

# Interface d'optimisation du choix de couleurs

## Interface Homme Machine - Master 1

François LEPAN  
Benjamin VAN RYSEGHEM

25 mars 2013

### Introduction

Ce rapport fait état de l'utilisation ainsi que de la description de l'interface. Nous verrons comment l'utiliser, quels procédés ont été mis en oeuvre ainsi qu'un UML résumant les classes principales ainsi que leurs fonctionnalités.

## 1 L'interface

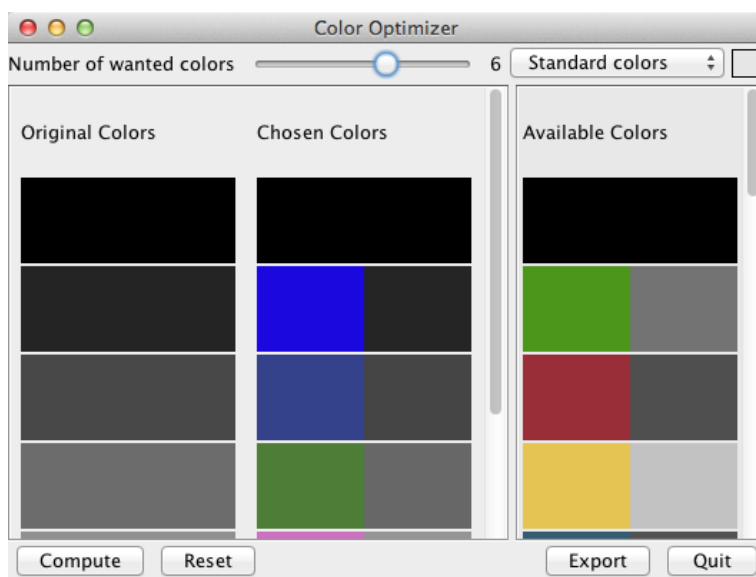


FIGURE 1 – L'interface d'optimisation du choix de couleurs

L'interface (*cf.* Fig. 1) choisie est très basique : il y a un curseur, une liste et le presse-papier de couleurs en haut, trois colonnes au centre et quatre boutons en bas.

Le curseur en haut permet de choisir le nombre de nuances de gris que l'application doit générer et la liste permet de choisir le type de couleurs à générer.

Au centre les deux colonnes à gauche représentent les couleurs que l'utilisateur va choisir. La colonne de gauche contient les nuances de gris générées par l'application et la colonne au centre contient le choix de l'utilisateur.

Pour choisir de nouvelles couleurs l'utilisateur n'aura qu'à prendre les couleurs depuis le panaché de couleurs proposées. Ce panaché est symbolisé par une liste qui se situe dans la colonne à droite de l'interface.

En bas se trouve les différents boutons. Les deux boutons de gauche servent à modifier la colonne du centre.

Le bouton *Compute* permet de calculer des couleurs ayant une nuance assez marquée. Le bouton *Reset* permet de vider cette colonne.

Ensuite le bouton *Export* copie dans le presse-papier les valeurs des couleurs choisies par l'utilisateur.

Et enfin le bouton *Quit* permet de quitter l'application.

## 2 L'utilisation

L'interface est très simple d'utilisation. L'utilisateur choisit le nombre de couleurs dont il a besoin via le curseur, créant ainsi un panaché de nuances de gris équitablement réparties, ainsi qu'un choix prédéfini pour chacune de ces couleurs, en fonction du générateur de couleurs choisi.

Si le choix calculé par l'application ne lui plaît pas, il peut à tout moment choisir de nouvelles couleurs via de simple glisser/déposer ou copier/coller parmi le panaché de choix<sup>1</sup>.

Il peut grâce aux boutons *Compute* et *Reset* recalculer ou remettre à zéro le choix des couleurs. Quand l'utilisateur est satisfait, il peut exporter son choix dans le presse-papier grâce au bouton *Export*.

L'utilisateur peut ensuite quitter l'application, soit en utilisant l'icône de la fenêtre, soit en pressant le bouton *Quit*.

## 3 Calcul du plus proche niveau de gris

L'algorithme de calcul du niveau de gris le plus proche est assez simple : pour l'ensemble des couleurs générées, nous calculons le niveau de gris, et sélectionnons la couleur étant la plus proche du niveau de gris souhaité :

$$\delta = | \text{niveau\_de\_gris} - (0.3 \times \text{couleur.rouge} + 0.59 \times \text{couleur.vert} + 0.11 \times \text{couleur.bleu}) |$$

## 4 Les procédés utilisés

Nous avons mis en place un glisser/déposer, un copier/coller ainsi que la création de composants *DualColorComponent* et sa sous classe *DraggableDualComponent*.

Le glisser/déposer nous paraissait essentiel pour que cette interface soit la plus facile à utiliser, plus intuitive. En effet l'utilisateur pourra prendre des couleurs parmi le panaché de couleurs pour les mettre dans la colonne de choix par un simple glisser/déposer.

Le copier/coller est un plus pour les utilisateurs qui préfèrent utiliser leur clavier. Un raccourci clavier permet également de coller rapidement plusieurs fois la même couleur.

---

1. colonne de droite

Nous avons aussi créer un composant nommé *DualColorComponent*. C'est ce composant qui alimente la colonne centrale ainsi que la colonne de droite. Cette classe est constituée de deux JPanel : l'un contenant la couleur, et l'autre son niveau de gris. Cette élément est très intéressant car il permet à l'utilisateur de toujours avoir une vue direct de la couleur et de son niveau de gris sans effort. Cela lui permettra de faire des choix plus rapide et précis, en connaissant de facto l'impact de son choix sur la représentation en niveau de gris.

Sa sous classe *DraggableDualComponent* permet de plus de pouvoir glisser la couleur vers un autre *DualColorComponent* (dans la liste de choix, ou le presse-papier de couleur).

## Conclusion

Cette interface répond bien au problème posé qui est d'aider l'utilisateur dans son choix de couleurs pour le passage en niveau de gris. L'interface permet de le faire sans effort (ou presque) et en un temps minimum grâce aux procédés mis en place.

## 5 Annexe

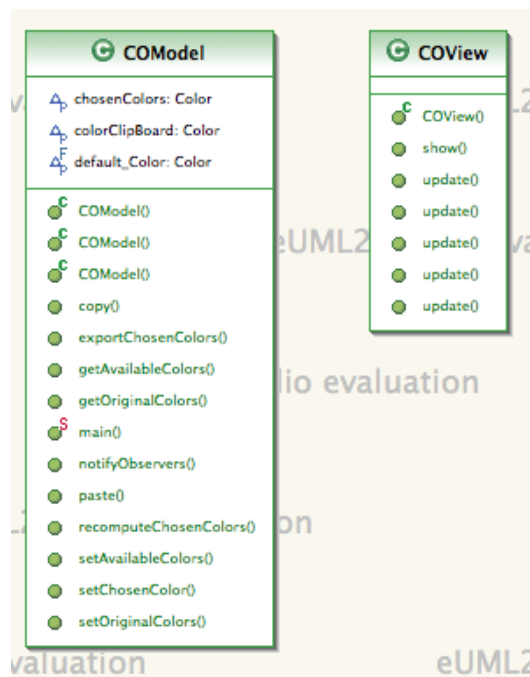


FIGURE 2 – UML du package core : le modèle et la vue

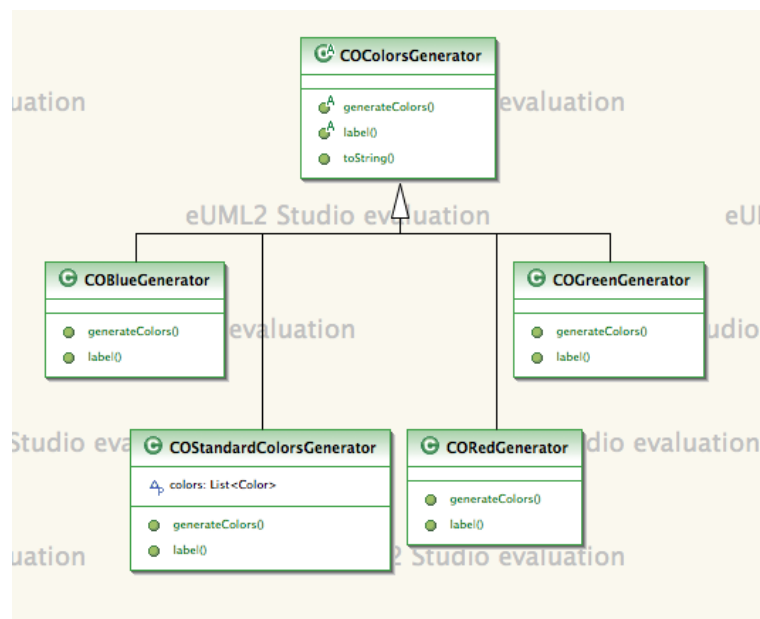


FIGURE 3 – UML du package `color.generator` : les classes qui s’occupent de générer les nuances de gris avec leurs couleurs respective

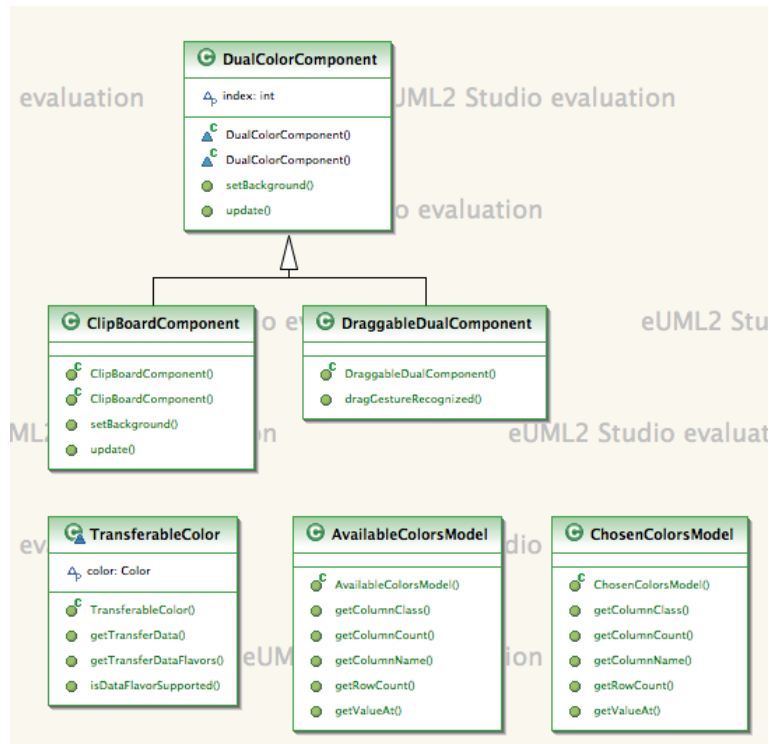


FIGURE 4 – UML du package support : les classes qui permettent le cliquer/glisser et copier/coller des couleurs ainsi que la classe DualColorComponent (toutes les classes ne sont pas présentes justes les essentiels)

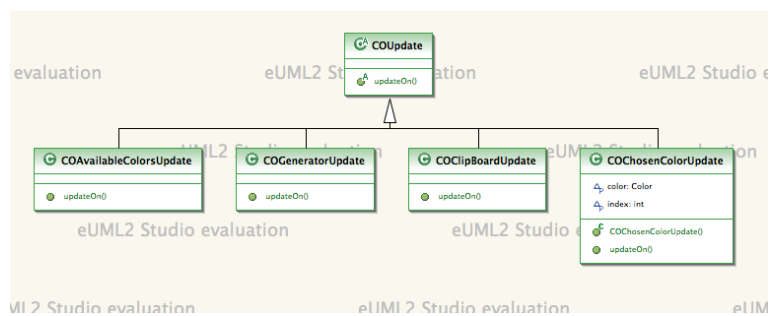


FIGURE 5 – UML du package update : classes permettant la mise à jour des composants