IUT Paris Descartes

2017-2018

Base de la Programmation Objet

Mémoire de Projet : Film ASCII Art

Maud GELLEE — Groupe 101

Arsène LAPOSTOLET — Groupe 112

Table des matières

I/ Présentation de l'application

Introduction :

L'art ASCII consiste à réaliser des images uniquement à l'aide des lettres et caractères spéciaux contenus dans le code ASCII.

Problématique :

Programmer une bibliothèque de classes Java de dessin et de création de films animés en ASCII Art.

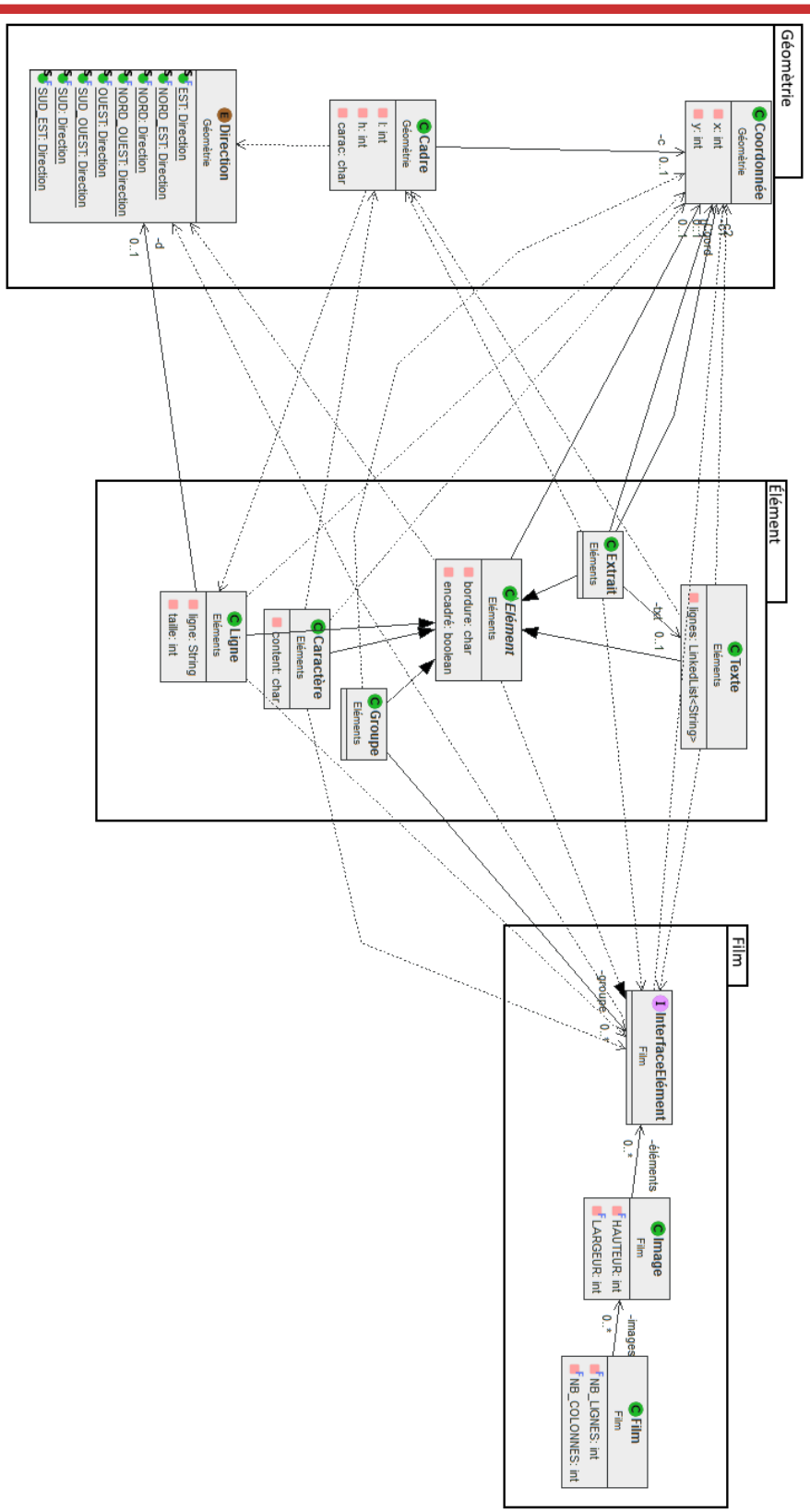
Rôle fonctionnel du projet :

Le but de ce projet est de programmer une bibliothèque de classes Java utilisables dans un projet. La bibliothèque doit permettre à un programmeur utilisateur de coder un programme principal permettant de créer des images en y ajoutant différents types d'éléments, puis de manipuler ces derniers pour créer un film.

Les entrées et sorties :

Une bibliothèque ne contient pas de programme principal, les entrées et sorties seront donc gérées par le programmeur utilisateur qui va créer un programme principal à partir de notre bibliothèque de classes.

II/ Diagramme UML



III/ Organisation des tests

Pour organiser les tests de l'application, nous avons, pour chaque classe, écrit des fichiers de test JUnit 5. Nous avons utilisé les mots clés *assertFalse* et *assertTrue* pour mettre en condition nos classes et mettre à l'épreuve leurs différentes fonctionnalités.

Bilan :

Tous les tests s'exécutent avec succès.

IV/ Citations

Classe Direction

Nous avons récupéré la classe énumérée *Direction* dans la série de TP sur la Chenille. Cela représente 35 lignes de code.

Classe Cadre

L'idée de la classe *Cadre* nous a été inspirée par un camarade (Sacha Froment). Nous l'avons cependant codée nous-même car elle repose sur nos propres composants. Cela représente 36 lignes de code.

V/ Bilan du projet

Pendant ce projet, nous avons renforcé nos connaissances de la conception et de la programmation en paradigme objet. Nous avons ainsi pu expérimenter l'utilisation de nouveaux composants comme les interfaces et de nouveaux concepts comme l'héritage, le découplage et la stabilité de paquetages.

De même nous avons découvert de nouvelles fonctionnalités de Java concernant la manipulation minutieuse de *char* et de *String* ainsi que de renforcer les bonnes pratiques de programmation et d'encapsulation que nous avions apprises auparavant.

Afin d'améliorer notre productivité et notre confort de travail collaboratif, nous avons, sur le conseil de nos professeurs, utilisé le système de gestion de version Git. Nous avons donc utilisé l'hébergement gratuit *framagit.org* ainsi que le plugin EGit d'Eclipse afin de disposer d'un client avec une interface graphique. Après avoir surmonté des difficultés de premier abord, nous avons pu profiter de la puissance de ce nouvel outil.

D'un point de vue plus général, ce projet a été une très bonne expérience ; elle nous a permis de confirmer les compétences déjà acquises en Java et en programmation objet ainsi que d'en acquérir de nouvelles dans de bonnes conditions. Le sujet, à la fois ludique et intéressant, nous a permis de ressentir l'esprit du développeur. De plus, la création d'une bibliothèque de classe était pour nous une expérience nouvelle, tout à fait enrichissante.

Enfin, l'aspect collaboratif de ce projet fait partie de ses points forts. En effet, la motivation mutuelle est un moteur puissant qui permet de surmonter tout type de difficulté. De plus, la résonnance de deux esprits focalisés sur le même objectif, communicant sur un problème, est une des formes les plus efficaces de réflexion. Chacun comblant les faiblesses de l'autre, nous avons pu nous tirer mutuellement vers le haut. Ayant déjà l'expérience du travail en commun, nous avons renforcé notre niveau de collaboration et cela nous a permis d'avancer encore plus, chacun connaissant les spécificités de l'autre.

IV/ Bilan du projet

Pour ce qui est des difficultés rencontrées, nous avons parfois eu des problèmes de compréhension du sujet, notamment concernant l'encadrement, et avec la construction d'images à partir d'une image existante. De plus, les choix d'encapsulation ont parfois été compliqués à faire pour nous. Comme dans tous les projets, des erreurs de conception nous ont conduits dans de mauvaises directions nous forçant à réimplémenter certaines fonctionnalités. Enfin, on peut citer un problème dans le *char[ ][ ]* qui n'étai pas initialisé, ainsi on voyait des espaces dans le fichiers texte mais ils n'étaient pas pris en compte dans CM-Player car les cases vides étaient à *null*. Nous avons réglé ce problème en initialiser le *char[ ] [ ]* avec des espaces grâce à *Arrays.fill()*.

Concernant les améliorations possibles du projet, nous n'avons par réussi à implémenter l'encadrement d'un groupe, à cause notamment de notre conception des lignes, il nous était difficile de délimiter avec certitude la surface des groupes. De plus, malgré tous les tests que nous avions effectués, nous n'avions sans doute pas remarqué toutes les fonctionnalités qui allaient manquer à notre bibliothèque, étant donné que nous ne l'avons pas utilisée en conditions réelles dans un projet.