

Blackjack

Variables Used

- shuff_card1 : The card value for the player's first card. A random number from 1 to 3.
- shuff_card2 : The card value for the player's second card. A random number from 1 to 13.
- shuff_suite1 : The suite for the player's first card. A random number from 1 to 4.
- shuff_suite2 : The suite for the player's second card. A random number from 1 to 4.
- card1Pic = 0 : Value for player's first card's image corresponding to its location in reddeck. A number from 1 to 52.
- card2Pic = 0 : Value for player's second card's image corresponding to its location in reddeck. A number from 1 to 52.
- Fir_tot : Total value of player's first 2 cards.
- Sec_tot : Total value of player's hit cards.
- playerTotal : Total value of all player cards.
- shuff_card3 : The card value for the player's hit card. A random number from 1 to 3. Keeps on changing as the player hits.
- shuff_suite3 : The suite for the player's hit card. A random number from 1 to 4. Keeps on changing as the player hits.
- card3Pic = 0 : Value for player's hit card's image corresponding to its location in reddeck. A number from 1 to 52. Keeps on changing as the player hits.
- shuff_cardAI1 : The card value for the AI's first card. A random number from 1 to 3.
- shuff_cardAI2 : The card value for the AI's second card. A random number from 1 to 13.
- shuff_suiteAI1 : The suite for the AI's first card. A random number from 1 to 4.
- shuff_suiteAI2 : The suite for the AI's second card. A random number from 1 to 4.
- Alcard1Pic = 0 : Value for AI's first card's image corresponding to its location in reddeck. A number from 1 to 52.
- Alcard2Pic = 0 : Value for AI's second card's image corresponding to its location in reddeck. A number from 1 to 52.
- AI_tot : Total value of AI's first 2 cards.
- Sec_tot_AI : Total value of AI's hit cards.
- shuff_cardAI3 : The card value for the AI's hit card. A random number from 1 to 3. Keeps on changing as the player hits.
- shuff_suiteAI3 : The suite for the AI's hit card. A random number from 1 to 4. Keeps on changing as the player hits.
- Alcard3Pic = 0 : Value for AI's hit card's image corresponding to its location in reddeck. A number from 1 to 52. Keeps on changing as the player hits.

Vectors Used

- cards() : Contains string names of all 13 cards in a suite. Eg. Ace, 2, 3, 4, 5 etc.
- suit() : Contains string names of all 4 suits. Eg. clubs, hearts etc.

- Value() : Contains numerical value of all 13 cards in a suite corresponding to their position in the cards() array. Eg. 11,2,3,4,5,6,7,8 etc.

Matlab Commands Used

- clc : Clears the command window.
- clear : Clears the workspace.
- load() : Loads the file containing images of a deck of cards.
- randi() : Generates one, or multiple random numbers between a given range.
- fprintf() : Prints the words and variables listed inside the parenthesis.
- input() : Asks user for input and stores response.
- figure,imshow() : Shows image listed in parenthesis in a new figure.
- set() : Used to set properties such as name of the figure.

The game starts with loading the card images. There are 3 vectors. One has card names, i.e Ace, 2, 3, 4 etc. as strings. The second has names of the 4 suites as strings and the final one has integer values of the cards in blackjack corresponding to the name array, i.e. 11,2,3,4,5,6,7,8,9,10,10,10,10. Then there are variables that have 2 random numbers from 1 to 13, which make the card value of the cards to be distributed to the player. Another 2 variables have a random number from 1 to 4 which make the suite of the 2 player cards. Then the numerical value of the cards is added and saved in a variable that tracks the total of the player. Then there are variables to store the numerical value of the card corresponding to its place in the reddeck array which contains images. Eg. Ace of Clubs is 1 and King of Diamonds is 52. To find the value, the program checks to see the suite number of the card, then finds the number of the card and gives it a value. Eg. If the card is a 9 of hearts, the program will see that the suite number is 3, it then notices the card number is 9. Then this means that this is the 35th element of reddeck, so the program will allot the value of 35 to the variable that tracks the image value of the card. It repeats this process for the second card as well. Once the 2 image values are attained, the program displays the image at this number from the reddeck array. Then it asks the user if he wants to stay or hit. If the user decides to hit, the program repeats the entire process to set up the hit card and display it as many times as the user wants to hit, or till he is busted. The program then stores the final total of the player and moves on to the AI. The entire process of finding the first two cards is repeated and the card values and images are shown. The AI then sees if his total is less than 17. If yes, he has to hit and stays if he is over 17. The process to add AI hit card is the same as the process of finding hit card of AI. The process continues until AI goes over 17 or busts. Then the final total of the AI is stored. Then the program checks if the AI has busted, if not it goes on to compare the final totals of the player and the AI, and declares the one with the higher total the winner.