using System.Collections.Generic; // Importa la librería System.Collections.Generic, que proporciona clases para colecciones genéricas, como listas.

using UnityEngine; // Importa el espacio de nombres UnityEngine, necesario para utilizar funcionalidades básicas de Unity, como 'Debug'.

public static class DataTranslator // Declara la clase estática DataTranslator. Al ser estática, no puede ser instanciada, solo puede accederse mediante su nombre.

{

  public static List<List<float>> TranslateData(List<List<float>> rawData, string exerciseName) // Define un método estático llamado TranslateData, que toma dos parámetros: 'rawData' (una lista de listas de flotantes) y 'exerciseName' (un string).

  {

    if (rawData == null || rawData.Count != 8) // Verifica si 'rawData' es nula o si su cantidad de listas no es igual a 8.

    {

      Debug.LogError("Los datos brutos deben contener exactamente 8 listas."); // Si la condición anterior es verdadera, muestra un error en la consola de Unity.

      return rawData; // Devuelve los datos originales si la validación falla.

    }

    List<List<float>> translatedData = new List<List<float>>(); // Crea una nueva lista de listas de flotantes, llamada 'translatedData', para almacenar los datos traducidos.

for (int i = 0; i < 8; i++) // Un bucle que recorre un rango de 0 a 7 (8 elementos en total).

{

translatedData.Add(new List<float>()); // Agrega una nueva lista vacía de flotantes en 'translatedData' para cada índice (total 8 listas vacías).

}

bool isPendulo = exerciseName.ToLower().Contains("péndulo"); // Verifica si el nombre del ejercicio contiene la palabra "péndulo" (ignorando mayúsculas y minúsculas). Esto se usa para identificar si el ejercicio es un "péndulo".

if (isPendulo) // Si el nombre del ejercicio contiene "péndulo":

{

translatedData[0] = new List<float>(rawData[0]); // Asigna la primera lista de 'rawData' a la primera lista de 'translatedData' (equivalente a Pitch1).

translatedData[1] = new List<float>(rawData[1]); // Asigna la segunda lista de 'rawData' a la segunda lista de 'translatedData' (equivalente a Roll1).

translatedData[2] = new List<float>(rawData[2]); // Asigna la tercera lista de 'rawData' a la tercera lista de 'translatedData' (equivalente a Yaw1).

}

else // Si el nombre del ejercicio no contiene "péndulo":

{

translatedData[0] = new List<float>(rawData[1]); // Asigna la segunda lista de 'rawData' a la primera lista de 'translatedData' (equivalente a Roll1).

translatedData[1] = new List<float>(rawData[0]); // Asigna la primera lista de 'rawData' a la segunda lista de 'translatedData' (equivalente a Pitch1).

translatedData[2] = new List<float>(rawData[2]); // Asigna la tercera lista de 'rawData' a la tercera lista de 'translatedData' (equivalente a Yaw1).

}

translatedData[3] = new List<float>(rawData[3]); // Asigna la cuarta lista de 'rawData' a la cuarta lista de 'translatedData' (equivalente a Pitch2).

translatedData[4] = new List<float>(rawData[5]); // Asigna la sexta lista de 'rawData' a la quinta lista de 'translatedData' (equivalente a Roll2).

translatedData[5] = new List<float>(rawData[4]); // Asigna la quinta lista de 'rawData' a la sexta lista de 'translatedData' (equivalente a Yaw2).

translatedData[6] = new List<float>(rawData[6]); // Asigna la séptima lista de 'rawData' a la séptima lista de 'translatedData' (equivalente a EMG).

translatedData[7] = new List<float>(rawData[7]); // Asigna la octava lista de 'rawData' a la octava lista de 'translatedData' (equivalente a tiempo).

return translatedData; // Devuelve la lista 'translatedData' con los datos correctamente traducidos y organizados.

}

}