

# Introducción a la programación

## Rapido y ...

February 6, 2023

### Introducción

El trabajo consiste en implementar un juego para escribir palabras rapidamente, cuyo objetivo es lograr que el usuario aprenda a teclear ubicando bien las manos en el teclado y sin mirarlo. Gran parte del juego ya está resuelto, solamente faltan implementar las funcionalidades más importantes.

## 1 El Juego

### Reglas del Juego

Se juega de a un jugador, que cuenta con 60 segundos para escribir la mayor cantidad de palabras que aparecen en pantalla. El jugador debe escribir la palabra en pantalla y si la palabra es la correcta, se deben sumar puntos al jugador y dicha palabra desaparece. Van apareciendo nuevas palabras en pantalla que van moviendose hacia la derecha hasta desaparecer. El archivo de muestra contiene países.

### Lo que ya esta implementado

El juego actualmente consta de un archivo con el programa principal. Este se encarga de capturar la entrada del teclado, llevar la cuenta de los puntos y del tiempo, así como también de dibujar en la pantalla.

El programa principal cuenta también con:

- una variable de tipo `lista` que es una matriz que guarda las palabras que están en pantalla.

Las letras están divididas en 3 grupos, las vocales, las consonantes comunes y las consonantes difíciles que otorgan distintos puntos al formar parte de la palabra que debe adivinar el jugador. El programa toma la palabra al azar desde el archivo `países.txt` donde, si el jugador ingresa la palabra correcta, la

jugada es válida y se suman puntos de acuerdo a la clasificación mencionada. Además, cada cierto tiempo aparecen nuevas palabras en pantalla (sin ocupar toda la pantalla, ni escaparse de ella ni sobreponerse).

Para la mayoría de estas tareas, el programa hace uso de una biblioteca de código llamada *PyGame*. Una biblioteca de código es un conjunto de sub-programas utilizados para desarrollar software. En particular PyGame es una biblioteca especialmente diseñada para el desarrollo de juegos interactivos en Python.

La posición  $(0, 0)$  de la pantalla es el vértice superior izquierdo, las  $x$  crecen hacia la derecha y las  $y$  crecen hacia abajo.

## Lo que falta implementar

Aun faltan implementar las funciones del archivo `funciones.py`. Estas funciones son utilizadas desde el programa principal.

La función `nuevoPais(listaPaises)` debe elegir al azar un país de la lista de países que fueron leídas previamente desde un archivo.

La función `actualizar(paisesEnPantalla, posiciones, listaPaises)` debe recibir la lista de países en pantalla, sus posiciones y la lista de todos los países. Debe eliminar los países y sus posiciones cuando empiezan a escaparse de la pantalla, también debe ir actualizando las posiciones de los países para que vayan moviéndose hacia la derecha y agregando nuevas en posiciones al azar, siempre con el cuidado de que las palabras no se superpongan, no aparezcan incompletas no llenen por completo la pantalla ni superen cierta cantidad de países para que el juego no sea muy difícil.

La función `esValida(candidata, listaPaises, paisesEnPantalla)`: debe recibir el país que escribe el usuario, la lista de países que se leyó del archivo y la lista de países de la pantalla y controlar si la palabra es correcta.

La función `quitar(candidata, paisesEnPantalla, posiciones)` debe recibir la palabra que escribe el usuario, la lista de países que hay actualmente en la pantalla y sus posiciones. Debe quitar a la palabra de la lista de países de la pantalla y también a la posición.

La función `procesar(candidata, listaPaises, paisesEnPantalla, posiciones)` debe recibir la palabra que escribe el usuario, la lista de países leída del archivo, la lista de países de la pantalla y sus posiciones. Debe controlar si la palabra es válida, es decir, si está en la pantalla y en caso afirmativo quitarla de la pantalla y sumar los puntos correspondientes.

La función `Puntos(candidata)` debe recibir un país y retornar el puntaje total correspondiente a la palabra formada según:

- cada vocal otorga 1 punto,
- cada consonante otorga 2 puntos, salvo las difíciles (j, k, q, w, x, y, z).
- cada consonante difícil (j, k, q, w, x, y, z) otorga 5 puntos.

La función `estaCerca(elem, lista)` es opcional, le podría servir para evitar superposiciones.

## 2 Cómo empiezo

### 2.1 Instalar PyGame

La versión más reciente de PyGame para Windows y Python 3.1 se descarga directamente desde <http://pygame.org/ftp/pygame-1.9.1.win32-py3.1.msi>. para diferentes versiones, chequear <http://www.pygame.org/download.shtml>. Al momento de instalar PyGame, el programa de instalación intentará reconocer la versión de Python instalada en la computadora. De no detectarla correctamente, se sugiere no continuar con la instalación, dado que PyGame no funcionará. En ese caso chequear que la versión de PyGame sea la indicada para la versión de Python. En nuestro caso es la 3.1.

### 2.2 Descargar archivos

Descargar del moodle de la materia el archivo comprimido con todos los archivos necesarios para el TP. Descomprimir todo el contenido del archivo en una carpeta y abrir los archivos `.py` con el PyScripter. El único archivo que se espera que modifiquen es `funciones.py` pero a la hora de ejecutar el proyecto, hay que ejecutar el archivo `principal.py`.

## 3 Consigna

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.

Sugerimos **fuertemente** probar y corregir las funciones más sencillas antes de encarar las funciones más complicadas.

## 4 Requisitos de aprobación y criterio de corrección

El presente trabajo debe realizarse Individualmente, se pueden usar TODAS las funciones auxiliares que desee. Para aprobar el trabajo, se deberá cumplir los siguientes items:

- El juego debe funcionar correctamente.
- El código debe ser claro. Es decir, las variables y funciones deben tener nombres que hagan fácil de entender el código a quien lo lea, y deben haber comentarios que ayuden al fácil entendimiento de cada porción de código. Además debe hacer funciones siempre que se considere necesario, y se evaluará el buen uso de las mismas.
- El código debe ser coherente. Es decir, no deben haber variables que no se usan, funciones que tomen parámetros que no necesitan, ciclos innecesarios, etc.

Nota: El correcto funcionamiento del juego no es suficiente para la aprobación del trabajo, son necesarios todos los items mencionados arriba.

## 5 Opcionales

Las siguientes funcionalidades del juego no son necesarias para la aprobación (con nota mínima), pero sirven para mejorar la nota del trabajo. De optar por hacerlas, se aplican las mismas reglas y criterios de corrección que para las funcionalidades básicas. Cualquier otra funcionalidad extra que se desee implementar debe ser antes consultada con los docentes.

### Longitud mínima

Toma de la lista de nombres solo aquellos que tengan más de cierta cantidad de caracteres.

### Velocidad

A medida que pasa el tiempo los nombres descienden con mayor velocidad.

### Por color

Cada país tiene un color aleatorio.

### Efectos de sonido

Hacer que el juego reproduzca efectos de sonido cuando sucedan los eventos más importantes: ingresa una palabra, ingresa una palabra con mucho puntaje, errores, etc.

## Mejores records

Hacer que el usuario luego de jugar tenga la opción de ingresar su nombre y se muestren los 10 mejores puntajes históricos con sus nombres.

## Diferentes niveles

Que el usuario tenga opciones de niveles donde se modifique el juego, por ejemplo:

- el usuario indica la longitud mínima (o máxima) de letras que deben tener las palabras,
- el usuario indica los puntos asignados a las letras difíciles (baja, media, alta).
- cambia la velocidad de descenso de los nombres.

## Juego predefinido

Leer de un archivo la tira de inicio y las nuevas tiras que renuevan la jugada.

## 6 Fecha de entrega

El trabajo debe ser entregado en el segundo llamado de FINAL.

## 7 Forma de entrega

El TP se considera correctamente entregado y apto para ser corregido si se cumple con una entrega impresa y una digital.

- La entrega impresa consiste de un informe impreso donde haya una introducción que explique de qué se trata el trabajo (explicado para alguien que no leyó el enunciado), que incluya el código de las funciones implementadas y una breve explicación de cada una de ellas junto con las dificultades de implementación con las que se encontraron. No se deben imprimir los casos de prueba. El informe también debe incluir las decisiones que hayan tomado ante diferentes alternativas posibles, cuáles fueron sus alternativas consideradas y por qué tomaron esas decisiones. El informe **no** debe incluir párrafos para ocupar lugar, explicaciones de funcionalidades básicas de Python, PyGame o de programación en general.