

## Approximer une image

Pour construire un dessin, on peut superposer des figures géométriques : par exemple superposer des cercles, des triangles, des carrés, des pentagones... On peut aussi considérer que l'intérieur des figures géométriques sont peintes d'une couleur, et en superposant les figures, les couleurs peuvent se mélanger. En superposant ces figures géométriques colorées, on pourrait donc réaliser des images.



Dans ce projet, on va s'intéresser à réaliser des images à l'aide de polygones : ce sont des figures géométriques qui sont formées par une séquence cyclique de segments consécutifs. Pour se simplifier la vie, on va aussi considérer des polygones convexes (si on prend deux points au hasard sur les côtés d'un polygone convexe, le segment qui relie ces deux points se trouve entièrement dans le polygone). Ceci nous empêchera d'avoir des polygones ou les côtés se coupent.

Le but du projet est de construire une image la plus semblable possible à une image cible. En particulier le but du projet est de trouver une image qui sera le plus semblable possible au tableau la Joconde de Léonard de Vinci à l'aide d'au plus 50 polygones convexes.

### Code existant

Le code utilise des fonctionnalités de JavaFX qui permet de faire des interfaces graphiques en Java. Vous devez donc utiliser une version de java 1.8 avec JavaFX version 2.

Nous livrons une classe `ConvexPolygon` qui modélise un polygone convexe et coloré à l'aide d'une couleur uniforme. Cette classe peut générer un polygone au hasard en spécifiant le nombre de sommets. La couleur et l'opacité de la couleur sont par défaut choisis au hasard.

L'opacité permet de mélanger des couleurs ou non : si l'opacité est 0%, votre forme géométrique sera complètement transparente. Si l'opacité est 100%, une forme géométrique placée auparavant pourra être complètement cachée.

Nous livrons une classe `Test.java` qui effectue les tâches suivantes :

- lecture de l'image en entrée
- exemple de génération d'une image : dans l'exemple, on génère 10 triangles (on génère des polygones convexes aléatoires qui ont trois sommets).
- exemple de comparaison entre deux images : l'idée de l'exemple est de sommer une erreur entre chaque pixel de l'image générée et chaque pixel de l'image cible. Ici, pour mesurer l'erreur, on a

choisi de mesurer la distance euclidienne entre les vecteurs (R,G,B)<sup>1</sup> qui représente le pixel cible et le pixel de l'image générée.

- exemple de stockage de l'image générée dans un fichier .png
- exemple d'affichage de l'image générée dans l'interface graphique.

## Travail à effectuer

Utilisez comme point de départ le code existant pour implémenter en Java une solution qui génère une image ressemblant le plus possible à l'image de la Joconde fournie.

Rédiger un rapport (trois pages maximum) qui contient

- Une description de votre algorithme et une justification des paramètres utilisés.
- Une discussion sur les raisons qui font que votre algorithme fonctionne ou non.
- La présentation des résultats que vous avez obtenus (image, temps de calcul)
- Une discussion sur ce qui a été le plus dur à implémenter pour vous et sur le niveau de difficulté du projet.

## Critères d'évaluation

L'évaluation se fera avant tout sur le travail du groupe, la pertinence et la compréhension des solutions que vous avez mises en place, que le résultat final soit très ressemblant ou non.

Le projet est à effectuer de manière préférentielle en binôme. Si vous présentez un projet en trinôme, puisque votre force de travail augmente de 50%, la notation sera beaucoup plus sévère à travail égal.

L'évaluation portera sur votre code, sur votre rapport, ainsi que sur une soutenance (qui sera si possible organisée avant les vacances). Cette soutenance ne demande aucune préparation de votre part. Elle durera une douzaine/quinzaine de minutes par groupe. La soutenance consistera en un échange au sujet de vos résultats, votre rapport, et du code. Si la soutenance fait apparaître qu'un des membres n'a pas beaucoup contribué, sa note pourra être revue à la baisse. Egalement, on pourra vous demander de montrer le code et de fournir les résultats que vous avez obtenu lors des exercices d'implémentation des TDs.

Ce projet compte pour 40% de la note de l'UE. Il est donc souhaitable que la note corresponde au travail de votre groupe, et non aux conseils d'autres groupes, d'autres étudiants ou d'internet. Si vous utilisez des sources (articles de recherche, posts sur internet, etc...), vous devez mentionner vos sources dans le rapport. Si j'ai des soupçons de plagiat, je n'hésiterai pas à saisir le conseil de discipline.

---

1. un pixel peut être représenté par le niveau de trois couleurs, par exemple rouge, vert et bleu : combinés ensemble, on peut donc déterminer la couleur du pixel

## Soumission

1. Le projet est à rendre le **5 décembre 2018** sur la plateforme myCourse. Tout retard sera pénalisé.
2. Vous devez soumettre deux fichiers :
  - une archive zip dont le nom est la concaténation des noms des auteurs et qui contiendra l'ensemble des fichiers sources. Si les auteurs sont Astérix et Obélix, le fichier zip sera donc nommé `asterix_obelix.zip`. Lorsqu'on décompresse le fichier zip, on devra trouver un répertoire nommé `asterix_obelix` contenant tous les fichiers sources. On devra pouvoir compiler et exécuter votre code depuis l'intérieur de ce répertoire.
  - un document au format pdf contenant votre rapport nommé `asterix_obelix.pdf`. N'oubliez pas d'indiquer le nom des auteurs du travail dans le rapport.

Vous serez pénalisé si vous ne suivez pas ces consignes.