Report On

Dice Roller GUI

Submitted in partial fulfillment of the requirements of the Course project in

Semester IV of Second Year Computer Engineering

by

Vedant Jagdale (Roll No. 61)

Jyoti Patel (Roll No. 66)

Ankul Tumsare (Roll No. 69)

Supervisor

Prof. Sneha Mhatre

**Vidyavardhini's College of Engineering & Technology**

**Department of Computer Engineering**



**(2024-25)**

**1**

**Vidyavardhini's College of Engineering & Technology**

**Department of Computer Engineering**

**CERTIFICATE**

This is to certify that the project entitled “Dice Roller GUI” is a bonafide work of "Vedant Jagdale (Roll No. 61), Jyoti Patel (Roll No. 66), Ankul Tumsare (Roll No. 69), Jidnyasa Naik (Roll No. 79)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester IV of Second Year Computer Engineering.

**Supervisor**

|  |  |  |
| --- | --- | --- |
| Prof. Sneha Mhatre |  |  |
|  |  |  |
| Dr Megha Trivedi  Head of Department | **2** | Dr. H.V. Vankudre  Principal |

**Abstract**

This project aims to design and implement a Graphical User Interface (GUI) for a Dice Roller application. The traditional dice rolling experience is transformed into a user-friendly interface, providing convenience and enhanced interaction for users. The project utilizes Python programming language and relevant libraries such as Tkinter for GUI development The project involves several stages of development, including conceptualization, design, implementation, testing, and documentation. Throughout the process, attention is paid to user experience and interface design principles to create an engaging and seamless interaction environment. By the completion of this project, a fully functional GUI-based Dice Roller application will be delivered, catering to the needs of tabletop gaming enthusiasts and users requiring random number generation in various scenarios. The project not only serves as a practical application but also demonstrates the utilization of GUI development techniques using Python, contributing to the advancement of software development skills.

Top of Form

**3**

**INDEX**

**Contents**

**1 Introduction**

1.1 Introduction

1.2 Problem Statement

**2 Proposed System**

2.1 Block diagram, its description and working [ER diagram]

2.2 Module Description

2.3 Brief description of software & hardware used and its programming

2.4 Code

**3** Results and conclusion

**4** References

**4**

**INTRODUCTION**

* 1. **Introduction**

This project focuses on developing a Graphical User Interface (GUI) for a Dice Roller application, aiming to modernize traditional tabletop gaming experiences. By leveraging programming languages and GUI frameworks, we aim to create an intuitive platform for generating random numbers, catering to gaming enthusiasts and users requiring random number generation.The project encompasses stages from conceptualization to implementation, emphasizing user experience principles and software development methodologies. Through innovation and customization, we aim to enhance gaming experiences while advancing our skills in software development.

* 1. **Problem Statement**

Despite the widespread popularity of tabletop gaming and the ubiquity of digital tools, the traditional method of rolling physical dice remains prevalent. However, this approach often presents limitations in terms of convenience, customization, and accessibility. Existing digital dice rollers lack intuitive interfaces and fail to meet the diverse needs of users, including gamers, educators, and professionals requiring random number generation.

**5**

**PROPOSED SYSTEM**

**Key features of the proposed system include:**

**1.Intuitive Interface:** The GUI presents an intuitive layout, allowing users to easily access and customize dice types, roll counts, and result summaries.

**2.Customization Options:** Users can select from a variety of dice types commonly used in tabletop gaming, including D4, D6, D8, D10, D12, and D20. They can also adjust the number of dice rolled and view the sum of the results for each roll.

**3.Responsive Design:** The GUI is designed to be responsive, ensuring seamless performance across different devices and screen sizes. Users can access the application from desktop computers, laptops, tablets, and smartphones.

**4.Visual Feedback:** The system provides visual feedback to users during the dice rolling process, including animations or sound effects to simulate the experience of rolling physical dice.

**5.Accessibility:** The GUI is designed with accessibility in mind, incorporating features such as keyboard shortcuts, screen reader compatibility, and customizable color schemes to accommodate users with different needs and preferences.

**2.2 Module Description:**

The Dice Roller GUI application is composed of essential modules to ensure a smooth user experience. The User Interface Module presents the interface and handles interactions, while the Dice Rolling Module generates random numbers based on user-defined parameters. The Customization Module allows users to tailor settings, and the Result Display Module showcases outcomes. Accessibility features are provided by the Accessibility Module, while error detection and handling are managed by the Error Handling Module. Finally, the Responsive Design Module ensures consistent usability across different devices. Together, these modules create a user-friendly and efficient Dice Roller GUI application.

**6**

**2.3 Description of Software & Hardware Used And Its Programming:**

**Software:**

**Python:** The primary programming language used for development, providing a versatile and efficient platform for building the application.

**Tkinter:** A standard GUI library for Python, utilized for creating the graphical user interface of the Dice Roller application.

**Integrated Development Environment (IDE):** Software tools like PyCharm, Visual Studio Code, or IDLE are employed for coding, debugging, and testing the application code efficiently.

**Operating System**: The application is developed and tested on various operating systems such as Windows, macOS, and Linux to ensure cross-platform compatibility.

**Hardware:**

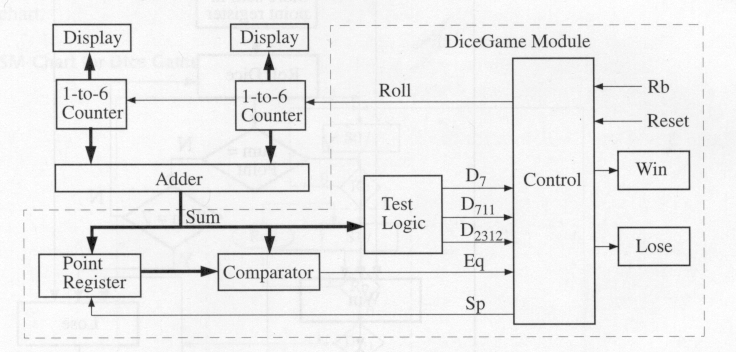
**Personal Computer:** A standard desktop or laptop computer is used as the primary development environment, equipped with adequate processing power and memory to support software development tasks.

**Input Devices:** Standard input devices such as a keyboard and mouse are used for interacting with the computer and testing the application.

**Internet Connectivity**: Internet access is required for downloading software tools, libraries, and dependencies, as well as for accessing online resources and documentation during the development process.

**7**

**Block Diagram:**

****

**8**

**2.4 Code**

**import tkinter**

**from PIL import Image, ImageTk**

**import random**

**import numpy as np**

**import pandas as pd**

**# Toplevel widget which represents the main window of an application**

**root = tkinter.Tk()**

**root.geometry('400x400')**

**root.title("Roll the Dice")**

**# Adding label into the frame**

**l0 = tkinter.Label(root, text="ROLL THE DICE")**

**l0.pack()**

**# Adding label with different font and formatting**

**l1 = tkinter.Label(root, text="TRY YOUR LUCK!", fg="light green",**

**bg="black",**

**font="Helvetica 16 bold italic")**

**l1.pack()**

**# Images**

**dice = ['die1.png', 'die2.png', 'die3.png', 'die4.png', 'die5.png', 'die6.png']**

**# Simulating the dice with random numbers between 0 to 6 and generating image**

**image1 = ImageTk.PhotoImage(Image.open(random.choice(dice)))**

**# Construct a label widget for image**

**label1 = tkinter.Label(root, image=image1)**

**label1.image = image1**

**# Packing a widget in the parent widget**

**label1.pack(expand=True)**

**# Function activated by button**

**def rolling\_dice():**

**image1 = ImageTk.PhotoImage(Image.open(random.choice(dice)))**

**# Update image**

**label1.configure(image=image1)**

**# Keep a reference**

**label1.image = image1**

**# Adding button, and command will use rolling\_dice function**

**button = tkinter.Button(root, text='Roll the Dice', fg='blue', command=rolling\_dice)**

**# Pack a widget in the parent widget**

**button.pack(expand=True)**

**# Call the mainloop of Tk**

**# Keeps window open**

**root.mainloop()**

**9**

**RESULTS AND CONCLUSION**

**Conclusion:**

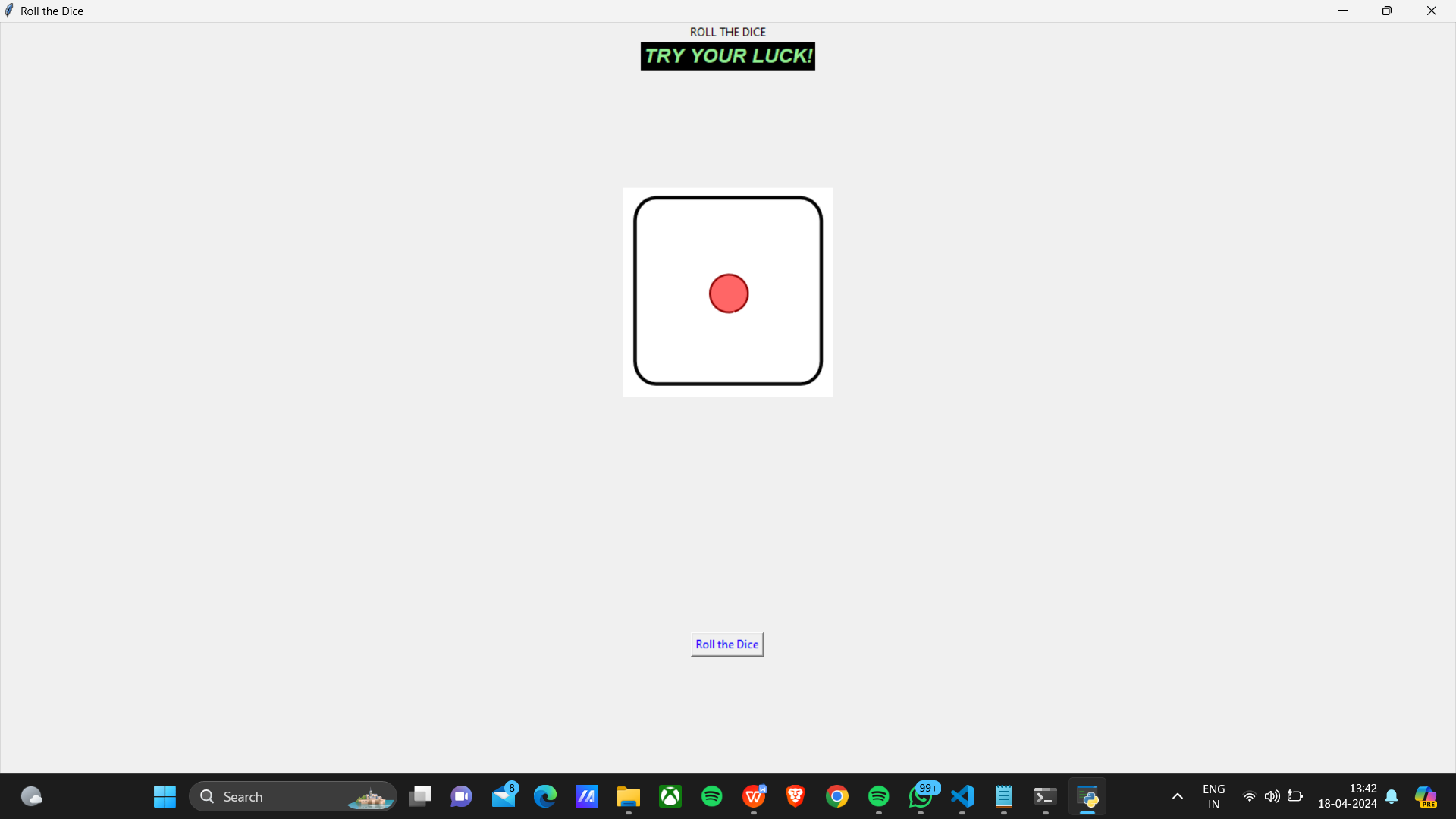
In conclusion, the Dice Roller GUI application represents a significant advancement in modernizing traditional tabletop gaming experiences and enhancing accessibility to random number generation tools. Through Python and Tkinter, we've created a user-friendly platform with modules ensuring seamless user interaction. Future iterations could include advanced features like dice customization and online platform integration. Overall, the application showcases technology's potential to enhance gaming experiences and streamline random number generation processes, making a meaningful impact in diverse contexts.

**Results:**

The Dice Roller GUI application offers a user-friendly solution for generating random numbers, modernizing traditional tabletop gaming experiences. Through Python and Tkinter, we've created a seamless platform with intuitive modules ensuring efficient user interaction. With features like dice customization and online platform integration in future iterations, the application showcases technology's potential to enhance gaming experiences. Overall, it stands as a meaningful advancement in streamlining random number generation processes and making gaming more accessible.

**10**

**Output**

****

**11**

**REFERENCES**

**Tkinter Documentation:** https://docs.python.org/3/library/tkinter.html

**W3Schools Tkinter Tutorial:** [https://www.w3schools.com/python/python\\_tkinter.asp](https://www.w3schools.com/python/python\\_tkinter.asp)

**JetBrains PyCharm:** https://www.jetbrains.com/pycharm/

**12**