

**Carrera de Buses en Tkinter con Python**

Este proyecto es un juego de carrera de buses desarrollado en Python utilizando la biblioteca tkinter para la interfaz gráfica. A continuación, se describen las importaciones, configuraciones, y funciones utilizadas en el código.

## 1. Importaciones

* **tkinter**: Para crear la interfaz gráfica de la aplicación.
* **random**: Para generar velocidades aleatorias que mueven los buses.
* **messagebox**: Para mostrar mensajes emergentes al usuario.

## 2. Configuración

* **Tamaño de la ventana**: Se define el ancho y alto de la ventana principal.
* **Línea de meta**: Se establece la posición de la línea de meta en el lienzo.

## 3. Ventana Principal

* Se crea la ventana principal usando tk.Tk() y se establece un título para la aplicación.

## 4. Canvas

* Se crea un lienzo negro (canvas) donde se dibujarán los elementos gráficos del juego.

## 5. Arte ASCII

* Se define una representación del bus utilizando arte ASCII, que se mostrará en la interfaz.

## 6. Entradas de Usuario

* Se crean cuadros de texto (Entry) para que el usuario ingrese los nombres de dos países, que representarán a los buses en la carrera.

## 7. Funciones

1. **draw\_bus(x, y, country\_name)**: Dibuja el bus en la posición (x, y) y muestra el nombre del país encima.
2. **draw\_track()**: Dibuja la pista de carrera y la línea de meta. Incluye carriles y líneas de separación.
3. **move\_buses()**: Mueve los buses aleatoriamente a lo largo de la pista y determina el ganador de la carrera. Si un bus llega a la línea de meta, se muestra el ganador.
4. **start\_race()**: Inicia la carrera al verificar que ambos nombres de países estén ingresados. Si no, muestra un mensaje de advertencia.
5. **reset\_race()**: Reinicia la carrera, restableciendo las posiciones de los buses y limpiando el lienzo.

## 8. Botones

* Se crean botones para iniciar y reiniciar la carrera, que ejecutan las funciones correspondientes al ser presionados.

## 9. Ejecutar

* Finalmente, se inicia el bucle principal de la aplicación con root.mainloop(), que mantiene la ventana abierta y en espera de interacciones del usuario.

**Crea un archivo llamado race.py**

import tkinter as tk

import random

from tkinter import messagebox

# Configuración inicial de la ventana

WIDTH = 600

HEIGHT = 400

FINISH\_LINE = WIDTH - 50

# Crear la ventana principal

root = tk.Tk()

root.title("Carrera de Buses")

# Cambiar el color de fondo del canvas a negro

canvas = tk.Canvas(root, width=WIDTH, height=HEIGHT, bg="black")

canvas.pack()

# Arte ASCII para el bus

bus\_ascii = [

" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",

" |,----.,----.,----.,--.\\",

" || || || || \\\\",

" |`----'`----'|----||----\\`.",

" [ | -||- \_\_|( |",

" [ ,--. |\_\_\_\_||.--. |",

" =-(( `))-----------(( `))=="

]

# Cuadros de texto para ingresar los nombres de los países

country1\_label = tk.Label(root, text="País 1:", bg="black", fg="white")

country1\_label.pack(pady=(10, 0)) # Espacio vertical arriba

country1\_entry = tk.Entry(root)

country1\_entry.pack(pady=(0, 10)) # Espacio vertical abajo

country2\_label = tk.Label(root, text="País 2:", bg="black", fg="white")

country2\_label.pack(pady=(10, 0)) # Espacio vertical arriba

country2\_entry = tk.Entry(root)

country2\_entry.pack(pady=(0, 10)) # Espacio vertical abajo

def draw\_bus(x, y, country\_name):

# Dibujar el nombre del país encima del bus

canvas.create\_text(x + 10, y - 20, text=country\_name,

anchor="nw", font=("Helvetica", 14), fill="white")

for i, line in enumerate(bus\_ascii):

canvas.create\_text(x, y + i \* 15, text=line,

anchor="nw", font=("Courier", 10), fill="white")

def draw\_track():

# Dibujar la pista

lane\_y\_positions = [50, 200] # Separar los carriles

lane\_height = 50

for lane\_y in lane\_y\_positions:

# Dibujar líneas de los carriles

canvas.create\_line(0, lane\_y, WIDTH, lane\_y, fill="white", width=2)

canvas.create\_line(0, lane\_y + lane\_height, WIDTH,

lane\_y + lane\_height, fill="white", width=2)

# Dibujar líneas de separación en la pista

for i in range(0, WIDTH, 40):

canvas.create\_line(i, lane\_y + lane\_height / 2, i +

20, lane\_y + lane\_height / 2, fill="black", width=2)

# Dibujar la línea de meta

canvas.create\_line(FINISH\_LINE, 0, FINISH\_LINE,

HEIGHT, fill="red", width=4)

# Posiciones iniciales

bus1\_x, bus1\_y = 0, 50

bus2\_x, bus2\_y = 0, 200 # Cambiar la posición vertical para más espacio

# Función para mover los buses

def move\_buses():

global bus1\_x, bus2\_x

# Mover buses con velocidad aleatoria

bus1\_x += random.randint(1, 10)

bus2\_x += random.randint(1, 10)

# Limpiar canvas, dibujar la pista y redibujar los buses

canvas.delete("all")

draw\_track()

draw\_bus(bus1\_x, bus1\_y, country1\_entry.get())

draw\_bus(bus2\_x, bus2\_y, country2\_entry.get())

if bus1\_x >= FINISH\_LINE and bus2\_x >= FINISH\_LINE:

winner = "¡Empate!"

elif bus1\_x >= FINISH\_LINE:

winner = f"¡{country1\_entry.get()} gana!"

elif bus2\_x >= FINISH\_LINE:

winner = f"¡{country2\_entry.get()} gana!"

else:

root.after(100, move\_buses) # Llamar nuevamente después de 100 ms

return

# Mostrar el resultado con separación

canvas.create\_text(WIDTH / 2, HEIGHT - 100, text=winner,

# Ajustar la posición vertical

font=('Helvetica', 24), fill="white")

# Función para iniciar la carrera

def start\_race():

global bus1\_x, bus2\_x

country1 = country1\_entry.get().strip()

country2 = country2\_entry.get().strip()

if not country1 or not country2:

messagebox.showwarning(

"Advertencia", "Por favor ingrese nombres para ambos países.")

return

bus1\_x, bus2\_x = 0, 0 # Reiniciar posiciones

move\_buses()

# Función para reiniciar la carrera

def reset\_race():

global bus1\_x, bus2\_x

bus1\_x, bus2\_x = 0, 0 # Reiniciar posiciones

canvas.delete("all") # Limpiar el canvas

draw\_track()

draw\_bus(bus1\_x, bus1\_y, country1\_entry.get())

draw\_bus(bus2\_x, bus2\_y, country2\_entry.get())

# Crear un Frame para organizar los botones

button\_frame = tk.Frame(root)

button\_frame.pack(pady=10) # Agregar un poco de espacio vertical

# Botón para iniciar la carrera

start\_button = tk.Button(

button\_frame, text="Iniciar Carrera", command=start\_race)

# Agregar espacio horizontal entre los botones

start\_button.pack(side="left", padx=10)

# Botón para reiniciar la carrera

reset\_button = tk.Button(

button\_frame, text="Reiniciar Carrera", command=reset\_race)

reset\_button.pack(side="left", padx=10)

# Dibujar la pista inicial

draw\_track()

# Ejecutar la aplicación

root.mainloop()

Ya puedes ejecutarlo usando python race.py.