

Exercice 03 – Le Pendu

Objectif

Appréhendez l'utilisation du **mocking** dans la réalisation de tests unitaires de type **NUnit** d'une classe **dépendant d'une autre classe pour son fonctionnement**.

Sujet

Vous devrez réaliser une série de tests unitaires basés sur le projet du pendu (en mode console). Pour ce faire, vous devrez utiliser le **mocking** en vous servant du package Nuget **Moq** qui vous permettra de simuler des parties afin de tester de façon unitaire votre classe **sans avoir à faire des tests d'intégration ou d'interface**.

Vous devrez ainsi tester après avoir mocké le fonctionnement du générateur de mots les trois méthodes de base de la classe **LePendu.cs**, et **ce sans modifier le code fourni**, qui sont :

- **GenererMasque()**
 - Vérifier que **le masque se génère correctement** lors de l'utilisation de cette méthode, en produisant le masque adapté au mot généré par la classe **GenerateurDeMot.cs**
- **TestChar()**
 - Vérifier que le **retour de la méthode est adapté en fonction de la saisie utilisateur** (un caractère faisant partie du mot à trouver devra retourner true alors qu'un caractère n'en faisant pas partie devra retourner false)
 - Vérifier que le **nombre d'essais restant diminue de façon à respecter les règles du pendu** (lors de la saisie d'un bon caractère, le nombre d'essais ne doit pas descendre, alors que lors de la saisie d'un caractère ne faisant pas partie du mot à trouver, le nombre d'essais doit descendre correctement)
 - Vérifier **la modification du masque lors de la saisie d'un caractère** faisant ou non partie du mot à trouver (le masque ne doit pas être modifié lors de la saisie d'un caractère ne faisant pas partie du mot à trouver mais doit être changé dans le cas contraire)
- **TestWin()**
 - Une partie dans laquelle l'utilisateur réalise une série de saisies utilisateurs amenant à un **scénario gagnant doit retourner un booléen de confirmation** de sa victoire (de valeur **true**).
 - Dans le cas contraire, le **booléen doit être de valeur false** pour infirmer de sa victoire et permettre ainsi la poursuite de la partie.
 - Dans le cas où un **utilisateur saisi trop de lettres ne faisant pas partie du mot recherché**, alors il doit **perdre**, quand bien même il finirait par saisir le bon mot à la fin.

```
3 références
public class LePendu
{
    private int _nbEssai;
    private string _masque;
    private string _motATrouve;

    3 références
    public int NbEssai { get => _nbEssai; }
    3 références
    public string Masque { get => _masque; }
    4 références
    public string MotATrouve { get => _motATrouve; }

    1 référence
    public LePendu()
    {
        _nbEssai = 10;
    }

    11 références
    public bool TestChar(char c) {...}

    3 références
    public bool TestWin() {...}

    1 référence
    public void GenererMasque(IGenerateur generateurDeMot) {...}
}
```