

# UNIVERSIDAD NACIONAL GENERAL SARMIENTO

Introducción a la programación  
TRABAJO PRÁCTICO

## **PyGame - TutiFrutiUNGS**

Comisión 07

Turno Noche

Grupo 4

*Bertani Luana*

*López Gonzalo*

*Peralta Laura*

Profesores: *Cinthya Anabela Cardozo, Luis Santiago Veronesi*

## Trabajo Práctico PyGame

### **Consigna:**

*Implementar las funciones requeridas para el correcto funcionamiento del juego **EsCorrecta** (), **unaAlAzar** (), **juegaCompu** ().*

*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.*

Los Simpson TutiFruti es un juego que ayuda a pensar, mejorar el vocabulario, estimula la memoria, el razonamiento y la concentración. Consiste en contestar en el menor tiempo posible y de manera correcta siete items: nombres, animales, colores, sustantivos comunes, países, marcas y capitales o provincias argentinas. Las palabras elegidas deberán comenzar con la letra que arroje el programa una vez finalizada la cuenta regresiva en la pantalla inicial. El participante tendrá a la computadora como contrincante y las elecciones que realice así como también el tiempo que dure el juego afectaran el puntaje final, el cual estará determinado de la siguiente manera:

- Se sumaran 10 puntos por cada palabra correcta
- Se restaran 5 puntos por cada palabra incorrecta
- En el caso de haber coincidencia respecto a la palabra elegida por la computadora, se sumaran solo 5 puntos
- Se restará un punto por cada segundo transcurrido. Ello se podrá observar en la parte superior izquierda de la pantalla, así como también del lado derecho, el puntaje que se acumulará con cada respuesta.

En la pantalla final figurarán las respuestas realizadas por el participante y por la computadora, los puntajes obtenidos según los aciertos, las coincidencias, los errores y el tiempo transcurrido. Asimismo, se hará mención en la parte inferior el record alcanzado hasta el momento.

Se han incorporado seis funciones que permiten el normal funcionamiento del juego con todas las características mencionadas anteriormente:

1. ***def unaAlAzar (abc):*** Retorna una letra al azar para que se use en el juego.

```
def unaAlAzar(abc):  
    letraElegida = random.choice (abc)  
    return letraElegida
```

2. ***def esCorrecta (palabraUsuario, letra, item, items, listaDeTodo):*** Verifica si la respuesta del usuario empieza con la letra indicada y también si se encuentra dentro de las disponibles en la lista de todas las opciones según el ítem correspondiente. En caso de que la palabra sea correcta, retornará 10 puntos, caso contrario le restara 5 puntos.

```
def esCorrecta(palabraUsuario, letra, item, items, listaDeTodo):  
    palabraUsuario = palabraUsuario.lower()  
    indiceDelItem = items.index (item)  
    opcionesDelItem = listaDeTodo[indiceDelItem]  
  
    if (palabraUsuario[0] == letra and palabraUsuario in  
opcionesDelItem):  
        return 10  
    return -5
```

3. ***def juegaCompu (letraAzar, listaDeTodo):*** Retorna una lista con una respuesta para cada categoría que cumpla con la condición de iniciar con la letra elegida en el comienzo del juego. En el caso de no hallar ninguna opción válida, devolverá como respuesta tres puntos seguidos "...".

```
def juegaCompu (letraAzar, listaDeTodo):  
    listaLetra = []  
    resultado = []  
  
    for idx in range (len (listaDeTodo)):  
        resultado.append ("...")  
  
        for palabra in listaDeTodo[idx]:  
            if palabra[0] == letraAzar:  
                listaLetra.append (palabra)  
  
        if len (listaLetra)>0:  
            resultado.pop (-1)  
            resultado.append (random.choice(listaLetra))  
            listaLetra=[]  
  
    return resultado
```

4. ***def guardar\_puntajes (puntajes) y def recuperar\_puntajes ():*** Estas funciones mantienen un historial con el mejor puntaje realizado y tiempo registrado.

```
def guardar_puntajes (puntajes):
    historial = open ("datos/historial.txt", "w")
    for puntaje, tiempo in puntajes:
        historial.write (str (puntaje)+","+tiempo+"\n")
    historial.close ()

def recuperar_puntajes ():
    puntajes = []
    historial = open ("datos/historial.txt", "r")

    for linea in historial:
        puntaje, tiempo = linea.rstrip ("\n").split (",")
        puntajes.append ((int (puntaje),tiempo))
    historial.close ()

    return puntajes
```

5. ***def cargarItems ():*** Lee los archivos de texto que contienen las opciones válidas para cada ítem o categoría y los carga en formato de lista a la variable listaDeTodo, la cual será usada durante el juego para verificar si las respuestas son correctas. Se indicó que al momento de abrir los archivos con la librería codecs, los caracteres deberán estar en formato utf-8, cuyo motivo se explicará mas adelante.

```
Def cargarItems ():
    nombres=[]
    animales=[]
    colores=[]
    sustantivos_comunes=[]
    paises=[]
    marcas=[]
    cap_prov_arg=[]

    #Nombres
    nombres_txt=codecs.open("items/nombres.txt", "r", "utf-8")
    datos_nombres=nombres_txt.read()
    nombres.append(datos_nombres)
    nombres_txt.close()
    nombres=nombres[0].split(", ")

    #Animales
    animales_txt=codecs.open("items/animales.txt", "r", "utf-8")
    datos_animales=animales_txt.read()
```

```
animales.append(datos_animales)
animales_txt.close()
animales=animales[0].split(", ")
```

```
#Colores
colores_txt=codecs.open("items/colores.txt", "r", "utf-8")
datos_colores= colores_txt.read ()
colores.append(datos_colores)
colores_txt.close()
colores=colores[0].split(", ")
```

```
#Sustantivos
sustantivos_comunes_txt= codecs.open ("items/sustantivos
comunes.txt", "r", "utf-8")
datos_sustantivos_comunes=sustantivos_comunes_txt.read()
sustantivos_comunes.append(datos_sustantivos_comunes)
sustantivos_comunes_txt.close()
sustantivos_comunes=sustantivos_comunes[0].split(", ")
```

```
#Países
países_txt=codecs.open("items/paises.txt", "r", "utf-8")
datos_países=países_txt.read()
países.append(datos_países)
países_txt.close()
países=países[0].split(", ")
```

```
#Marcas
marcas_txt=codecs.open("items/marcas.txt", "r", "utf-8")
datos_marcas=marcas_txt.read()
marcas.append(datos_marcas)
marcas_txt.close()
marcas=marcas[0].split(", ")
```

```
#Capitales
cap_prov_arg_txt=codecs.open ("items/capitales y provincias
Argentinas.txt", "r", "utf-8")
datos_cap_pro_arg=cap_prov_arg_txt.read()
cap_prov_arg.append(datos_cap_pro_arg)
cap_prov_arg_txt.close()
cap_prov_arg=cap_prov_arg[0].split(", ")
```

```
listaDeTodo= [nombres, animales, colores, sustantivos_comunes,
países, marcas, cap_prov_arg]
```

```
return listaDeTodo
```

6. ***def dibujarPresentacion (screen, imgPresentacion, segundos):*** La presentación del juego es lograda por esta función, la cual recibe por parámetro la ventana del juego, una imagen y los segundos que son usados en la cuenta regresiva inicial.

```
def dibujarPresentacion (screen, imgPresentacion, segundos):  
    screen.blit (imgPresentacion, (0, 0))  
    segundos *= -1  
  
    defaultFontMUYGRANDE = pygame.font.Font  
    (pygame.font.get_default_font(), TAMANO_LETRA_MUYGRANDE)  
  
    RenCtaRegresiva = defaultFontMUYGRANDE.render (str (segundos), 1,  
    COLOR_BLANCO)  
  
    screen.blit (RenCtaRegresiva, (396, 115))
```

Por otra parte, se realizaron modificaciones en el código del programa principal, de los archivos de configuración y extras:

- Python contempla el sistema de codificación ASCII (Código Estadounidense Estándar para el Intercambio de Información) de 7 bits, el cual no incluye ni “ñ” ni acentos ni ningún otro carácter que no esté en el teclado americano. Por tal motivo, se declaró la codificación de caracteres Unicode UTF-8 (Unicode Transformation Format-8) para admitir caracteres no compatibles con ASCII incorporando al inicio del programa principal lo siguiente:

```
#!/usr/bin/env python  
# -*- coding: utf-8 -*-
```

- Se adicionó en la variable “abc” a la letra “ñ”. Para lograr renderizarla en mayúscula se sustituyó la fuente tipográfica predeterminada de Pygame por la fuente Arial ya que la versión 1.9.1 no la admite.
- Se agregó un nuevo ciclo, cuyo objetivo es habilitar la posibilidad de reiniciar el juego. También se encuentran definidas las variables que se usan, las cuales se resetean en caso de que se inicie un juego nuevo.

```
while True:
```

```
    if juegoNuevo:
```

```
        # Controladores de ciclo  
        juegoNuevo = False  
        presentacion = True  
        habilitarReinicio = False  
        ctaRegresiva = 5  
        i = 0
```

```
        # Tiempo total del juego  
        gameClock = pygame.time.Clock()  
        totaltime = 0  
        fps = FPS_INICIAL
```

```
        # Variables  
        puntos = 0  
        palabraUsuario=""  
        eleccionUsuario= []  
        eleccionCompu= []  
        aciertos = 0  
        incorrectas = 0  
        segundos = 0  
        letraAzar = unaAlAzar(abc)
```

```
        # Musica  
        pygame.mixer.music.load("sonidos/intro.mp3")  
        pygame.mixer.music.play()  
        aycaramba= pygame.mixer.Sound ("sonidos/aycaramba.wav")
```

- Se anuló la función dameLetraApretada y se substituyó por unicode.

```
#letra = dameLetraApretada(e.key)  
letra = e.unicode  
palabraUsuario += letra
```

- Se modificó el modo de realizar el conteo de los segundos para permitir el reinicio por cada juego o ronda.

```
#segundos = pygame.time.get_ticks () / 1000  
segundos = (math.ceil ((totaltime / 100)/10) - ctaRegresiva)
```

- Se modificaron tamaños y colores de las fuentes desde el archivo Configuraciones.
- En la función **dibujarSalida ()** del archivo **extras.py**, se agregaron líneas de código para renderizar los resultados del juego, así como también, verificar si se logró un nuevo récord y en tal caso llamar a la función **guardar\_Puntajes ()**.

### **# Resultados**

```

ptsCoincidencia = 0
for idx in range (0, (len (eleccioncompu))):
    if eleccioncompu [idx] == eleccionUsuario[idx]:
        ptsCoincidencia += 10

total = puntos + ptsCoincidencia - segundos

renAcierto = defaultFont.render ("Aciertos:      " + str(aciertos)
    + "pts", 1, COLOR_TEXTO)

renCoincidenecia = defaultFont.render ("Coincidencias:      " + str
    (ptsCoincidencia) + "pts", 1, COLOR_TEXTO)

renIncorrectas = defaultFont.render ("Incorrectas:      " + str
    (incorrectas) + "pts", 1, COLOR_TEXTO)

renDescTiempo = defaultFont.render ("Tiempo:      - " + str
    (segundos) + "pts", 1, COLOR_TEXTO)

renTotal = defaultFont.render ("TOTAL:      " + str
    (total) + "pts", 1, COLOR_TEXTO)

screen.blit (renAcierto, (100, 300))
screen.blit (renCoincidenecia, (100, 330))
screen.blit (renIncorrectas, (100, 360))
screen.blit (renDescTiempo, (100, 390))
screen.blit (renTotal, (150, 430))

```

### **# Record**

```

ultimo_record = recuperar_puntajes ()
record = ultimo_record [0][0]
tiempo = ultimo_record [0][1]

```



***# Renderizar nuevo record***

*if total > record:*

*# Musica ganador*

*pygame.mixer.music.load ("sonidos/ta-ra-ra-ra-hey.mp3")*

*pygame.mixer.music.play ()*

*renFelicidades = defaultFont.render("FELICIDADES NUEVO  
RECORD", 1, COLOR\_LETRAS)*

*renNuevoRecord = defaultFont.render ("El nuevo record es de:  
" + str (total) + "pts", 1, COLOR\_LETRAS)*

*screen.blit (renFelicidades, (400, 350))*

*screen.blit (renNuevoRecord, (400, 400))*

***#Guardar nuevo record***

*puntajes = [(total, str (int (segundos)))]*

*guardar\_puntajes (puntajes)*

***# Renderizar record anterior***

*else:*

*# Musica perdedor*

*pygame.mixer.music.load("sonidos/ouch..mp3")*

*pygame.mixer.music.play()*

*renRecord= defaultFont.render("Tu record anterior fue de " +  
str (record)+ "pts, en " + tiempo + " segundos", 1,  
COLOR\_LETRAS)*

*screen.blit (renRecord, (100, 470))*

***# Jugar de nuevo***

*renReiniciar = defaultFont.render ("PRESIONE ENTER PARA JUGAR DE  
NUEVO", 1, COLOR\_LETRA)*

*screen.blit (renReiniciar, (100, 550))*

- Se extendieron los archivos de texto de cada categoría con variedad de palabras.
- Finalmente, se agregaron sonidos que acompañan al participante durante la ronda de juego.