

Exercice 03B – Bowling

Objectif

Poursuivre l'apprentissage du **mocking** via la réalisation d'une série de tests d'une application de **bowling virtuel** jouant sur l'**aléatoire**.

Sujet

Sans en modifier le code ni le fonctionnement, vous devrez réaliser une série de tests unitaires basés sur la classe **Frame.cs** fournie. Pour ce faire, vous vous servirez du framework **NUnit** ainsi que du package NuGet **Moq** dans le but de proposer plusieurs tests visant à expérimenter la robustesse de l'application. Pour simuler le jeu et éviter les valeurs aléatoires, vous mockerez le fonctionnement de la classe générant les lancers, en passant notamment par son interface (**IGenerator**).

Ces tests devront permettre de vérifier :

- S'il s'agit d'une série standard (round 1 par exemple)
 - Le **premier** lancer d'une série doit augmenter le **score** de la série
 - Le **second** lancer d'une série doit augmenter le **score** de cette série
 - En cas de **strike**, il ne doit **pas être possible de lancer de nouveau** au cours de cette même série
 - En cas de lancers standards, il ne doit **pas être possible de lancer plus de 2 fois**
- S'il s'agit d'une série finale (dernier round)
 - En cas de **strike**, il doit être possible de **lancer une nouvelle fois** au cours d'une série
 - En cas de **strike** puis de lancer, le **score** est censé augmenter en accord avec le résultat du lancer
 - En cas de **strike** puis d'un lancer, il doit être possible de **lancer une nouvelle fois**
 - En cas de **strike** puis de lancer, le **score** est censé augmenter en accord avec le résultat
 - En cas de **spare**, il doit être possible de **lancer une nouvelle fois** au cours d'une série
 - En cas de **spare** puis de lancer, le **score** est censé augmenter en accord avec le résultat du lancer
 - En cas de **lancers standards**, il ne doit pas être possible de **lancer plus de 4 fois**

```
27 références
public class Frame
{
    private int _score;
    private List<Roll> _rolls;
    private bool _lastFrame;
    private IGenerator _generator;

    5 références | 5/5 ayant réussi
    public int Score { get; }

    1 référence
    public bool LastFrame { get => _lastFrame; set => _lastFrame = value; }

    19 références | 10/10 ayant réussi
    public List<Roll> Rolls { get => _rolls; set => _rolls = value; }

    11 références | 11/11 ayant réussi
    public Frame(IGenerator g, bool lastFrame) { }

    3 références
    private void MakeRoll(int max) { }

    11 références | 11/11 ayant réussi
    public bool Roll() { }
}
```