**Seminario de Lenguajes: Python**

**Trabajo final: Informe**

**Bautista Bodrone - Juan Cruz Galati M. – Alejandro Zambrano Taus**

**Grupo 12**

CONTENIDOS

[Introducción. 2](#_Toc20102936)

[Temas investigados. 2](#_Toc20102937)

[Problemas y soluciones surgidas. 2](#_Toc20102938)

[Conclusiones y trabajos futuros. 2](#_Toc20102939)

[Referencias. 2](#_Toc20102940)

|  |
| --- |
| **Introducción.** |

Se trata de un Juego de tipo Sopa de Letras desarrollado en el lenguaje Python y utilizando los módulos PySimpleGUI y Pattern, y los Sensores para transmitir información al usuario. Es un juego simple pero entretenido y que facilitará el trabajo a docentes de nivel primario a enseñar diferentes palabras del español y su clasificación, de forma didáctica e interactiva.

El programa está estructurado de la siguiente manera:

* **Menú principal**: Desde él se define el tema1, se accede a la configuración y al juego.
* **Configuración**: Desde él se modifican las configuraciones particulares para la ejecución del juego, tales como: cantidad de palabras por tipo de palabra, orientación de las palabras, colores para cada tipo de palabra, entre otras.
  + Posee una configuración por defecto, que se carga al iniciar el programa.
* **Juego**: Desde él se genera la matríz para el propio juego basándose en las configuraciones establecidas (o la configuración por defecto). La interfaz fue desarrollada teniendo en cuenta que sea lo más intuitiva posible.

|  |
| --- |
| **Temas investigados.** |

* **PySimpleGUI**:
  + El primer módulo en ser analizado. Se discutió la mejor distribución de las secciones principales (juego, menú y configuración) y la comunicación entre ellas. Luego, los PopUps de cada una y, por último, la gama de colores a utilizar en cada tema.
    - Cada interfaz fue programada individualmente, buscando el mayor dinamismo posible.
* **Pattern.es**: Se aprovecharon los métodos provistos por este módulo para el reconocimiento del tipo de palabras mediante Wiktionary y el propio diccionario de Pattern.
* **Otros módulos**:
  + Se utilizaron módulos provistos por la propia librería de Python 3.6.8, tales como os, random, time, nunpy, entre otros.

|  |
| --- |
| **Problemas y soluciones surgidas.** |

* Sistema Operativo:
* Versión de Python y compatibilidad con Pattern:
  + Hay un error de Pattern que produce problemas para ejecutarlo en Python 3.7.
  + **Solución**: Se aclara en el LEEME, entre los requisitos, la necesidad de usar la versión 3.6.8 de Python.
* PySimpleGUI (configuración):
  + En algunas versiones de PySimpleGUI hay un error que afecta al método Update de los Radio Button Element, que impide su correcta actualización.
  + **Solución**: En vez de actualizar la ventana, se cierra y se crea una nueva ventana en cada ejecución. Es poco estético e ineficiente, pero es la única forma de usar este tipo de inputs.
  + **Actualización**: El error fue solucionado en la versión más reciente de PySimpleGUI (agosto 2019). De todos modos mantendremos la solución anterior para asegurar compatibilidad en computadoras que no cuenten con la última versión de PySimpleGUI.
* Tamaño de la matriz (sopa de letras):
  + La Sopa de letras es una matriz cuadrada que se genera tomando como base el número de letras de la palabra más larga y la cantidad de palabras a ubicar (una fila/columna por palabra, como mínimo), sumando una o dos casillas extra de forma aleatoria. Esto producía dos problemas:
    - **1.** Que la ventana pudiera ser excesivamente grande si la palabra era un poco larga.
      * **Solución**: Si la matriz posee más de 12 casillas por línea, reduce el tamaño de cada casilla.
    - **2.** Que la ventana pudiera ser excesivamente grande por la cantidad de palabras a ubicar.
      * **Solución:** Se limitó la cantidad de palabras a 15. Esto se consiguió limitando a 5 la cantidad de palabras por tipo que se pueden seleccionar desde la configuración.
* Reconocimiento de palabras de Pattern:
  + El módulo Pattern considera todas las palabras que no logra reconocer como Sustantivos (lo que tiene sentido, por si son nombres).
  + **Solución**: Se considera que la lista de palabras guardada en el diccionario de Pattern y Wiktionary.

|  |
| --- |
| **Conclusiones y trabajos futuros.** |

Python es un lenguaje de programación muy versátil, con la capacidad de utilizar multitud de librerías para cualquier necesidad. La realización de este trabajo nos planteó un reto que nos permitió comprobar nuestros conocimientos adquiridos durante la cursada y nos planteó instancias para poder investigar y encontrar soluciones propias a imprevistos. El trabajo en grupo fue muy importante, obteniendo soluciones e ideas ingeniosas gracias a la discusión y planteamiento desde diferentes puntos de vista.

Sin dudas a este programa le quedan muchas mejoras que podrían realizarse, muchas puramente estéticas, y la implementación de diversas funcionalidades que podrían lograr que cumpla mejor con su funcionalidad de facilitar la enseñanza.

Algunas actualizaciones que podrían hacerse son:

* Mostrar una ventana con la solución en caso de perder.
* Guardar la Lista de Palabras en un archivo separado.
  + En caso de no existir el archivo al buscarlo, crearlo con una lista de palabras por defecto (actualmente hace algo similar, pero en el mismo archivo de configuración).
* Mejorar la interacción con el usuario incluyendo sonidos.
* Mejorar la interfaz gráfica con un diseño infantil.
  + Planeábamos realizar esto usando una temática de selva, con un mono como mascota principal.

|  |
| --- |
| **Referencias.** |

* **PySimpleGUI:** 
  + <https://pypi.org/project/PySimpleGUI/>
* **Licencia:** 
  + <https://www.gnu.org/licenses/licenses.es.html>
  + <https://www.gnu.org/licenses/gpl-howto.html>
* **Juego:**
  + <https://docs.scipy.org/doc/numpy/reference/generated/numpy.transpose.html#numpy-transpose>