

ROLL DICE

PREPARED FOR

Proof of Concept

PREPARED BY

ANVESH CHANDUPATLA

LAKSHMI MUVVA

PRASAD GINNARAPU

MADHU GADDAM

DIVYA THANDRA

JAGADEESWARA REDDY VENNAPUSA

INDEX

TITLE	Page No
I. OVERVIEW	3
II. BACKGROUND	3
III. PROBLEM/NEEDS.....	3
IV. INSTALLATION INSTRUCTIONS	4
V. FUNCTIONS/MODULES	4
VI. ALGORITHM FOR ROLLING DICE	5
VII. SOURCE CODE	6
VIII. CONCLUSION	8

I. OVERVIEW

The aim of this POC (Proof of Concept) is to build a virtual roll dice. It has 6 sides and each side represented by a unique number from 1 to 6. It can be rolled in four ways they are Right, Left, Up and Down. Initially, the user prompted a dice with a random face values followed by four options to enter one of them. If user gives any input from prompted options, the dice should be rolled/changed accordingly the user input. And updated values should be printed.

II. BACKGROUND

A dice is a small throwable object with multiple resting positions, used for generating random number. This makes dice suitable as gambling devices for games like craps, or for use in non-gambling tabletop games. A traditional die is an often rounded cube, with each of its six faces showing a different number. The design as a whole is aimed at the die providing a randomly determined integer from one to six, each of those values being equally likely. A variety of similar devices are also described as dice; such specialized dice may have polyhedral or irregular shapes and may have faces marked with symbols instead of numbers. They may be used to produce results other than one through six. Loaded and crooked dice are designed to favor some results over others for purposes of cheating or amusement.

III. PROBLEM/NEEDS

Application should plot the dice with initial positions



Fig: Screenshot of Demo virtual dice.

User should be prompted

Press

R -> to roll right side

L -> to roll left side

U -> to roll upside

D -> to roll Downside

Pick your move:

Example:

Press

R -> to roll right side

L -> to roll left side

U -> to roll upside

D -> to roll Downside

Q -> to quit

Pick your move: R

then the position of the dice should change accordingly.

And print the latest position

If user picks other than the options(R, L, U, D) it should prompt 'INVALID MOVE'

If user picks 'Q' it should print all the moves and the latest position.

IV. INSTALLATION INSTRUCTIONS:

- Go to browser and Download Python IDLE latest version.
- Go to this link [Download Python | Python.org](https://www.python.org/downloads/).

V. FUNCTIONS/MODULES

Random Module:

The `random` module is a built-in module to generate the pseudo-random variables. It can be used perform some action randomly such as to get a random number, selecting a random elements from a list, shuffle elements randomly, etc.

To use this module first import Random Module from library.

```
import random
```

`shuffle()`:

The `random.shuffle()` method randomly reorders/shuffles the elements in a list.

We can import this from random module.

Functions:

roll_right (): This function created for roll right the dice

roll_left (): This function created for roll left the dice

roll_up(): This function created for roll up the dice

roll_down(): This function created for roll down the dice

random. shuffle () : To shuffle the list elements for every take.

Click:

R -To roll right side

L - To roll left side

U -To roll upside

D -To roll Downside

Q -> to quit

Pick your move:

Pick your move: R

then the position of the dice should change accordingly.

And print the latest position

If user picks other than the options (R, L, U, D) it should prompt 'INVALID MOVE'

If user picks 'Q' it should print all the moves and the latest positions.

VI. ALGORITHM FOR ROLLING DICE

step 1: First import the random module for shuffling dice.

step 2: Take a list like [1,2,3,4,5,6] store in a variable. Here, 'lst '

step 3: Take an empty list. This is used to store the performed dice actions.

Step 4: Take a variable to count the number of dice rolls performed. Here, "count "

step 5: Define a function for dice design and take parameter of list variable.

step 6: To draw a dice use print function and format function to make dice design.

step 7: Print the Individual value on dice by using index. [Positions: right, left, up, bottom].

step 8: call the random.shuffle() function with list variable parameter to shuffle the elements for every time

step 9: Take another four functions to rotate right, left, up and down and all functions should be return updated elements then swap the values for changing the dice positions.

step 10: And then Write a while loop to pick the particular move.

step 11: And user should be prompted "Enter the direction to rotate".

step 12: Write the if condition to check the particular move or not.

step 13: count every move after rolled out.

step 15: When the user enter Q for quit, all the performed roll are added to an empty list and then print the all moves of dice.

step 16: Finally, if user Enter any wrong input it shows an "Invalid move" message and again asks the "Enter the direction to rotate".
(until and unless user give Quit, it keeps on asking user to enter direction.)

VII. SOURCE CODE:

```
import random
lst=[1,2,3,4,5,6]
lst1=[]
count=0
def dice(lst):
    print(format(lst[1], '>25'))
    print(format('+-----+', '>30'))
    print(format('/|', '>18'), format('/|', '>11'))
    print(format('/ |', '>18'), format(lst[2], '>3'), format('/ |', '>7'))
    print(format('+--+-----+', '>27'), ' +', lst[0])
    print(format(lst[5], '>13'), format('| /', '>3'), format('| /', '>11'))
    print(format('| /', '>17'), format(lst[4], '>4'), format('| /', '>6'))
    print(format('|/', '>16'), format('|/', '>11'))
    print(format('+-----+', '>27'), format(lst[3], '>20'), sep='\n')

#random.shuffle(lst)
dice(lst)
print("Enter the Direction you want")
print("R -> to roll right side\nL -> to roll left side\nU -> to roll upside\nD -> to roll\nDownside\nQ -> to Quit")
```

```

def roll_right(right): #This function rotates the cube towards right
    temp=lst[4]        #initially front value stored into temporary variable, so front facing
value will be empty,
    lst[4]=lst[5]      #now front value changed by left value
    lst[5]=lst[2]      #left value changed by back value
    lst[2]=lst[0]      #back value shifted to right
    lst[0]=temp        #front value shifted to right
    r=right.copy()
    lst1.append(['Right ',r])
    dice(right)
    return right

```

```

#    print(right)
# Right_rotation = roll_right(lst)
#print("Right rotated",Right_rotation)

```

```

def roll_left(left):
    temp=lst[5]
    lst[5]=lst[4]
    lst[4]=lst[0]
    lst[0]=lst[2]
    lst[2]=temp
    l=left.copy()
    lst1.append(['Left ',l])
    dice(left)

```

```

    return left
# Left_rotation = roll_left(lst)
# print("Left rotated",Left_rotation)

```

```

def roll_up(up):
    temp=lst[4]        # front stored in temp, so front facing is empty now,
    lst[4]=lst[3]      # bottom to front, bottom is empty
    lst[3]=lst[2]      # back to bottom, back is empty
    lst[2]=lst[1]      # top to back, now top is empty
    lst[1]=temp        # front is now changed to top face
    u=up.copy()
    lst1.append(['Up ',u])
    dice(up)

```

```

    return up
# Up_rotation = roll_up(lst)

```

```

# print("Up rotated",Up_rotation)

def roll_down(down):
    temp=lst[3]      # ==> bottom stored in temp, so bottom face is empty now,
    lst[3]=lst[4]    #----->bottom to front, front is empty
    lst[4]=lst[1]    #----->top to front, top is empty
    lst[1]=lst[2]    #----->back to top, now back is empty
    lst[2]=temp      #----->back is now changed to top face
    d=down.copy()
    lst1.append(['Down ',d])
    dice(down)

    return down
# Down_rotation = roll_down(lst)
# print("Down rotated",Down_rotation)

while True:
    option = input("Enter the direction to rotate:").upper()
    if option == 'R':
        count+=1
        print("Right rotation",roll_right(lst))
    elif option == 'L':
        count+=1
        print("Left rotation",roll_left(lst))
    elif option == 'U':
        count+=1
        print("Up rotation",roll_up(lst))
    elif option == 'D':
        count+=1
        print("Down rotation",roll_down(lst))
    elif option == 'Q':
        print(count,"Rotations done!")
        print(lst1)
        for i,j in lst1:
            print(i)
            dice(j)
        break
    else:
        print("Invalid Move")

```

VIII. CONCLUSION:

Finally, we print the all moves of dice and according to user requirements.