

Licenciatura en Sistemas

Trabajos Práctico

TutiFruti UNGS

Introducción a la Programación

(Primer semestre, año 2020)

El siguiente trabajo presenta el juego TutiFruti. El jugador debe escribir una palabra que comience con una letra dada al azar correspondiente al ítem indicado, ya sea colores, países, etc. La idea es que los jugadores mejoren sus tiempos con este programa.

Integrantes: Gonzalo Galarza galarzajota71@gmail.com
 Arroyo Guadalupe guadalupenicol102@gmail.com

1.Introducción

Este trabajo tuvo como eje conductor, comprender las diferentes funciones necesarias para que el programa principal del juego TutiFruti en PyGame se pueda ejecutar correctamente. Utilizamos como marco teórico además de los módulos aportados por la cátedra, algún apoyo virtual.

En una primera etapa establecimos el problema planteado según cada función y el requerimiento específico, luego analizamos tanto las posibilidades como las dificultades.

Una vez que identificamos la solución más conveniente realizamos las pruebas necesarias. Logrando que el programa se ejecute.

2.Desarrollo

En un principio teníamos que completar la función **unaAlAzar** que toma como parámetro a la lista **abc**. Buscamos como elegir un elemento al azar de la lista compuesta por el abecedario. Teníamos dos opciones: usar un **bucle for**, o bien utilizar una función propia de Python. Finalmente decidimos hacer el código más simple, usando la función **random.choice()** tomando como parámetro a la lista **abc**, e igualándola a la variable **azar**. Devolviendo la letra elegida, como se muestra en la imagen.

```
from configuracion import *
from principal import *
import math
import random

def unaAlAzar(abc): #La funcion unaAlAzar debe elegir una letra al alazar de la lista abc del programa princial
    azar=random.choice(abc) #Elige una letra al azar de la lista abc
    return(azar) #Devuelve esa letra
```

Seguimos con la función **esCorrecta**, la cual recibe como parámetros **palabraUsuario**, **letra**, **item**, **items** y **listaDeTodo**, esta función es la encargada de retornar los puntajes. Intentamos resolverla varias veces de diversas formas, pero siempre terminábamos con un error o devolvía puntaje 0.

```
def esCorrecta(palabraUsuario, letra, item, items, listaDeTodo):# Esta funcion se encarga de retornar los puntajes
    if palabraUsuario[0]==letra: #se fija que la primer letra de la palabra del usuario sea igual a letra al azar
        i=0 # Inicializo la variable i que tomará los valores de la lista items en cada iteracion
        while i < len(items): # Mientras i sea menor a la longitud de la lista items
            if items[i]==item: # Se fija si la lista items en la posicion i es igual al item correspondiente
                if palabraUsuario in listaDeTodo[i]: # Si la palabra del usuario está en el item correcto
                    return(10) # Devuelve 10 puntos
                else: ## Parte mal ubicada en el codigo
                    return(0) # Sino devuelve 0 puntos
            i=i+1 # Aumento de variable
        else:
            return(-5) # Si la primer letra de la palabra del usuario no es igual a la letra al azar resta 5 puntos
```

Repasamos la estructura, hasta que encontramos que una parte del código estaba mal ubicada dentro de la función, como podemos ver en la captura. Al modificarla el programa comenzó a sumar y descontar los puntos correctamente.

```
def esCorrecta(palabraUsuario, letra, item, items, listaDeTodo): # Esta funcion se encarga de retornar los puntajes
    if palabraUsuario[0]==letra: #se fija que la primer letra de la palabra del usuario sea igual a letra al azar

        i=0 # Inicializo la variable i que tomará los valores de la lista items en cada iteracion
        while i < len(items): # Mientras i sea menor a la longitud de la lista items

            if items[i]==item: # Se fija si la lista items en la posicion i es igual al item correspondiente
                if palabraUsuario in listaDeTodo[i]: # Si la palabra del usuario está en el item correcto
                    return(10) # Devuelve 10 puntos

            else:
                return(0) # Sino devuelve 0 puntos
            i=i+1 # Aumento de variable
    else:
        return(-5) # Si la primer letra de la palabra del usuario no es igual a la letra al azar resta 5 puntos
```

Después analizamos la función **juegaCompu**, ésta recibe como parametros **letraAzar** y **listaDeTodo**, debiendo retornar una lista con una palabra al azar de cada item que comience con la letra estipulada. Un error que tuvimos fue al incrementar la variable de control en el ciclo while, la cual no aumentaba ya que estaba mal ubicada dentro del programa.

Fuera de esto la función **juegaCompu** no tuvo mayores complicaciones.

```
def juegaCompu(letraAzar, listaDeTodo): # JuegaCompu retorna una lista con una palabra al azar de cada item que empieze con la letra al azar
    salida=[] # Es donde se guardarán las palabras de cada item que comiencen con la letra indicada

    i=0 # Inicializo la variable i que tomará los valores de la listaDeTodo en cada iteracion
    while i < len(listaDeTodo): # Mientras i sea menor a la longitud de la listaDeTodo
        palabra="" # Creo la variable palabra que por ahora esta vacia

        for elemento in listaDeTodo[i]: # La variable elemento tomará los valores que guarda cada item en cada una de las repeticiones
            if elemento[0]==letraAzar: # Si el elemento comienza con la letra estipulada
                palabra=elemento # La variable palabra toma el valor del elemento al que se la esta igualando

        salida.append(palabra) # agrega la palabra en la lista salida en caso de cumplir lo anterior o en caso de no tener palabra para algun item agrega una cadena vacia
        i=i+1 # Aumento la variable para pasar a la siguiente lista
    return(salida) # Devuelve la lista salida
```

Para un mejor desarrollo de el programa, decidimos agregarle lectura de archivos con el objetivo de tener una mayor variedad de respuestas. Así le dimos inicio a la función **lectura**, la cual mediante archivos podíamos obtener mas variedad de respuestas a la hora de jugar. El inconveniente con esta función fue el salto de línea que tienen las listas por defecto a la hora de leer un archivo, lo cual nos trajo muchas fallas en el juego. Después de investigar varios sitios web, lo arreglamos y finalmente pudimos corregirlo.

```
def lectura(nombreArchivo): # La funcion lectura abre el archivo que tiene como parametro
    archivo=open(nombreArchivo+".txt","r") # Abrimos el archivo txt
    mostrar=[]
    elementos=archivo.readlines() # Elementos es una lista con todos valores del archivo
    for linea in elementos: # La variable linea toma cada valor que contiene elementos
        if linea[-1] == '\n': # verifica si hay un salto de linea
            linea = linea[:-1] # Borra el el salto de linea
            mostrar.append(linea) # Agrega a la lista mostrar cada valor
    return(mostrar) # Devuelve mostrar
    archivo.close() # Cerramos el archivo
```

En cuanto a la imagen de fondo de pantalla, después de días de investigación, pudimos encontrar un código algo complejo y adaptarlo a nuestro programa para que funcione correctamente. Tuvimos que entender el mecanismo del funcionamiento para ajustarlo a nuestra necesidad.

La implementación del sonido luego de analizar algunas alternativas posibles para nuestro juego decidimos poner una canción que tenga la misma duracion del juego.

```
pygame.mixer.music.play(1)
imagendefondo=pygame.image.load("imagendefondo.jpg").convert()
```

Le agregamos mas ítems al programa principal para hacer el juego más entretenido para los jugadores.

```
items=["colores","países","animales","equipos argentinos","marcas de autos","simpsons"]
colores= lectura("colores")
países= lectura("países")
animales= lectura("animales")
equipos= lectura("equipos argentinos")
marcasAutos=lectura("marcas de autos")
simpsons=lectura("simpsons")
```

3. Conclusiones:

Fue un desafío para nosotros programar un videojuego, teniendo en cuenta que recién empezamos. Al investigar sobre los obstáculos que se nos fueron presentando aprendimos nuevas formas de enfrentarlos y resolverlos. Tuvimos que trabajar sobre programas ya creados, lo cual nos complicó un poco. Entendemos que esto nos servirá en la vida profesional cuando tengamos que vincularnos con otros sectores y trabajar de forma colaborativa.

Anexo:

Implementar las funciones requeridas para el correcto funcionamiento del juego. Pensar e implementar funciones auxiliares que resuelvan tareas intermedias, de forma tal que el código sea más claro, sencillo, ordenado, legible y fácil de corregir.

Las funciones que reciben listas como parámetros deberán también chequear que dichas listas permanezcan en el estado correcto luego de utilizada la función.