

Manual de Instalación y Uso del Código: Simulación de la Trayectoria de una Bomba Lanzada desde un Avión

1. Requisitos del Sistema

Antes de instalar y ejecutar el código, asegúrese de que su sistema cumpla con los siguientes requisitos:

- **Sistema Operativo:** Windows, macOS o Linux
- **Python:** Versiones 3.6 o superiores.
- **Bibliotecas necesarias:**
 - **Tkinter:** Para la creación de la interfaz gráfica de usuario (generalmente preinstalada con Python).
 - **Matplotlib:** Para la visualización de gráficos y animación.
 - **NumPy:** Para realizar los cálculos matemáticos necesarios.
 - **Pillow (PIL):** Para manejar imágenes en la interfaz gráfica.

2. Instalación de Dependencias

Si no tienes las bibliotecas necesarias instaladas en tu entorno de Python, sigue estos pasos:

Paso 1: Instalación de Python

Si aún no tienes Python instalado, puedes descargar la última versión desde el sitio web oficial:

[Descargar Python](#)

Paso 2: Instalación de las bibliotecas requeridas

Puedes instalar las bibliotecas necesarias utilizando pip (el gestor de paquetes de Python). Abre una terminal o línea de comandos y ejecuta los siguientes comandos:

```
pip install matplotlib numpy pillow
```

Nota: Tkinter generalmente se instala por defecto con Python, pero si no lo tienes, puedes instalarlo de la siguiente manera:

- En sistemas Linux: `sudo apt-get install python3-tk`
- En sistemas Windows/macOS, Tkinter debería estar incluido automáticamente.

3. Configuración del Entorno de Trabajo

Asegúrate de tener el código fuente del programa guardado en una carpeta de tu elección. Si has descargado el código desde algún repositorio o fuente, asegúrate de descomprimirlo o guardarlo correctamente.

El archivo principal del programa debe tener una extensión `.py`, por ejemplo, `simulacion_bomba.py`.

4. Ejecución del Código

Para ejecutar el código, sigue estos pasos:

Paso 1: Abrir la terminal o línea de comandos

- En **Windows**, abre la "Command Prompt" o "PowerShell".
- En **macOS o Linux**, abre la terminal.

Paso 2: Navegar a la carpeta donde está el código

Usa el comando `cd` para navegar hasta la carpeta donde tienes guardado el archivo `.py`. Ejemplo:

cd /ruta/a/tu/codigo

Paso 3: Ejecutar el archivo

Una vez en la carpeta correcta, ejecuta el código usando el siguiente comando:

```
python simulacion_bomba.py
```

Paso 4: Interfaz gráfica

Al ejecutar el código, aparecerá una ventana con una interfaz gráfica de usuario (GUI) construida con Tkinter. A través de esta interfaz podrás ingresar los parámetros de entrada como la velocidad del avión y la altura desde la cual se lanza la bomba.

5. Uso del Programa

Paso 1: Ingreso de parámetros

En la interfaz gráfica, se te pedirá que ingreses los siguientes parámetros:

1. **Velocidad del avión (V):** La velocidad a la que el avión se desplaza horizontalmente (en metros por segundo).
2. **Altura del avión (Ha):** La altura desde la que el avión lanza la bomba (en metros).
3. **Dimensiones del cañón:**
 - **Ancho superior del cañón.**
 - **Profundidad del cañón.**
 - **Ancho de la base del cañón.**

Paso 2: Validación de entrada

El programa validará que los valores introducidos sean correctos. Si alguno de los valores no es numérico o es menor o igual a cero, se mostrará un mensaje de error y el programa no continuará hasta que se corrija.

Paso 3: Iniciar la simulación

Una vez que hayas ingresado todos los parámetros, haz clic en el botón "Iniciar Simulación". El programa comenzará a calcular la trayectoria de la bomba y mostrará una animación en tiempo real.

Paso 4: Observación de la simulación

La animación mostrará lo siguiente:

- El avión se mueve horizontalmente a la velocidad indicada.
- En el momento que el avión suelta la bomba, esta comienza a seguir una trayectoria parabólica influenciada por la gravedad.
- El cañón (representado como un trapecio invertido) se mostrará en el gráfico.

Paso 5: Finalización de la simulación

La simulación finalizará cuando la bomba toque el suelo (cuando la posición vertical de la bomba sea cero). En ese momento, la animación se detendrá.

6. Manejo de Excepciones

El programa incluye manejo de excepciones que asegura:

- Los valores introducidos son numéricos y mayores a cero.

- Si un valor inválido es ingresado, el programa muestra un mensaje de error y detiene la simulación hasta que se ingrese un valor válido.

7. Posibles Mejoras Futuras

Aunque la simulación proporciona un modelo físico sólido, se podrían considerar mejoras como:

- Agregar la resistencia del aire en la trayectoria de la bomba para hacer el modelo más realista.
- Implementar un control para modificar la dirección del viento, lo que afectaría el movimiento de la bomba.

8. Solución de Problemas Comunes

- **No se visualiza la interfaz gráfica:** Asegúrate de tener instalada la biblioteca Tkinter y de estar ejecutando el código con una versión compatible de Python.
- **Error en la validación de entradas:** Verifica que todos los valores ingresados sean numéricos y positivos. Si el error persiste, asegúrate de que las entradas sean del tipo correcto.
- **Problemas con las bibliotecas:** Si alguna biblioteca no se instala correctamente, verifica tu conexión a Internet o ejecuta el comando de instalación nuevamente.