

Application Color contrast calculator

15 DECEMBRE

IUT Dijon

Créé par : Alexandre Guidet

Table des matières	
Description du besoin	3
Références.....	4
Analyse.....	5
Cas d'utilisation	5
Classes	5
Maquettes	6
Séquence générale d'utilisation	7
Analyse technique	7
Conception architecturale	8
Conception détaillée.....	8
Fonctionnalité : choisir une couleur	9

Description du besoin

Le but est de développer une petite application Windows simple, fonctionnant sous Windows 10, afin de calculer facilement le contraste entre deux couleurs, obtenues grâce à une « pipette », dans le but d'indiquer si le contraste est suffisant, au sens des préconisations du WCAG sur l'accessibilité.

Une application équivalente (mais sur le web) : [Contrast Finder, trouvez le bon contraste pour vos couleurs, accessibilité web \(WCAG / RGAA\) \(contrast-finder.org\)](https://contrast-finder.org)

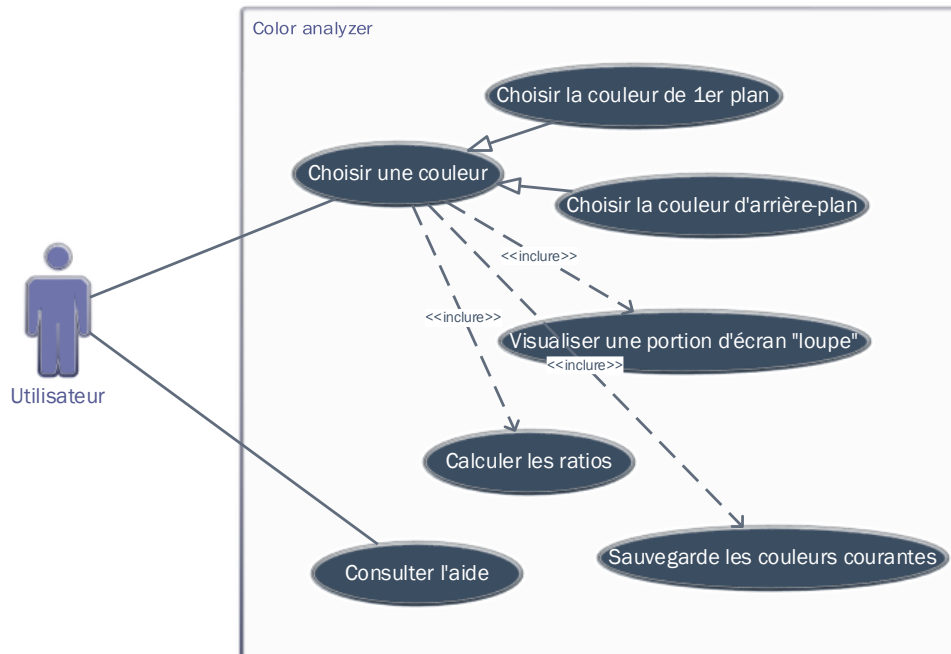
Références

1. Chand, Manesh. **WPF Magnifier control**. *C# corner*. [En ligne] 29 09 2012.
<https://www.c-sharpcorner.com/UploadFile/mahesh/wpf-magnifier-control/>.
2. multiple. **How to build a magnifier in WPF C# 4.0**. *Stack Overflow*. [En ligne] 2011.
<https://stackoverflow.com/questions/6713853/how-to-build-a-magnifier-in-wpf-c-sharp-4-0>.
3. authors, Multiple. **Capture the screen shot using .NET**. *Stack Overflow*. [En ligne] 2011. **c# - Capture the screen shot using .NET - Stack Overflow**.
4. Planetcalc. **Luminance relative et rapport de contraste**. *Planetcalc*. [En ligne]
<https://fr.planetcalc.com/7779/>.
5. Egorozh. **WPF ColorPicker**. *GitHub*. [En ligne]
<https://github.com/egorozh/Egorozh.ColorPicker>.
6. Guidet, Alexandre. **Sérialisation JSON en .NET**. *Base de connaissances IUT informatique Dijon*. [En ligne] <https://iutdijon.u-bourgogne.fr/intra/info/bc/serialisation-json-en-net/>.
7. —. **Sauvegarde fichiers dans applications .NET**. *Base de connaissances IUT informatique Dijon*. [En ligne] <https://iutdijon.u-bourgogne.fr/intra/info/bc/sauvegarde-fichier-dans-les-applications-net/>.
8. W3C. **WCAG**. *W3C*. [En ligne] <https://www.w3.org/Translations/WCAG20-fr/#visual-audio-contrast>.

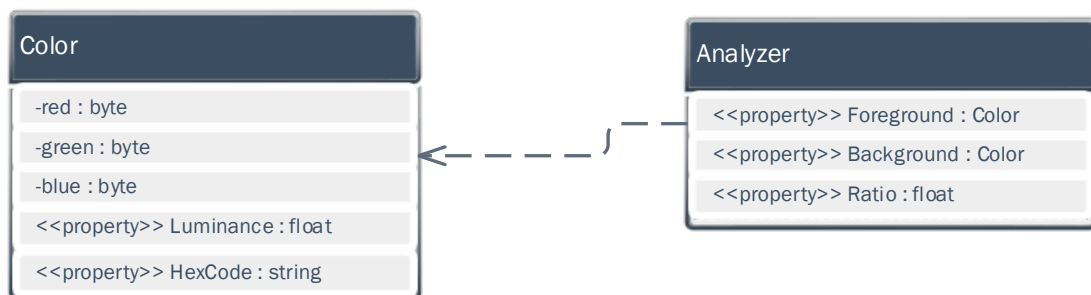
Dépôt du projet : <https://github.com/dept-info-iut-dijon/color-constrast/>

Analyse

Cas d'utilisation



Classes



Maquettes

Color Contrast analyzer

Couleur de premier plan

Rouge

128

#808080

Vert

128

Bleu

128

Choose

Pick

Couleur d'arrière-plan

Rouge

128

#808080

Vert

128

Bleu

128

Choose

Pick

Ratio de contraste :

4,5:1

Recommandations WCAG :

Texte petit

Exemple de texte

✓

✗

AA

AAA

Texte grand


Exemple de texte

✓

✗

A propos de...

Analyseur de contraste de couleurs



Version 1.0

Alexandre Guidet - 2021

<https://github.com/dept-info-iut-dijon/color-contrast/>

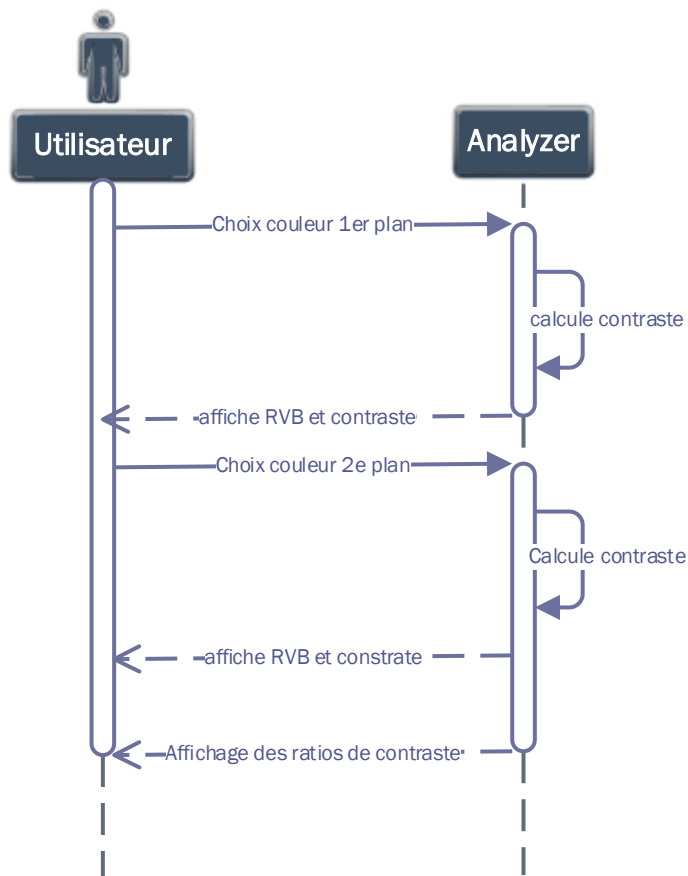
Utilise le composant ColorPicker par Egorozh

<https://github.com/egorozh/Egorozh.ColorPicker>

Le but de l'application est de calculer le taux de contraste entre une couleur d'avant-plan et une couleur d'arrière-plan, afin de pouvoir respecter les recommandations WCAG sur l'accessibilité. Différentes possibilités sont offertes pour choisir la couleur (saisie des composantes RVB, du code HTML, choix dans une palette, « pipette » sur l'écran...), le calcul du ratio est automatique. Un texte d'exemple utilisant les couleurs choisies est fourni. Des icons indiquent directement si les recommandations WCAG sont respectées ou non.

6

Séquence générale d'utilisation



Analyse technique

L'application nécessite de savoir capturer, à la souris, la couleur d'une autre application (mode « pipette », avec loupe intégrée (1) (2)) pour fournir une couleur RVB standard. Il est également possible de régler la couleur avec les composantes directement (rouge/vert/bleu) voire la couleur HTML, ou un choix via la boîte de dialogue standard de choix de couleur. La capture de l'écran (3) doit permettre de prendre une couleur dans n'importe quelle application, y compris le bureau windows. Une fois la couleur rvb déterminée, il faut savoir comment calculer le contraste (4). Il est également possible d'utiliser un composant de type « color picker » tiers tel que (5).

L'application sera développée en C#, en utilisant .NET et le framework WPF pour l'affichage. Les données (couleurs choisies) seront simplement sauvegardées en format texte (JSON) dans le dossier des données de l'application.

L'application est déposée sur GitHub :

[dept-info-iut-dijon/color-contrast: Simple color contrast analyzer, in C# \(github.com\)](https://github.com/dept-info-iut-dijon/color-contrast)

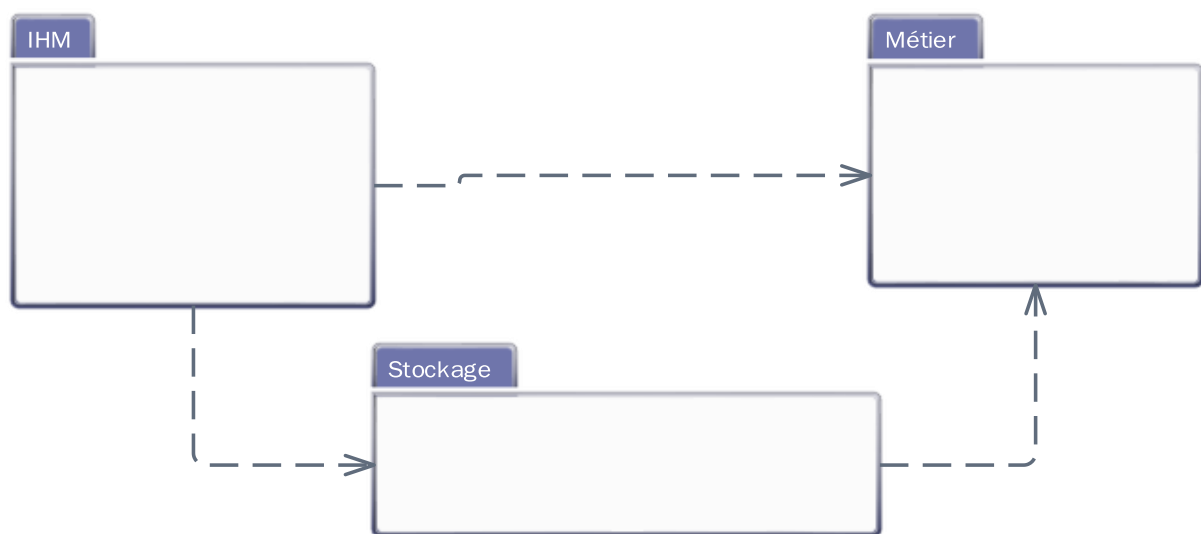
Conception architecturale

Nous allons utiliser le modèle d'architecture MVVM qui présente l'avantage d'être natif en WPF.

L'application utilisera une partie IHM et une partie métier.

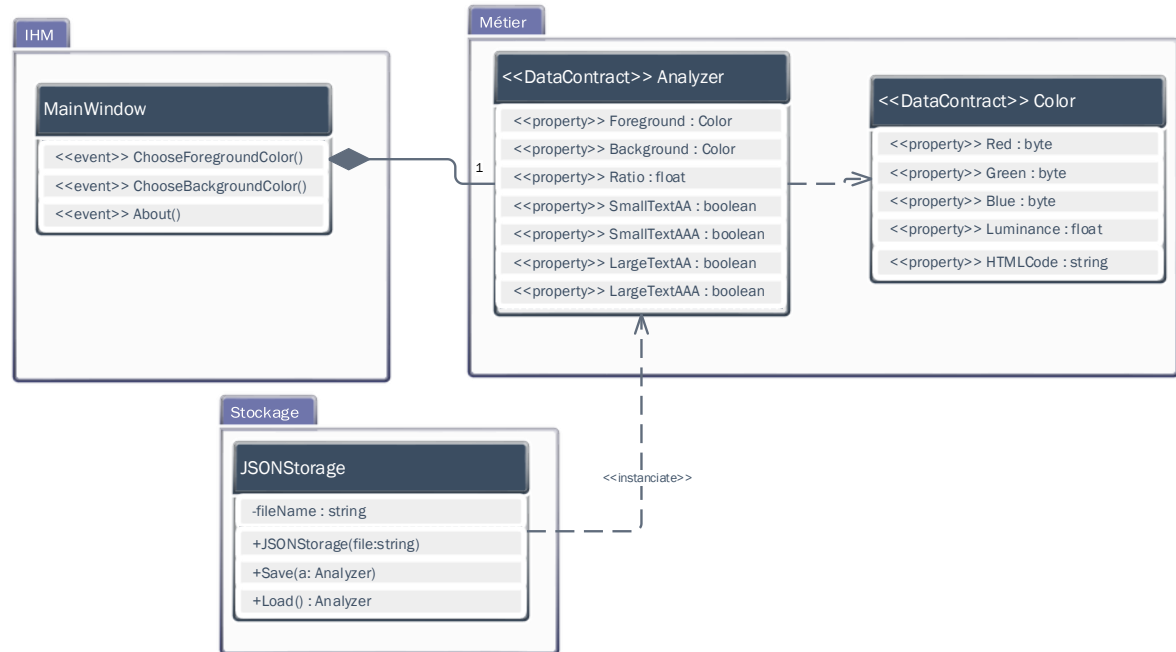
La partie IHM contiendra la gestion de l'écran, la gestion de la capture (pipette) et de la loupe. Les « VM » seront placés dans cette partie.

La partie métier contiendra les calculs RVB / contraste, et donc le « modèle ». La partie stockage ne contient que le code nécessaire pour la persistance des données.

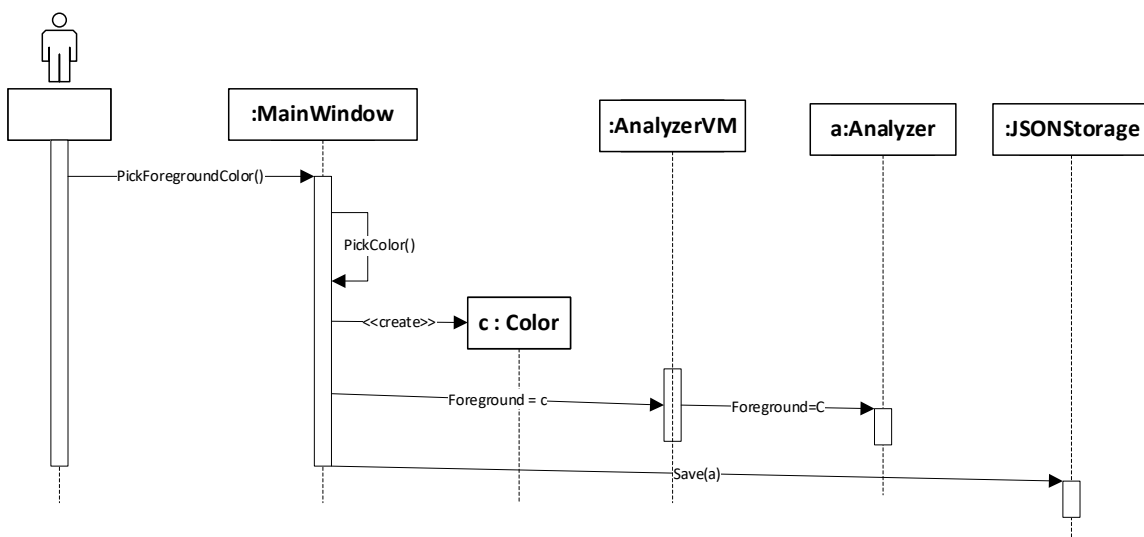


Conception détaillée

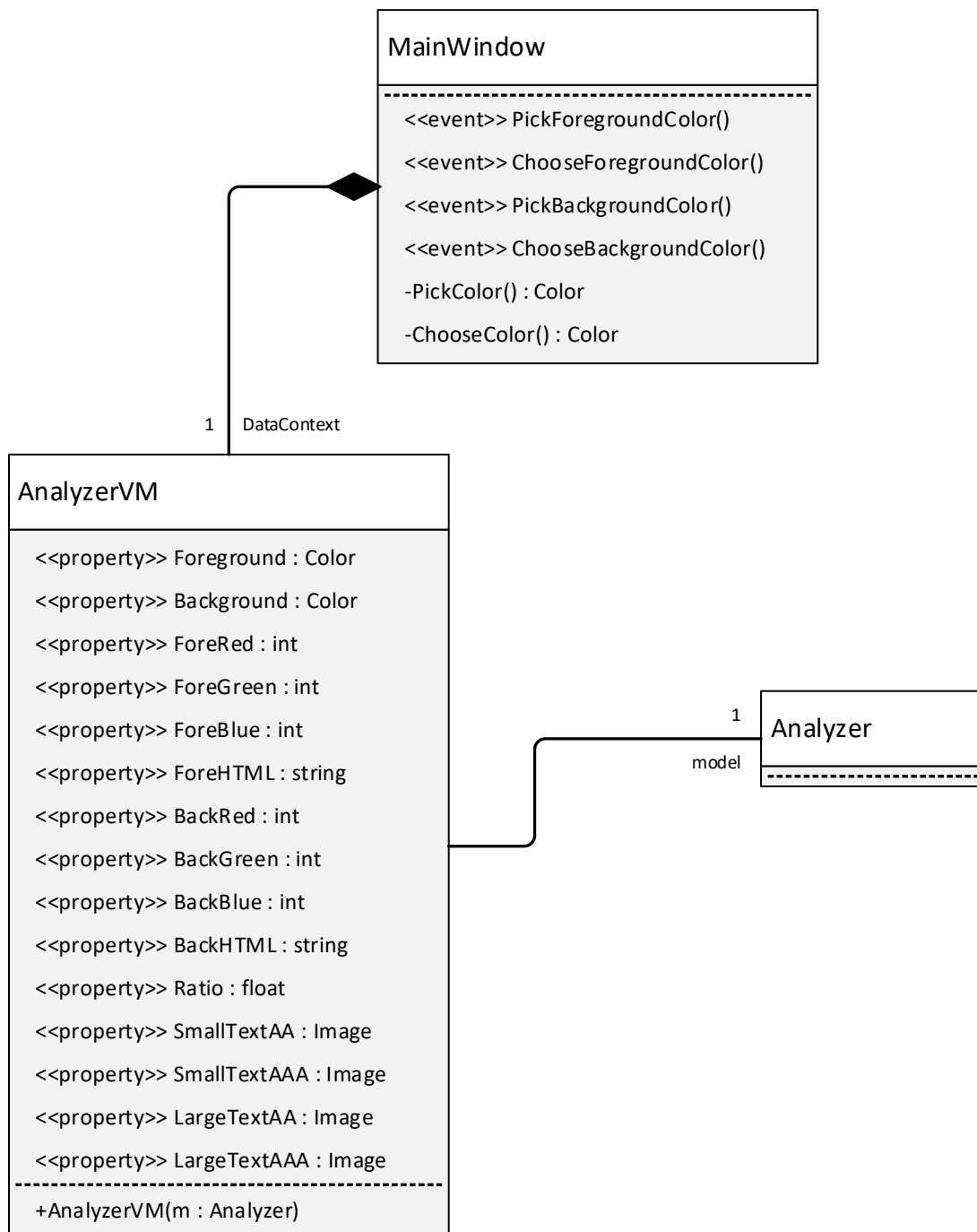
Au départ, il va y avoir une classe dans chaque couche, qui sera la façade de cette dernière.



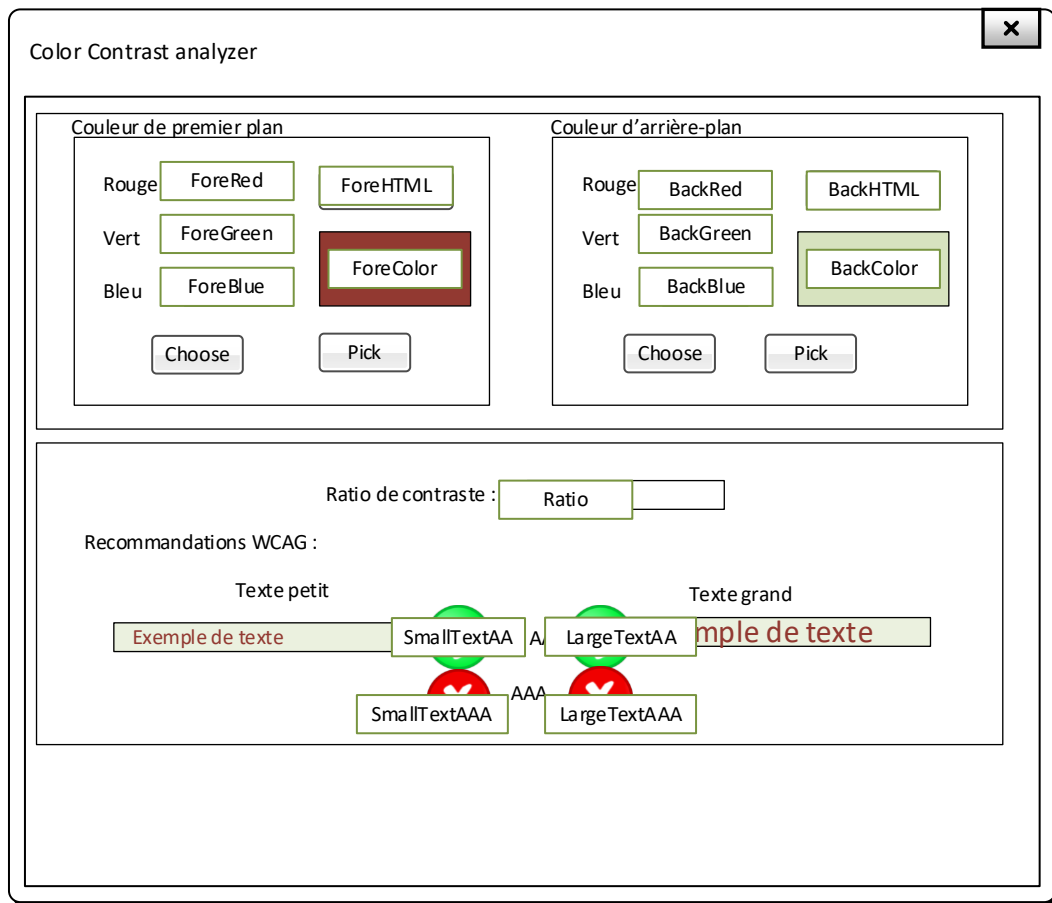
Fonctionnalité : choisir une couleur



La séquence sera la même pour la couleur de fond, une factorisation avec une méthode privée est donc obligatoire. Pour synchroniser efficacement la vue avec les modifications du modèle, suivant MVVM, il sera créé un « vue modèle ».



Les contrôles de la vue principale seront reliés par « data binding » à une instance de AnalyzerVM et à ses (nombreuses) propriétés.



Pour la fenêtre « à propos » elle ne contient pratiquement que du texte statique, sauf le numéro de version qui serait intéressant d'aller chercher automatiquement. Un « `viewmodel` » va donc être ajouté pour fournir ce numéro de version.