';
                app.dealerCard4.Visible = 'off';
                app.dealerCard5.Visible = 'off';
                app.HiddenDealerCard.Visible = 'off';
                app.dealerTotalCount.Value = 0;
                app.playerTotalCount.Value = 0;
                app.dealerTotalCount.Visible = 'off';
                app.chip_img.Visible = 'off';
                app.player_next_draw = 3;
                app.dealer_next_draw = 3;
                app.player_hand = 0;
                app.dealer_hand = 0;
                app.Deck = 1:52;
            end
        end
```

reset: called whenever a game finishes.

- If you run out of money, then it will show “game over” message and “new game” button will be visible at the center of the screen
- Otherwise resets the visibility of game entities (player cards, dealer cards, etc..)

Multiplayer Code

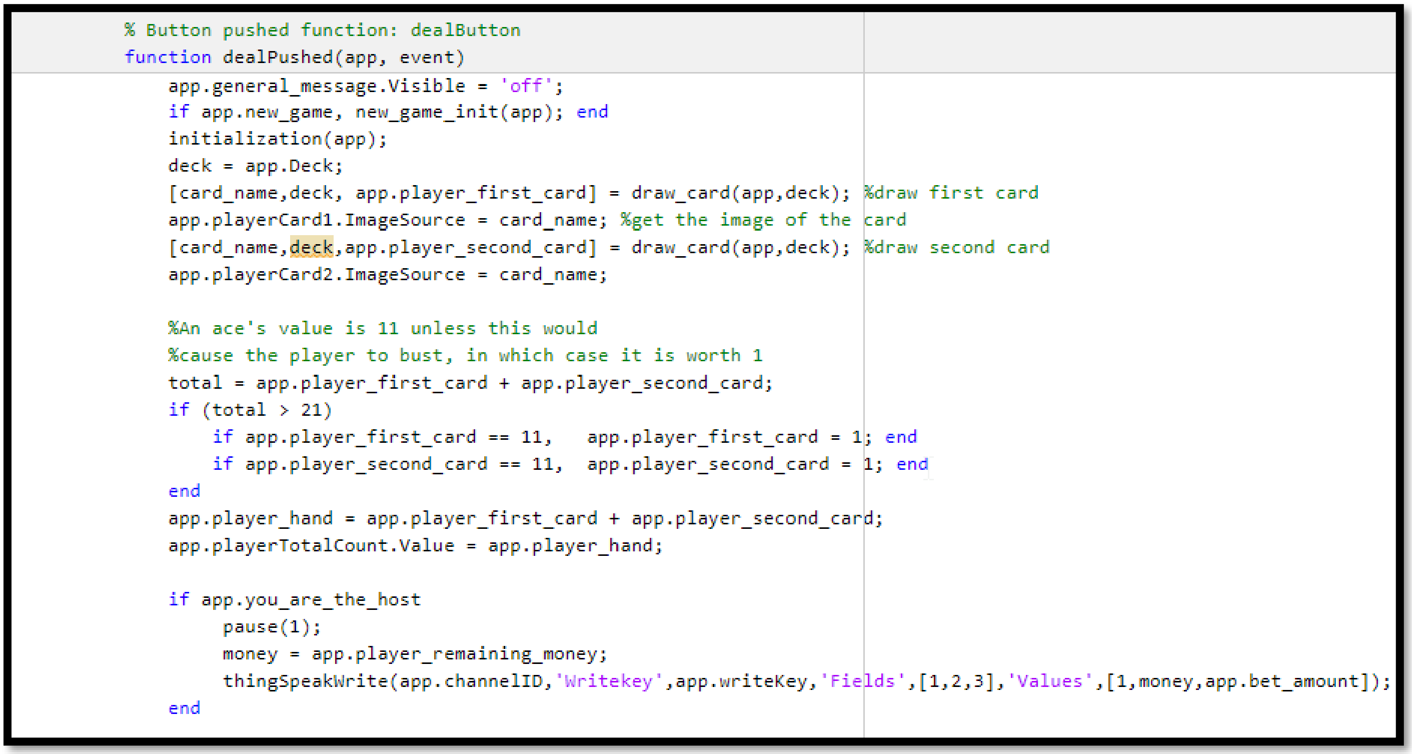


Start BlackJackApp
Home Screen Music
multiplayer
Start Game
1st person to press start
2nd person to press start
Waiting for Player 2 to join game
Waiting for Player 1 to Deal
Player 1
Change Bet
Enter Bet
Confirm New Bet
Deal
Player 2
Stand (End turn)
Hit (Add 1 Card)
Hit (Add 1 Card)
Stand (End turn)
Resets Hand and automatically deals again
Bust (Get over 21)
Waiting for opponet to press stand
Bust (Get over 21)
Resets Hand and automatically deals again
Player 1
Player 2
+1 Your Wins & gain money betted
+1 Opponet Wins & lose money betted
+1 TIES
+1 Your Wins & gain money betted
+1 Opponet Wins & lose money betted
YOU WIN Screen + Sound
YOU LOSE Screen + Sound
TIE Screen
YOU WIN Screen + Sound
YOU LOSE Screen + Sound
Game Over. You Run Out of Money

- Flowchart explaining the Multiplayer game flow

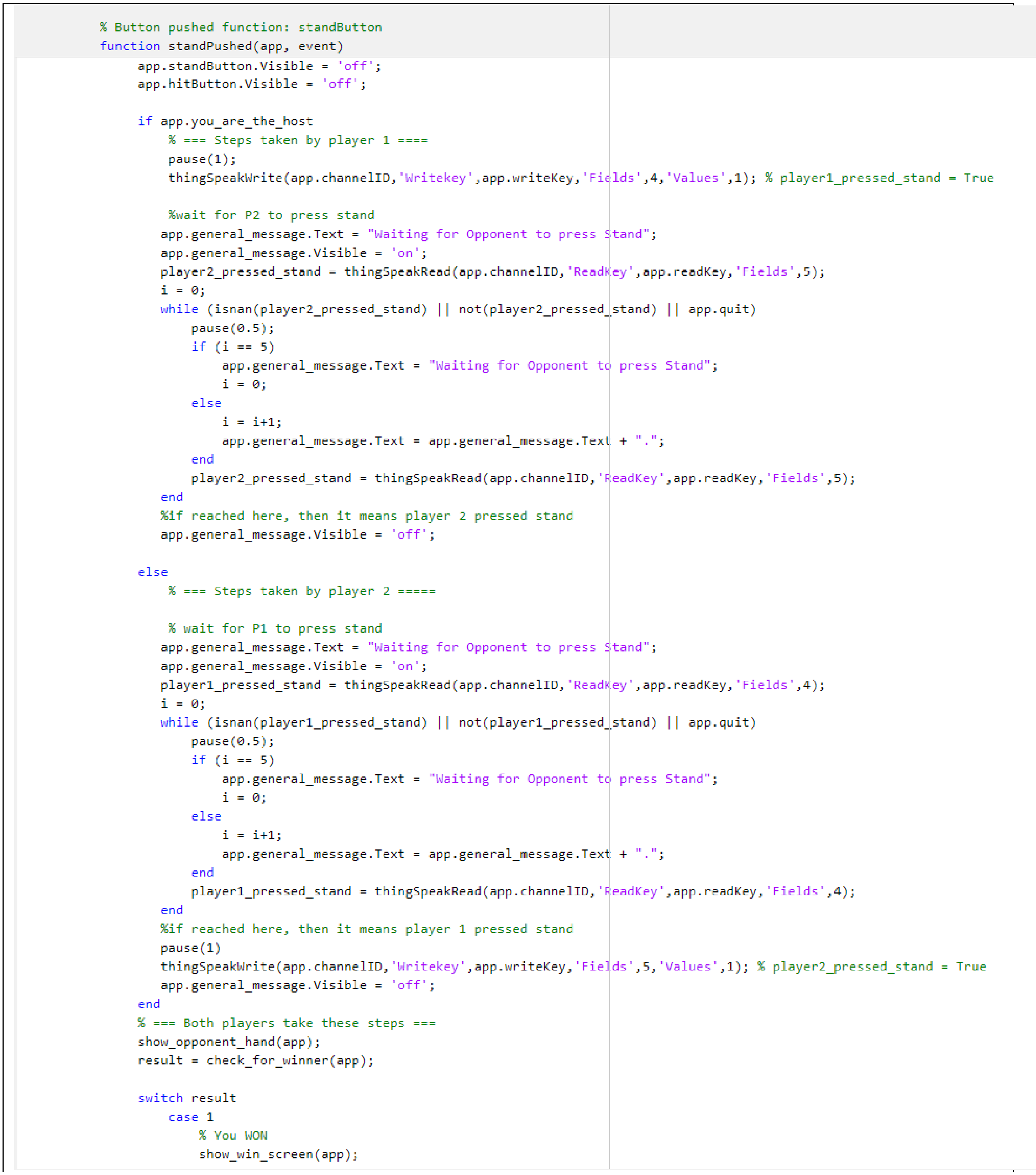
The functions:

- **Note:** the following functions : (changeBetPushed, ConfirmNewBetButton, tutorialButton, hitpushed, draw_card, new_game_init, intializaiton, show_win_screen, show_loss_screen, reset) are the same as the singleplayer so we wont' explain them here

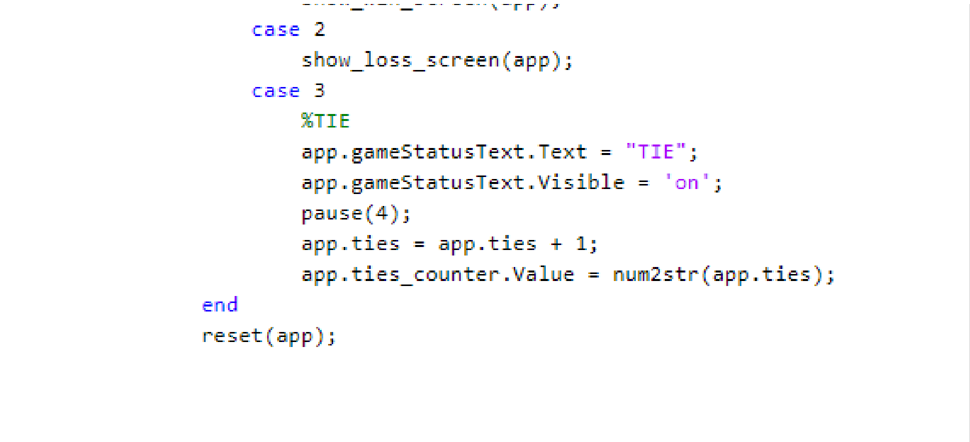


```
        % Button pushed function: dealButton
        function dealPushed(app, event)
            app.general_message.Visible = 'off';
            if app.new_game, new_game_init(app); end
            initialization(app);
            deck = app.Deck;
            [card_name,deck, app.player_first_card] = draw_card(app,deck); %draw first card
            app.playerCard1.ImageSource = card_name; %get the image of the card
            [card_name,deck,app.player_second_card] = draw_card(app,deck); %draw second card
            app.playerCard2.ImageSource = card_name;

            %An ace's value is 11 unless this would
            %cause the player to bust, in which case it is worth 1
            total = app.player_first_card + app.player_second_card;
            if (total > 21)
                if app.player_first_card == 11,   app.player_first_card = 1; end
                if app.player_second_card == 11,  app.player_second_card = 1; end
            end
            app.player_hand = app.player_first_card + app.player_second_card;
            app.playerTotalCount.Value = app.player_hand;

            if app.you_are_the_host
                 pause(1);
                 money = app.player_remaining_money;
                 thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',[1,2,3],'Values',[1,money,app.bet_amount]);
            end
```

dealPushed: The same as the single player except we added the thingspeak functionality. If you the host (ie player 1), then you the function will push total money and bet amount to things speak (field 2 and 3 respectively), and infrom the Player2 that you pressed the deal button, so that they can continue the game (since player2 was waiting for you to press the deal button)



```
        % Button pushed function: standButton
        function standPushed(app, event)
             app.standButton.Visible = 'off';
             app.hitButton.Visible = 'off';

             if app.you_are_the_host
                 % === Steps taken by player 1 ====
                 pause(1);
                 thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',4,'Values',1); % player1_pressed_stand = True

                 %wait for P2 to press stand
                app.general_message.Text = "Waiting for Opponent to press Stand";
                app.general_message.Visible = 'on';
                player2_pressed_stand = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',5);
                i = 0;
                while (isnan(player2_pressed_stand) || not(player2_pressed_stand) || app.quit)
                    pause(0.5);
                    if (i == 5)
                        app.general_message.Text = "Waiting for Opponent to press Stand";
                        i = 0;
                    else
                        i = i+1;
                        app.general_message.Text = app.general_message.Text + ".";
                    end
                    player2_pressed_stand = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',5);
                end
                %if reached here, then it means player 2 pressed stand
                app.general_message.Visible = 'off';

             else
                 % === Steps taken by player 2 =====

                 % wait for P1 to press stand
                app.general_message.Text = "Waiting for Opponent to press Stand";
                app.general_message.Visible = 'on';
                player1_pressed_stand = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',4);
                i = 0;
                while (isnan(player1_pressed_stand) || not(player1_pressed_stand) || app.quit)
                    pause(0.5);
                    if (i == 5)
                        app.general_message.Text = "Waiting for Opponent to press Stand";
                        i = 0;
                    else
                        i = i+1;
                        app.general_message.Text = app.general_message.Text + ".";
                    end
                    player1_pressed_stand = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',4);
                end
                %if reached here, then it means player 1 pressed stand
                pause(1)
                thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',5,'Values',1); % player2_pressed_stand = True
                app.general_message.Visible = 'off';
             end
             % === Both players take these steps ===
             show_opponent_hand(app);
             result = check_for_winner(app);

             switch result
                 case 1
                     % You WON
                     show_win_screen(app);
```



```
        show_win_screen(app);
    case 2
        show_loss_screen(app);
    case 3
        %TIE
        app.gameStatusText.Text = "TIE";
        app.gameStatusText.Visible = 'on';
        pause(4);
        app.ties = app.ties + 1;
        app.ties_counter.Value = num2str(app.ties);
end
reset(app);
```

standPushed: This is callback function, called whenever the player presses the “stand” button signaling the end of their turn.

if it’s player1, then the function does the following:

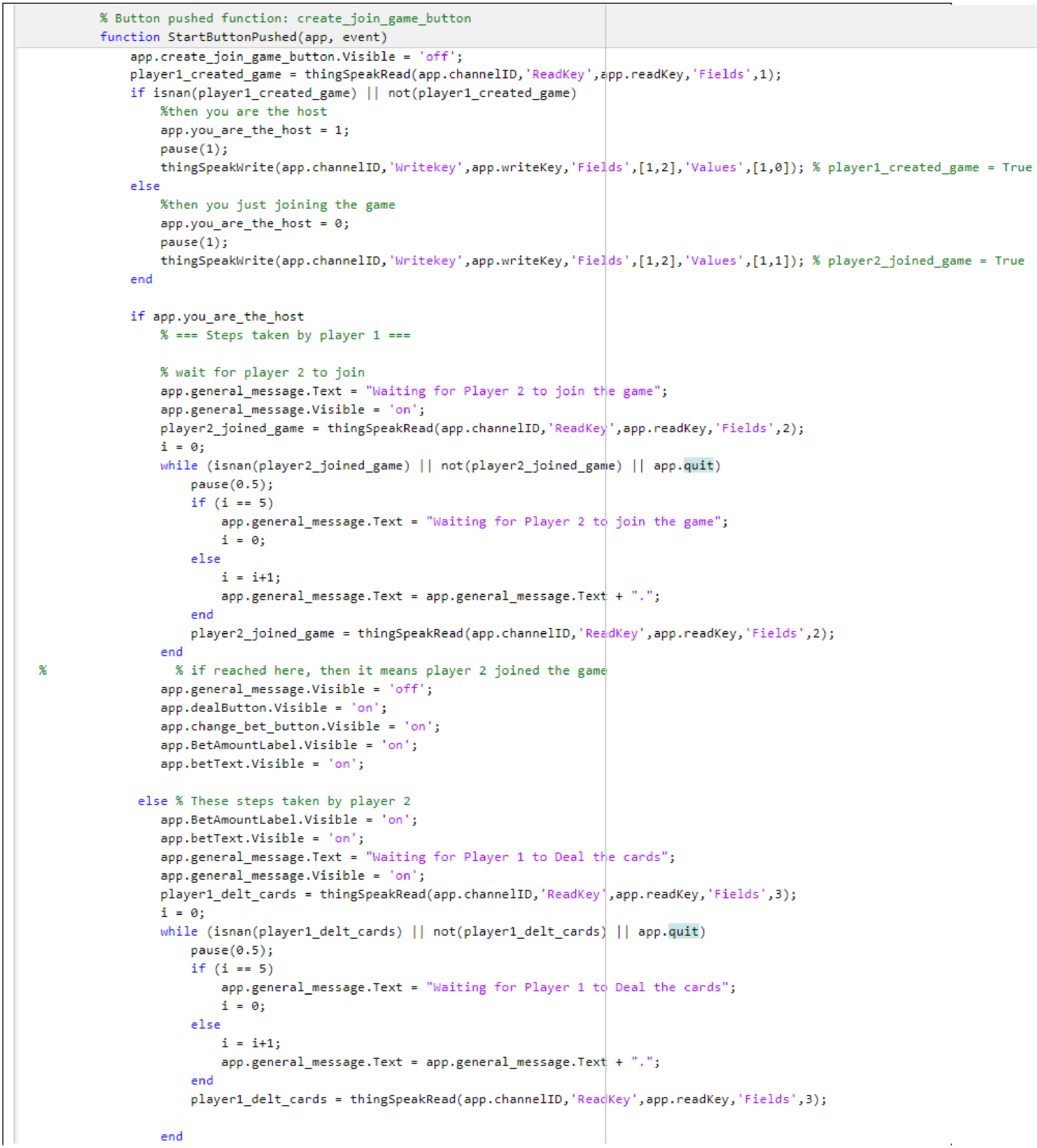
- Let player2 know that you pressed stand by pushing to thingspeak
- Wait in while loop for player2 to press stand by consistently checking the thingspeak for changes
- While waiting it’s does show dots animation to indicate to the player that the screen is not freezing

If it’s player2, then function does the following:

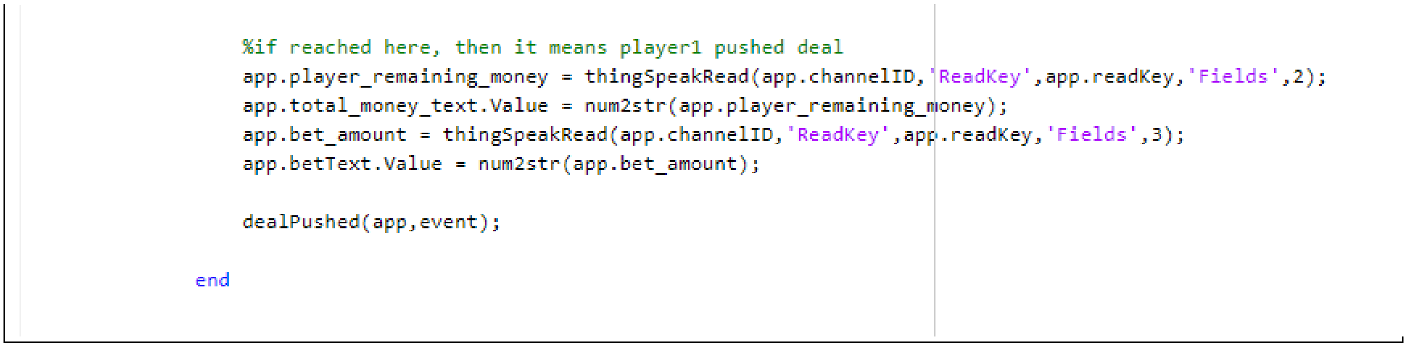
- Wait in while loop for player1 to press stand by consistently checking the thingspeak for changes + do dots animation

After that the function does the following:

- After both player pressed stand, reveal the hands by calling “show_opponent_hand” function, and check for winner by calling “check_for_winner” function.
- Show the win or loss screen, then start a new game by calling reset function



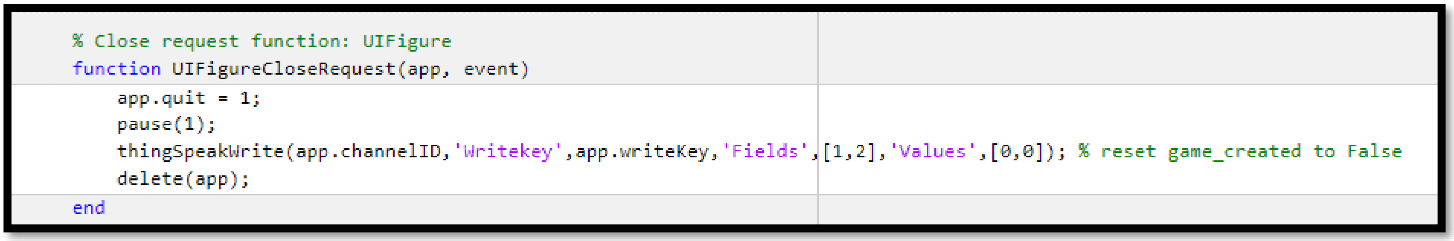
```
        % Button pushed function: create_join_game_button
        function StartButtonPushed(app, event)
            app.create_join_game_button.Visible = 'off';
            player1_created_game = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',1);
            if isnan(player1_created_game) || not(player1_created_game)
                %then you are the host
                app.you_are_the_host = 1;
                pause(1);
                thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',[1,2],'Values',[1,0]); % player1_created_game = True
            else
                %then you just joining the game
                app.you_are_the_host = 0;
                pause(1);
                thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',[1,2],'Values',[1,1]); % player2_joined_game = True
            end

            if app.you_are_the_host
                % === Steps taken by player 1 ===

                % wait for player 2 to join
                app.general_message.Text = "Waiting for Player 2 to join the game";
                app.general_message.Visible = 'on';
                player2_joined_game = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',2);
                i = 0;
                while (isnan(player2_joined_game) || not(player2_joined_game) || app.quit)
                    pause(0.5);
                    if (i == 5)
                        app.general_message.Text = "Waiting for Player 2 to join the game";
                        i = 0;
                    else
                        i = i+1;
                        app.general_message.Text = app.general_message.Text + ".";
                    end
                    player2_joined_game = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',2);
                end
%                 % if reached here, then it means player 2 joined the game
                app.general_message.Visible = 'off';
                app.dealButton.Visible = 'on';
                app.change_bet_button.Visible = 'on';
                app.BetAmountLabel.Visible = 'on';
                app.betText.Visible = 'on';

             else % These steps taken by player 2
                app.BetAmountLabel.Visible = 'on';
                app.betText.Visible = 'on';
                app.general_message.Text = "Waiting for Player 1 to Deal the cards";
                app.general_message.Visible = 'on';
                player1_delt_cards = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',3);
                i = 0;
                while (isnan(player1_delt_cards) || not(player1_delt_cards) || app.quit)
                    pause(0.5);
                    if (i == 5)
                        app.general_message.Text = "Waiting for Player 1 to Deal the cards";
                        i = 0;
                    else
                        i = i+1;
                        app.general_message.Text = app.general_message.Text + ".";
                    end
                    player1_delt_cards = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',3);

                end
```



```
                %if reached here, then it means player1 pushed deal
                app.player_remaining_money = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',2);
                app.total_money_text.Value = num2str(app.player_remaining_money);
                app.bet_amount = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',3);
                app.betText.Value = num2str(app.bet_amount);

                dealPushed(app,event);

            end
```

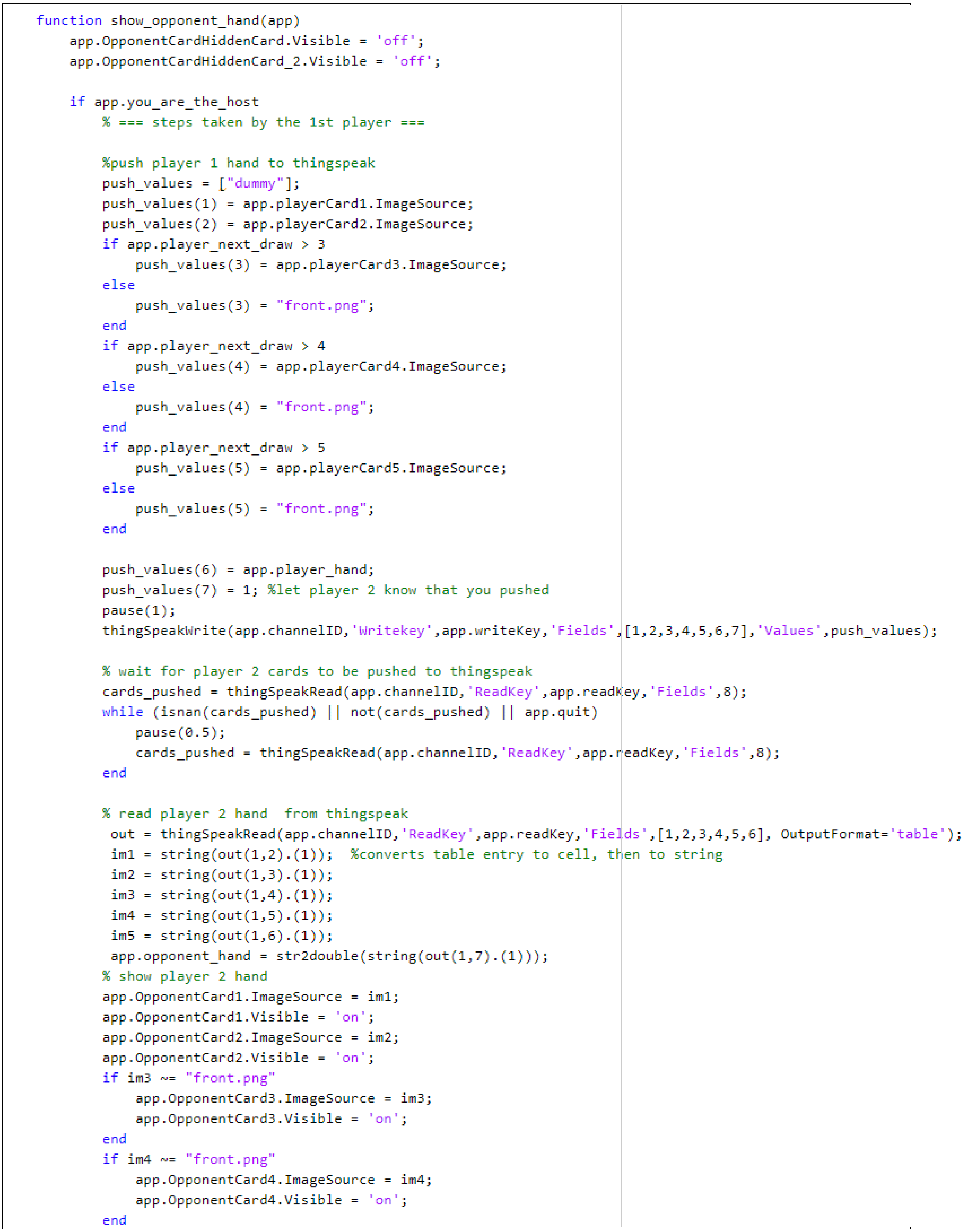
StartButtonPushed: this is callback function, called whenever the player presses the “START” button at the beginning of the game.

- Hide the “START” button
- Whoever clicked the start first will be the player1 (ie the host). Player1 will push to thingspeak “player1_create_game = True” by setting “field 1” to value 1, then the player1 will be waiting in a while loop for player2 to join.
- If it’s player2, then the function pushes to thingspeak to inform player1 that the player2 joined the game
- After player2 joins the game, the “deal” and “change bet” buttons will be visible to the player1.
- player2 will be waiting in while loop for player1 to deal the cards and constantly checking thingspeak for changes. When the player1 deals the card, player2 will be informed, and exit the loop, then pull the bet amount from thingspeak, then hit and stand button will be visible to player2.

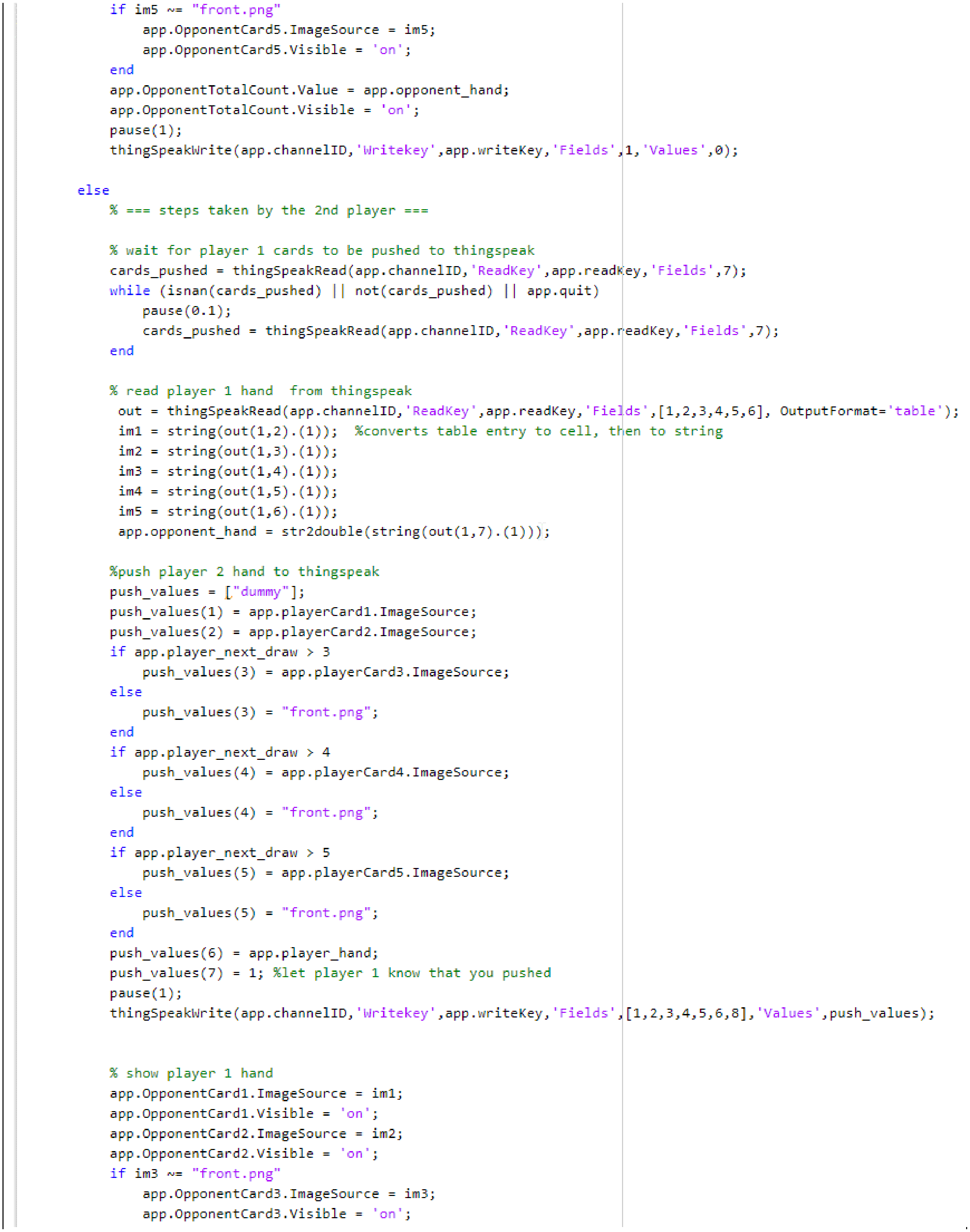


```
        % Close request function: UIFigure
        function UIFigureCloseRequest(app, event)
            app.quit = 1;
            pause(1);
            thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',[1,2],'Values',[0,0]); % reset game_created to False
            delete(app);
        end
```

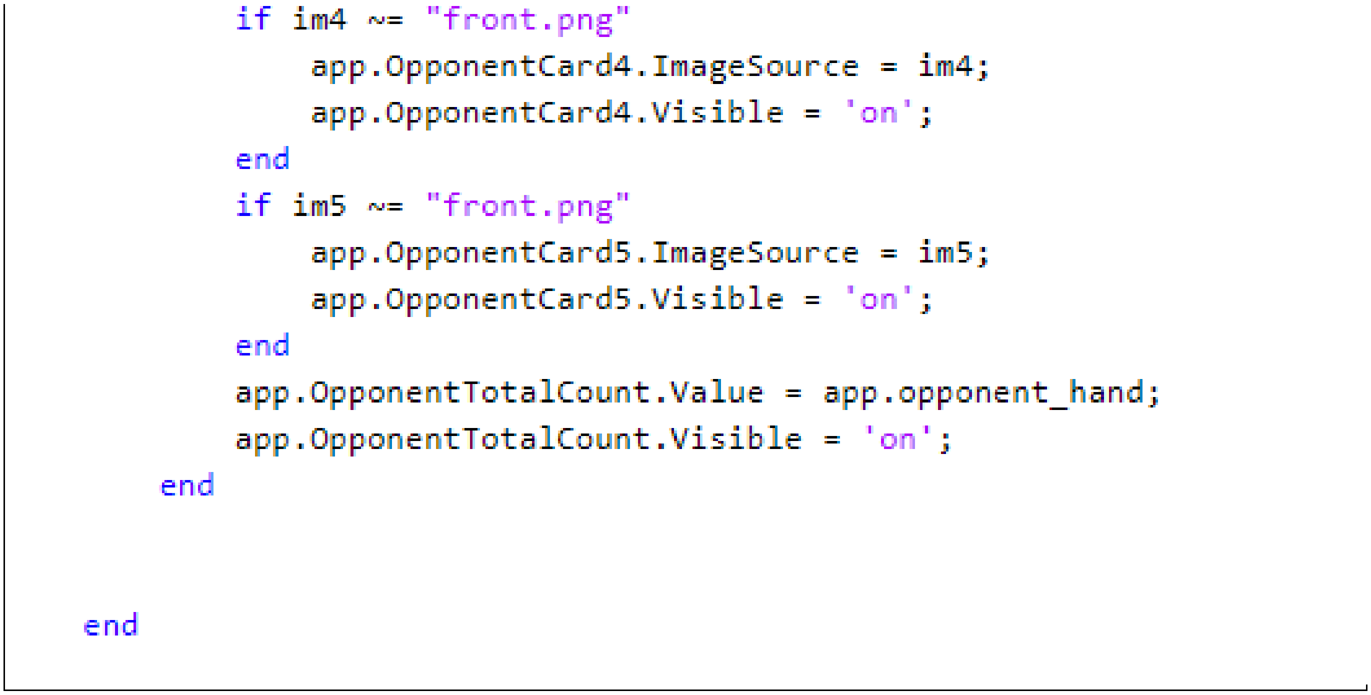
UIFigureCloseRequest: called whenever you close the UI Figure, it resets the thingspseak fields to default state, and then deltes the app.



```
function show_opponent_hand(app)
    app.OpponentCardHiddenCard.Visible = 'off';
    app.OpponentCardHiddenCard_2.Visible = 'off';

    if app.you_are_the_host
        % === steps taken by the 1st player ===

        %push player 1 hand to thingspeak
        push_values = ["dummy"];
        push_values(1) = app.playerCard1.ImageSource;
        push_values(2) = app.playerCard2.ImageSource;
        if app.player_next_draw > 3
            push_values(3) = app.playerCard3.ImageSource;
        else
            push_values(3) = "front.png";
        end
        if app.player_next_draw > 4
            push_values(4) = app.playerCard4.ImageSource;
        else
            push_values(4) = "front.png";
        end
        if app.player_next_draw > 5
            push_values(5) = app.playerCard5.ImageSource;
        else
            push_values(5) = "front.png";
        end

        push_values(6) = app.player_hand;
        push_values(7) = 1; %let player 2 know that you pushed
        pause(1);
        thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',[1,2,3,4,5,6,7],'Values',push_values);

        % wait for player 2 cards to be pushed to thingspeak
        cards_pushed = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',8);
        while (isnan(cards_pushed) || not(cards_pushed) || app.quit)
            pause(0.5);
            cards_pushed = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',8);
        end

        % read player 2 hand  from thingspeak
         out = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',[1,2,3,4,5,6], OutputFormat='table');
         im1 = string(out(1,2).(1));  %converts table entry to cell, then to string
         im2 = string(out(1,3).(1));
         im3 = string(out(1,4).(1));
         im4 = string(out(1,5).(1));
         im5 = string(out(1,6).(1));
         app.opponent_hand = str2double(string(out(1,7).(1)));
        % show player 2 hand
        app.OpponentCard1.ImageSource = im1;
        app.OpponentCard1.Visible = 'on';
        app.OpponentCard2.ImageSource = im2;
        app.OpponentCard2.Visible = 'on';
        if im3 ~= "front.png"
            app.OpponentCard3.ImageSource = im3;
            app.OpponentCard3.Visible = 'on';
        end
        if im4 ~= "front.png"
            app.OpponentCard4.ImageSource = im4;
            app.OpponentCard4.Visible = 'on';
        end
```



```
        if im5 ~= "front.png"
            app.OpponentCard5.ImageSource = im5;
            app.OpponentCard5.Visible = 'on';
        end
        app.OpponentTotalCount.Value = app.opponent_hand;
        app.OpponentTotalCount.Visible = 'on';
        pause(1);
        thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',1,'Values',0);

    else
        % === steps taken by the 2nd player ===

        % wait for player 1 cards to be pushed to thingspeak
        cards_pushed = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',7);
        while (isnan(cards_pushed) || not(cards_pushed) || app.quit)
            pause(0.1);
            cards_pushed = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',7);
        end

        % read player 1 hand  from thingspeak
         out = thingSpeakRead(app.channelID,'ReadKey',app.readKey,'Fields',[1,2,3,4,5,6], OutputFormat='table');
         im1 = string(out(1,2).(1));  %converts table entry to cell, then to string
         im2 = string(out(1,3).(1));
         im3 = string(out(1,4).(1));
         im4 = string(out(1,5).(1));
         im5 = string(out(1,6).(1));
         app.opponent_hand = str2double(string(out(1,7).(1)));

        %push player 2 hand to thingspeak
        push_values = ["dummy"];
        push_values(1) = app.playerCard1.ImageSource;
        push_values(2) = app.playerCard2.ImageSource;
        if app.player_next_draw > 3
            push_values(3) = app.playerCard3.ImageSource;
        else
            push_values(3) = "front.png";
        end
        if app.player_next_draw > 4
            push_values(4) = app.playerCard4.ImageSource;
        else
            push_values(4) = "front.png";
        end
        if app.player_next_draw > 5
            push_values(5) = app.playerCard5.ImageSource;
        else
            push_values(5) = "front.png";
        end
        push_values(6) = app.player_hand;
        push_values(7) = 1; %let player 1 know that you pushed
        pause(1);
        thingSpeakWrite(app.channelID,'Writekey',app.writeKey,'Fields',[1,2,3,4,5,6,8],'Values',push_values);


        % show player 1 hand
        app.OpponentCard1.ImageSource = im1;
        app.OpponentCard1.Visible = 'on';
        app.OpponentCard2.ImageSource = im2;
        app.OpponentCard2.Visible = 'on';
        if im3 ~= "front.png"
            app.OpponentCard3.ImageSource = im3;
            app.OpponentCard3.Visible = 'on';
```



```
            if im4 ~= "front.png"
                app.OpponentCard4.ImageSource = im4;
                app.OpponentCard4.Visible = 'on';
            end
            if im5 ~= "front.png"
                app.OpponentCard5.ImageSource = im5;
                app.OpponentCard5.Visible = 'on';
            end
            app.OpponentTotalCount.Value = app.opponent_hand;
            app.OpponentTotalCount.Visible = 'on';
        end


    end
```

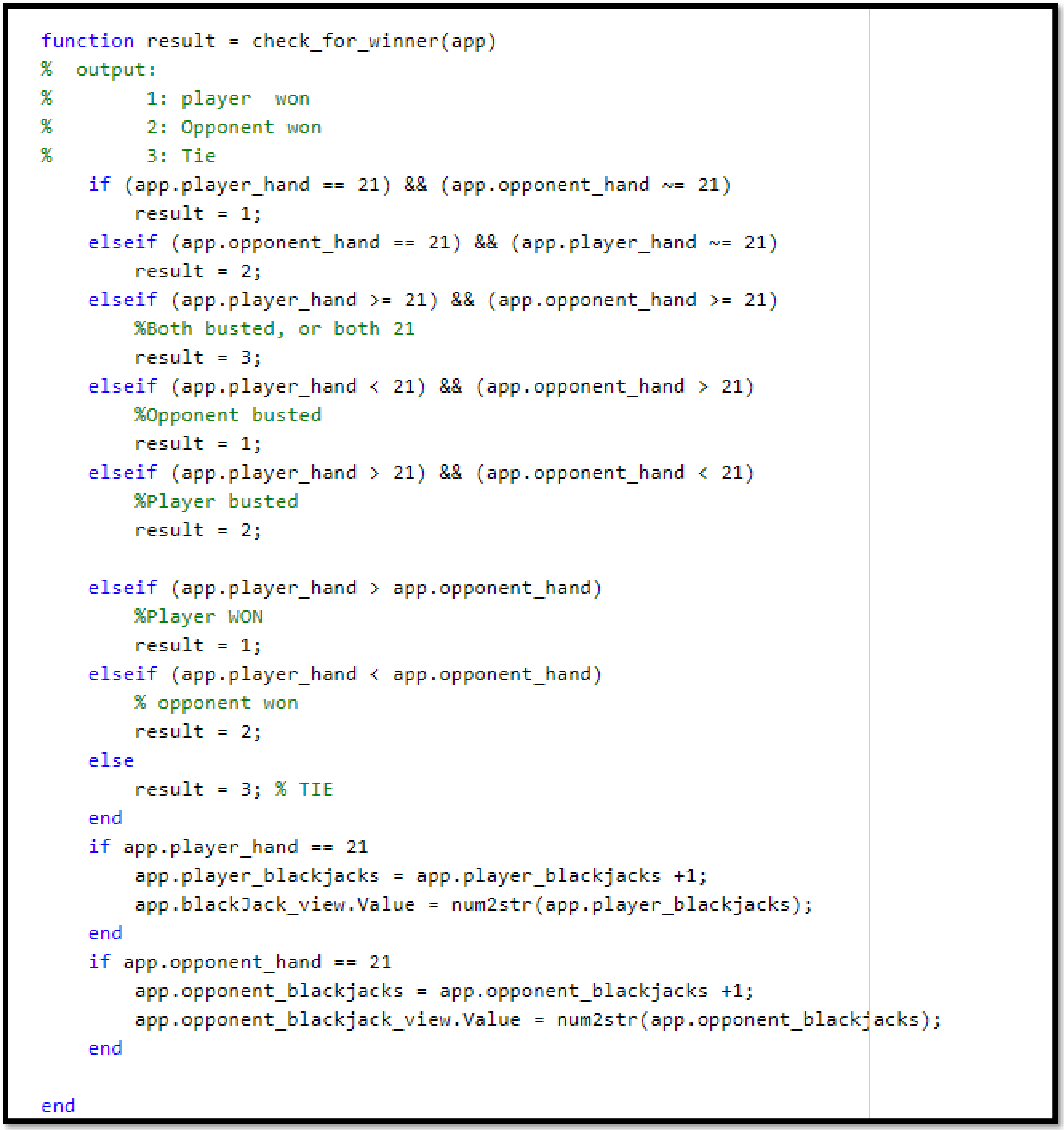
Show_opponent_hand: this function pushed player cards to thingspeak, and reveals the opponent hand

If it’s player1, then the function does the following:

- Push player1 cards and hand value to thingspeak
- Let player2 know that did the push to thingspeak
- Wait for player2 cards to be pushed to thingspeak
- Then read player2 cards and hand value
- Show player2 hand

If it’s player2, then the function does the following:

- Wait for player1 cards to be pushed to thingspeak
- Read player 1 cards and hand value from thingspeak
- Then push player2 cards and hand value to thingspeak
- Let player1 know that you pushed
- Show player 1 hand



```
function result = check_for_winner(app)
%  output:
%        1: player  won
%        2: Opponent won
%        3: Tie
    if (app.player_hand == 21) && (app.opponent_hand ~= 21)
        result = 1;
    elseif (app.opponent_hand == 21) && (app.player_hand ~= 21)
        result = 2;
    elseif (app.player_hand >= 21) && (app.opponent_hand >= 21)
        %Both busted, or both 21
        result = 3;
    elseif (app.player_hand < 21) && (app.opponent_hand > 21)
        %Opponent busted
        result = 1;
    elseif (app.player_hand > 21) && (app.opponent_hand < 21)
        %Player busted
        result = 2;

    elseif (app.player_hand > app.opponent_hand)
        %Player WON
        result = 1;
    elseif (app.player_hand < app.opponent_hand)
        % opponent won
        result = 2;
    else
        result = 3; % TIE
    end
    if app.player_hand == 21
        app.player_blackjacks = app.player_blackjacks +1;
        app.blackJack_view.Value = num2str(app.player_blackjacks);
    end
    if app.opponent_hand == 21
        app.opponent_blackjacks = app.opponent_blackjacks +1;
        app.opponent_blackjack_view.Value = num2str(app.opponent_blackjacks);
    end

end
```

Check_for_winner: this functions checks whether the player won (1), opponent won (2), or it’s tie (3)