

Correction de `BMPReader`

Gestion des arguments

Il suffit dans un premier temps de tester le cas où : il n'y a pas d'argument dans `string[] args` ou plus de 2 arguments. On a le nombre d'élément dans `args` grâce à la propriété `count`. Afficher le message d'erreur si le nombre d'argument n'est pas correct avec `Console.WriteLine()` et quitter (*return débranchant*).

Ensuite on essaye d'ouvrir le fichier en appelant le constructeur de la classe `public FileStream(string path, FileMode mode)`, si le fichier n'existe pas une exception sera déclanchée : `FileNotFoundException`. On gère ce cas avec un bloc `try-catch` pour afficher un message d'erreur et quitter.

Vérification du *magic number*

Après avoir initialisé un `BinaryReader` en utilisant le `FileStream`. On vérifie le *magic number*.

Pour vérifier la *magic number* la méthode `ReadChars` pour lire 2 caractères semble appropriée (on peut aussi utiliser deux fois `ReadChar`).

On teste les deux caractères, s'ils ne correspondent pas à un *magic number* de fichier `BMP` i-e `BM`, alors on affiche le message d'erreur et on quitte.

Affichage des informations

Par le même procédé que ci-dessus on utilisera les méthodes

`ReadInt32`, `ReadInt16` pour récupérer les entiers des champs

`Image Width`, `Image Height` et `Bits per Pixel`, les autres champs sont simplement ignorés car leurs informations ne nous intéressent pas dans ce TP.

L'utilisation de `BinaryReader` met à jour le curseur de lecture du fichier.

En fonction de la taille des champs (à ignorer) on fera attention à rester aligné. Sinon les données récupérées seront fausses.

Une fois les informations affichées dans la console, on arrive à la fin du programme, ça quitte.