

Proyecto PY02 – Trivia

Curso: Estructura de Datos

Fecha de entrega: 30 de marzo del 2023, 5:00 p.m.

Objetivo

Construir un programa que presente una interfaz para jugar trivia.

Implementación

El programa debe cumplir con las siguientes reglas:

- **Carga de datos**
El juego leerá un conjunto de archivos de texto plano desde una ubicación especificada como parámetro al ejecutar el programa.
 - Todos los archivos dentro de ese directorio serán considerados.
 - El nombre de cada archivo representa el nombre de la categoría correspondiente.
- **Formato de archivo**
Cada archivo contiene preguntas de selección múltiple, con el siguiente formato por línea:
 - **P:** indica una pregunta
 - **O:** indica una opción de respuesta
 - **R:** indica el número de la respuesta correcta
- **Configuración de juego**
 - El juego puede ser de una o múltiples categorías.
 - Pueden participar entre 1 y 6 jugadores, que deben identificarse al iniciar el juego.
 - Se elige la cantidad de rondas a jugar.
 - Debe haber al menos rondas * jugadores preguntas disponibles para poder iniciar el juego.
- **Mecánica del juego**
 - En juegos multicategoría, cada jugador elige la categoría de su pregunta.

- Las preguntas se asignan aleatoriamente y no se pueden repetir si ya fueron respondidas correctamente.
 - Preguntas incorrectas pueden repetirse.
 - Puntos por pregunta correcta: 1 x número de ronda.
 - Gana quien tenga más puntos al finalizar las rondas.
 - **Desempates**
 - Si hay empate, se juegan rondas adicionales solo con los jugadores empatados.
 - Se pueden usar preguntas de categorías no seleccionadas si es necesario.
 - Si se agotan todas las preguntas, se declara empate.
-

Tecnología

- El código debe compilarse con BCC 10.2 para Windows.
 - Debe funcionar en Windows 10.
 - Las estructuras de datos deben ser lineales dinámicas implementadas por el grupo.
 - No se permite el uso de string.h ni de estructuras de datos externas.
 - Solo se permiten las librerías vistas en clase.
-

Grupos de trabajo

- Grupos de 2 a 3 estudiantes.
 - Se debe publicar la conformación del grupo en el foro del curso.
 - Trabajos de grupos no autorizados no serán evaluados.
-

Datos administrativos

- **Entrega:**
 - Vía asignación en el sitio del curso.

- Se debe entregar un enlace al repositorio con código y documentación.
 - Solo serán válidos los commits anteriores a la fecha límite.
 - Entregas tardías perderán 2 puntos por día de retraso.
 - Repositorio:
 - El repositorio debe estar compartido con el profesor y el asistente (Hansol Antay - @hros19).
 - Toda la documentación debe estar dentro del repositorio.
-

Evaluación

- Se usará la rúbrica oficial del curso.
 - El proyecto debe funcionar de forma autónoma, sin intervención de los desarrolladores.
 - La calidad del código y la documentación será evaluada cuidadosamente.
-

Recomendaciones

- Comunicación: Colabore con su equipo y consulte con otros estudiantes o el profesor.
 - Tiempo: No deje el proyecto para último momento.
 - Código limpio: Mantenga un estilo ordenado y entendible.
 - Trabajo en equipo: Dividan tareas y apoyen el progreso conjunto.
 - Práctica: Practique programación estructurada y lógica de resolución de problemas.
-

¡Mucho éxito con el trabajo!

Empiecen con tiempo suficiente, analicen y diseñen la solución antes de comenzar a programar. Un código bien planeado desde la primera línea facilitará el éxito del proyecto.