

# SIMATS ENGINEERING

CSA0880-python programming for web application  
CAPSTONE PROJECT-JUNE 2024

## DICE ROLLING SIMULATOR

GUIDED BY :

R. YUVARANI

PRESENTED BY :

1.C.SAI MURALI[192211951]

2.S.SARATH SAI[192210051]

3.M.ROHIT[192211415]

# ABSTRACT

- ▶ Dice rolling simulator is a software that mimics the action of rolling dice. It strives to faithfully recreate the element of chance associated with real life dice rolling through accurate simulations.
- ▶ This feature allows user to experience a realistic and interactive dice rolling experience.
- ▶ A virtual dice roller that generates random numbers like a real dice. Once you hit on the roll button, the simulator picks a random number that match those dice faces.
- ▶ The dice rolling simulator offers a versatile and immersive platform for simulating dice rolls with precision and realism.
- ▶ PRNG algorithm that provides sufficient randomness for the intended application.

# INTRODUCTION

- ▶ The Dice Rolling Simulator is a simple Python program that simulates the experience of rolling a six-sided dice.
- ▶ It provides a text-based interface where users can roll the dice and see the result displayed on the screen.
- ▶ The program allows users to roll the dice repeatedly, providing them with an opportunity to experience the randomness of dice rolls in a virtual environment.
- ▶ The simulator utilizes python's random module to generate random numbers, simulating the outcome of rolling a dice with sides.
- ▶ Overall, the Dice Rolling Simulator provides a simple yet entertaining way for users to experience the thrill of rolling a dice in a virtual setting, making it suitable for educational purposes, gaming enthusiasts, or anyone looking for a fun diversion.

# ALGORITHM

- ▶ Import the necessary modules ``random`` and ``time``.
- ▶ Define a function ``roll_dice()`` that generates a random number between 1 and 6, simulating the roll of a six-sided dice.
- ▶ Define a function ``print_dice(number)`` that prints a graphical representation of the dice face corresponding to the given number.
- ▶ Define the ``main()`` function:
  - a. Print a welcome message.
  - b. Start an infinite loop.
  - c. Prompt the user to press Enter to roll the dice or 'q' to quit.
  - d. If the user presses Enter, roll the dice, print a message indicating that the dice is rolling, wait for a second (using ``time.sleep(1)``), print the graphical representation of the rolled dice, and print the number rolled.
- ▶ If the user inputs 'n' when asked if they want to roll again, print a farewell message and break out of the loop.
- ▶ Call the ``main()`` function if the script is executed directly.
- ▶ This algorithm outlines the flow of the code, guiding how it operates and interacts with the user.

# PROGRAM

```
import random
import time
def roll_dice():
    return random.randint(1, 6)
def print_dice(number):
    if number == 1:
        print(" ----- ")
        print("| | ")
        print("| • | ")
        print("| | ")
        print(" ----- ")
    elif number == 2:
        print(" ----- ")
        print("| • | ")
        print("| | ")
        print("| • | ")
        print(" ----- ")
```



```
    print(" ----- ")
    print("| • |")
    print("| • |")
    print("| • |")
    print(" ----- ")
elif number == 4:
    print(" ----- ")
    print("| • • |")
    print("| |")
    print("| • • |")
    print(" ----- ")
elif number == 5:
    print(" ----- ")
    print("| • • |")
    print("| • |")
    print("| • • |")
    print(" ----- ")
elif number == 6:
    print(" ----- ")
    print("| • • |")
    print("| • • |")
    print("| • • |")
    print(" ----- ")
```

```
def main():
    print("Welcome to the Dice Rolling Simulator!")
    while True:
        input("Press Enter to roll the dice (Press 'q' to quit): ")
        dice_number = roll_dice()
        print("\nRolling the dice...")
        time.sleep(1)
        print_dice(dice_number)
        print("\nYou rolled:", str(dice_number))
        if input("\nRoll again? (y/n): ").lower() == 'n':
            print("Thanks for playing!")
            Break
if __name__ == "__main__":
    main()
```

# OUTPUT

```
Welcome to the Dice Rolling Simulator!  
Press Enter to roll the dice (Press 'q' to quit):
```

```
Rolling the dice...
```

```
-----  
|  •  •  |  
|      |  
|  •  •  |  
|      |  
-----
```

```
You rolled: 4
```

```
Roll again? (y/n): y  
Press Enter to roll the dice (Press 'q' to quit):
```

```
Rolling the dice...
```

```
-----  
|  •    |  
|      |  
|      • |  
|      |  
-----
```

```
You rolled: 2
```

```
Roll again? (y/n): n  
Thanks for playing!
```



# RESULT AND DISCUSSION

The experiment with the dice rolling simulator demonstrates how well it generates random results that resemble conventional dice rolls. After a great deal of testing, the simulator continuously generated a wide range of outcomes that were representative of the inherent unpredictability of dice-based games. The distribution of roll results verified the quality of the simulator's random number generation process, as it nearly matched the expected probability for each face of the dice. Nonetheless, few disparities were observed, emphasizing the intrinsic fluctuation in probabilistic procedures and the necessity for more improvement in subsequent rounds.

# CONCLUSION

The dice rolling simulator experiment has demonstrated its ability to accurately simulate random dice rolls, providing a reliable tool for various applications such as gaming and statistical analysis. The experiment revealed that the simulator effectively generates outcomes reflective of the probabilities associated with traditional dice, albeit with occasional minor discrepancies. Despite these minor imperfections, the simulator's overall performance and user experience were deemed satisfactory, showcasing its potential value as a versatile tool for simulating random events.

**THANK YOU**