

Master 1

MIAGE

PROJET INTELLIGENCE ARTIFICIELLE:
APPROXIMER UNE IMAGE

Organisation du travail :

1. Choix d'un algorithme
2. Fonctionnement de l'algorithme
3. Test
4. Difficultés rencontrées

Réalisé par :

Aminata DIA

Ndeye Amy Diakhaté DIOP

Professeur :

Stéphane AIRIAU

Année Universitaire: 2018-2019

Introduction

Dans ce projet, il s'agit d'approximer une image en utilisant au plus 50 polygones convexes sous JavaFX. Dans notre cas il sera question de construire une image la plus semblable possible à l'image du tableau de la Joconde de Léonard de Vinci.

1) Choix d'un algorithme

Plusieurs solutions permettent de réaliser ce projet (Genetic algorithm, Hill Climbing, ...). Notre choix s'est porté sur l'algorithme génétique. Cet algorithme qui se base sur les principes de gènes et de chromosomes, nous a permis d'approximer le plus possible l'image source et l'image générée par la mutation de polygones convexes.

2) Fonctionnement de l'algorithme

Initialement, 50 polygones ont été générés au hasard. Dans chaque étape d'**optimisation**, nous avons, d'abord, une **phase d'évaluation** où chaque chromosome possède un fitness qui est une comparaison pixel par pixel avec l'image source. Ensuite nous passons à la **phase de sélection** accomplie par clonage. Il s'en suit la **phase de mutation** qui permet de modifier soit la couleur du polygone soit la position de ses sommets. L'opération d'optimisation reprend jusqu'à l'obtention d'une image la plus ressemblante possible à l'image source.

3) Test

Nous avons exécuté notre algorithme sur plusieurs types d'images. L'image finale générée est une assez bonne approximation de l'image originale et l'algorithme a fonctionné raisonnablement. Ce pendant, L'algorithme prend beaucoup de temps à s'exécuter. Ci-dessous des images obtenue au cours du processus avec des intervalles de plus de 500 générations.



413 générations, 24s



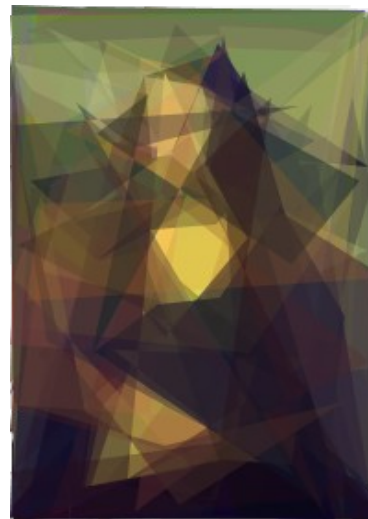
877 générations, 74s



1230 générations, 128s



2000 générations, 326s



2752 générations, 666s



3000 générations, 1003s

4) Difficultés

Nous avons eu à rencontrer quelques difficultés dans la réalisation du projet.

- L'utilisation de Java FX, étant une nouveauté pour nous, il nous a fallu du temps pour bien comprendre les bases.
- Une autre difficulté pour nous a été de trouver l'algorithme adéquat. Dans un premier temps nous avons commencé à utiliser l'algorithme de Hill Climbing mais ce dernier paraissait un peu complexe pour nous. Finalement nous avons opter pour l'algorithme génétique qui, néanmoins, a nécessité un temps de compréhension.

Sources :

<https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>

[https://www.google.com/url?](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKewi17fODnJHfAhVPhxoKHUMSAn0QFjAAegQICBAB&url=http%3A%2F%2Fgenetics.io%2F&usg=AOvVaw0R0ieMHRpNTE_UBI1XI3RE)

[sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKewi17fODnJHfAhVPhxoKHUMSAn0QFjAAegQICBAB&url=http%3A%2F%2Fgenetics.io%2F&usg=AOvVaw0R0ieMHRpNTE_UBI1XI3RE](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKewi17fODnJHfAhVPhxoKHUMSAn0QFjAAegQICBAB&url=http%3A%2F%2Fgenetics.io%2F&usg=AOvVaw0R0ieMHRpNTE_UBI1XI3RE)

<https://github.com/miezekatze/evolisa/tree/master/src>