

Introducción a la programación

Pasa palabra?

May 21, 2018

Introducción

El trabajo consiste en adivinar la palabra que aparece en pantalla, en su versión inicial cada * oculta una letra. El archivo de donde se sacan las palabras que van a aparecer en pantalla son los subtítulos de la película Matrix de 1999. 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 escribir la mayor cantidad de palabras que aparecen en pantalla. El jugador debe escribir la palabra en pantalla y si es la correcta, se deben sumar puntos (si acierta exactamente la palabra se le suman 5 veces la longitud de dicha palabra, si la palabra que escribe cumple todos los requisitos pero no es exactamente la palabra de los subtítulos se le suma la longitud de la palabra) y se cambia de palabra. Si es incorrecta, se le restan puntos, exactamente la longitud de la palabra que debía adivinar. Es decir, con las letras de la pantalla seguramente se pueden formar muchas palabras y puede ocurrir que aunque las letras formaban originalmente una palabra el usuario puede sugerir otra y deberá ser considerado válida. Por ejemplo si el archivo contiene la palabra *entero*, y en pantalla se ve ***t*ro* y el usuario escribe *satiro* o *futuro*, el juego debe considerar correcta la palabra y sumar 6 puntos.

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 brindan dos archivos de

texto con palabras, uno con subtítulos de la película Matrix y otro con palabras del diccionario.

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

Lo primero es leer y limpiar el archivo de los subtítulos. El programa debe leer el archivo `matrix1999.srt` y quedarse con las palabras de mayor longitud de cada línea, de esas palabras elegir una para que aparezca oculta (con algunos `*`) en pantalla. Debe eliminar los caracteres conflictivos como acentos, signos de admiración y símbolos que no forman palabras y se usan para saber en qué momento de la película debe aparecer ese subtítulo. Luego debe hacer funcionar el juego, deben aparecer palabras ocultas en pantallas y si el jugador ingresa la palabra correcta, la jugada es válida y se suman puntos. También debe mostrar la longitud de la palabra, esto va a cobrar mucha importancia en los ejercicios adicionales.

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

La función `elegirLaMasLarga(listaPalabras)` recibe una lista de palabras y devuelve la palabra de mayor longitud

La función `frasesToPalabras(listaFrases)` recibe una lista, donde cada elemento es una frase, y devuelve una lista con todas las cadenas que componen las frases.

La función `quitarCaracteresEspeciales(linea)` recibe una cadena y cambia los caracteres conflictivos como tildes, signos de admiración, etc.

La función `lectura(archivo, listaFrases)` lee el archivo de subtítulo.

La función `lectura2(archivo2, listaPalabrasDiccionario)` lee el archivo del diccionario

La función `nuevaPalabra(palabras)` recibe una lista con todas las palabras y devuelve una de ellas.

La función `esValida(candidata, oculta, palabra)` debe recibir la palabra que escribe el usuario, la palabra con * y la palabra correcta. Debe controlar si la palabra es correcta, es decir, si cumple los requisitos.

La función `Puntos(candidata)` debe recibir una palabra y retornar el puntaje total correspondiente a la palabra formada según: si coincide exactamente con la palabra de los subtítulos devuelve 5 veces la longitud de la palabra, si aparece en el diccionario devuelve la longitud de la palabra y si es errónea resta la longitud de la palabra

La función `procesar(candidata, oculta, palabra, listaPalabrasPeli, listaPalabrasDiccionario)` debe recibir la palabra que escribe el usuario, la palabra oculta que aparece en pantalla, la palabra original, la lista de palabras del archivos de subtítulos y la lista de palabras del diccionario. Debe controlar si la palabra es válida, y sumar o restar los puntos correspondientes. Devuelve el puntaje obtenido.

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 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.

Longitud mínima

Toma de la lista de palabras solo aquellas que tengan más de cierta cantidad de caracteres.

Mas oculta

Los asteriscos reemplazan a uno o mas caracteres.

correcta

Muestra en pantalla la palabra oculta cuando el usuario no acierta.

Efectos de sonido

Hacer que el juego reproduzca efectos de sonido cuando sucedan los eventos más importantes: acierta, es correcta pero no está en la lista de las palabras del subtítulo, 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,
- cambia la cantidad de letras ocultas.
- se otorga un tiempo por palabra.

Juego predefinido

Leer archivos con pocas palabras para poder testear el juego.

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 2018 *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.