This project is about customizing the yathzee game in assembly languege.

In main proc,ý used code label called loop1 to work main proc 12 times in total.To count each step of game,i increased loop count memory label and at the and of main proc,i simply compared it with 12. After that,Total score that achieved by user is calculated and printed.Finally the code asks the user if he wants to play again or not.Also in the beginning part there are some steps to clear memory labels.

In each step,the code asks user to press enter to roll a dice 'works with only the enter key'.After rolling a set of die,it asks user if he\she wants to reroll or just fill the score table.If user calls the char 'y' then the code asks for the number of dies to be rerolled and the dies to be rerolled in order,two times.In this step RerollADice and DisplayDices procs are used for rerolling and displaying dies afterwards.After rerolling or notreroll ing then the code goes code label called 'l6' to ask for which combination ll be choosed.Simply,user is asked for a number that represents the combination he\she choosees.Then eax is compared with sequence of 1-12.When eax equals on of them code jumps relevant code label and calculates the current die set.After caclculating,that relevant label jumps to 'final' code label and stores score in score memory label set .And 11 times more.

PROC 1:ScoreCheck
Is used in a purpose of checking whether the number representing combination entered by user is used before or not.I used another ScoreC array to hold checked memory labels.If its used,the code moves the value '1' in the ScoreC in the main proc.
If ýts used before,ebx is changed into 1.If not ,ebx is set by 0.

PROC 2:RerollADice

The dies that ll be rerolled is pointed in RerollD
 memory label in main proc.In here,i scan RerollD
and the parts that is '1' is relatively rerolled a
gain in die set.At the end we set all values in Re
rollD by 0.

PROC 3:RollADice
Simply randomizes 5 values between 1-6 and stores t
hem in Dices memory label set.

PROC 4:DisplayDices
DÝsplays dies in order using Dices memory label.

PROC 5:Display
Displays the menu.

PROC 6:CalOnes
Calculate Ones is used for calculating the score o
f the die set in ones Combination.Ebx is set by 0
in the beginning.Adds '1'to ebx if the set has one
.Calculated score is stored in ebx.

PROC 7:CalTwos
Is used for calculating the score of the die set i
n twos Combination.Adds '2'to ebx if the set has o
ne.Calculated score is stored in ebx.

PROC 8:CalThrees
Is used for calculating the score of the die set i
n threes Combination.Adds '3'to ebx if the set has
 one.Calculated score is stored in ebx.

PROC 9:CalFours
Is used for calculating the score of the die set i
n fours Combination.Adds '4'to ebx if the set has
one.Calculated score is stored in ebx.

PROC 10:CalFives
Is used for calculating the score of the die set i
n fives Combination.Adds '5'to ebx if the set has
one.Calculated score is stored in ebx.

PROC 11:CalSixes
Is used for calculating the score of the die set i
n sixes Combination.Adds '6'to ebx if the set has
one.Calculated score is stored in ebx.

PROC 12:ThreeOfAKind
Is used for calculating the score of die set in Th
ree of a kind combination.It scans Dices memory la
bel set and for each value adds 1 to relevant part
 of DiceCount memory label set.If any part is grea
ter than 3 or equal to the 3 that means the set sa
tisfies the three of a kind combination.If not,ýt
doesnt satisfies that combination.
If the die set suits,all dies are sumed up and stor
ed in ebx.If not,ebx is set by 0.
At the end,DiceCount values are set by 0.

PROC 13:FourOfAKind
Almost same with the proc three of a kind.Only che
cks the label part is greater than 4 or equal to t
he 4.

PROC 14:Yathzee
Again almost same with the proc three of a kind,bu
t only when one of the parts is equal to 5,ebx is
set by 50.Else,ebx is set by 0,.

PROC 15:FourInARow
If the die set suits four in a row,it has to inclu
de 1 2 or 3.So if it has one of these ,it scans th
e set 1 by 1 for next values and increments ecx ev
ery time it has next value.For example,after findi
ng 1 in the array it scans for 2,if it finds 2 it
scans for 3 and so on.At the same time it increase
s the ecx by 1.If ecx is greater equal to the 4 th
en it goes to memory label l3 and sets ebx by 20.I
f its lower than 4,it checks for 2 and 3.Finaly if
 it doesnt finds and row,the code sets ebx by 0.

PROC 16:FiveInARow
Almost same with FourInARow but it compares ecx wi
th 5 this time instead of 4.If its equal to 5 then

ebx is set by 30.Else,ebx is set by 0.

PROC 17:AnyThing
Simply finds the total of Dices set and store it i
n ebx.