

Cégep du Vieux-Montréal

Le 25 mai 2020

# ***Rapport de projet***

*Application de détection de couleurs*



Par

***Carl Genest***

Présenté à

***Jean-Christophe Demers***

Dans le cadre du cours

Projet Synthèse

420-C61-IN

### *Présentation générale de l'application*

L'application de détection de couleurs développée a été nommée Crayorescent en raison des crayons de couleur utilisés pour la détection. Pour en faire usage, l'utilisateur suit les étapes décrites dans le logiciel et bientôt, l'application sera en mesure de reconnaître une couleur qui est présentée à la caméra en direct.

### *Résumé du développement pendant la session*

Les objectifs de la planification ont largement été modifiés, les classes du diagramme UML ont été revues, mais les cas d'usage et l'interface graphique sont restés les mêmes.

Une difficulté importante a été de faire en sorte que le flux vidéo en direct soit intégré à un widget de l'interface. La méthode la plus simple qui a été trouvée sur StackOverflow utilisait Pyside 2 plutôt que PyQt 5 et pour cette raison le transfert de l'un à l'autre a été opéré et ce, très facilement. Ce code utilisait un QTimer qui est dépendant de la boucle du fil d'exécution principal (thread that has an event loop). De plus, les variables reliées aux boutons étaient plus facilement utilisables si les fonctions en faisant usage se trouvaient dans le même fichier que la fonction d'affichage d'une page. La structure initiale des classes a alors grandement été affectée par cet aspect pratique lors de l'implémentation. Le design pattern MVC initialement prévu a été abandonné, mais l'objet d'accès à la base de données (DAO) est resté utile. L'extraction de couleur, les valeurs de pixels ainsi que le calcul de moyenne de pixels ont été remplacés par le partitionnement en k-moyennes et le KNN est assez petit pour être une simple fonction.

L'idée d'implémenter un comptage des crayons a été explorée, mais elle nécessitait une prédiction quant à la position des objets dans la vidéo en plus d'une détection des formes et cela aurait été une charge considérable de travail additionnel. L'idée d'ajouter un graphique tridimensionnel avec Matplotlib aurait été intéressante, mais elle a été abandonnée faute de temps en fin de session.

OpenCV est la librairie omniprésente dans ce projet et sa page sur la fonction de K-Means est une source importante puisqu'elle constitue le cœur du fonctionnement de l'application (voir références).

### *Fonctionnalités*

La fonctionnalité abandonnée est celle du graphique. Les autres fonctionnalités sont toutes parfaitement fonctionnelles. Il s'agit de la prise de photos, de la préparation des données et de la détection de couleurs. La prise de photos permet de voir en direct ce qui est montré à la caméra. Elle permet de prendre une photo, d'y associer le nom d'une couleur, de l'enregistrer dans la base de données et de la faire apparaître dans la liste des photos gardées. On peut d'ailleurs supprimer une des photos sélectionnées dans cette liste. La préparation des données permet de voir si le masque s'applique bien à chacune des photos et la sélection est triée par couleur. On peut alors extraire les valeurs des couleurs pour toutes les photos en un ensemble de données ou aller sélectionner des plages de couleur différente pour la détection d'une couleur donnée. On peut aussi supprimer un ensemble de données de la base de données et le sélectionnant dans la liste affichée. Les valeurs enregistrées pour chacun des ensembles de données sont aussi affichées. Les configurations des masques peuvent être changées avec les différentes barres de défilement pour ensuite être enregistrées. La détection de couleur permet la détection en temps réel de la couleur. Un ensemble de données précédemment enregistré peut être sélectionné pour servir à la détection.

### *Améliorations possibles*

L'élimination de répétition dans le code aurait été souhaitable et une adoption du patron de conception de type stratégie pour encapsuler des algorithmes et les réutiliser plus efficacement aurait été une nette amélioration de la qualité du code. Outre cela, on peut remarquer que l'on doit poser un seul crayon à la fois au lieu de plusieurs : une amélioration en ce sens aurait été possible.

On remarque que les choix de couleur sont limités. Cela est dû au fait que d'autres couleurs comme le brun, le violet et le bleu foncé ont des plages de valeurs qui sont limitées et rapprochées. Le blanc et le noir ne sont pas implémentés à cause des couleurs du fond et du masque respectivement qui créent la confusion. Si l'on sélectionne une plage de rouge qui correspond aux valeurs de l'extrême inférieur dans les masques (vis-à-vis les teintes dites *hue*), le risque de confusion avec la couleur orange est élevé. À l'inverse, la plage de rouge supérieure évite totalement cette confusion. On comprend évidemment que les valeurs de rouge des deux extrêmes font partie d'un continuum selon le modèle HSV. On aurait d'ailleurs pu y ajouter une fonctionnalité d'ajout de choix de couleurs possibles. On remarque aussi que des imperfections peuvent être décelées dans la page de préparation, mais que les changements de masques sont appliqués à ce qui est vu par la caméra, ce qui oblige l'utilisateur à faire des allers-retours entre les deux pages. Ceci aurait pu être refactorisé pour en améliorer l'expérience.

### Auto-évaluation

- 1) J'estime que le présent projet synthèse est à la hauteur de la formation dont j'ai bénéficié. Il est une preuve de ma maîtrise des GUI, de la base de données et de concepts d'intelligence artificielle tels que vus dans les cours de génie logiciel, de langages de bases de données et d'intelligence artificielle. Le projet pousse l'apprentissage un peu plus loin en utilisant la vision par ordinateur et je suis d'avis que cela représente un défi à la hauteur des attentes liées au niveau collégial.
- 2) Mes attentes en début de formation étaient de pouvoir faire un logiciel utilisable sur Windows sachant que je pourrais faire du web et des tâches simples de réseau. J'avais aussi comme attente de pouvoir intégrer de l'intelligence artificielle dans ces programmes. En vue de ces deux attentes particulières je dois dire que je suis très satisfait.
- 3) Pour l'agréabilité de l'interface et le succès vis-à-vis la détection de couleur, je me donne 80%. Les points retranchés s'expliqueraient par la qualité non optimale du code.