PROJECTE AC6

Mòdul: MP03B - Programació

UF: UF5

Professor: Marc Callejón

Data d'entrega: 02/04/2025

Mètode d'entrega: Git

Deberéis desarrollar una aplicación que simule un juego de slots vinculado con la gestión de órdenes de producción de robots. Tendrás que utilizar el paradigma de la programación orientada a objetos, colecciones, LINQ, expresiones lambda y clases genéricas.

Tu simulación debe mostrar caracteres por consola que representarán la tirada de los modelos en tiempo real. (unas 10 llamadas a la api con Sleep variable)
https://learn.microsoft.com/es-es/dotnet/api/system.threading.thread.sleep?view=net-9.0

Ejemplo:
[R2D2] [C3PO] [BB8]

Se establecen las siguientes Reglas del Juego:

Generación de robots:

Cada tirada de la máquina de slots genera una orden aleatoria de producción.
Los símbolos del slot representan los modelos posibles: R2D2, C3PO, BB8.

Para realizar la tirada se debe consultar el siguiente endpoint:
randomnumberapi.com/api/v1.0/random?min=1&max=4&count=3

El jugador puede iniciar la tirada con una tecla

Puntuación:

Si se obtiene una combinación de 3 símbolos iguales, se genera automáticamente una nueva orden con un bonus de 10 puntos.

Si se obtienen 2 símbolos iguales, se genera una orden con un bonus de 5 puntos.

Combinaciones sin ninguna coincidencia no otorgan puntos.

Generación de órdenes: Se genera un robot de cada tipo indicado por los slots, ejemplos:

[R2D2] [C3PO] [BB8] 1 robot de cada tipo.

[R2D2] [C3PO] [R2D2] 2 robots R2D2 y 1 robot C3PO. + 5 puntos

Duración del Juego:

El juego tiene una duración de 10 tiradas Se contabilizan los puntos acumulados al finalizar la partida y el número de robots generados con los siguientes puntos: (3 puntos para R2D2, 2 puntos para C3PO, 1 punto para BB8)

Clases Genéricas:

Se debe implementar una clase genérica que permita gestionar la creación de órdenes de producción.

La clase genérica debe aceptar diferentes tipos de modelos y almacenar las órdenes en una colección.

La clase debe incluir métodos para agregar órdenes, consultar las órdenes generadas y calcular puntos totales, puntuación máxima y puntuación mínima.

Al finalizar la partida:

Se deben mostrar todos los datos almacenados en la clase genérica, consulta la clase File de la documentación de .net para guardar en un fichero de texto las puntuaciones y el nombre de los usuarios. Muestra el top 3.

https://learn.microsoft.com/en-us/dotnet/api/system.io.file?view=net-9.0

Clases:

C3PO:

Propiedades:

```
int: nombre (puedes usar un id, controla en número de robots
generados y genéralo de manera incremental),
string: modelo,
date: fecha de creación (aleatorio)
```

Métodos:

```
NumberOfRepairs() (Número aleatorio de reparaciones)
ShowData()
```

R2D2:

Propiedades:

int: nombre (puedes usar un id, controla en número de robots generados y genéralo de manera incremental),
string: modelo,
date: fecha de creación (aleatorio)
int: número de versión

Métodos:

NumberOfBattles() (Número aleatorio de combates)
ShowData()

BB8:

Propiedades:

int: nombre (puedes usar un id, controla en número de robots generados y genéralo de manera incremental),
string: modelo,
date: fecha de creación (aleatorio)
float: número de versión

Métodos:

NumberOfFlights() (Número aleatorio de combates)
NumberOfRepairs() (Número aleatorio de reparaciones)
ShowData()