

**MINI PROJECT**

**NAME OF THE PROJECT:-**

Dice Rolling Simulator

**NAME**:-P.V.N.SIRISHA

**REG NO**:-192210231

**CONSULTANT PROFSSOR:-**

[DR.GANESH RAMCHANDRAN]

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENT** | **PAGE NO** |

|  |  |  |
| --- | --- | --- |
| 1 | ABSTRACT | 3 |
| 2 | ABOUT PYTHON | 4 |
| 3 | INTRODUCTION | 5 |
| 4 | FLOW CHART | 6 |
| 5 | CODE | 7-9 |
| 6 | OUTPUT SCREEN | 10 |
| 7 | CONCLUSION | 11 |

# ABSTRACT:-

* In This Project We Desgined A Dice Game Which Will Useful For

Kids.And it is related to mathematics with some fun.

* Coming to the game when the user press the **‘Roll Dice’** button the two dices will be rolled and displayed on screen.

* Then the user have to add the two values of the dices and enter in the text box provided and press submit button.

* If the user entered value is correct a popup message box will come with a message “Congratulations,You are correct ”, if the user enters Wrong value a popup message box will come with a message

“Sorry,you are wrong”.

In the end, our project isn't just about understanding Rolling dice. It's about creating a game that's easy to enjoy and sharing what we discovered with everyone who loves playing games.

Student signature

# [student name] INTRODUCTION:-

**ABOUT PYTHON**:-

* Python is a high-level, general-purpose, and interpreted programming language used in various sectors including machine learning, artificial intelligence, data analysis, web development, and many more.
* Python is known for its ease of use, powerful standard library, and dynamic semantics. It also has a large community of developers who keep on contributing towards its growth.
* The major focus behind creating it is making it easier for developers to read and understand, also reducing the Lines of code.
* Python has plenty of features that make it the most demanding and more popular. Let’s read about a few of the best features that

Python has:

* Easy to read and understand
* Interpreted language
* Object-oriented programming language
* Free and open-source
* Versatile and Extensible
* Multi-platform
* Hundreds of libraries and frameworks.
* Flexible, supports GUI
* Dynamically typed

# Introduction To Tkinter:-

Python Offers multiple options for developing GUI(Graphical User Interface). out of all the GUI Methods, tkinter is the most commonly used method. it is a standard python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. Creating a GUI application using Tkinter is an easy task. All We need to do is perform the following steps.

# To create a GUI using tkinter:-

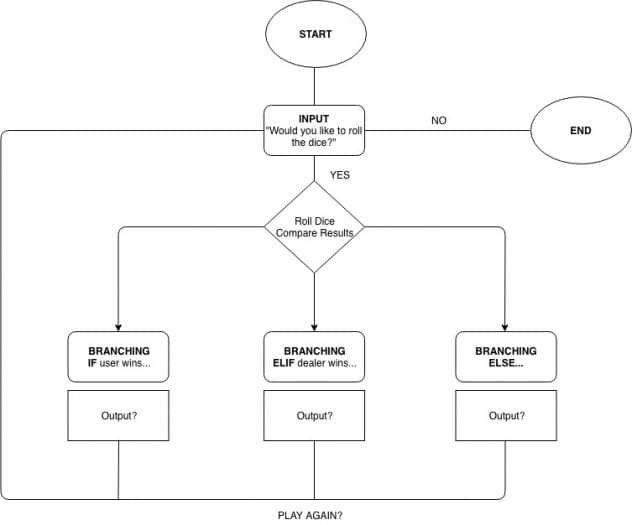
* importing the tkinter module
* create the main window
* Add one or more widgets to the main window
* Enter the main event loop to take action against each event triggered by the user.

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

# About Game: -

In the game we have 2 dices and it will roll simultaneously when user press roll dice button and it will give output. The user has to add the values displayed on the dice and enter the correct value in the text box/entry box provided in the screen. the program will check the user entered value with the correct value. if the user entered value is correct it displays a message box with the message "Congratulations You are correct" and if the user enters wrong or incorrect value it displays a message box with the message " Sorry you are wrong".

## FLOW CHART



## CODE:-

from tkinter import \* from tkinter import messagebox import random

global total root = Tk() root.title('Add The Dice') root.geometry("500x500") # Get The Dice Number def get\_number(x):

if x == '\u2680': return (1) elif x == '\u2681': return (2) elif x == '\u2682': return (3) elif x == '\u2683': return (4) elif x == '\u2684': return (5) elif x == '\u2685': return (6) # Roll the dice def roll\_dice(): global total

# Roll Random Dice d1 = random.choice(my\_dice) d2 = random.choice(my\_dice) # Determine Dice Number sd1 = get\_number(d1) sd2 = get\_number(d2) # Update Labels dice\_label1.config(text=d1) dice\_label2.config(text=d2)

# Update Total Label total = sd1 + sd2

#print(type(total), type(e1.get()))

# Submit Function def Submit(): if int(e1.get()) == total:

messagebox.showinfo(root, "congratulations, you are correct")

else:

messagebox.showinfo(root, "sorry,you are wrong") e1.delete(0, END) # Create A Dice List my\_dice = ['\u2680', '\u2681', '\u2682', '\u2683', '\u2684', '\u2685', ]

# Create a Frame my\_frame = Frame(root) my\_frame.pack()

# Create Dice Labels dice\_label1 = Label(my\_frame, text='', font=("times", 100), fg="blue") dice\_label1.grid(row=0, column=0, padx=5) sub\_dice\_label1 = Label(my\_frame, text="") sub\_dice\_label1.grid(row=1, column=0) dice\_label2 = Label(my\_frame, text='', font=("times", 100), fg="red") dice\_label2.grid(row=0, column=1, padx=5) sub\_dice\_label2 = Label(my\_frame, text="") sub\_dice\_label2.grid(row=1, column=1)

# Create Roll Button

my\_button = Button(root, text="Roll Dice", command=roll\_dice, font=("Helvetica",

20),fg="blue",bg="yellow") my\_button.pack(pady=10)

# creating Entry labels

ans = Label(root, text="Enter Total Of 2 Dices and press

submit",font=("times",15),fg="cyan",bg="black") ans.pack()

# creating Text Boxes e1 = Entry(root) e1.pack() # creating Button submit = Button(root, text="Submit", command=Submit, font=("times", 15)) submit.pack() # Roll the dice roll\_dice()

root.mainloop()

## OUTPUT SCREEN:-

**Test**

**case:**

**-**

**1**

**Test**

**case:**

**-**

**2**



## CONCLUSION: -

* This game helps a lot for Childrens, they can play and do addition(maths) at a time.
* The children like this game and parents will be able to teach maths (Additions concept) their children in very simple and understandable way.
* This project serves as a great introduction to basic Python programming concepts, including functions, user input handling, and conditional statements. It also demonstrates the use of the random module for generating random values.