

UNIVERSITY INSTITUTE *of*  
**COMPUTING**  
*Asia's Fastest Growing University*

NAAC  
GRADE **A+**  
ACCREDITED UNIVERSITY

# **A PROJECT REPORT** **PYTHON**

MASTER OF COMPUTER APPLICATION

SUBMITTED BY

**MD SHAHJAD RAJA**

UID – 24MCA20291

CHANDIGARH UNIVERSITY, MASTER OF COMPUTER APPLICATION,

NORTH CAMPUS, PUNJAB(INDIA)

25 OCTOBER 2024



**CHANDIGARH**  
**UNIVERSITY**

Discover. Learn. Empower.

# **BANK MANAGEMENT SYSTEM USING Tkinter**

---

## **INDEX PAGE**

1. Introduction
  2. Literature Review
  3. Description of the Program
  4. Result
  5. Conclusion
  6. References
-

## **ABSTRACT**

The Snake Water Gun Game is a classic three-choice game similar to Rock, Paper, Scissors, where each option defeats or loses to the other. This project converts the game into a graphical user interface (GUI) using Python's Tkinter library. The player selects one of the options (Snake, Water, or Gun), and the computer randomly selects an option. The program then determines whether the player wins, loses, or the game ends in a draw, displaying the results on the screen.

## **INTRODUCTION**

The Snake Water Gun Game is a simple game that tests the user's luck. Each of the three options has a set of outcomes when compared with the other two. The goal is to create an enjoyable user interface using Tkinter for an interactive and visually appealing game. This program enhances the user experience by replacing the traditional console-based input/output method with graphical components such as buttons and labels.

## **DESCRIPTION**

The program is built using the Tkinter library in Python. It includes:

### **1. Game Logic:**

- **Choices:** The player chooses Snake (s), Water (w), or Gun (g) using buttons.
- **Outcome:** The computer randomly selects one of the options. The game compares the player's choice and the computer's choice to determine the result: win, lose, or draw.

### **2. Graphical User Interface:**

- A window is created with buttons for each option (Snake, Water, Gun).
- The result is displayed on the screen after each round.

### 3. Flow of Interaction:

- The user clicks on one of the buttons to select their choice.
- The computer randomly picks a choice.
- The result is shown on the screen through a label that updates after each round.

## CODE IMPLEMENT

```
import tkinter as tk
from tkinter import messagebox
import random

class SnakeWaterGunGame:
    def __init__(self, root):
        self.root = root
        self.root.title("Snake Water Gun Game")
        self.root.geometry("400x300")

        self.title_label = tk.Label(root, text="Snake, Water, Gun Game", font=("Arial", 16))
        self.title_label.pack(pady=10)

        self.instruction_label = tk.Label(root, text="Choose one option:", font=("Arial", 12))
        self.instruction_label.pack(pady=10)

        self.snake_button = tk.Button(root, text="Snake", width=10, command=lambda:
self.play_game('s'))
        self.snake_button.pack(pady=5)

        self.water_button = tk.Button(root, text="Water", width=10, command=lambda:
self.play_game('w'))
        self.water_button.pack(pady=5)

        self.gun_button = tk.Button(root, text="Gun", width=10, command=lambda:
self.play_game('g'))
        self.gun_button.pack(pady=5)

        self.result_label = tk.Label(root, text="", font=("Arial", 12))
        self.result_label.pack(pady=20)

    def play_game(self, youstr):

        computer = random.choice([-1, 0, 1])
        youDict = {"s": 1, "w": -1, "g": 0}
```

```
reverseDict = {1: "snake", -1: "water", 0: "gun"}

    you = youDict[youstr]

    result_text = f"You chose {reverseDict[you]}. Computer chose  
{reverseDict[computer]}.\n"

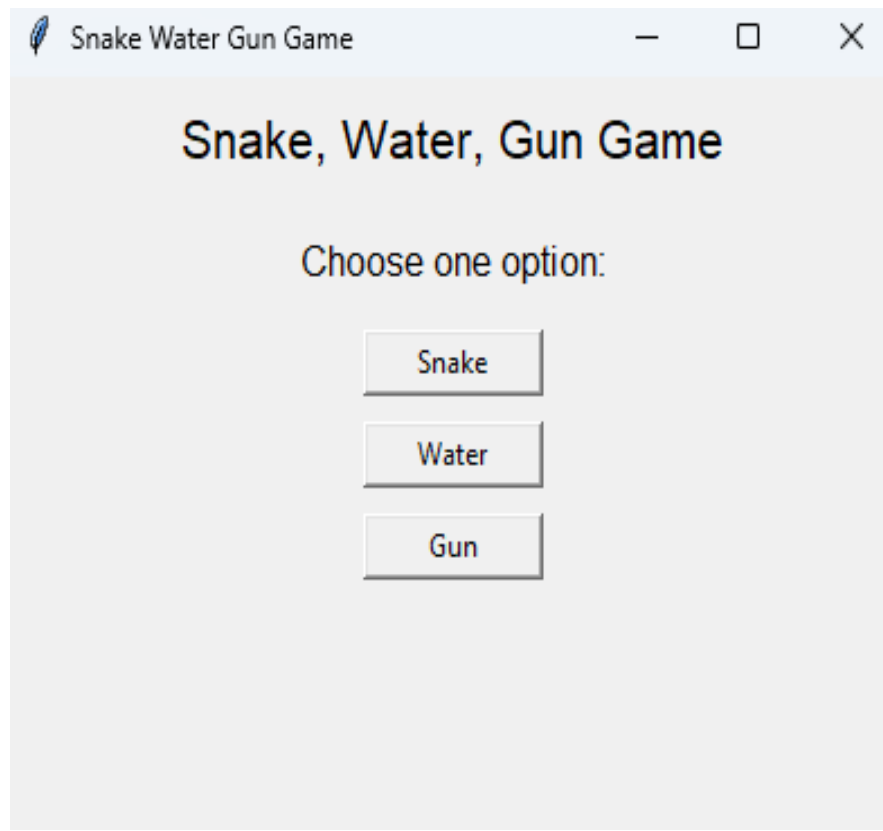
    if computer == you:
        result_text += "It's a draw!"
    else:
        if (computer == -1 and you == 1) or (computer == 1 and you == 0) or (computer == 0  
and you == -1):
            result_text += "You win!"
        else:
            result_text += "You lose!"

    self.result_label.config(text=result_text)

if __name__ == "__main__":
    root = tk.Tk()
    game = SnakeWaterGunGame(root)
    root.mainloop()
```



## OUTPUT



## **LITERATURE REVIEW**

Graphical User Interfaces (GUIs) in games are critical in providing an intuitive and engaging experience for users. Tkinter, the standard GUI library for Python, allows developers to create easy-to-use windows and handle event-driven programming effectively. Research has shown that simple games with user-friendly interfaces tend to have higher engagement rates. Furthermore, incorporating randomness in games increases their replay ability, as seen in classic games such as Rock-Paper-Scissors, from which Snake Water Gun derives its concept.

## **RESULT**

The Snake Water Gun Game is fully functional and allows users to interact with the program by clicking buttons rather than typing input. Each outcome (win, lose, or draw) is displayed immediately after the user's selection, providing instant feedback. The graphical user interface enhances the user experience by making the game more engaging and enjoyable compared to a traditional console version.

## **CONCLUSION**

The Snake Water Gun Game is fully functional and allows users to interact with the program by clicking buttons rather than typing input. Each outcome (win, lose, or draw) is displayed immediately after the user's selection, providing instant feedback.

The graphical user interface enhances the user experience by making the game more engaging and enjoyable compared to a traditional console version.

## **REFERENCES**

- Tkinter Documentation: <https://docs.python.org/3/library/tkinter.html>
- Random Module in Python: <https://docs.python.org/3/library/random.html>
- Snake Water Gun Game Concept: Derived from the classic game Rock-Paper-Scissors.