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

TEC | Tecnológico de Costa Rica

Proyecto Programado #3

Ahorcado versión Gráfica y Clases

Contenido

Índice de Figuras

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)

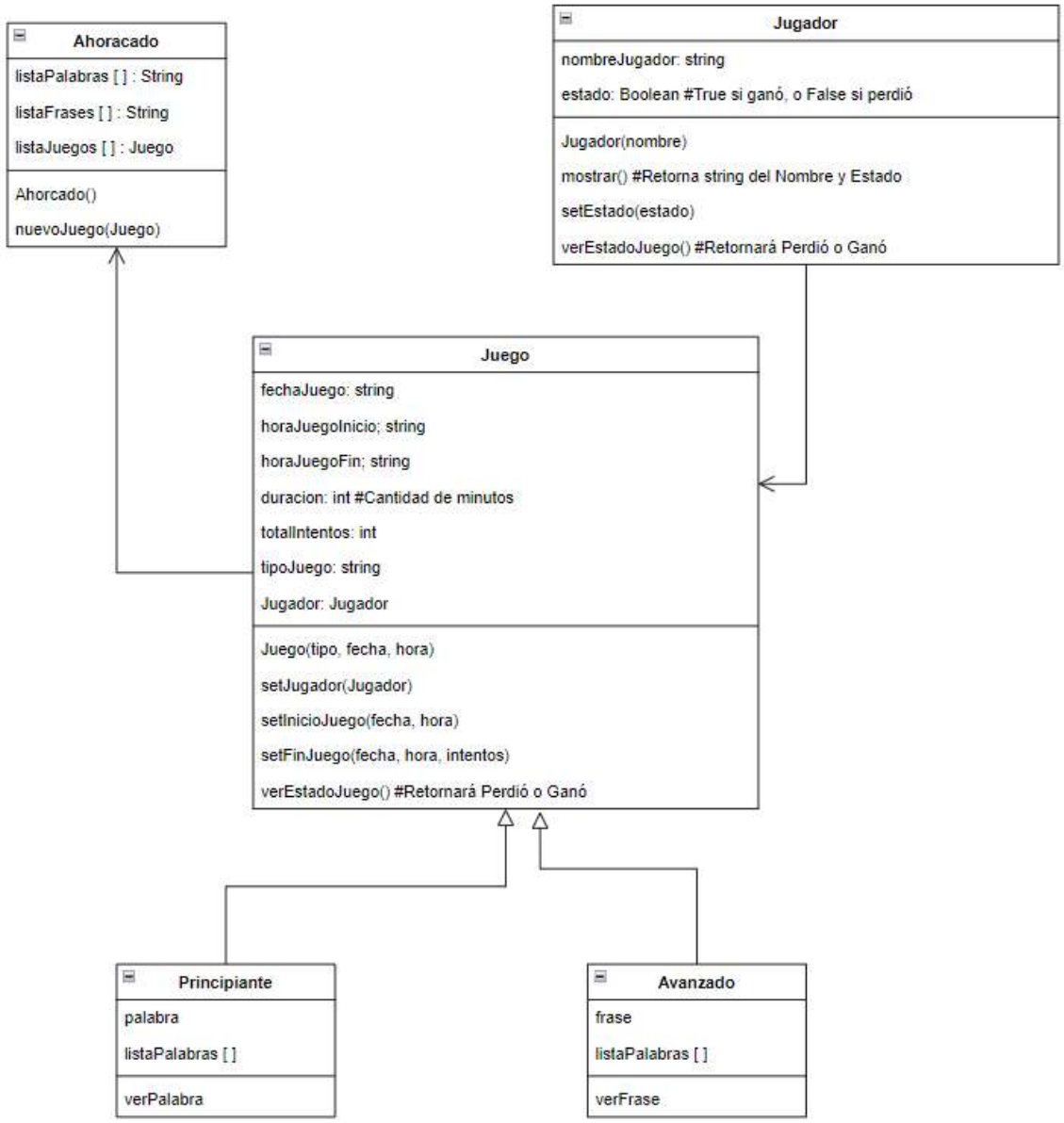
2. Software por desarrollar

Su trabajo consiste en implementar una aplicación de escritorio para administrar el juego utilizando el lenguaje **Python**, y programación **Iterativa y Orientada a Objetos.** A continuación, se muestra la pantalla principal del juego.

Figura 1 Pantalla inicio

En la siguiente imagen se muestra el cómo será el diseño de las clases para la adaptación de esta nueva versión del juego de ahorcado

Figura 2 Diagrama de clases del Ahorcado



Lo que se muestran son los atributos y métodos básicos, pero no limita a que se agreguen más según decisiones de desarrollo del grupo

3. Opciones Administrativas

Una vez seleccionado este, 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):

Figura 3 Menú administrativo

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 en una lista.

No pueden existir **palabras** repetidas y no pueden **eliminarse palabras** que hayan sido asociados a algún juego. Para poder modificar o eliminar, debe de hacerse seleccionando el item desde el List

Figura 4 Gestión de palabras

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 en una lista.

No pueden existir **frases** repetidas y no pueden **eliminarse frases** que hayan sido asociados a algún juego. Para poder modificar o eliminar, debe de hacerse seleccionando el item desde el List

Figura 5 Gestión de frases

4. Opciones del jugador

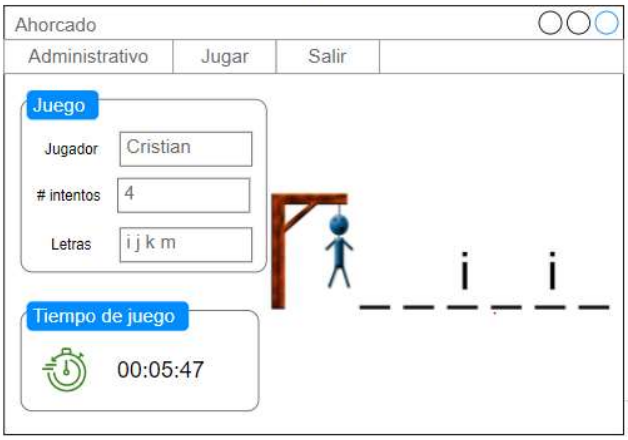
4.1. Nuevo Juego

Una vez seleccionado, con ayuda de los **widgets correctos**, solicitar el nombre del **usuario**, obtenido este, seleccionar el modo de jugo: **(Principiante o Avanzado) combo box.**

- Si selecciona **Principiante**, seleccionará al azar una palabra de la lista de palabras
- Si selecciona **Avanzado**, seleccionará al azar una frase de la lista de frases

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**.

Figura 6 Pantalla de jugador



Ya no será necesario el cronómetro

4.2. Historia del Juego

En esta opción, se mostrará en pantalla la historia de cómo se creó 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.

Figura 7 Ventana de historia del juego

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

Esta sección se dejará a la creatividad del grupo de trabajo del cómo resolver este problema, es decir el cómo guardará esta información para luego ser mostrada en la pantalla

Figura 8 Estadísticas

4.4. Ayuda

En esta opción, de la misma forma que la **sección 4.2**, mostrará una explicación al usuario de cómo jugar este juego. El archivo donde se guardará esta información se llamará **"Ayuda.txt"**

Figura 9 Ventana de ayuda

5. Salir

Finaliza el juego

6. Funciones adicionales

- **5 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. Esta implementación debe ser con CLASES**

- **5 puntos adicionales** Si se implementa con éxito el cronómetro

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**.
- Debe de crear las clases explicadas en la primera sección
- Se deben manejar **mensajes claros** al usuario **(Messagebox)**.
- Realizar **validaciones de captura de campos**. Deben hacer uso de mecanismos de validación eficientes y mostrarlos con **mensajes claros** al usuario **(Messagebox)**.
- Número de código o identificador es único y autogenerado
- Toda función ***bult-in*** están permitidas.
- El proyecto debe ser guardado en el repositorio del GitHub:
    - Taller Grupo 60: https://classroom.github.com/a/qzuXwEKL
    - Taller Grupo 61: https://classroom.github.com/a/rBJs8XXa

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). **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. El desglose de la evaluación del proyecto programado:

1. Documentación interna 4 ptos.
2. Documentación externa 6 ptos.
3. Funcionalidad 80 ptos (ver detalle en Software a Desarrollar)
4. Revisión del proyecto (según completitud del proyecto) 10 ptos.

10. Forma de trabajo

El trabajo se debe realizar de forma **individual o en parejas**.

11. Aspectos administrativos

Existen 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. Además, en el 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 o estudiantes
d. 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 **80%** **0**

Rubro	Valor	Obtenido	Observaciones
Opciones Administrativas			
Control de acceso	**10**		
Gestión de palabras	**15**		

FUNCIONALIDAD	**80%**	**0**	
Rubro	**Valor**	**Obtenido**	**Observaciones**
Gestión de frases	**15**		
Opciones de Jugador			
Nuevo Jugo	**10**		
Jugar	**20**		
Historia	**5**		
Estadísticas	**20**		
Ayuda	**5**		
Extras			
Manejo de idioma	**5**		
Estadísticas gráficas	**10**		
TOTAL	**115**	**0**	
DOCUMENTACIÓN INTERNA	**4%**		
DOCUMENTACIÓN EXTERNA	**6%**		
REVISIÓN DEL PROYECTO	**10%**		
NOTA FINAL	**100%**	**0%**	

13. Entrega del proyecto

La entrega y revisión del proyecto será el **lunes 20 de junio**, posteriormente se compartirá la hora para revisión del proyecto. **El proyecto debe de estar en el repositorio máximo una hora antes de la revisión.**

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