Guide de bonnes pratiques en C# et en développement de façon générale :

Que ce soit pour développer vos projets en DotNet ou en Javascript, ces bonnes pratiques et règles vous seront utiles si vous êtes plusieurs à travailler sur un même projet ou que votre code est transmis à un autre développeur.

1) Les conventions de style :

Les conventions de style permettent d'attribuer la même typographie entre les variables, champs, fonction, etc.

Il existe 3 grandes façons d'écrire nos accesseurs :

- Pascal case : 1ères lettres de chaque mot sont en majuscules. Exemple : "BackColor"

- Camel case : 1ère lettre est minuscule, puis chaque 1ère lettre de chaque mot est majuscule. Exemple : "backColor"

- Uppercase : Toutes les lettres sont en majuscules. Exemple : "IO"

Vous trouverez les conventions pour chaque type d'accesseurs dans le document ci-dessous (en DOTNET) :

https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-1.1/x2dbyw72%28v%3dvs.71%29

Ces conventions peuvent êtres réutilisables dans tous les langages.

2) Sensibilité à la casse :

La première règle consiste à ne pas créer de noms de variable qui se différencie d'un autre uniquement par des majuscules : Par exemple : "var myVar = 0;" et "var MYVAR = 0;" sont faux parce que ce qui les différencient sont uniquement des majuscules/minuscules.

Cette règle marche aussi pour les paramètres des fonctions, et le nom des fonctions.

3) Lisibilité du code et commentaires

Cette partie traite de toutes les petites astuces à faire lorsque on code (qui deviendront des réflexes au fur et à mesure).

*a) Fluidité du code*

Pour être compréhensible, un code de projet doit être :

- bien écrit, pas de fautes d'orthographe ou de grammaire dans les fonctions, variables ou même les commentaires.

- l'indentation du code : décaler son code petit à petit pour le rendre lisible. La plupart des IDE (Interface de développement) le font automatiquement lorsque on retourne à la ligne dans une fonction par exemple.

- Utiliser un IDE permettra de s'y retrouver plus facilement dans son projet et de bénéficier de fonctionnalités

*b) Gestion des commentaires pour autrui et soi-même*

Les commentaires font aussi partie de la vie d’un développeur malgré le fait que ceux-ci sont généralement moins séduisants à produire.

Ils existent 2 types de commentaires :

- Ceux qui se trouvent avant chaque fonctions, méthodes qui permettent de définir la fonction, de quand elle date, qui l’a écrite. Ils sont généralement multilignes pour avoir la place verticalement pour bien tout expliquer.

- Ceux qui explique des fonctionnalités bien précises et qui sont trouvables sur la ligne même.

Ce sont des bonnes pratiques qui sont enseignées et utilisées par la plupart des développeurs. Cela permet de gagner du temps et de la compréhension lorsque on travaille sur un projet à plusieurs ou que nous utilisons du code d’une autre personnes.

https://blog.cellenza.com/dotnet/organiser-son-code-dans-sa-solution-visual-studio/

https://anceret-matthieu.fr/2018/01/16/csharp-bonnes-pratiques/

https://framablog.org/2009/11/21/commentaires-code-source-programmation/