Undergraduate Elective Course

A Lottery Mini Program by Python

Beijing Institute of Technology

Ming-Jian Li 李明健

mingjianli@bit.edu.cn

mingjianli.com

1. Introduction

2. Method

3. Results

4. Discussion

Introduction

I am a teacher for AI and Simulation Science in BIT

*Picture generated by Midjourney AI

Introduction

I need 10 students to make presentation for the final class

*Picture generated by Midjourney AI

Introduction

But after several weeks, I still have no volunteers

*Picture from *My So-Called Life*

Introduction

So I decide to choose 10 lucky guys

*Picture from *Mr. and Mrs. Smith*

Introduction

Through a lottery draw !

*Picture from *Bing*

Introduction

Let ~~fate~~ probability decide everything !

Pierre-Simon Laplace

Thomas Bayes

Method

The code creates a simple GUI application where users can click "Start" to begin a **random scrolling of names** and "Stop" to **freeze on a randomly selected name**, simulating a **lottery draw**.

Tools: Python for coding

TKinter library for graphic interface

Random library for lottery draw

Kimi for assisting in writing code

`tkinter` — Python interface to Tcl/Tk

Source code: Lib/tkinter/__init__.py

The `tkinter` package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and `tkinter` are available on most Unix platforms, including macOS, as well as on Windows systems.

Running `python -m tkinter` from the command line should open a window demonstrating a simple Tk interface, letting you know that `tkinter` is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded. See the source code for the `_tkinter` module for more information about supported versions.

Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.

TKinter library used for graphic interface

Method

Import Libraries: The code starts by importing the Tkinter library for GUI components and the random library for selecting names randomly.

Global Variable: A global variable `running` is defined to control the loop that changes the displayed name.

Name List: A list `names` contains 43 names, which are the participants in the lottery.

LotteryStart Function: This function sets the `running` variable to `True` and defines an inner function `update` that randomly selects a name from the `names` list and updates the text of the label `lab2` with the chosen name. The `update` function is called recursively every 1 millisecond to keep changing the name as long as `running` is `True`.

LotteryStop Function: This function sets the `running` variable to `False`, stopping the name change loop. It then randomly selects a final name and sets the text of `lab2` to display this name, effectively "freezing" the display on a random name.

Method

GUI Setup: The main window is created with a title and a specific geometry. Two labels are created: `lab1` to display "Lottery !" and `lab2` to display the randomly selected names. The labels are placed on the grid layout with specific row and column configurations.

Buttons: Two buttons are created, `btn_start` and `btn_stop`, which when clicked, call the `LotteryStart` and `LotteryStop` functions, respectively. These buttons are also placed on the grid layout.

Grid Layout Configuration: The grid layout is configured to have two columns, each with equal weight, ensuring that the buttons and labels are distributed evenly.

Main Loop: The `mainloop` function is called on the window object to start the Tkinter event loop, which waits for user interactions and updates the GUI accordingly.

Method

```python
import tkinter as tk
import random

# Global variable to control the loop of name changes
running = False

# List of all people's names
names = [
    [illegible]
]
```

Method

```
def LotteryStart():
    global running
    running = True
    def update():
        if running:  # Only update the name when running is True
            name = random.choice(names)
            lab2.config(text=name, fg="#192367")
            window.after(1, update)  # Update the name every 1
millisecond
    update()  # Start the name update

def LotteryStop():
    global running
    running = False
    name = random.choice(names)
    lab2.config(text=name, fg="#192367")  # Freeze on a random
name when stopped
```

Method

```
# Create the main window
window = tk.Tk()
window.title("Lottery Mini Program")
window.geometry("800x600")

# Create a label to display the lottery result
lab1 = tk.Label(window, text="Lottery !", foreground="#4758d2",
font=("Arial", 32))
lab1.grid(row=0, column=0, columnspan=2, pady=20)

lab2 = tk.Label(window, font=("Arial", 150))
lab2.grid(row=1, column=0, columnspan=2, pady=30)
```

Method

```
# Create a button that calls the Lottery function when clicked
btn_start = tk.Button(window, text="Start", font=("Arial", 22),
foreground="white",
                      background="#55c08f", command=LotteryStart)
btn_start.grid(row=3, column=0, sticky="ew", padx=10, pady=20)

btn_stop = tk.Button(window, text="Stop", font=("Arial", 22),
foreground="white",
                      background="#c05555", command=LotteryStop)
btn_stop.grid(row=3, column=1, sticky="ew", padx=10, pady=20)

# Configure the grid layout to have two columns
window.grid_columnconfigure(0, weight=1)
window.grid_columnconfigure(1, weight=1)

# Enter the main loop
window.mainloop()
```

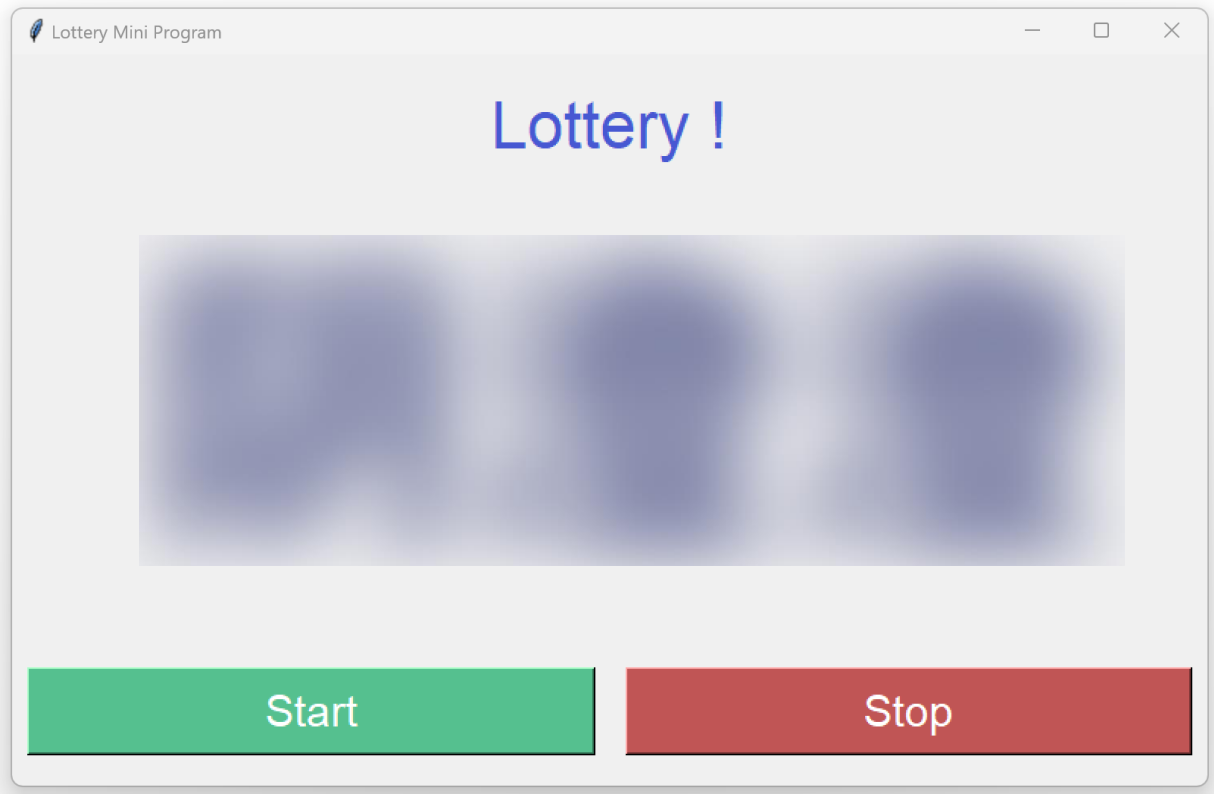
1. Introduction

2. Method

3. Results

4. Discussion

Result



Click “Start” to begin a **random scrolling of names** and “Stop” to **freeze on a randomly selected name.**

1. Introduction

2. Method

3. Results

4. Discussion

Discussion

Conclusion: A GUI application has been established to do a lottery draw for students making presentations.

Discussion: The function is stable, and a new feature can be added to prevent already selected names from reappearing in the future.

Thank you !

Probability bless you !