

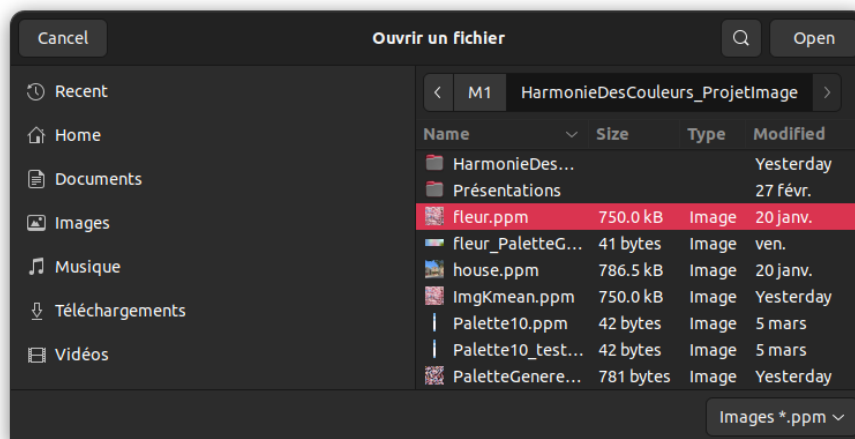
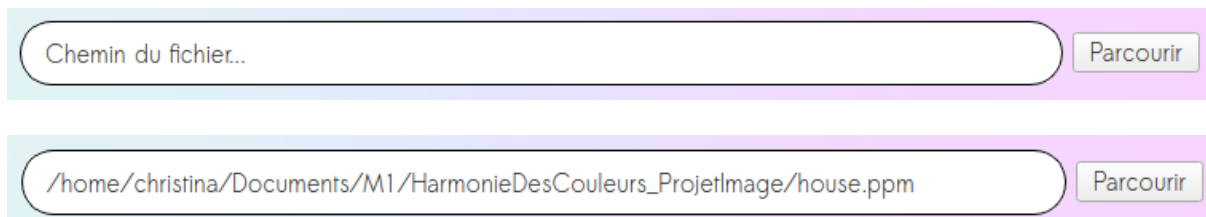
Harmonie des Couleurs

Compte Rendu 3

Développement de l'application avec Qt Creator :

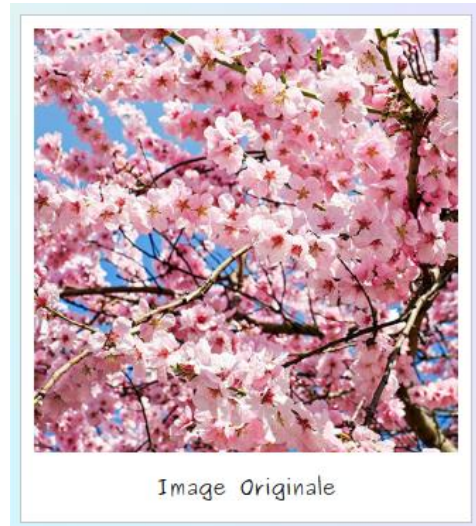
Nous avons décidé que nous utiliserions Qt Creator pour notre interface graphique. Nous avons choisi Qt Creator plutôt qu'un autre car Thibault était plus familier à cet environnement de développement intégré. Cependant, pour nous répartir les tâches, c'est Christina qui a mis en place les éléments nécessaires à la base de notre application, donc une découverte totale de sa part.

La première étape a été de créer un bouton « parcourir » permettant d'ouvrir une fenêtre d'explorateur de fichier `QFileDialog` afin de sélectionner parmi nos fichiers une image dans le format choisi (ici *.ppm). Nous avons ensuite créé une barre de texte `QString` afin que, après validation de l'étape précédente, nous ayons le chemin du fichier image. Nous avons utilisé des `QHBoxLayout`, `QPushButton` et des `QLabel` pour faire cela.



À noter que l'on a mis une option permettant de ne montrer que les fichiers au format *.ppm et que la fenêtre met en effet que les fichiers correspondants.

Dans un second temps, par les moyens de `QFrame` et `QLabel`, nous avons fait un design de type « polaroid ». Lorsqu'il n'y a pas d'image sélectionnée, nous avons l'inscription « *Choisir une image...* » dans un fond grisé, dans le cas contraire, l'image apparaît aux bonnes dimensions. Une inscription « *Image Originale* » apparaît en légende afin de faire comprendre à l'utilisateur que cette image sera celle comparée.

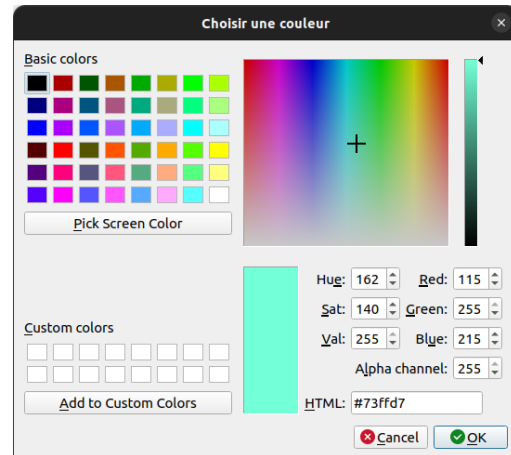
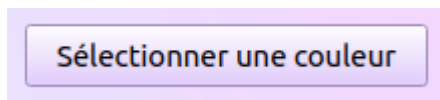
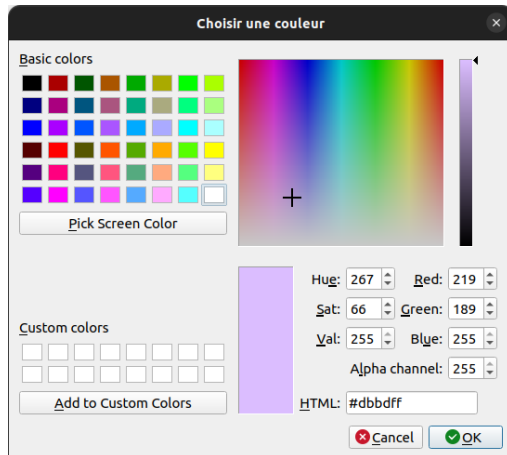


Nous avons également implémenté un bouton « Sélectionner une couleur » afin d'ouvrir la palette de couleurs et choisir arbitrairement une couleur. Le bouton prend ainsi en fond la couleur de la valeur sélectionnée. Cela est permis grâce notamment à `QColorDialog`. Il est à noter qu'avec Ubuntu, il est nécessaire de désactiver l'interface native avec la `QColorDialog::DontUseNativeDialog` afin d'avoir un plus large et plus précis choix de couleurs.



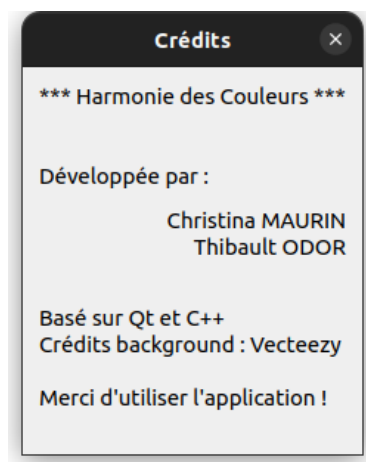
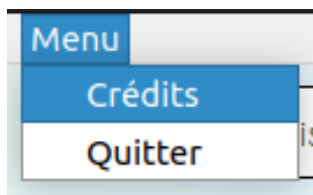
On ne souhaite pas avoir l'interface ci-dessus.

En utilisant l'option *DontUseNativeDialog* :



Ensuite, nous avons créé un `QLabel` permettant de mettre un fond à notre application. Nous voulions faire un fond dégradé aux couleurs pastels afin que cela reste agréable à l'œil. Nous avons donc sélectionné une image sur internet, en particulier libre d'usage si l'on crédite le site. Nous avons notamment dû passer par `QLabel` et ne pas modifier directement le `QCentralWidget` car modifier la feuille de style *stylesheet* faisait hériter tous ses paramètres aux autres éléments enfants. Par exemple, notre barre affichant le chemin n'avait plus son fond blanc.

Comme le site internet de notre *background* spécifiait un crédit, nous avons décidé d'utiliser notre barre de menu et d'implémenter un « Menu > Crédits » avec `QAction`. Grâce à un `QDialog` lors du déclenchement du « bouton », nous avons pu ouvrir une nouvelle fenêtre permettant d'afficher les crédits. Avec `setText()` nous avons pu écrire le texte que nous souhaitions, avec des balises html.

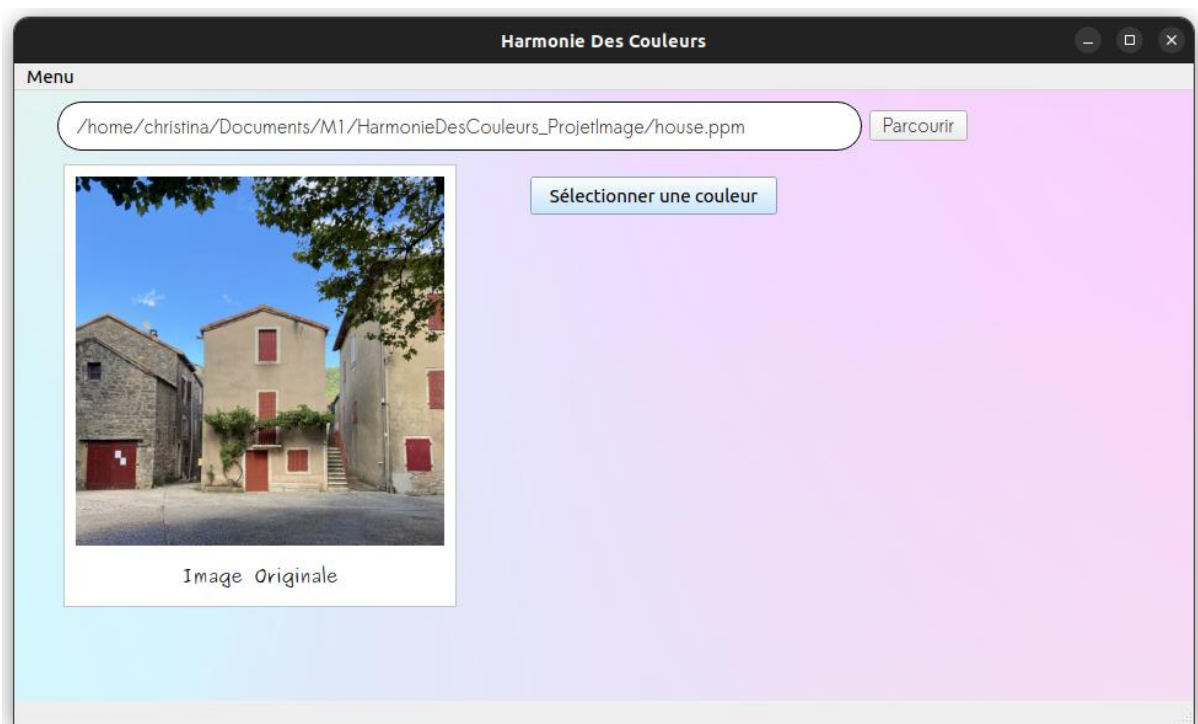


Savoir faire cela nous semble intéressant afin, plus tard, de pouvoir générer une fenêtre pour écrire un manuel d'utilisation du programme.

Également, nous avons fait dans ce menu déroulant une option « Quitter » qui permet d'avoir une autre façon de fermer l'application.

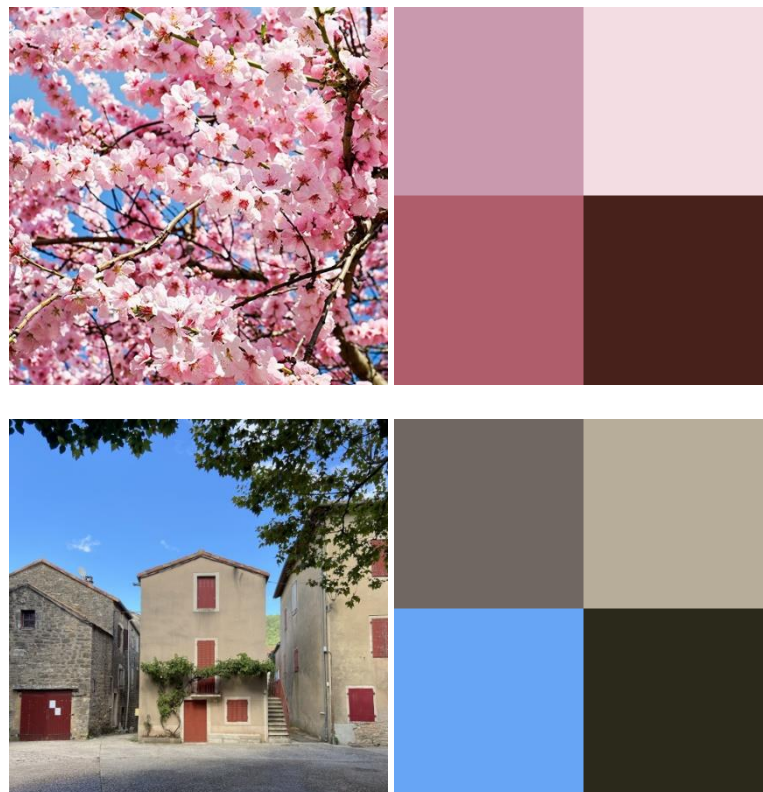
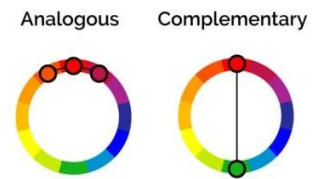
Pour le moment, il n'y a aucun lien entre notre application et le code que nous avons écrit précédemment.

Pour le rendu de l'image modifiée, nous aurons un deuxième affichage de l'image elle aussi sous forme « polaroid ».

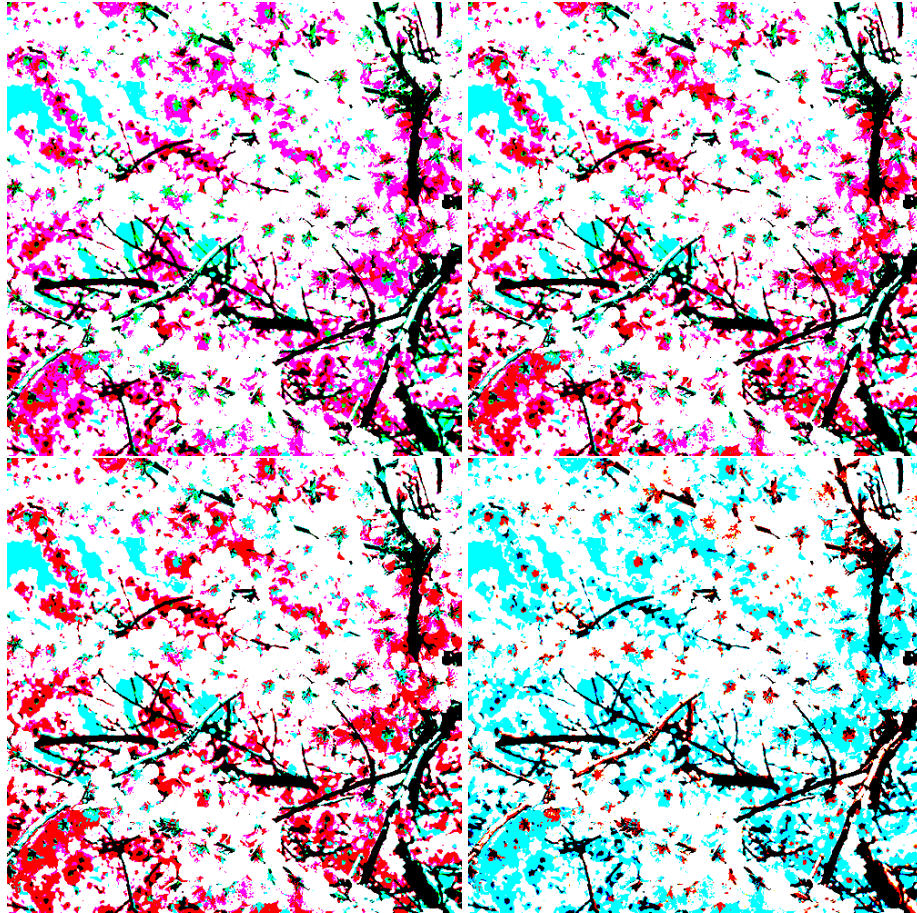


Algorithme :

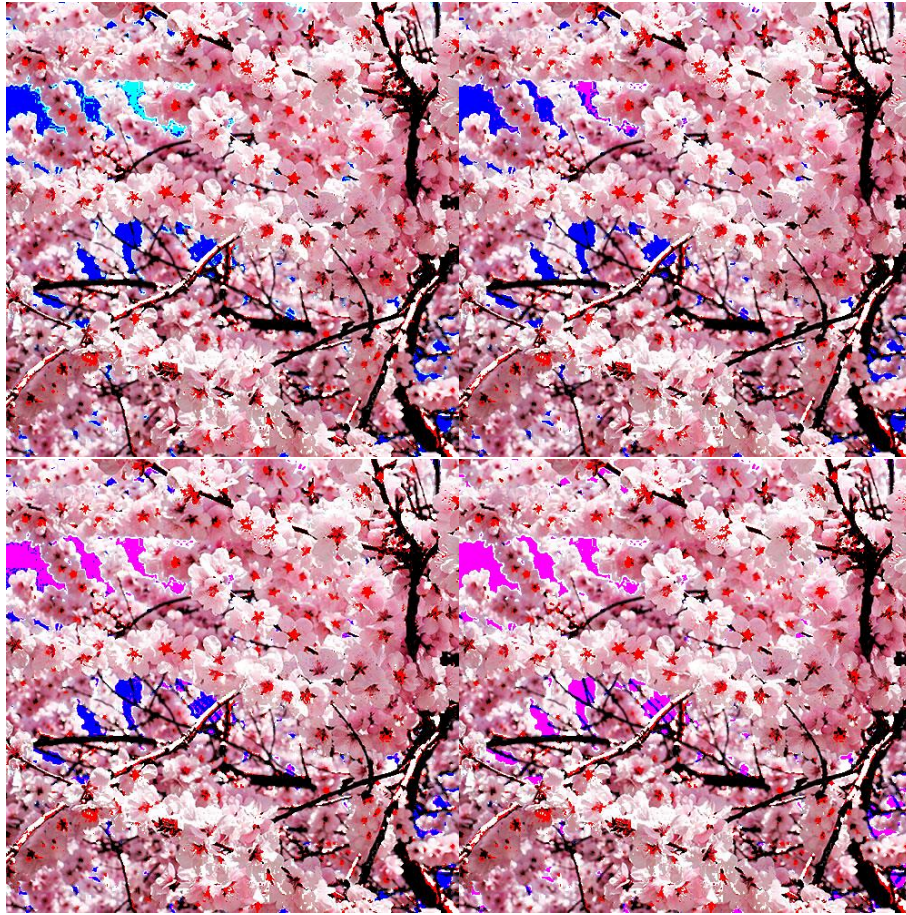
Cette semaine nous avons également commencé à mettre au point deux types d'harmonisations, l'harmonisation Analogique et l'harmonisation complémentaire. Mais également une recherche de couleur dominante par K-mean, qui nous sera très utile afin de proposer à l'utilisateur qu'elles couleurs sont les plus pertinentes après qu'il ai déposé l'image dans l'application.



Exemple de couleurs dominantes



Images utilisant les couleurs complémentaires



Images utilisant les couleurs analogues avec une largeur de 100

Pour le moment nous obtenons des résultats plutôt décevants, cela est dû au manque d'optimisation et la non prise en compte de l'harmonisation spatiale qui sera à étudier plus en détail prochainement.

Pour la semaine prochaine :

Pour la semaine prochaine nous prévoyons de faire la connexion entre notre application et les fonctions utiles que nous avons précédemment codées. Nous continuerons d'améliorer nos algorithmes.

Sources :

<https://www.techniques-de-peintre.fr/theorie-des-couleurs/harmonie-coloree/>

https://en.wikipedia.org/wiki/HSL_and_HSV

<https://doc.qt.io/>