

INFORME INTRODUCCIÓN A

LA PROGRAMACIÓN:

TAREA 2

Nombre: Pía Contreras Guerrero

Rut: 19.840.187-0

Sección: 10

ÍNDICE

	Página
1) Diseño de la solución	3-7
2) Lista de funciones	8
3) Ejemplos de prueba	9-12
4) Experiencia	13

DISEÑO DE LA SOLUCIÓN

Para comenzar a programar el juego del ahorcado lo primero que hice fue importar módulos, entre los cuales se encuentran “random” para poder elegir una palabra al azar el en texto “palabra entregada por la profesora, el módulo “sys” para poder salir del juego, “Tkinter” para poder dibujar y abrir nuevas ventanas, y csv para poder leer el archivo de texto.

```
import random
from Tkinter import *
import sys
import csv
```

Luego de creados los módulos cree la ventana donde irá el juego, le puse de nombre “El ahorcado”, junto con ello cree un frame llamo “frameprincipal” que “contiene” la ventana.

```
ventana =Tk()
ventana.title ("El ahorcado")
frameprincipal=Frame(ventana)
frameprincipal.pack()
```

Para poder dibujar cree una variable llamada canvas la cual me permitiría hacer las líneas y los ovalos en la ventana vacía creada.

```
canvas=Canvas(ventana, width=600, height=600, bg="white")
```

Para que la ventana estuviese lista, tuve que crear un botón que diera la opción de empezar de nuevo, además de un lugar donde se ingresaran las letras, un Label que me permitiera que las letras se mostraran de nuevo, y cerrar la ventana para que luego corriera.

```

titulo=Label(frameprincipal, text="El ahorcado")
titulo.pack(side=LEFT)

etiqueta=Label(ventana, text="Ingrese letra:")
etiqueta.pack()

texto=StringVar()
letra=Entry(ventana, textvariable=texto)
letra.bind("<Return>", letras)
letra.pack()

Palabra=Label(ventana, text="Palabra: " + rayitas())
Palabra.pack()

letrasJugadas=Label(ventana, text="Letras Jugadas: ")
letrasJugadas.pack()

LetrasJugadas=Label(ventana)
LetrasJugadas.pack()

boton=Button(ventana, text="Empezar de nuevo", command=jugarDeNuevo)
boton.pack()

ventana.mainloop()

```

Para el dibujo tuve que crear con la variable canvas líneas y ovalos y dar la forma a un colgado, el código es el siguiente:

```

from Tkinter import *

ventana=Tk()

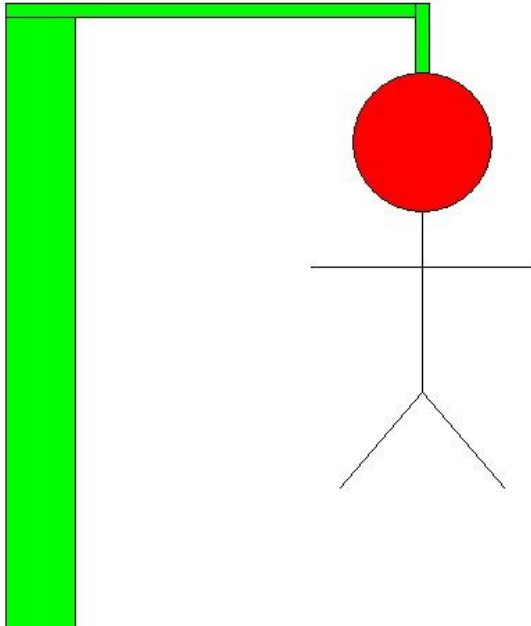
canvas=Canvas(ventana, width=800, height=1000)
canvas.pack()

canvas.create_line(400, 200, 400, 330, fill="black")
canvas.create_line(400, 330, 340, 400, fill="black")
canvas.create_line(400, 330, 460, 400, fill="black")
canvas.create_oval(350, 100, 450, 200, fill="red")
canvas.create_line(320, 240, 480, 240, fill="black")
canvas.create_rectangle(100, 50, 150, 600, fill="green")
canvas.create_rectangle(100,50, 400, 60, fill="green")
canvas.create_rectangle(395, 50, 405, 100, fill="green")

ventana.mainloop()

```

El cual se ve de la siguiente manera en la ventana:



Luego de creado el interfaz del ahorcado abrí el archivo donde estaban contenidas las palabras para así luego se pudieran elegir aleatoriamente para que el juego se lleve a cabo, además con csv se podían leer línea a línea, cree una lista para que cada elemento que se leyera, se guardara en esta lista, con ello cree un contador que luego permitiría que el elemento 0 de la lista tomara el valor de la primera palabra de la fila.

```
listaPalabras=[0]*10
p=0
palabra= open("palabras.txt")
linea=csv.reader(palabra, delimiter=" ")
p=palabra.read()
e=random.choice(p)
for fila in linea:
    listaPalabras[0]=fila[0]
    p+=1
    random.choice(listaPalabra)
    i=[0]
    for i in range(0, len(listaPalabras)):
        print listaPalabras[i]
palabra.close()
```

Para que los buttons y labels aparecieran de manera correcta tuve que crear tres importantes funciones:

```
def Letras(n):
    letra=texto.get()
    letrasJugadas.config(text=letra)
    return

def jugarDeNuevo():
    etiqueta.config(text="")
    return

def letras(x):
    letra=texto.get()
    letrasJugadas.config(text=letra)
    return

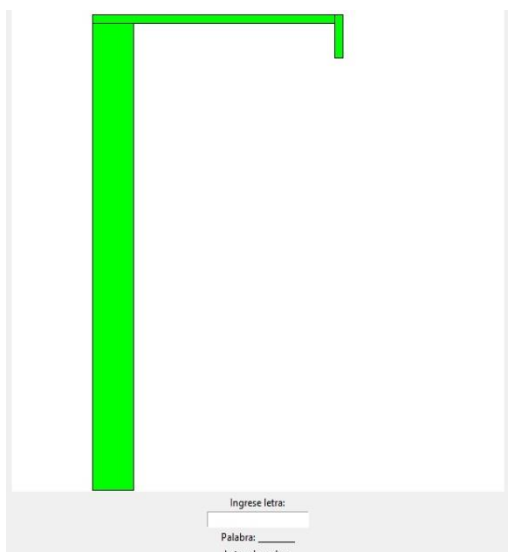
def rayitas():
    return " _ "*cantidad
```

Hasta aquí el juego aún no envía palabras al azar, pese a estar guardados en una lista así que decidí escribir la lista a mano, y de aquí usar random.choice() de la siguiente manera:

```
listaPalabras=["laberinto", "cocodrilo", "nandu", "zapatilla", "bicicleta", "armadillo", "aconcagua", "sencillo", "zorzal", "compromiso"]
palabraElegida=random.choice(listaPalabras)
tupalabra=""
vidas=5
print palabraElegida
print len(palabraElegida)
print [palabraElegida]

i=0
while i<len(palabraElegida):
    letra=palabraElegida[i]
    i=i+1
    print letra
cantidad=len(palabraElegida)
```

Hasta este momento la ventana se veía de esta manera:



Y la parte de donde el programa envía la palabra al azar se ve de la siguiente forma:

```
labyrintho
9
['labyrintho']
l
a
b
e
r
i
n
t
o
```

Para poder dibujar el ahorcado, cree una función que a medida que se fueran cumpliendo ciertas condiciones se fuese dibujando el ahorcado:

```
#dibujo_muneco: num(0-4)->dibujo
#permite que a medida que se vaya cumpliendo ciertas condiciones
#se vaya dibujando el ahorcado
#ejemplo:dibujo_muneco(4) debería dibujar la cabeza
def dibujo_muneco(x):
    if x==4:
        canvas.create_oval(350, 100, 450, 200, fill="red")
    if x==3:
        canvas.create_line(320, 240, 480, 240, fill="black")
    if x==2:
        canvas.create_line(400, 330, 460, 400, fill="black")
    if x==1:
        canvas.create_line(400, 330, 340, 400, fill="black")
    if x==0:
        canvas.create_line(400, 200, 400, 330, fill="black")
    return
```

Para que fuese leyendo letra a letra y fuese identificando si la letra ingresada estaba correcta o no utilicé un ciclo for que identifique que la letra ingresada esté en la palabra elegida al azar, el cual es de la siguiente manera:

```
for letra in palabraElegida:
    if(letra in palabraElegida==True):
        letra in rayitas
    else:
        vidas-=1
        dibujo_muneco(vidas)
```

LISTA DE FUNCIONES

Algunas de las funciones utilizadas fueron:

1) rayitas:

```
#rayitas:str->"_"
#rayitas crea la cantidad de lineas por letra de cada palabraElegida(al azar)
#ejemplo:rayitas(nandu) debería devolver _ _ _ _ _
def rayitas():
    return "_"*cantidad
```

2) jugarDeNuevo:

```
#permite comenzar el juego de nuevo
def jugarDeNuevo():
    etiqueta.config(text="")
    return
```

3) letras:

```
#ayuda al momento de poder ingresar la letra
def letras(x):
    letra=texto.get()
    letrasJugadas.config(text=letra)
    return
```

4) dibujo_muneco:

```
#dibujo_muneco: num(0-4)->dibujo
#permite que a medida que se vaya cumpliendo ciertas condiciones
#se vaya dibujando el ahorcado
#ejemplo:dibujo_muneco(4) debería dibujar la cabeza
def dibujo_muneco(x):
    if x==4:
        canvas.create_oval(350, 100, 450, 200, fill="red")
    if x==3:
        canvas.create_line(320, 240, 480, 240, fill="black")
    if x==2:
        canvas.create_line(400, 330, 460, 400, fill="black")
    if x==1:
        canvas.create_line(400, 330, 340, 400, fill="black")
    if x==0:
        canvas.create_line(400, 200, 400, 330, fill="black")
    return
```


EJEMPLOS DE PRUEBA

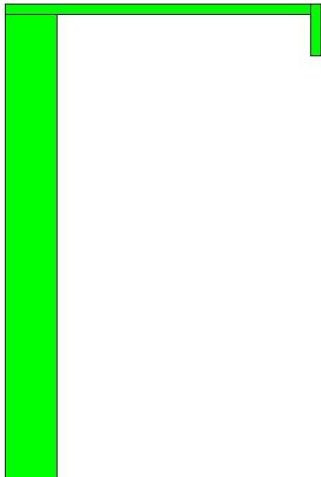
1) Caso 1: Gana en todas las letras

Cuando el jugador gana, no se dibuja el muñeco, aparece un mensaje que dice “HAS GANADO” y abajo (no se alcanza a ver en la foto) aparecen las letras jugadas y la palabra jugada.

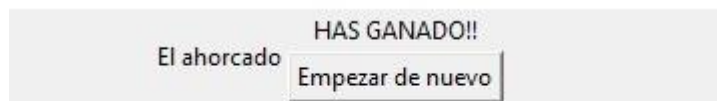
La palabra al azar se ve de la siguiente forma:

```
nandu  
5  
['nandu']  
n  
a  
n  
d  
u
```

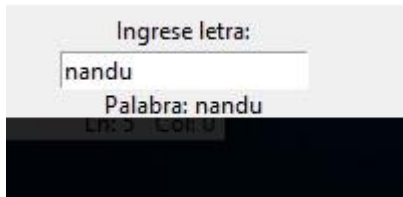
El dibujo en la ventana se ve así:



El frame principal se percibe de la siguiente forma:



Y en el label donde se ingresan las letras aparece así (no se logra percibir las letras ingresadas abajo):

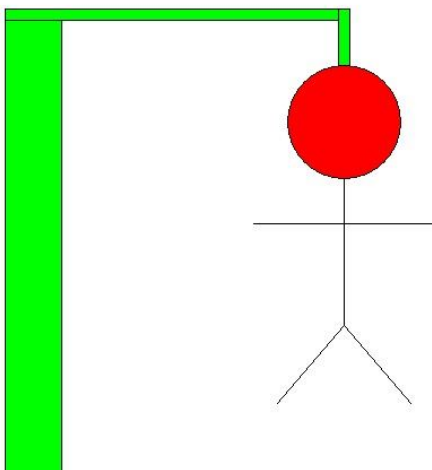


2) Caso 2: Se equivoca en todas las letras

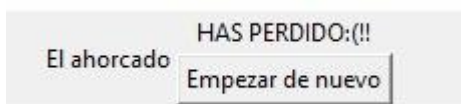
La palabra al azar se ve de la siguiente forma:

```
nandu  
5  
['nandu']  
n  
a  
n  
d  
u
```

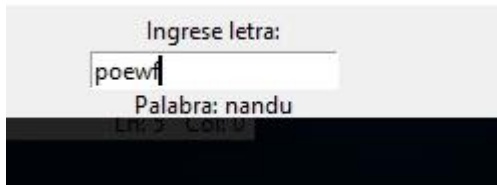
El dibujo cuando se pierde, en este caso cuando se equivoca en todas las letras luce así en la ventana:



El frame principal cuando se pierde muestra un mensaje que dice “HAS PERDIDO😞”:



Y el “entry”, es decir donde se pueden introducir las letras, se ve de la siguiente manera (donde se muestran las letras ingresadas aparece abajo y no se logra percibir):

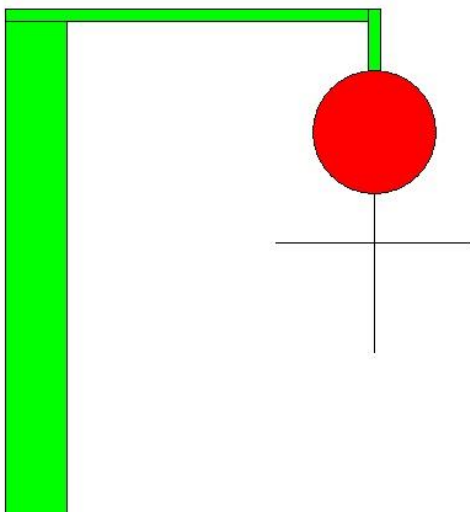


3) Caso 3: Se equivoca en algunas letras

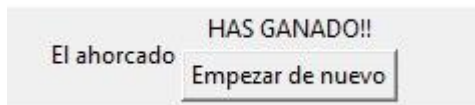
La letra al azar se ve de la siguiente forma:

```
zapatilla
9
['zapatilla']
z
a
p
a
t
i
l
l
a
```

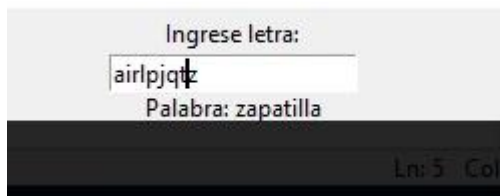
El dibujo al equivocarse en tres ocasiones, dibuja tres partes del cuerpo, la cabeza, los brazos y el cuerpo y se ve así:



El frame principal muestra el mensaje “HAS GANADO” ya que pese a equivocarse tres veces, pudo acertar en la palabra sin que las vidas llegaran a cero:



Por ultimo donde se ven e introducen las letras se ve de la siguiente forma (las letras ingresadas están muy abajo y no se alcanzan a ver):



EXPERIENCIA

En mi experiencia programando el juego del ahorcado en Python me pareció que fue una instancia de mucho aprendizaje, donde se pudo usar todo lo aprendido.

Al principio fue un poco difícil como abordar los problemas pero luego, investigando por internet, preguntando y probando, por ejemplo como dibujar, se hace más amigable.

En lo personal me costó el tema de entender las coordenadas al dibujar, además de entender que hacían los frame en particular.

Pude que el juego eligiera palabras al azar de una lista escrita en Python, pero no pude que la eligiera abriendo el archivo con `x=(open"...")`.

Hacer el juego del ahorcado fue un desafío grande para mí, entendí mucho más como funciona el for y como crear ventanas y dentro de ellas botones, lugares especiales donde se pudieran ingresar caracteres, etc.