Ingeniería en Computación
Taller de Programación
Semestre 1, 2022
Profesor: Ing. Cristian Campos Agüero

TEC | Tecnológico de Costa Rica

Proyecto Programado #1
Ahorcado

1. Introducción

Es uno de los juegos de mayor dominio o conocimiento de muchas personas, en la cual consiste en adivinar una palabra o frase, la persona quien adivina tiene una serie de intentos finitos (mientras se dibuja a un ahorcado)

Fuente: https://es.wikipedia.org/wiki/Ahorcado_(juego)

2. Software por desarrollar

Su trabajo consiste en implementar una aplicación de escritorio para administrar el juego utilizando el lenguaje **Python**, y programación **Iterativa**

El sistema inicia digitando la función **ahorcado()** y desplegará un **Menú Principal** donde se puede ingresar a las:

- (A) Opciones administrativas,
- (J) Opciones de jugador
- (S) Salir.

Si selecciona ingresar a las opciones administrativas o normales se desplegará un menú para las diferentes opciones, y dentro de esos menús de podrá devolver al Menú Principal.
Para todas las opciones del menú, debe permitir las opciones tanto en **mayúsculas** como **minúsculas**

3. Opciones Administrativas

Una vez seleccionado la **opción A**, el usuario deberá ingresar una **clave** de acceso (**debe estar generada y guardada en un archivo, este debe llamarse Acceso.txt**) y una vez digitado correctamente se deben habilitar las siguientes funcionalidades (Opciones administrativas):

- (P) Gestión de Palabras
- (F) Gestión de Frases
- (R) Retornar

3.1. Gestión de Palabras

El sistema debe permitir dar mantenimiento de las palabras, se debe permitir **incluir, eliminar o modificar cada una de ellas, además de mostrarlas**. La información que se debe almacenar será:

- Identificador (consecutivo autogenerado)
- Palabra

No pueden existir **palabras** iguales de la misma manera identificadores repetidos y no pueden **eliminarse palabras** que hayan sido asociados a algún juego.

Debe existir un archivo llamado "**Palabras.txt**" y el formato interno debe ser de la siguiente forma:

1, CASA
2, MUNDO
3, PUERTA

Para poder modificar o eliminar, debe de hacerse por medio del código

3.2. Gestión de Frases

El sistema debe permitir dar mantenimiento de las frases, se debe permitir **incluir, eliminar o modificar cada una de ellas, además de mostrarlas**. La información que se debe almacenar será:

- Identificador (consecutivo autogenerado)
- Frases

No pueden existir **frases** iguales de la misma manera identificadores repetidos y no pueden **eliminarse frases** que hayan sido asociados a algún juego.

Debe existir un archivo llamado "**Frases.txt**" y el formato interno debe ser de la siguiente forma:

1, CASA VERDE
2, PLANETA TIERRA
3, HOLA MUNDO

Para poder modificar o eliminar, debe de hacerse por medio del código

3.3. Retornar

Esta opción regresa al menú anterior

4. Opciones del jugador

En esta opción aparecerá el siguiente menú:

```
Bienvenido al juego del Ahorcado
=================================================================
 (N)uevo juego | (H)istoria | (E)stadísticas | (A)yuda | (R)etornar
=================================================================
```

4.1. Nuevo Juego

Una vez seleccionado, con ayuda de los inputs, solicitar el nombre del **usuario**, obtenido este, seleccionar el modo de jugo: **((P)rincipiante o (A)vanzado).**

- Si selecciona **Principiante**, seleccionará al azar una palabra del archivo "**Palabras.txt**"
- Si selecciona **Avanzado**, seleccionará al azar una palabra del archivo "**Frases.txt**"

Obtenido estos dos datos, se procede a jugar.
Se debe pintar en pantalla los datos como el **nombre del jugador, la palabra o frase, el número de intentos, la lista de letras usadas, el resultado y al ahorcado**.

```
Bienvenido al juego del Ahorcado
=================================================================
 (N)uevo juego | (H)istoria | (E)stadísticas | (A)yuda | (R)etornar
=================================================================

 ---------------            Jugador: Cristian
 |           |
 |           O              PALABRA:  _ A _ A _
 |         --|--
 |           |              Intento: 15
 |          / \            Letras usadas: fwx5
 |         /   \
 |
 |                          PERDISTE
 ===============
```

Finalizado el juego, se guardará en un archivo llamado **"Juego.txt"** los siguientes datos bajo el siguiente formato:

Código, Nombre del jugador, modo, Palabra o frase, total intentos, lista de letras usadas, resultado

Por ejemplo:

1, Cristian, Principiante, tecnologico, 8, ["A", "I", "T", "N", "E", "O","C","L"],ganador

4.2. Historia del Juego

En esta opción, se mostrará en pantalla la historia de cómo se creo este juego, la información debe cargarse desde un archivo llamado **"Historia.txt"**, el contenido de este lo pueden encontrar como en Wikipedia u otros sitios en la internet.

Al momento de mostrarse en pantalla debe ser en forma ordenada, el despliegue de pantalla no debe sobrepasar los **70 caracteres de largo**.

Al final de desplegar, debe solicitar al usuario **en presionar la tecla C** para continuar y volver a mostrar el menú de Opciones del Jugador

4.3. Estadísticas de Juegos

Se mostrará los siguientes datos:

1. La palabra con el total de derrotas al jugador
2. La frase con el total de derrotas al jugador
3. El total de juegos a nivel general
4. El total de juegos a nivel principiante
5. El total de juegos a nivel avanzado
6. El total general de juegos ganados y perdidos

Se recomienda al programador, guardar este dato en un archivo llamado **"Estadísticas.txt"**.

4.4. Ayuda

En esta opción, de la misma forma que la **sección 4.1**, mostrará una explicación al usuario de cómo jugar este juego. El archivo donde se guardará esta información se llamará **"Ayuda.txt"**

4.5. Retornar

Esta opción regresa al menú anterior

5. Salir

Finaliza el juego

6. Funciones adicionales

Se darán **15 puntos adicionales**, es decir si las opciones anteriores fueron **desarrolladas con éxito.** Esta nueva funcionalidad consistirá en el manejo en dos idiomas, inglés y español. En el momento de crear un nuevo juego, se le solicitará al usuario en qué idioma será la palabra para buscar y que además toda la **interfaz del juego también esté en inglés**

7. Aspectos técnicos

El proyecto deberá estar escrito en el lenguaje de programación **Python**. En caso de requerir librerías adicionales para compilar y ejecutar el programa, deberán **especificarlo** en la documentación, ya que de lo contrario se descontarán puntos en la evaluación. Al iniciar el programa se carga la información de disco y al salir se guarda (o una la manera permita que todos los datos registrados sean persistentes).
Y considerar lo siguiente:

- Se debe utilizar **iteración** en el desarrollo del proyecto, se permiten estructuras como **while y for**.
- Se deben manejar **mensajes claros** al usuario.
- Realizar validaciones de captura de campos. Deben hacer uso de mecanismos de validación eficientes
- Número de código o identificador es único y autogenerado
- Toda función ***bult-in*** de Python que deseen utilizar debe ser **validada** con el profesor. Las funciones **eval, append, Split, strip, pop, len** o **in** no están permitidas.
- El proyecto debe ser guardado en el repositorio del GitHub (https://classroom.github.com/a/c8kpgTD).

8. Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de estos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, restricciones y objetivo.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: **instrucciones de compilación, ejecución y uso**.
3. Pruebas de funcionalidad: (***Youtube***)
4. Descripción del problema.
5. Diseño del programa: decisiones de desarrollo, algoritmos usados.
6. Librerías usadas: creación de archivos, etc.
7. Análisis de resultados: lista detallada de objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
8. **Bitácora** (autogenerada, por ejemplo, en **GitHub** con **commit** por usuario incluyendo comentario, o bien, con otra herramienta que se muestra con imágenes lo desarrollado). **Al menos 6 commits**
9. Conclusión (es)

9. Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación. El proyecto programado tiene un valor de **15%** de la nota final, en el rubro de Proyectos.

Desglose de la evaluación del proyecto programado:

1. Documentación interna 4 ptos.
2. Documentación externa 6 ptos.
3. Funcionalidad 75 ptos (ver detalle en Software a Desarrollar)
4. Revisión del proyecto (según completitud del proyecto) 10 ptos.
5. Hora de Entrega 5 ptos.

10. Forma de trabajo

El trabajo se debe realizar de forma **individual**.

11. Aspectos administrativos

Crear 2 carpetas llamadas **documentación** y **programa**, en la primera deberá incluir el documento **PDF** solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de este proyecto programado.
Deben crear un archivo llamado **README.md**, este archivo debe contener la siguiente información:

a. Nombre del curso
b. Número de semestre y año lectivo
c. Nombre del Estudiante
d. Número de carné del estudiante
e. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada): [Deplorable|Regular|Buena|MuyBuena|Excelente|Superior]

Con este enlace https://docs.github.com/es/github/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax le ayudará a dar formato al archivo README.md

12. Rúbrica de evaluación

FUNCIONALIDAD **75%** **0**

Rubro	Valor	Obtenido	Observaciones
Opciones Administrativas			
Control de acceso	**5**		
Gestión de palabras	**15**		
Gestión de frases	**15**		

FUNCIONALIDAD	**75%**	**0**	
Rubro	**Valor**	**Obtenido**	**Observaciones**
Retornar	**5**		
Opciones de Jugador			
Nuevo Jugo	**10**		
Jugar	**20**		
Historia	**5**		
Estadísticas	**15**		
Ayuda	**5**		
Retornar	**5**		
Extras			
Manejo de idioma	**15**		
TOTAL	**115**	**0**	
DOCUMENTACIÓN INTERNA	**4%**		
DOCUMENTACIÓN EXTERNA	**6%**		
REVISIÓN DEL PROYECTO	**10%**		
HORA DE ENTREGA	**5%**		
NOTA FINAL	**100%**	**0%**	

13. Entrega

En el rubro de "Hora de Entrega" valdrá 5 puntos de la nota total del proyecto, según la siguiente escala:

a. Si se entrega antes de las 11:55:55 PM del martes **03 de mayo de 2022**, 5 puntos.
b. Si se entrega antes de las 11:55:55 AM el **04 de mayo de 2021**, 2.5 puntos.
c. Si se entrega antes de las 11:00:00 PM el **04 de mayo de 2021**, 0 puntos.
   Después de este punto, **NO SE ACEPTARÁN** más trabajos.

Todo el contenido de cada proyecto debe ser 100% original y en caso de plagio se asignará nota cero.