

Introducción a la programación

El juego del orden del subtítulo

18 de octubre de 2019

Introducción

El trabajo consiste en hacer un juego en el que se debe adivinar el orden de aparición de un subtítulo de una película.



**Equipo de Protección
Ambiental**

Gran parte del juego ya está resuelto, faltan implementar las funcionalidades más importantes.

1. El Juego

Reglas del Juego

Se juega de a un jugador, que cuenta con 60 segundos para sumar la mayor cantidad de puntos. Debe escribir una palabra que se encuentre dentro de la línea del subtítulo que supone se ubica luego de la que aparece primero en pantalla.

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. Se brinda un archivo de texto con el subtítulo de la película de los Simpsons en un formato que se debe respetar y limpiar al leerlo.

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 subprogramas 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

Lo primero que se debe hacer es leer el archivo. Luego debe hacer funcionar el juego, para ello debe completar las funciones vacías obligatorias.

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

La función `lectura(archivo, subtítulo, n)` se queda solo con los subtítulos de al menos longitud n y lo filtra quitando caracteres especiales.

La función `seleccion(subtítulo)` elige un subtítulo al azar, y devuelve una lista compuesta por ese subtítulo, el siguiente y otro distinto.

La función `puntos(n)` devuelve el puntaje correspondiente dependiendo de la seguidilla de correctas que haya alcanzado el jugador

La función `procesar(palabraUsuario, mostrada, siguiente, otra, correctas)` chequea que sea correcta la opción elegida por el jugador, es decir, que pertenece solo a la frase siguiente y devuelve el puntaje correspondiente.

2. Cómo empiezo

2.1. Instalar PyGame

La versión más reciente de PyGame para Windows y Python 3.2 se descarga directamente desde <http://pygame.org/ftp/pygame-1.9.2a0.win32-py3.2.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 3.2.

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 al principio es `funcionesVACIAS.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 en grupos de 2 o 3 integrantes. 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.
- Los casos de prueba deben ser completos y debe ser posible correrlos de nuevo por cualquiera.

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.

Sin Repetidos

Evita que se repitan subtítulos.

Longitud mínima

Exige que las palabras tengan como mínimo una cierta longitud.

Efectos de sonido

Hacer que el juego reproduzca efectos de sonido cuando sucedan los eventos más importantes: acierta, seguidilla de aciertos o 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,
- se elige la película.
- se agregan mas opciones de subtítulos equivocados.

Juego predefinido

Leer un subtítulo con pocas líneas para poder testear el juego.

Burla

Si el usuario falla le explica el error con sonidos diferentes o carteles.

6. Fecha de entrega

El trabajo debe ser entregado en la fecha estipulada en el cronograma, recordar que es requisito hacer pre-entregas.

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.
- La entrega digital consiste en enviar un e-mail de la siguiente manera:
 1. A la cuenta de correo electrónico de los docentes de la comisión.
 2. **Con asunto:** entrega-tp 2019 *nombre – estudiantes*.
 3. Adjuntando por separado un archivo comprimido con los archivos que conforman el código fuente de **todo** el programa y un archivo con el informe (preferentemente en formato pdf). El código debe cargar sin errores. Si el trabajo es muy pesado recomendamos subirlo a google drive, dropbox, etc y pasarnos el link.

Nota: Se acepta una sólo entrega por grupo, de modo que si un grupo entrega su trabajo más de una vez, los docentes elegirán cuál de las versiones entregadas corregir.