BlackJack.m

%BlackJack Twenty-One

%Main Function

%Written by Ishmael Leon

% Christopher Reid

% Scott Copley

% Jordane MCcalla

%December XX, 2017

%Constants to be initialized

close all;

clear global;

clc;

% global keisha;

%global - variables can be used in every function under the current folder

global CARDS;

global BANKROLL;

global BJDEALER;

global BJPLAYER;

global BJBOARD;

global QUITGAME;

global GENERAL;

BJBOARD.MaxDelay = 3;

% with global variables, we can use the dot feature to create a structure

% of that variable

%Open the graphics interface that a player uses to setup the game

Init = BJStartValues;

BJPLAYER.Total = 0;

BJDEALER.Total = 0;

BJPLAYER.Status = 'Stand';

BJPLAYER.HandsPlayed = 0;

BJPLAYER.Wins = 0;

BJPLAYER.Loses = 0;

BJPLAYER.Pushes = 0;

BJPLAYER.Percent = 0;

BANKROLL.Starting = 0;

BANKROLL.HandsLeft = Init.NumHands;

BANKROLL.Starting = Init.Bankroll;

BANKROLL.Unit = Init.BetUnit; % Point A . Let us find Init.BetUnit

CARDS.DecksUsed = Init.NumDecks;

% disp(Init.PN);

BJPLAYER.PN = Init.PN;

BJBOARD.Delays = BJBOARD.MaxDelay - BJBOARD.MaxDelay\*Init.GameSpeed/100; % Controls the speed of the game

clear Init;

%First checking if it is a multi deck game - number of card packs to be

%used

if CARDS.DecksUsed == 1

CARDS.MultiDeck = 0;

else

CARDS.MultiDeck = 1;

end

%Declaring and initializing constant variable shuffle

CARDS.Shuffle = 1; %Cards Shuffle when = 1

BANKROLL.History = BANKROLL.Starting; %Ensures that startup bank amount is the same

Bet = BANKROLL.Unit; %Inital Bet

BANKROLL.InsuranceBet = 0;

QUITGAME = 0; %Ensures proper execution of our sentinal function

%Program ends when 1 - To aid the functioning of the

%conditional while loop (while ~quitgame)... run program

FirstHand = 1;

AllVisible = 0; %When 1, all objects will be displayed on the board (for debug) - To ensure the proper functioning of the different elements on the board

BJPLAYER.XIncrement = [];

BJPLAYER.XIncrement(1) = 0; %The increment in x direction between

BJPLAYER.XIncrement(2) = 0; %dealt cards - The spacing between

%the cards of a hand on the board (horizontally)

%Call the function that creates our playing board

Help;

uiwait;

handles = BJBoardCreation(Bet);

% Therefore handles takes in an assignment of the output of

%BJ BJBoardCreation(Bet)

% Let us go to BJBoardCreation() !!...

pause(.2);

try

while ~QUITGAME

%Prepares array of cards randomly

if CARDS.Shuffle == 1

ShuffleNow(CARDS.DecksUsed);

CARDS.Shuffle = 0;

CARDS.CardsLeft = CARDS.DecksUsed\*52; %Total number of cards

CARDS.NextCard = 1; %Next card to be dealt

end

%Variables initialized with every hand

BJPLAYER.Total = 0; %Player's Card Total

BJDEALER.Total = 0; %Dealer's Card Total

BJPLAYER.Ace = 0; %Becomes 1 if Player has an Ace used as 11

BJDEALER.Ace = 0; %Becomes 1 if Dealer has an Ace used as 11

BJDEALER.Ace\_2 = 0;

BJDEALER.Ten = 0;

BJPLAYER.Hand=[]; %Player's Hand

BJDEALER.Hand=[]; %Dealer's Hand

BJPLAYER.handle=[]; %Player's card handles

BJDEALER.handle=[]; %Dealer's card handles

BJPLAYER.Blackjack = 0; %Becomes 1 if Player has Blackjack

BJDEALER.Blackjack = 0; %Becomes 1 if Dealer has Blackjack

BJPLAYER.Bust = 0; %Becomes 1 if Player Busts

BJDEALER.Bust = 0; %Becomes 1 if Dealer Busts

BJPLAYER.NumCards = 0; %Number of cards the player holds

BJDEALER.NumCards = 0; %Number of cards the dealer holds

endhand = 0; %If not 0, hand ends

BJPLAYER.X1 = 250; %Initialize Player's Card Coordinates

BJPLAYER.Y1 = 80;

BJPLAYER.SPLITx = 450; %Player's 2nd hand coordinates for splitting

BJPLAYER.SPLITy = 80;

BJDEALER.X1 = 250; %Initializes Dealer's Card Coordinates

BJDEALER.Y1 = 400;

BJPLAYER.Winner = 0; %0=push, 1=Player wins, 2=Dealer wins

BJPLAYER.Splits = 0; %Becomes 1 if the player has used his one split

BJPLAYER.CurrentHand = 1; %Always 1, unless playing the 2nd hand after a split

BJPLAYER.handle\_get = 0;

BJDEALER.handle\_get = 0;

BJPLAYER.handle = [];

BJDEALER.handle = [];

GENERAL.cwidth = 104;

GENERAL.cheight = 151;

BJPLAYER.XIncrement(1) = 0; %The increment in x direction between

BJPLAYER.XIncrement(2) = 0;

%Wait for the player to press Deal or Quit

if FirstHand

%Set starting money to be the current bank amount

BANKROLL.Money = BANKROLL.Starting;

uiwait(gcf);

set(handles.DealButton,'visible','off');

set(handles.QuitButton,'visible','off');

set(handles.ResultTxt,'visible','off');

set(handles.InsuranceButton,'visible','off');

end

%Determine Player's wager

if (BANKROLL.Money > 0)

Bet = str2double(get(handles.BetText,'string')); % If having some sort of money

if (Bet > BANKROLL.Money)

Bet = BANKROLL.Money; % Whatever you can afford

set(handles.BetText,'string',num2str(Bet));

elseif (Bet <= 0) && (BANKROLL.Money >= BANKROLL.Unit)

Bet = BANKROLL.Unit; % If you can afford the initial bet

set(handles.BetText,'string',num2str(Bet));

elseif (Bet <= 0)

Bet = BANKROLL.Money; % Bet becomes whatever money is in the bank

end

else

endhand = 1;

set(handles.OutofMoneyTxt,'visible','on'); % if money in bank is 0

end

%This defines the action of each hand % if endhand is 0 - con

if ~endhand

%Deal 2 Cards to player

%First Card

BJDealCard('player',handles,0,0);

%Second Card

BJDealCard('player',handles,0,0);

% We can use BJDealCard to deal a card either to a player or to

% the dealer !!

% Here we are dealing to the player !!!

%Check Player's Hand for ace/blackjack and compensate for it

if (CARDS.Value(CARDS.NextCard-2) == 1) %If the value of your first card is a 11, check it as an ace

if (BJPLAYER.Total + 10) > 21

BJPLAYER.Total = BJPLAYER.Total;

elseif (BJPLAYER.Total + 10) <= 21

BJPLAYER.Total = BJPLAYER.Total + 10;

BJPLAYER.Ace = 1; % Counts existence of an ace used as 11

end

elseif (CARDS.Value(CARDS.NextCard-1) == 1) %If the value of your second card is a 11

if (BJPLAYER.Total + 10) > 21

BJPLAYER.Total = BJPLAYER.Total;

elseif (BJPLAYER.Total + 10) <= 21

BJPLAYER.Total = BJPLAYER.Total + 10;

BJPLAYER.Ace = 1; % Counts existence of an ace used as 11

end

end

if BJPLAYER.Total == 21 % Checks for blackjack in a players hand

set(handles.PlayerBJTxt,'visible','on'); % Updates the gui for the occurence of blackjack

endhand = 1; % Ends the current hanad of the game

BJPLAYER.Blackjack = 1; % Counts existence of blackjack with a player

end

set(handles.PlayerTotalTxt,'string', num2str(BJPLAYER.Total),'visible','on'); % updates the player's total after the hand

% on gui

%Deal 2 Cards to dealer, first is face down

%First Card

BJDealCard('dealer',handles,1,0);

%Second Card

BJDealCard('dealer',handles,0,0);

% Here we are dealing to the dealer !!

% As observed to be the first parameter to BJDealCard !!

% Check Dealer's Hand for ace/blackjack and compensate for it

if (CARDS.Value(CARDS.NextCard-2) == 1)

if (BJDEALER.Total + 10) > 21

BJDEALER.Total = BJDEALER.Total;

elseif (BJPLAYER.Total + 10) <= 21

BJDEALER.Total = BJDEALER.Total + 10;

BJDEALER.Ace = 1; % Counts existence of an ace used as 11

end

elseif (CARDS.Value(CARDS.NextCard-1) == 1)

if (BJDEALER.Total + 10) > 21

BJDEALER.Total = BJDEALER.Total;

elseif (BJDEALER.Total + 10) <= 21

BJDEALER.Total = BJDEALER.Total + 10;

BJDEALER.Ace = 1;

% BJDEALER.Ace\_2 = 1;

end

end

if(CARDS.Value(CARDS.NextCard-1) == 10)

BJDEALER.Ten = 1;

end

if BJDEALER.Ten == 1

set(handles.InsuranceButton,'visible','on');

set(handles.InsuranceBox,'visible','on');

set(handles.Reset,'visible','on');

end

set(handles.DealerTotalTxt,'string',[num2str( CARDS.Value(CARDS.NextCard-1) ) '+' ]);

if (CARDS.Value(CARDS.NextCard-2) == 10 && CARDS.Value(CARDS.NextCard-1) == 1) || ...

(CARDS.Value(CARDS.NextCard-1) == 10 && CARDS.Value(CARDS.NextCard-2) == 1)

BJDEALER.Total = 21;

end

%If player has blackjack, show the dealer's down card

if BJPLAYER.Blackjack == 1

set(handles.BJDEALER.Card\_1,'CData',CardGet( CARDS.Value(CARDS.NextCard-1) ,CARDS.Suit(CARDS.NextCard-1) ) );

set(handles.PlayerTotalTxt,'string',num2str(BJPLAYER.Total));

end

end

%Player's Turn

while ~endhand

%Make Selection Buttons visible

set(handles.HitButton,'visible','on');

set(handles.StandButton,'visible','on');

set(handles.DoubleButton,'visible','off');

set(handles.SplitButton,'visible','off');

if BJPLAYER.NumCards(BJPLAYER.CurrentHand) == 2

if (CARDS.Value(BJPLAYER.Hand(BJPLAYER.CurrentHand,1)) == CARDS.Value(BJPLAYER.Hand(BJPLAYER.CurrentHand,2))) && ~BJPLAYER.Splits

set(handles.SplitButton,'visible','on');

end

set(handles.DoubleButton,'visible','on');

end

%Wait for Hit, Stand, Double,Insurance or Split button to be pressed

if BJPLAYER.Total(BJPLAYER.CurrentHand) < 21

uiwait(gcf);

elseif BJPLAYER.Total == 21

BJPLAYER.hitorstand = 6; %If player already has 21, just stand

BJPLAYER.Status = 'Stand';

set(handles.StatusTxt,'string',BJPLAYER.Status);

end

if BJPLAYER.hitorstand == 12

BANKROLL.InsuranceBet = 0;

end

if BJPLAYER.hitorstand == 7

Bet(1) = 2\*Bet(1);

set(handles.BetText,'string',num2str(Bet(1)));

set(handles.StatusTxt,'string','Double');

BJDealCard('player',handles,0,0);

break

end

if BANKROLL.InsuranceBet ~= 0

set(handles.InsuranceBox,'string',num2str(BANKROLL.InsuranceBet));

else

...

end

if BJPLAYER.hitorstand == 9

BANKROLL.InsuranceBet = BANKROLL.InsuranceBet + 25;

end

if BANKROLL.InsuranceBet < 0.5\*BANKROLL.Money

set(handles.InsuranceBox,'string',num2str(BANKROLL.InsuranceBet));

else

msgbox('Not enough cash');

pause(2);

close(msgbox);

set(handles.InsuranceBox,'string','25');

end

%Manages betting with buttons

if(BJPLAYER.hitorstand == 2)

if(BANKROLL.Money > 25)

Bet = Bet + 25;

elseif (BANKROLL.Money < 25)

end

elseif(BJPLAYER.hitorstand == 3)

if(BANKROLL.Money > 50)

Bet = Bet + 50;

elseif (BANKROLL.Money < 50)

end

elseif(BJPLAYER.hitorstand == 4)

if(BANKROLL.Money > 100)

Bet = Bet + 100;

elseif (BANKROLL.Money < 100)

end

elseif (BJPLAYER.hitorstand == 11)

Bet = 0;

elseif (BJPLAYER.hitorstand == 10)

Bet = BANKROLL.Unit;

end

if BANKROLL.Money > Bet

set(handles.BetText,'string', num2str(Bet));

else

set(handles.BetText,'string','25');

msgbox('Not enough cash');

% pause(2);

close(msgbox);

end

%If the player hits or doubles down, deal another card

if (BJPLAYER.hitorstand == 5)

%Call the hit function

BJPLAYER.Status = 'Hit';

set(handles.StatusTxt,'string',BJPLAYER.Status);

BJPlayerHits(handles);

end

% if BJPLAYER.Total == 21 % Checks for blackjack in a players hand

% set(handles.PlayerBJTxt,'visible','on'); % Updates the gui for the occurence of blackjack

% endhand = 1; % Ends the current hand of the game

% BJPLAYER.Blackjack = 1; % Counts existence of blackjack with a player

% end

%If the player stands or has already busted or hit 21, player's

%turn ends unless he split and has another hand to go

if ((BJPLAYER.hitorstand == 6) || (BJPLAYER.Total(BJPLAYER.CurrentHand) >= 21))

BJPLAYER.Status = 'Stand';

set(handles.StatusTxt,'string',BJPLAYER.Status);

if (~BJPLAYER.Splits || ((BJPLAYER.Splits+1) == BJPLAYER.CurrentHand))

break

elseif (BJPLAYER.Splits+1) > BJPLAYER.CurrentHand

BJPLAYER.CurrentHand = BJPLAYER.CurrentHand + 1;

set(handles.PlayerTotalSplitTxt(1),'foregroundcolor',[.5 .5 .5]);

set(handles.PlayerTotalSplitTxt(2),'foregroundcolor',[0 0 0]);

end

end

%If the player has split, re-distribute his two cards, and play

%both new hands

if (BJPLAYER.hitorstand == 8)

BJPLAYER.Status = 'Split';

set(handles.StatusTxt,'string',BJPLAYER.Status);

BJPLAYER.Splits = BJPLAYER.Splits + 1;

%Double the bet

Bet(1) = 2\*Bet(1);

%Can't get blackjack after a split

BJPLAYER.Blackjack(1:2) = 0;

%Move cards in split position

set(BJPLAYER.handle(1),'pos',[BJPLAYER.X1 BJPLAYER.Y1 GENERAL.cwidth GENERAL.cheight],'visible','on');

set(BJPLAYER.handle(2),'pos',[BJPLAYER.SPLITx BJPLAYER.SPLITy GENERAL.cwidth GENERAL.cheight],'visible','on');

BJPLAYER.handle = [];

%If aces were split, the hand ends after two new cards are dealt

%Also, set ace field to 1 and update the totals

if (CARDS.Rank(BJPLAYER.Hand(1)) == 1) && (CARDS.Rank(BJPLAYER.Hand(2)) == 1)

endhand = 1;

BJPLAYER.Ace(1) = 1;

BJPLAYER.Ace(2) = 1;

BJPLAYER.Total(1) = 11;

BJPLAYER.Total(2) = 11;

else

BJPLAYER.Ace(1) = 0;

BJPLAYER.Ace(2) = 0;

BJPLAYER.Total(1) = CARDS.Value(CARDS.NextCard-4);

BJPLAYER.Total(2) = CARDS.Value(CARDS.NextCard-3);

end

%Update player total outputs

set(handles.PlayerTotalTxt,'visible','off');

set(handles.PlayerTotalSplitTxt(1),'string',num2str(BJPLAYER.Total(1)),'visible','on','foregroundcolor','b');

set(handles.PlayerTotalSplitTxt(2),'string',num2str(BJPLAYER.Total(2)),'visible','on');

BJPLAYER.Hand = [];

BJPLAYER.Hand(1,1) = CARDS.NextCard - 4;

BJPLAYER.Hand(2,1) = CARDS.NextCard - 3;

pause(BJBOARD.Delays)

%Fix BJPLAYER.NumCards

BJPLAYER.NumCards(1) = 1;

BJPLAYER.NumCards(2) = 1;

%Deal next 2 cards

BJDealCard('player',handles,0,0);

pause(BJBOARD.Delays);

BJPLAYER.CurrentHand = 2;

BJDealCard('player',handles,0,BJPLAYER.SPLITx);

BJPLAYER.CurrentHand = 1;

%If the player hits or doubles down, deal another card

if (BJPLAYER.hitorstand == 5)

%Call the hit function

BJPLAYER.Status = 'Hit';

set(handles.StatusTxt,'string',BJPLAYER.Status);

BJPlayerHits(handles);

end

if BJPLAYER.Splits > 0

BJPLAYER.Splits = 1;

end

%Look for aces

for n = 1:BJPLAYER.Splits+1

for m = 1:max(BJPLAYER.NumCards)

if (CARDS.Value(BJPLAYER.Hand(n,m)) == 1) && ~BJPLAYER.Ace(n)

BJPLAYER.Ace(n) = 1;

BJPLAYER.Total(n) = BJPLAYER.Total(n) + 10;

end

end

end

set(handles.PlayerTotalSplitTxt(1),'string',num2str(BJPLAYER.Total(1)) );

set(handles.PlayerTotalSplitTxt(2),'string',num2str(BJPLAYER.Total(2)) );

pause(BJBOARD.Delays);

end

end

if BJPLAYER.Splits > 0

BJPLAYER.Splits = 1;

end

for n = 1:BJPLAYER.Splits+1

if BJPLAYER.Total(n) > 21

set(handles.PlayerTotalTxt,'string','Bust');

set(handles.PlayerTotalSplitTxt(n),'string','Bust');

BJPLAYER.Bust(n) = 1;

else

BJPLAYER.Bust(n) = 0;

end

end

set(handles.HitButton,'visible','off');

set(handles.StandButton,'visible','off');

set(handles.DoubleButton,'visible','off');

set(handles.SplitButton,'visible','off');

%Dealer's Turn

%Flip dealer's down card and output his new total

set(handles.BJDEALER.Card\_1,'CData',CardGet( CARDS.Value(BJDEALER.Hand(1)),CARDS.Suit(BJDEALER.Hand(1)) ) );

%Check Dealer's Hand for ace/blackjack and compensate for it

if ( CARDS.Value( BJDEALER.Hand(1) ) == 10 && CARDS.Value( BJDEALER.Hand(2) ) == 1 ) || ...

( CARDS.Value( BJDEALER.Hand(2) ) == 10 && CARDS.Value( BJDEALER.Hand(1) ) == 1 )

BJDEALER.Total = 21;

BJDEALER.Blackjack = 1;

end

set(handles.DealerTotalTxt,'string',num2str(BJDEALER.Total));

if (BJDEALER.Blackjack)

if BJPLAYER.Blackjack

set(handles.BothBJTxt,'visible','on');

else

set(handles.DealerBJTxt,'visible','on');

end

%Show down card if dealer has Blackjack

set(handles.BJDEALER.Card\_1,'CData',CardGet( CARDS.Value(CARDS.NextCard-2),CARDS.Suit(CARDS.NextCard-2) ) );

set(handles.DealerTotalTxt,'string',num2str( num2str(BJDEALER.Total) ) );

set(handles.DealerBJTxt,'visible','on');

% BJDEALER.Blackjack = 1;

end

%Dealer hits on 16 or less and soft 17 (17 with Ace as 11) unless player

%has busted or had blackjack

while ((BJDEALER.Total < 17) || ((BJDEALER.Total == 17) && (BJDEALER.Ace))) && ~all(BJPLAYER.Bust) && ~all(BJPLAYER.Blackjack)

BJDealerHits(handles);

end

%Check for dealer bust

if BJDEALER.Total > 21

set(handles.DealerTotalTxt,'string','Bust','visible','on');

BJDEALER.Bust = 1;

% elseif BJDEALER.Total == 21

% set(handles.DealerBJTxt,'visible','on');

% BJDEALER.Blackjack = 1;

end

%Determine who wins

BJDetermineWinner;

%Bankroll changes based on results, and update Result txt

for n = 1:length(BJPLAYER.Winner)

switch BJPLAYER.Winner(n)

case 1

BJPLAYER.Wins = BJPLAYER.Wins + 1;

BANKROLL.Money = BANKROLL.Money + Bet(n) + BANKROLL.InsuranceBet;

set(handles.ResultTxt,'string','You Won!','ForegroundColor','b','visible','on');

case 2

BJPLAYER.Loses = BJPLAYER.Loses + 1;

BANKROLL.Money = BANKROLL.Money - Bet(n) - BANKROLL.InsuranceBet;

set(handles.ResultTxt,'string','You Lost!','ForegroundColor','b','visible','on');

otherwise

BJPLAYER.Pushes = BJPLAYER.Pushes + 1;

set(handles.ResultTxt,'string','It''s a Push','ForegroundColor','b','visible','on');

end

set(handles.InsuranceButton,'visible','off');

set(handles.InsuranceBox,'visible','off');

set(handles.Reset,'visible','off');

set(handles.WinsBox,'string',num2str(BJPLAYER.Wins));

set(handles.LosesBox,'string',num2str(BJPLAYER.Loses));

set(handles.PushesBox,'string',num2str(BJPLAYER.Pushes));

set(handles.BankrollTxt,'string',num2str(BANKROLL.Money),'visible','on');

break

end

%Update result txt when player split

if BJPLAYER.Splits == 1

if all(BJPLAYER.Winner == 1)

BJPLAYER.Wins = BJPLAYER.Wins + 2;

set(handles.ResultTxt,'string','You Win Both!','ForegroundColor','b','visible','on');

elseif BJPLAYER.Winner(1) == 2 && BJPLAYER.Winner(2) == 2

BJPLAYER.Loses = BJPLAYER.Loses + 2;

set(handles.ResultTxt,'string','You Lose Both!','ForegroundColor','b','visible','on');

elseif all(BJPLAYER.Winner == 0)

BJPLAYER.Pushes = BJPLAYER.Pushes + 2;

set(handles.ResultTxt,'string','You Push Both!','ForegroundColor','b','visible','on');

elseif any(BJPLAYER.Winner == 1) && any(BJPLAYER.Winner == 2)

BJPLAYER.Wins = BJPLAYER.Wins + 1;

BJPLAYER.Loses = BJPLAYER.Loses + 1;

set(handles.ResultTxt,'string','Win one, Lose one!','ForegroundColor','b','visible','on');

elseif any(BJPLAYER.Winner == 1) && any(BJPLAYER.Winner == 0)

BJPLAYER.Wins = BJPLAYER.Wins + 1;

BJPLAYER.Pushes = BJPLAYER.Pushes + 1;

set(handles.ResultTxt,'string','Win one, Push one!','ForegroundColor','b','visible','on');

elseif any(BJPLAYER.Winner == 0) && any(BJPLAYER.Winner == 2)

BJPLAYER.Pushes = BJPLAYER.Pushes + 1;

BJPLAYER.Loses = BJPLAYER.Loses + 1;

set(handles.ResultTxt,'string','Push one, Lose one!','ForegroundColor','b','visible','on');

end

set(handles.InsuranceButton,'visible','off');

set(handles.InsuranceBox,'visible','off');

set(handles.WinsBox,'string',num2str(BJPLAYER.Wins));

set(handles.LosesBox,'string',num2str(BJPLAYER.Loses));

set(handles.PushesBox,'string',num2str(BJPLAYER.Pushes));

end

if (BANKROLL.Money <= 0)

set(handles.OutofMoneyTxt,'visible','on');

QUITGAME = 1;

end

%Reduce # of hands to go by 1. If we're now at 0, set endhand to 1.

%If we were at 0 before, just keep going since there's no set # of

%hands

if BANKROLL.HandsLeft

BANKROLL.HandsLeft = BANKROLL.HandsLeft - 1;

set(handles.HandsLeftBox,'string',num2str(BANKROLL.HandsLeft));

if ~BANKROLL.HandsLeft

QUITGAME = 1;

msgbox('Game has ended, Hands are finished');

end

end

BJPLAYER.HandsPlayed = BJPLAYER.HandsPlayed + 1;

set(handles.HandsPlayedBox,'string',num2str(BJPLAYER.HandsPlayed));

BJPLAYER.Percent = (BJPLAYER.Wins/BJPLAYER.HandsPlayed)\*100;

set(handles.WinningPercentBox,'string',[num2str(round(BJPLAYER.Percent)) '%']);

%Get ready for next hand

if ~QUITGAME

%Check Card Penetration

%In single deck, shuffles after 66% of cards are used

%In multi deck, shuffles after 75% of cards are used

if (~CARDS.MultiDeck) && (CARDS.CardsLeft < (1/3\*CARDS.DecksUsed\*52))

CARDS.Shuffle = 1;

set(handles.ShuffleTxt,'visible','on');

pause(2);

set(handles.ShuffleTxt,'visible','off');

end

if (CARDS.MultiDeck) && (CARDS.CardsLeft < (1/4\*CARDS.DecksUsed\*52))

CARDS.Shuffle = 1;

set(handles.ShuffleTxt,'visible','on');

pause(2);

set(handles.ShuffleTxt,'visible','off');

end

%Ask player if he wants to deal again

if QUITGAME == 0

set(handles.DealButton,'visible','on');

set(handles.QuitButton,'visible','on');

uiwait(gcf);

end

if QUITGAME

break

end

else

pause(BJBOARD.Delays\*2);

end

%Get ready for next hand

FirstHand = 0;

BANKROLL.Bet = str2double(get(handles.BetText,'string'));

% Clear the extra stuff from the board to prepare for the next hand

% Remove the player's cards

set(handles.BJPLAYER.Card\_1,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_2,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_3,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_4,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_5,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_6,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_7,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_8,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_9,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_10,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_11,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJPLAYER.Card\_12,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJDEALER.Card\_1,'style','pushbutton','visible','off' ,'CData', '');

set(handles.BJDEALER.Card\_2,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJDEALER.Card\_3,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJDEALER.Card\_4,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJDEALER.Card\_5,'style','pushbutton','visible','off', 'CData', '');

set(handles.BJDEALER.Card\_6,'style','pushbutton','visible','off', 'CData', '');

BJPLAYER.Total = 0;

BJPLAYER.Ace = 0;

BJDEALER.Total = 0;

BJDEALER.Ace = 0;

BJPLAYER.Hand = [];

BJDEALER.Hand = [];

BJPLAYER.handle = [];

BJPLAYER.handle = [];

endhand = 0;

BJPLAYER.Splits = 0;

BANKROLL.InsuranceBet = 0;

% Clear the hand results

set(handles.ResultTxt,'string','','visible','off');

set(handles.QuitButton,'visible','off');

set(handles.StatusTxt,'string','','visible','on');

set(handles.PlayerTotalTxt,'string','','visible','on');

set(handles.PlayerTotalSplitTxt(1),'visible','off');

set(handles.PlayerTotalSplitTxt(2),'visible','off');

set(handles.DealerTotalTxt,'string','');

set(handles.PlayerBJTxt,'visible','off');

set(handles.DealerBJTxt,'visible','off');

set(handles.BothBJTxt,'visible','off');

set(handles.OutofMoneyTxt,'visible','off');

set(handles.NoDDMoneyTxt,'visible','off');

set(handles.ShuffleTxt,'visible','off');

set(handles.InsuranceButton,'visible','off');

set(handles.InsuranceBox,'string','','visible','off');

set(handles.Reset,'visible','off');

BANKROLL.History(end+1) = BANKROLL.Money;

end

close all;

delete(gcf);

catch

return

end

BJActionSelect.m

%This function is called when the Hit, Stand and Double boxes are pressed

function BJActionSelect(Action)

global BJPLAYER

BJPLAYER.hitorstand = Action; %Save which button was pressed

uiresume(gcf); %Un-pause

end

BJBoardCreation

function [handles] = BJBoardCreation(Bet)

%output from BJBoardCreation --- handles !

% Creates the BlackJack Playing Board

global BANKROLL;

global BJBOARD;

global BJPLAYER;

global BJDEALER;

% Bet = 50;

Creator = 'I-J-S-C';

Date = '1/12/17';

close all;

% What does BJBoardCreation outputs ??

% It outputs handles !!

% Therefore anything with handles.Keisha ... we have access to ! .. Thus ..

% We have access to the gui !!!

handles.Main = gcf; % gcf - Get current figure

% BJBOARD.pos=get(gcf,'pos');

BJBOARD.scr=get(0,'screensize');

set(handles.Main,'name','BlackJack','menubar','none','numbertitle','off','pos',[BJBOARD.scr(3)\*0.25 60 810 640] );

% create an axes that spans the whole gui

handles.Main2 = axes('unit', 'normalized', 'position',[0 0 1 1]);

% import the background image and show it on the axes

bg = imread('Back.jpg');

imagesc(bg);

% prevent plotting over the background and turn the axis off

set(handles.Main2,'handlevisibility','off','visible','off')

% making sure the background is behind all the other uicontrols

uistack(handles.Main2, 'bottom');

handles.Author = uicontrol('style','text','string',strcat('Created By:',' ',Creator),'pos',[680 95 115 15],'fontsize',8, 'backgroundcolor', [0 1 0]);

handles.Date = uicontrol('style','text','string',Date,'pos',[715 80 50 15],'fontsize',8,'backgroundcolor', [0 1 0]);

handles.DealerText = uicontrol('style','text','string','Dealer','pos',[350 600 90 25],'fontsize',15,'horiz','c', 'backgroundcolor', [0 1 0]);

handles.YouText = uicontrol('style','text','string','You','pos',[352 305 90 25],'fontsize',15,'horiz','c', 'backgroundcolor', [0 1 0]);

handles.BankTxt = uicontrol('style','text','string','Bank','pos',[ 700 600 90 25],'fontsize',15,'horiz','l','horiz','c', 'backgroundcolor', [0 1 0]);

handles.BankrollTxt = uicontrol('style','text','string',['$' num2str(BANKROLL.Starting)],'pos',[718 575 60 20],'fontsize',15,'horiz','c', 'backgroundcolor', [0 1 0]);

% handles.HelpButton = uicontrol('style','pushbutton','string','Help','Position', [700 510 90 25],'fontsize',15,'visible','on','Callback','BJActionSelect(1);', 'backgroundcolor', 'w');

chip25 = imread('chip25.png');

chip50 = imread('chip50.png');

chip100 = imread('chip100.png');

handles.BetText = uicontrol('style','text','string',Bet,'pos',[55 150 90 25],'fontsize',15,'horiz','c','backgroundcolor',[0 1 0]);

handles.BetButton25 = uicontrol('style','pushbutton','Position',[30 480 150 150],'CData',chip25,'visible','on','Callback','BJActionSelect(2);');

handles.BetButton50 = uicontrol('style','pushbutton','Position',[30 330 152 148],'CData',chip50,'visible','on','Callback','BJActionSelect(3);');

handles.BetButton100 = uicontrol('style','pushbutton','Position',[30 180 147 150], 'CData',chip100,'visible','on','Callback','BJActionSelect(4);');

handles.group = uibuttongroup('visible','on','Position',[0 0 1 .1],'SelectionChangedFcn',@bselection);

handles.HitButton = uicontrol('style','pushbutton','string','Hit','Position', [170 7 180 50],'fontsize',15,'visible','on','Callback','BJActionSelect(5);','backgroundcolor','w');

handles.DealButton = uicontrol('style','pushbutton','string','Deal','Position',[360 7 180 50],'fontsize',15,'visible','on','Callback','uiresume(gcf)','backgroundcolor','w'); %Deal Pushbutton

handles.StandButton = uicontrol('style','pushbutton','string','Stand','Position', [10 7 150 50],'fontsize',15,'visible','on','Callback','BJActionSelect(6);','backgroundcolor','w');

handles.DoubleButton = uicontrol('style','pushbutton','string','Double','Position', [550 7 150 50],'fontsize',15,'visible','on','Callback','BJActionSelect(7);','backgroundcolor','w');

handles.SplitButton = uicontrol('style','pushbutton','string','Split','Position', [710 7 90 50],'fontsize',15,'visible','on','Callback','BJActionSelect(8);','backgroundcolor','w');

handles.QuitButton = uicontrol('style','pushbutton','string','Quit','Position',[700 430 90 25],'fontsize',15,'horiz','c','visible','on','backgroundcolor','w','UserData',1,'CallBack','QuitGame'); %Quit Pushbutton

handles.HandsLeftTxt = uicontrol('style','text','string','Hands Left:','pos',[650 380 90 25],'backgroundcolor','w','horiz','c','fontsize',15);

handles.HandsLeftBox = uicontrol('style','text','string',num2str(BANKROLL.HandsLeft),'pos',[760 380 40 25],'Callback','HandsLeft(handles)','fontsize',15);

handles.Statistics = uicontrol('style','text','string','Statistics','pos',[700 310 90 25],'fontsize',15);

handles.HandsPlayed = uicontrol('style','text','string','Hands Played','pos',[610 270 130 25],'fontsize',11);

handles.HandsPlayedBox = uicontrol('style','text','string',num2str(BJPLAYER.HandsPlayed),'pos',[760 270 40 25],'horiz','c','fontsize',11);

handles.Wins = uicontrol('style','text','string','Wins','pos',[610 240 130 25],'fontsize',11);

handles.WinsBox = uicontrol('style','text','string',num2str(BJPLAYER.Wins),'pos',[760 240 40 25],'horiz','c','fontsize',11);

handles.Loses = uicontrol('style','text','string','Loses','pos',[610 210 130 25],'fontsize',11);

handles.LosesBox = uicontrol('style','text','string',num2str(BJPLAYER.Loses),'pos',[760 210 40 25],'horiz','c','fontsize',11);

handles.Pushes = uicontrol('style','text','string','Pushes','pos',[610 180 130 25],'fontsize',11);

handles.PushesBox = uicontrol('style','text','string',num2str(BJPLAYER.Pushes),'pos',[760 180 40 25],'horiz','c','fontsize',11);

handles.WinningPercent = uicontrol('style','text','string','Winning %','pos',[610 150 130 25],'fontsize',11);

handles.WinningPercentBox = uicontrol('style','text','string',[ num2str(BJPLAYER.Percent) '%' ],'pos',[760 150 40 25],'horiz','c','fontsize',11);

handles.InsuranceButton = uicontrol('style','pushbutton','string','Insurance','foregroundcolor','b','backgroundcolor','w','Position',[452 295 92 50],'fontsize',15,'visible','off','Callback','BJActionSelect(9);');

handles.InsuranceBox = uicontrol('style','text','string',num2str(BANKROLL.InsuranceBet),'foregroundcolor','b','backgroundcolor','w','Position',[452 265 90 25],'fontsize',15,'horiz','c','visible','off');

handles.Reset = uicontrol('style','pushbutton','string','Reset','foregroundcolor','b','backgroundcolor','w','Position',[452 345 92 25],'fontsize',15,'horiz','c','visible','off','Callback','BJActionSelect(12);');

handles.StatusTxt = uicontrol('style','text','string',BJPLAYER.Status,'pos',[700 470 90 25],'fontsize',15,'horiz','c','backgroundcolor',[0 1 0],'visible','on');

handles.PlayerName = uicontrol('style','text','string',BJPLAYER.PN,'pos',[700 540 90 25],'fontsize',15,'horiz','c','backgroundcolor',[ 0 1 0],'visible','on');

handles.PlayerTotalTxt = uicontrol('style','text','string',num2str(BJPLAYER.Total),'pos',[352 270 90 25],'fontsize',15,'horiz','c','backgroundcolor',[0 1 0]); %Player's total

handles.DealerTotalTxt = uicontrol('style','text','string',num2str(BJDEALER.Total),'pos',[350 565 90 25],'fontsize',15,'horiz','c','backgroundcolor',[0 1 0]);

handles.ReplayButton = uicontrol('style','pushbutton','string','Replay Bet','Position',[21 80 75 60],'fontsize',10,'horiz','r','backgroundcolor','w','Callback','BJActionSelect(10);'); %Replay Bet Pushbutton

handles.ResetButton = uicontrol('style','pushbutton','string','Reset Bet','pos',[106 80 75 60],'fontsize',10,'horiz','r','backgroundcolor','w','Callback','BJActionSelect(11);'); %Reset Bet Pushbutton

handles.PlayerBJTxt = uicontrol('style','text','string','You have BlackJack!','pos',[420 215 190 45],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off');

handles.DealerBJTxt = uicontrol('style','text','string','Dealer has BlackJack!','pos',[440 510 205 45],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off');

handles.BothBJTxt = uicontrol('style','text','string','Both players have BJ!','pos',[300 430 205 45],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off');

handles.OutofMoneyTxt = uicontrol('style','text','string','You''re out of money!','pos',[300 335 205 45],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off');

handles.NoDDMoneyTxt = uicontrol('style','text','string','Can''t Double Down!','pos',[300 335 205 45],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off');

handles.ShuffleTxt = uicontrol('style','text','string','Shuffling...','pos',[300 335 205 45],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off');

handles.ResultTxt = uicontrol('style','text','string','YOU WIN FORTIS!','pos',[300 335 205 45],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off');

handles.PlayerTotalSplitTxt(1) = uicontrol('style','text','string','Your Total :','pos',[250 240 140 25],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off'); %Player's total for hand 1 when splitting

handles.PlayerTotalSplitTxt(2) = uicontrol('style','text','string','Your Total :','pos',[430 240 140 25],'fontsize',15,'horiz','c','foregroundcolor','b','backgroundcolor',[0 1 0],'visible','off'); %Player's total for hand 2 when splitting

handles.BJPLAYER.Card\_1 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_2 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_3 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_4 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_5 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_6 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_7 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_8 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_9 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_10 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_11 = uicontrol('style','pushbutton','visible','off');

handles.BJPLAYER.Card\_12 = uicontrol('style','pushbutton','visible','off');

handles.BJDEALER.Card\_1 = uicontrol('style','pushbutton','visible','off');

handles.BJDEALER.Card\_2 = uicontrol('style','pushbutton','visible','off');

handles.BJDEALER.Card\_3 = uicontrol('style','pushbutton','visible','off');

handles.BJDEALER.Card\_4 = uicontrol('style','pushbutton','visible','off');

handles.BJDEALER.Card\_5 = uicontrol('style','pushbutton','visible','off');

handles.BJDEALER.Card\_6 = uicontrol('style','pushbutton','visible','off');

handles.BJDEALER.Card\_7 = uicontrol('style','pushbutton','visible','off');

if ~BANKROLL.HandsLeft %If there are no hands left, update on gui

set(handles.HandsLeftBox,'string','');

end

BJDealCard

function [handlegot] = BJDealCard(recipient,handles,facedown,Split)

% Take into four parameters (inputs) !

% Recipient : - Dealer / Player

% Handles : - Handles in Blackjack.m

% facedown - For the dealer !!

% if facedown = 1 ... get facedown card ....

% if facedown = 0 ... no facedown card ...

% the fourth capability of BJDealCard is to split !

global BJPLAYER % KEISHA INTERNATIONAL !

global BJDEALER % KEISHA INTERNATIONAL !

global CARDS % KEISHA INTERNATIONAL !

global GENERAL % KEISHA INTERNATIONAL !

% handles = BJBoardCreation;

% BJPLAYER.handle\_get = 0;

% recipient = 'player';

handlegot = 0; %initializes variables to zero

add = 0; % initializes variables to zero

if ~exist('facedown','var') || ~facedown

%if facedown variable does not exist/ real ... or if facedown == 0

Rank = CARDS.Rank(CARDS.NextCard);

else

Rank = 0;

end

if BJPLAYER.Splits == 1

add = 30;

end

switch recipient

case 'player'

if BJPLAYER.handle\_get == 0

% BJPLAYER.XIncrement = [ row matrix with two elements ]

% [ 30 30 ]

BJPLAYER.XIncrement(1) = 0;

BJPLAYER.XIncrement(2) = 0;

handlegot = handles.BJPLAYER.Card\_1;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.handle\_get == 1

set(handles.BJPLAYER.Card\_2,'CData','','visible','off');

set(handles.BJPLAYER.Card\_3,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_2;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.handle\_get == 2

set(handles.BJPLAYER.Card\_3,'CData','','visible','off');

set(handles.BJPLAYER.Card\_4,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_3;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

if BJPLAYER.Splits == 0

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

else

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(30+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

end

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.handle\_get == 3

set(handles.BJPLAYER.Card\_4,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_4;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

if BJPLAYER.Splits == 0

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

else

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+Split+30), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

end

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

if Split == 0

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

else

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 60;

end

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.handle\_get == 4

if BJPLAYER.CurrentHand == 1

% disp(BJPLAYER.XIncrement(1));

set(handles.BJPLAYER.Card\_6,'CData','','visible','off');

set(handles.BJPLAYER.Card\_5,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_5;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)-add+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.CurrentHand == 2

set(handles.BJPLAYER.Card\_6,'CData','','visible','off');

set(handles.BJPLAYER.Card\_5,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_5;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+BJPLAYER.SPLITx), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

end

elseif BJPLAYER.handle\_get == 5

if BJPLAYER.CurrentHand == 1

set(handles.BJPLAYER.Card\_6,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_6;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)-add+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.CurrentHand == 2

set(handles.BJPLAYER.Card\_6,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_6;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+BJPLAYER.SPLITx), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

end

elseif BJPLAYER.handle\_get == 6

if BJPLAYER.CurrentHand == 1

set(handles.BJPLAYER.Card\_7,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_7;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)-add+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.CurrentHand == 2

set(handles.BJPLAYER.Card\_7,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_7;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+BJPLAYER.SPLITx), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

end

elseif BJPLAYER.handle\_get == 7

if BJPLAYER.CurrentHand == 1

set(handles.BJPLAYER.Card\_8,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_8;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)-add+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.CurrentHand == 2

set(handles.BJPLAYER.Card\_8,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_8;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+BJPLAYER.SPLITx), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

end

elseif BJPLAYER.handle\_get == 8

if BJPLAYER.CurrentHand == 1

set(handles.BJPLAYER.Card\_9,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_9;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)-add+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.CurrentHand == 2

set(handles.BJPLAYER.Card\_9,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_9;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+BJPLAYER.SPLITx), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

end

elseif BJPLAYER.handle\_get == 9

if BJPLAYER.CurrentHand == 1

set(handles.BJPLAYER.Card\_10,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_10;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)-add+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.CurrentHand == 2

set(handles.BJPLAYER.Card\_10,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_10;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+BJPLAYER.SPLITx), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

end

elseif BJPLAYER.handle\_get == 10

if BJPLAYER.CurrentHand == 1

set(handles.BJPLAYER.Card\_11,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_11;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)-add+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.CurrentHand == 2

set(handles.BJPLAYER.Card\_11,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_11;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+BJPLAYER.SPLITx), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

end

elseif BJPLAYER.handle\_get == 11

if BJPLAYER.CurrentHand == 1

set(handles.BJPLAYER.Card\_12,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_12;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(1)-add+BJPLAYER.X1), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(1) = BJPLAYER.XIncrement(1) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

elseif BJPLAYER.CurrentHand == 2

set(handles.BJPLAYER.Card\_12,'CData','','visible','off');

handlegot = handles.BJPLAYER.Card\_12;

BJPLAYER.handle(BJPLAYER.handle\_get + 1) = handlegot;

BJPLAYER.NumCards(BJPLAYER.CurrentHand) = BJPLAYER.NumCards(BJPLAYER.CurrentHand) + 1;

BJPLAYER.Hand(BJPLAYER.CurrentHand,BJPLAYER.NumCards(BJPLAYER.CurrentHand)) = CARDS.NextCard;

Card\_Spec = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

set(BJPLAYER.handle(BJPLAYER.handle\_get + 1),'pos',[(BJPLAYER.XIncrement(2)+BJPLAYER.SPLITx), BJPLAYER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJPLAYER.XIncrement(2) = BJPLAYER.XIncrement(2) + 30;

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + CARDS.Value(CARDS.NextCard);

BJPLAYER.handle\_get = BJPLAYER.handle\_get + 1;

end

end

case 'dealer'

if BJDEALER.handle\_get == 0

BJDEALER.XIncrement = 0;

handlegot = handles.BJDEALER.Card\_1;

BJDEALER.handle(BJDEALER.handle\_get + 1) = handlegot;

BJDEALER.NumCards = BJDEALER.NumCards + 1;

BJDEALER.Hand(BJDEALER.NumCards) = CARDS.NextCard;

Card\_Spec\_1 = CardGet(0,CARDS.Suit(CARDS.NextCard));

set(BJDEALER.handle(BJDEALER.handle\_get + 1),'pos',[(BJDEALER.XIncrement+BJDEALER.X1), BJDEALER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec\_1,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJDEALER.XIncrement = BJDEALER.XIncrement + 30;

BJDEALER.Total = BJDEALER.Total + CARDS.Value(CARDS.NextCard);

BJDEALER.handle\_get = BJDEALER.handle\_get + 1;

elseif BJDEALER.handle\_get == 1

set(handles.BJDEALER.Card\_2,'CData','','visible','off');

set(handles.BJDEALER.Card\_3,'CData','','visible','off');

handlegot = handles.BJDEALER.Card\_2;

BJDEALER.handle(BJDEALER.handle\_get + 1) = handlegot;

BJDEALER.NumCards = BJDEALER.NumCards + 1;

BJDEALER.Hand(BJDEALER.NumCards) = CARDS.NextCard;

Card\_Spec\_1 = CardGet(Rank,CARDS.Suit(CARDS.NextCard));

set(BJDEALER.handle(BJDEALER.handle\_get + 1),'pos',[(BJDEALER.XIncrement+BJDEALER.X1), BJDEALER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec\_1,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJDEALER.XIncrement = BJDEALER.XIncrement + 30;

BJDEALER.Total = BJDEALER.Total + CARDS.Value(CARDS.NextCard);

BJDEALER.handle\_get = BJDEALER.handle\_get + 1;

elseif BJDEALER.handle\_get == 2

set(handles.BJDEALER.Card\_3,'CData','','visible','off');

set(handles.BJDEALER.Card\_4,'CData','','visible','off');

handlegot = handles.BJDEALER.Card\_3;

BJDEALER.handle(BJDEALER.handle\_get + 1) = handlegot;

BJDEALER.NumCards = BJDEALER.NumCards + 1;

BJDEALER.Hand(BJDEALER.NumCards) = CARDS.NextCard;

Card\_Spec\_1 = CardGet(Rank,CARDS.Suit(CARDS.NextCard));

set(BJDEALER.handle(BJDEALER.handle\_get + 1),'pos',[(BJDEALER.XIncrement+BJDEALER.X1), BJDEALER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec\_1,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJDEALER.XIncrement = BJDEALER.XIncrement + 30;

BJDEALER.Total = BJDEALER.Total + CARDS.Value(CARDS.NextCard);

BJDEALER.handle\_get = BJDEALER.handle\_get + 1;

elseif BJDEALER.handle\_get == 3

set(handles.BJDEALER.Card\_4,'CData','','visible','off');

handlegot = handles.BJDEALER.Card\_4;

BJDEALER.handle(BJDEALER.handle\_get + 1) = handlegot;

BJDEALER.NumCards = BJDEALER.NumCards + 1;

BJDEALER.Hand(BJDEALER.NumCards) = CARDS.NextCard;

Card\_Spec\_1 = CardGet(Rank,CARDS.Suit(CARDS.NextCard));

set(BJDEALER.handle(BJDEALER.handle\_get + 1),'pos',[(BJDEALER.XIncrement+BJDEALER.X1), BJDEALER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec\_1,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJDEALER.XIncrement = BJDEALER.XIncrement + 30;

BJDEALER.Total = BJDEALER.Total + CARDS.Value(CARDS.NextCard);

BJDEALER.handle\_get = BJDEALER.handle\_get + 1;

elseif BJDEALER.handle\_get == 4

set(handles.BJDEALER.Card\_5,'CData','','visible','off');

handlegot = handles.BJDEALER.Card\_5;

BJDEALER.handle(BJDEALER.handle\_get + 1) = handlegot;

BJDEALER.NumCards = BJDEALER.NumCards + 1;

BJDEALER.Hand(BJDEALER.NumCards) = CARDS.NextCard;

Card\_Spec\_1 = CardGet(Rank,CARDS.Suit(CARDS.NextCard));

set(BJDEALER.handle(BJDEALER.handle\_get + 1),'pos',[(BJDEALER.XIncrement+BJDEALER.X1), BJDEALER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec\_1,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJDEALER.XIncrement = BJDEALER.XIncrement + 30;

BJDEALER.Total = BJDEALER.Total + CARDS.Value(CARDS.NextCard);

BJDEALER.handle\_get = BJDEALER.handle\_get + 1;

elseif BJDEALER.handle\_get == 5

set(handles.BJDEALER.Card\_6,'CData','','visible','off');

handlegot = handles.BJDEALER.Card\_6;

BJDEALER.handle(BJDEALER.handle\_get + 1) = handlegot;

BJDEALER.NumCards = BJDEALER.NumCards + 1;

BJDEALER.Hand(BJDEALER.NumCards) = CARDS.NextCard;

Card\_Spec\_1 = CardGet(Rank,CARDS.Suit(CARDS.NextCard));

set(BJDEALER.handle(BJDEALER.handle\_get + 1),'pos',[(BJDEALER.XIncrement+BJDEALER.X1), BJDEALER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec\_1,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJDEALER.XIncrement = BJDEALER.XIncrement + 30;

BJDEALER.Total = BJDEALER.Total + CARDS.Value(CARDS.NextCard);

BJDEALER.handle\_get = BJDEALER.handle\_get + 1;

elseif BJDEALER.handle\_get == 6

set(handles.BJDEALER.Card\_7,'CData','','visible','off');

handlegot = handles.BJDEALER.Card\_7;

BJDEALER.handle(BJDEALER.handle\_get + 1) = handlegot;

BJDEALER.NumCards = BJDEALER.NumCards + 1;

BJDEALER.Hand(BJDEALER.NumCards) = CARDS.NextCard;

Card\_Spec\_1 = CardGet(Rank,CARDS.Suit(CARDS.NextCard));

set(BJDEALER.handle(BJDEALER.handle\_get + 1),'pos',[(BJDEALER.XIncrement+BJDEALER.X1), BJDEALER.Y1, GENERAL.cwidth, GENERAL.cheight],'CData', Card\_Spec\_1,'visible','on');

CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) = CARDS.CardCount(CARDS.Rank(CARDS.NextCard)) - 1;

BJDEALER.XIncrement = BJDEALER.XIncrement + 30;

BJDEALER.Total = BJDEALER.Total + CARDS.Value(CARDS.NextCard);

BJDEALER.handle\_get = BJDEALER.handle\_get + 1;

end

end

CARDS.NextCard = CARDS.NextCard + 1;

CARDS.CardsLeft = CARDS.CardsLeft - 1;

end

BJDealerHits

function BJDealerHits(handles)

global CARDS

global BJDEALER

global BJBOARD

%Deal Card

set(handles.BJDEALER.Card\_3,'CData','');

set(handles.BJDEALER.Card\_4,'CData','');

BJDealCard('dealer',handles,0,0);

%Check and compensate for aces

if (CARDS.Rank(CARDS.NextCard) == 1) && (BJDEALER.Total < 12)

BJDEALER.Total = BJDEALER.Total + 10;

BJDEALER.Ace = 1;

end

if (BJDEALER.Ace == 1) && (BJDEALER.Total > 21)

BJDEALER.Total = BJDEALER.Total - 10;

BJDEALER.Ace = 0;

end

%Update dealer's total

set(handles.DealerTotalTxt,'string',num2str(BJDEALER.Total) );

%Update Hi-Lo count

% BJUpdateHiLoCount(handles,CARDS.NextCard-1);

pause(BJBOARD.Delays);

BJDetermineWinner

function BJDetermineWinner

%Determines the winner of the blackjack hand

global BJPLAYER

global BJDEALER

global BANKROLL

% global QUITGAME

for n=1:length(BJPLAYER.Total)

if BJPLAYER.Blackjack(n) && ~BJDEALER.Blackjack %Player has Blackjack

BJPLAYER.Winner(n) = 1;

if ~BJPLAYER.Splits %Only pay out 1.5x if it wasn't a split

BANKROLL.Bet(n) = BANKROLL.Bet(n)\*1.5;

end

elseif ~BJPLAYER.Blackjack(n) && BJDEALER.Blackjack %Dealer has Blackjack

BJPLAYER.Winner(n) = 2;

elseif BJPLAYER.Blackjack(n) && BJDEALER.Blackjack %Both have Blackjack

BJPLAYER.Winner(n) = 0;

elseif BJPLAYER.Bust(n) %Player Busts

BJPLAYER.Winner(n) = 2;

elseif BJDEALER.Bust %Dealer Busts

BJPLAYER.Winner(n) = 1;

elseif (BJPLAYER.Total(n) > BJDEALER.Total) %Player beats Dealer

BJPLAYER.Winner(n) = 1;

elseif (BJDEALER.Total > BJPLAYER.Total(n)) %Dealer beats Player

BJPLAYER.Winner(n) = 2;

elseif (BJPLAYER.Total(n) == BJDEALER.Total) %Push

BJPLAYER.Winner(n) = 0;

end

% QUITGAME = 0;

end

BJPlayerHits

function BJPlayerHits(handles)

%Performs the required actions when a player decides to hit, on the hand

%specified by Player.CurrentHand

global BJPLAYER

% global BJDEALER

global CARDS

global BANKROLL

%If the player doubled down but doesn't have the money, call it

%a hit

if (BJPLAYER.hitorstand == 7) && (BANKROLL.Money < 2\*BANKROLL.Bet(BJPLAYER.CurrentHand))

set(handles.NoDDMoneyTxt,'visible','on');

BJPLAYER.hitorstand = 5;

end

%Deal another card

set(handles.BJPLAYER.Card\_3,'CData','');

set(handles.BJPLAYER.Card\_4,'CData','');

BJDealCard('player',handles,0,0);

%Check and compensate for Aces

if (CARDS.Rank(CARDS.NextCard-1) == 1) && (BJPLAYER.Total(BJPLAYER.CurrentHand) < 12)

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) + 10;

BJPLAYER.Ace(BJPLAYER.CurrentHand) = 1;

end

if (BJPLAYER.Ace(BJPLAYER.CurrentHand) == 1) && (BJPLAYER.Total(BJPLAYER.CurrentHand) > 21)

BJPLAYER.Total(BJPLAYER.CurrentHand) = BJPLAYER.Total(BJPLAYER.CurrentHand) - 10;

BJPLAYER.Ace(BJPLAYER.CurrentHand) = 0;

end

%Update the appropriate string with the new total

if BJPLAYER.Splits

set(handles.PlayerTotalSplitTxt(BJPLAYER.CurrentHand),'string',num2str(BJPLAYER.Total(BJPLAYER.CurrentHand)));

else

set(handles.PlayerTotalTxt,'string',num2str(BJPLAYER.Total));

end

%Double the bet if it was a double down

if BJPLAYER.hitorstand == 7

BANKROLL.Bet(BJPLAYER.CurrentHand) = BANKROLL.Bet(BJPLAYER.CurrentHand)\*2;

end

BJStartValues

function Init = BJStartValues

%Copyright - Ishmael Leon, Scott Copely, Jordane McCalla, Christopher Reid

%Creates a GUI that allows the player to determine the initial values to be

%used in the game

[s,Fs]=audioread('jazz.wav');

% File Location - 'jazz.wav' ... Current folder

% Assigned to an array/matrix having only two elements called s & Fs

% Having that the output from audioread is a row matrix with only 2

% elements

player=audioplayer(s,Fs);

global BJPLAYER

% clear BJPLAYER.hitorstand\_intro;

%Initialize Default values

Init.NumHands = 0;

Init.Bankroll = 5000;

Init.BetUnit = 5;

Init.NumDecks = 1;

Init.GameSpeed = 50; %As a percentage

Init.PN = 'Fortis <3';

BoxSize = [530 180 260 340]; % dimensions of the box - [distance from left , distance from bottom. width, height]

%Initialize the box

color = [.4 .2 .4]; % Colour mixtures - red green blue - purple - "RGB TRIPLET"

% red = [ 1 0 0];

% blue = [0 0 1];

% green = [ 0 1 0];

% purple = [ 20 0 20 ];

handles.IntroBox = gcf; % Brings up a blank fig

%gcf - current figure handle .... graphical current figure

set(handles.IntroBox,'name','BlackJack Setup','menubar','none','numbertitle','off','color',color,'pos',BoxSize);

%Display the appropriate options and wait for the player to press Begin

handles.InitNumHandsTxt = uicontrol('style','text','string','Hands','fontsize',12,'pos',[50 BoxSize(4)-65 98 25],'backgroundcolor',color,'foregroundcolor','w','horiz','r');

handles.InitNumHandsBox = uicontrol('style','edit','string',num2str(Init.NumHands),'pos',[100 BoxSize(4)-80 50 18], 'horiz', 'c');

handles.Player = uicontrol('style','text','string','Player''s Name:','pos',[40 BoxSize(4)-40 120 25],'backgroundcolor',color,'foregroundcolor','w','horiz','r','fontsize',12);

handles.PlayerBox = uicontrol('style','edit','string',Init.PN,'pos',[165 BoxSize(4)-35 80 18],'horiz','c','fontsize',12);

handles.InitBankrollOpTxt = uicontrol('style','text','string','Bankroll Options:','fontsize',12,'pos',[69 BoxSize(4)-110 200 20],'backgroundcolor',color,'foregroundcolor','w','horiz','l');

handles.InitBankrollTxt = uicontrol('style','text','string','Bankroll:','fontsize',8,'pos',[25 BoxSize(4)-130 98 15],'backgroundcolor',color,'foregroundcolor','w','horiz','r');

handles.InitBankrollBox = uicontrol('style','edit','string',num2str(Init.Bankroll),'pos',[130 BoxSize(4)-130 50 18], 'horiz', 'c');

handles.InitBetUnitTxt = uicontrol('style','text','string','Bet Unit:','fontsize',8,'pos',[25 BoxSize(4)-150 98 15],'backgroundcolor',color,'foregroundcolor','w','horiz','r');

handles.InitBetUnitBox = uicontrol('style','edit','string',num2str(Init.BetUnit),'pos',[130 BoxSize(4)-150 50 18],'horiz','c');

handles.InitCardOpTxt = uicontrol('style','text','string','Card Options:','fontsize',12,'pos',[69 BoxSize(4)-180 200 20],'background',color,'foregroundcolor','w','horiz','l');

handles.InitNumDecksTxt = uicontrol('style','text','string','Number of Decks:','fontsize',8','pos',[63 BoxSize(4)-200 98 15],'backgroundcolor',color,'foregroundcolor','w','horiz','r');

handles.InitNumDecksBox = uicontrol('style','edit','string',num2str(Init.NumDecks),'pos',[161 BoxSize(4)-200 50 18],'horiz','c');

play(player);

handles.InitGamePlayOpTxt = uicontrol('style','text','string','Gameplay Options:','fontsize',12,'pos',[70 BoxSize(4)-230 200 20],'backgroundcolor',color,'horiz','l','foregroundcolor','w');

handles.InitGameSpeedTxt = uicontrol('style','text','string','Game Speed:','fontsize',8','pos',[43 BoxSize(4)-250 98 15],'backgroundcolor',color,'horiz','r','foregroundcolor','w');

handles.InitGameSpeedBox = uicontrol('style','edit','string',num2str(Init.GameSpeed),'pos',[142 BoxSize(4)-250 30 18],'horiz','c');

handles.InitGameSpeedTxt2 = uicontrol('style','text','string','%','fontsize',8','pos',[171.9 BoxSize(4)-250 20 15],'backgroundcolor',color,'foregroundcolor','w');

% handles.Help = uicontrol('style','pushbutton','string','Help','fontsize',12,'pos',[50 45 160 20],'callback','BJActionSelectIntro(1);');

handles.InitBegin = uicontrol('style','pushbutton','string','Start','fontsize',12,'pos',[50 30 160 20],'callback','uiresume;');

uiwait;

%Temporarily waits/pauses the game for an input from the user

stop(player); % Stops music on exiting

handles.IntroBox2 = figure; % Brings up a blank fig for the second title screen with only blackjack image

set(handles.IntroBox2,'name','','menubar','none','numbertitle','off','color',color);

imshow('blackjack.jpg');

robot = java.awt.Robot;

robot.keyPress(java.awt.event.KeyEvent.VK\_ALT); %// send ALT

robot.keyPress(java.awt.event.KeyEvent.VK\_SPACE); %// send SPACE

robot.keyRelease(java.awt.event.KeyEvent.VK\_SPACE); %// release SPACE

robot.keyRelease(java.awt.event.KeyEvent.VK\_ALT); %// release ALT

robot.keyPress(java.awt.event.KeyEvent.VK\_X); %// send X

robot.keyRelease(java.awt.event.KeyEvent.VK\_X); %// release X

pause(5);

% When begin is pressed,

% save the values to the appropriate place,

% while checking to make sure all inputs are numeric

if ~isempty(get(handles.PlayerBox,'string'))

Init.PN = get(handles.PlayerBox,'string');

end

if ~isempty(str2double(get(handles.InitNumHandsBox,'string'))) % convert to number first, checking if it is empty before assigning

Init.NumHands = str2double(get(handles.InitNumHandsBox,'string')); % assign to our declared variable for future use

end

if str2double(get(handles.InitNumHandsBox,'string')) > 6

Init.Numhands = 6;

end

if str2double(get(handles.InitNumHandsBox,'string')) <=0

Init.Numhands = 1;

end

if ~isempty(str2double(get(handles.InitBankrollBox,'string')))

Init.Bankroll = str2double(get(handles.InitBankrollBox,'string'));

end

if str2double(get(handles.InitBankrollBox,'string')) < 1000

Init.Bankroll = 1000; % Ensures the proper starting bankroll

end

if ~isempty(str2double(get(handles.InitBetUnitBox,'string')))

Init.BetUnit = str2double(get(handles.InitBetUnitBox,'string'));

end

if str2double(get(handles.InitBetUnitBox,'string')) <= 0

Init.BetUnit = 25;

end

if str2double(get(handles.InitBetUnitBox,'string')) > 500 % Ensures the proper starting betting unit

Init.BetUnit = 500;

end

if ~isempty(str2double(get(handles.InitNumDecksBox,'string')))

Init.NumDecks = str2double(get(handles.InitNumDecksBox,'string'));

end

if str2double(get(handles.InitNumDecksBox,'string')) <=0

Init.NumDecks = 1;

end

if str2double(get(handles.InitNumDecksBox,'string')) > 6

Init.NumDecks = 6;

end

if ~isempty(str2double(get(handles.InitGameSpeedBox,'string')))

Init.GameSpeed = str2double(get(handles.InitGameSpeedBox,'string'));

end

if str2double(get(handles.InitGameSpeedBox,'string')) <=0

Init.GameSpeed = 50; % Ensures proper game speed

end

if str2double(get(handles.InitGameSpeedBox,'string')) > 100

Init.GameSpeed = 50;

end

close all

CardGet

function [ccard] = CardGet(RANK,SUIT)

% left bottom corner (X0,Y0)

% top right corner (X1, Y1)

% CSIZE - size dimensions of a card

% obj - card itself

% BJBoardCreation;

% global CARDS;

%

% RANK = 0;

% SUIT = CARDS.Suit(CARDS.NextCard);

Card = SUIT;

if RANK == 11

Rank = 'J';

concat = [Rank Card '.jpg'];

ccard = imread(concat);

elseif RANK == 1

Rank = 'A';

concat = [Rank Card '.jpg'];

ccard = imread(concat);

elseif RANK == 12

Rank = 'Q';

concat = [Rank Card '.jpg'];

ccard = imread(concat);

elseif RANK == 13

Rank = 'K';

concat = [Rank Card '.jpg'];

ccard = imread(concat);

else

Rank = RANK;

concat = [num2str(Rank) Card '.jpg'];

ccard = imread(concat);

end

% figure;

% imshow(ccard);

HandsLeft

function HandsLeft(handles)

%function for updating the value of Bankroll.HandsLeft when the

%HandsLeftBox is modified

global BANKROLL;

BANKROLL.HandsLeft = str2double(get(handles.HandsLeftBox,'string'));

Help

function [] = Help()

f = figure('menu','none','toolbar','none','numbertitle','off', 'name','BLACKJACK');

fid = fopen('Help.txt');

ph = uipanel(f,'Units','normalized','position',[0 0 1 1],'title',...

'Blackjack Help');

lbh = uicontrol(ph,'style','listbox','Units','normalized','position',...

[0 0 1 1],'FontSize',9);

indic = 1;

while 1

tline = fgetl(fid);

if ~ischar(tline)

break

end

strings{indic}=tline;

indic = indic + 1;

end

fclose(fid);

set(lbh,'string',strings);

set(lbh,'Value',1);

set(lbh,'Selected','on');

QuitGame

function QuitGame

global QUITGAME;

% set(handles.QuitButton,'UserData',1);

QUITGAME = 1;

uiresume(gcf);

end

ShuffleNow

function ShuffleNow(NumDecks)

%This function takes the number of decks needed as an input, and outputs a

%structure named Cards that contains the value (a number from 1 to 10),

%rank (1 number from 1 to 13), and suit (the character c, h, s or d).

%Value and Rank are different because in blackjack, Jacks, Queens and Kings

%are all worth the same (i.e. Their rank would be 11, 12, or 13, but their

%value is always 10).

% NumDecks = 6;

%Initialize variables

global CARDS

CARDS.Rank = randperm(52\*NumDecks); % Random number out of a row matrix of 52 to 142 cards

CARDS.CardCount(1:13)=0; %Sets a row matrix of 13 values to zero

for n = 1:(52\*NumDecks)

%This sets the card's suit based on the original number's remainder

%when dividing by 4 -> 0 is clubs, 1 is hearts, 2 is spades, 3 is

%diamonds.

switch mod(CARDS.Rank(n), 4)

case 0

CARDS.Suit(n) = 'C';

case 1

CARDS.Suit(n) = 'H'; % Method in randomizing the card's suit by using a remainder calculation

case 2

CARDS.Suit(n) = 'S';

case 3

CARDS.Suit(n) = 'D';

otherwise

fprintf('%s\n','CARDS.Suit error');

CARDS.Suit(n) = 'X';

end

%This sets the card's value such that the first 1/13th of the numbers = 1 (ace),

%next 1/13th of numbers = 2, etc. Also, 11 = Jack, 12 = Queen, 13 = King.

%It is stored in matrix CardArray.

if (CARDS.Rank(n) <= 1\*NumDecks\*4)

CARDS.Rank(n) = 1;

CARDS.Value(n) = 1;

CARDS.CardCount(1)= CARDS.CardCount(1) + 1;

elseif (CARDS.Rank(n) <= 2\*NumDecks\*4)

CARDS.Rank(n) = 2;

CARDS.Value(n) = 2;

CARDS.CardCount(2) = CARDS.CardCount(2) + 1; % Ensures that the value occurence of cards

elseif (CARDS.Rank(n) <= 3\*NumDecks\*4) % Within a deck occurs a maximum of 4 times randomly

CARDS.Rank(n) = 3;

CARDS.Value(n) = 3;

CARDS.CardCount(3) = CARDS.CardCount(3) + 1; %Keeps track of the cards occurence in the 13 zeroed row matrix

elseif (CARDS.Rank(n) <= 4\*NumDecks\*4) %By adding one to the position

CARDS.Rank(n) = 4;

CARDS.Value(n) = 4;

CARDS.CardCount(4) = CARDS.CardCount(4) + 1;

elseif (CARDS.Rank(n) <= 5\*NumDecks\*4)

CARDS.Rank(n) = 5;

CARDS.Value(n) = 5;

CARDS.CardCount(5) = CARDS.CardCount(5) + 1;

elseif (CARDS.Rank(n) <= 6\*NumDecks\*4)

CARDS.Rank(n) = 6;

CARDS.Value(n) = 6;

CARDS.CardCount(6) = CARDS.CardCount(6) + 1;

elseif (CARDS.Rank(n) <= 7\*NumDecks\*4)

CARDS.Rank(n) = 7;

CARDS.Value(n) = 7;

CARDS.CardCount(7) = CARDS.CardCount(7) + 1;

elseif (CARDS.Rank(n) <= 8\*NumDecks\*4)

CARDS.Rank(n) = 8;

CARDS.Value(n) = 8;

CARDS.CardCount(8) = CARDS.CardCount(8) + 1;

elseif (CARDS.Rank(n) <= 9\*NumDecks\*4)

CARDS.Rank(n) = 9;

CARDS.Value(n) = 9;

CARDS.CardCount(9) = CARDS.CardCount(9) + 1;

elseif (CARDS.Rank(n) <= 13\*NumDecks\*4) %All cards worth 10

CARDS.Value(n) = 10;

if (CARDS.Rank(n) <= 10\*NumDecks\*4)

CARDS.Rank(n) = 10;

CARDS.CardCount(10) = CARDS.CardCount(10) + 1;

elseif (CARDS.Rank(n) <= 11\*NumDecks\*4)

CARDS.Rank(n) = 11;

CARDS.CardCount(11) = CARDS.CardCount(11) + 1;

elseif (CARDS.Rank(n) <= 12\*NumDecks\*4)

CARDS.Rank(n) = 12;

CARDS.CardCount(12) = CARDS.CardCount(12) + 1;

elseif (CARDS.Rank(n) <= 13\*NumDecks\*4)

CARDS.Rank(n) = 13;

CARDS.CardCount(13) = CARDS.CardCount(13) + 1;

end

end

% CARDS.NextCard = 1;

% obj = CardGet(CARDS.Rank(CARDS.NextCard),CARDS.Suit(CARDS.NextCard));

% imshow(obj);

%

end