Suplementary Material: The source code for the prototype

```
from tkinter import*
import math
import os
from PIL import ImageTk,ImageColor, Image
from pygame import mixer

#Importar imágenes  #almacenamiento rutas de carpetas
#Carga de directorio/carpeta
#Carpeta principal
carpeta_principal=os.path.dirname(__file__) #para que la ruta se haga di-
námica, y se puedan abrir en cualquier os
print(carpeta_principal)

#Carpeta e imagenes
contenedor_imagen=os.path.join(carpeta_principal, "Imagenes") #join sirve
para concatenar las rutas

#Iniciar mixer en pygame
mixer.init()

#Cargar sonido con condicional
try:
    sonido_explosion=mixer.Sound("explosion.wav")
except:
    print("El archivo no existe o no se encuentra")
    sonido_explosion=None

ventana=Tk()
ventana.title("PROGRAMA 1" )
ventana.geometry("650x600")
#ventana.iconbitmap("canon.ico")

#Cargar imágen
Tanque=ImageTk.PhotoImage(Image.open(os.path.join(contenedor_imagen, "Tan-
que.jpeg"))
.resize((140,100)))

Bandera=ImageTk.PhotoImage(Image.open(os.path.join(contenedor_imagen,
"bandera.jpg"))
.resize((50,50)))

Fuego=ImageTk.PhotoImage(Image.open(os.path.join(contenedor_imagen, "lla-
mas.jpg"))
.resize((50,50)))
```

```
lienzo=Canvas(ventana, bg="white")

lienzo.create_rectangle(0, 540,600, 600, tags="linea", fill="green")
def nueva_linea(event):

    lienzo.delete("linea")

    lienzo.create_rectangle(0, event.height - 60, event.width,
event.height, tags="linea", fill="green")

lienzo.bind("<Configure>", nueva_linea)

lienzo.create_image(5,570,image=Tanque,anchor="sw")
lienzo.create_image(600,535,image=Bandera,anchor="se", tag="bandera")

# Posic_ini = 20
x1_ini = 135
y1_ini = 500
x2_ini = 145
y2_ini = 510

#velociion inicial
vx_ini = 50
vy_ini =50
t = 0

# incremento en el tiempo
dt = 0.1

#incremento en x
delta_x = vx_ini*dt
# Posicion inicial en y
y_ante = (-1)*(y2_ini + vy_ini*t - 5*t*t)

movimiento_id=None

def circulo():
    global vx_ini, vy_ini, delta_x, y_ante, t, circulo1


    try:
        Vo = float(vel_ini_x_modulo.get())
        angulo = float(angulo_disparo.get())

        #Angulo a radianes para que poder calcular las componentes, python
trabaja en radianes
        radianes=math.radians(angulo)
        vx_ini=Vo*math.cos(radianes)
```

```
        vy_ini=Vo*math.sin(radianes)

    except ValueError:
        print("Error: Ingresa valores numéricos.")
        return  # Sale de la función si no hay valores válidos

    delta_x = vx_ini * dt
    t = 0
    y_ante = (-1) * (y2_ini + vy_ini * t - 5 * t * t)


    circulo1 = lienzo.create_oval(x1_ini, y1_ini, x2_ini, y2_ini,
fill="green")
    print("\n\n\n")
    movimiento_id = ventana.after(50, movimiento)
    t = 0
    # Posicion inicial en y
    y_ante = (-1)*(y2_ini + vy_ini*t - 5*t*t)


def movimiento():
    global circulo1, movimiento_id, y_ante, t, Bandera

    #Para mover el circulo
    y = (-1)*(y2_ini + vy_ini*t - 5*t*t)
    delta_y =  y - y_ante
    lienzo.move(circulo1, delta_x, delta_y)

    #Actualización de coordenadas
    x1, y1, x2, y2=lienzo.coords(circulo1)
    print("las coordenadas son: ", x2, y2)
    print("el incremento: ", delta_x, delta_y, t, "\n")

    #Verificar que la bandera existe
    bandera_items=lienzo.find_withtag("bandera")

    if bandera_items:
        x1_bandera, y1_bandera = lienzo.coords(bandera_items[0])

        if abs(x1 - x1_bandera )<50 and abs(y1 -  y1_bandera) <50:
            lienzo.delete("bandera")

            if Fuego:
                lienzo.create_image(x1_bandera, y1_bandera, image=Fuego,
anchor="se", tag="explosion")

            if sonido_explosion:
                sonido_explosion.play()
```

```
            ventana.after_cancel(movimiento_id)
            return

    t = t+dt
    y_ante = y

    if  y2 >540:
        lienzo.delete(circulo1)
        ventana.after_cancel(movimiento_id)
    else:
        movimiento_id=ventana.after(50, movimiento)

def X():
    ventana.destroy()

MiLabel_x_i=Label(lienzo, text="Introduce la velocidad inicial (Km/h): " )
MiLabel_x_i.place(x=20, y=10)
vel_ini_x_modulo=Entry(lienzo)
vel_ini_x_modulo.place(x=250, y=10)

MiLabel_y_i=Label(lienzo, text="Introduce el ángulo de disparo (grados):
")
MiLabel_y_i.place(x=20, y=40)
angulo_disparo=Entry(lienzo)
angulo_disparo.place(x=250, y=40)

B_Start=Button(ventana, text="Start", font="Roboto", padx=8, pady=5, com-
mand=circulo, relief="sunken")
B_Start.place(relx=0.25, rely=0.95, anchor=CENTER)

B_Cerrar=Button(ventana, text="Cerrar",font="Roboto", padx=8, pady=5, com-
mand=X,relief="sunken")
B_Cerrar.place(relx=0.75, rely=0.95, anchor=CENTER)

lienzo.pack(fill=BOTH, expand=True)
ventana.mainloop()
```