DESAFIO: Puntos Extras

Objetivo: Desarrollar un programa en Python que simule el juego de la Generala, incluyendo el uso de módulos por import, clases e interfaz por consola.

Descripción del juego:

La Generala es un juego de dados popular que involucra la estrategia y la suerte. El objetivo del juego es acumular la mayor cantidad de puntos en tres rondas. Cada ronda se compone de tres turnos, en los que cada jugador puede tirar los dados hasta tres veces para obtener una combinación de puntos.

Características del programa:

- **Simulación de tiradas de dados:** El programa debe simular el lanzamiento de cinco dados, utilizando un generador de números aleatorios.
- **Implementación de clases:** El programa debe definir clases para representar los elementos del juego, como dados, jugadores y el marcador.
- **Interfaz por consola:** El programa debe proporcionar una interfaz de usuario basada en consola para interactuar con los jugadores, mostrar las tiradas de dados, las opciones disponibles y el puntaje actual.
- **Puntuación:** El programa debe implementar las reglas de puntuación para las diferentes combinaciones posibles, incluyendo Generala, poker, full, escalera, color y tríos.
- **Rondas y turnos:** El programa debe gestionar las rondas y turnos del juego, permitiendo a cada jugador realizar sus tiradas y tomar decisiones estratégicas.
- **Finalización del juego:** El programa debe determinar el ganador al final de las tres rondas, mostrando el puntaje final de cada jugador.

Módulos por import (minimos):

- random: Para generar números aleatorios para las tiradas de dados.

Clases (minimas):

- Dado: Representar un dado con atributos como valor y cara actual.
- Jugador: Representar un jugador con atributos como nombre, puntaje total y puntaje por ronda.
- Marcador: Representar el marcador del juego, llevando la cuenta del puntaje de cada jugador y las combinaciones válidas.

Interfaz por consola:

- Utilizar funciones de impresión para mostrar mensajes, opciones y resultados al jugador.
- Solicitar entrada al jugador para indicar acciones como tirar los dados o seleccionar combinaciones.

Reglas de puntuación:

- **Generala (50 puntos):** Cinco dados del mismo valor.
- **Poker (40 puntos):** Cuatro dados del mismo valor.
- **Full (30 puntos):** Tres dados del mismo valor y dos dados de otro valor.
- **Escalera (20 puntos):** Cinco dados consecutivos.
- **Color (20 puntos):** Cinco dados de cualquier valor, pero del mismo color.
- **Trío (10 puntos):** Tres dados del mismo valor.
- **Dos pares (5 puntos):** Dos pares de dados diferentes.
- **Par (1 punto):** Un par de dados del mismo valor.

Rondas y turnos:

- El juego se compone de tres rondas.
- Cada ronda se compone de tres turnos por jugador.
- En cada turno, el jugador puede tirar los dados hasta tres veces.
- Después de cada tirada, el jugador puede elegir entre anotar una combinación, guardar algunos dados para la siguiente tirada o descartar todos los dados.

Finalización del juego:

- El juego termina después de tres rondas.
- El jugador con mayor puntaje total es el ganador.

Consideraciones adicionales:

- El programa puede incluir opciones para visualizar el historial de tiradas y combinaciones anotadas por cada jugador.
- Se puede implementar un modo de juego para un solo jugador contra la computadora.

Recursos adicionales:

- https://m.youtube.com/watch?v=RQ9Bf_Vq6cc
- https://ruibalgames.com/wp-content/uploads/2015/10/Reglas-de-juego-%E2%80%93-Generala-Real.pdf
- https://github.com/PyGithub/PyGithub

Entrega:

- El programa debe entregarse como un archivo de código Python (.py).
- Se debe incluir un breve manual de usuario que explique cómo ejecutar el programa y las opciones disponibles.

Evaluación:

El proyecto será evaluado en base a los siguientes criterios:

- **Funcionalidad:** El programa debe cumplir con todas las funcionalidades descritas en el enunciado.
- **Diseño:** El código debe estar bien estructurado, utilizando clases, funciones y módulos de manera adecuada.
- **Legibilidad:** El código debe ser legible y fácil de entender, utilizando comentarios y docstrings.
- **Robustez:** El programa debe manejar errores y excepciones de manera adecuada.
- **Documentación:** El código debe estar documentado de manera clara y concisa.

La entrega debe hacerse por mail a diego.ambrossio@unab.edu.ar con copia a angel.bianco@unab.edu.ar.

FECHA MAXIMA 12 de mayo 23:59 hs

¡Buena suerte!