

05/05/2025

Présent :

Ethan Ducournau  
Hugo Deleporte  
Matisse Jean Marie

Décrire aussi les classes avec les /\*\* (une ou deux lignes)

Chaque attribut doit être commenté

Toutes les méthodes publiques : javadoc obligatoire, pas forcément pour les privées (bien pour nous)

Tests unitaires :

checkCollisions() : au moins deux tests unitaires à faire (cas où il y a et il y a pas des collisions)

findCluster() : tester la composante connexe (cas où il y en a et où il y en a pas)

ne pas tester toutes les fonctions, montrer qu'on sait utiliser JUNIT.

faire deux trois tests par fonctions

Rapport environ 20 pages (sans les annexes)

rajouter une section sur les technologies utilisées (entre gestion de projet et développement) (sur les IDE, java, setbuilder (dire pourquoi ? Avoir des menus sympas, bonne séparation des modèles), expliquer pourquoi ces choix là).

Montrer qu'on a bien séparé le modèle de l'affichage : très important ! C'est sur cela qu'il faut appuyer, montrer qu'on a bien séparé les deux choses.

pas d'explications de codes dans les rapports, mais faire un diagramme de classe, c'est très important.

Ne pas forcément expliquer toutes les classes, mais seulement les principales

pour les attributs, garder aussi que les principaux, pas besoin de tout expliquer

pareil pour les méthodes, pas besoin de mettre toutes les méthodes, mettre seulement les méthodes les plus importantes

ce n'est pas la syntaxe java qu'il faut mettre, mais la syntaxe UML

quand on met une méthode, mettre sa signature

il faut que le diagramme de classe explique comment les classes sont agencées, cela montre comment le code fonctionne

raconter les problèmes rencontrés lors du processus de codage

important : les patrons de conception de codage (on doit parler du modèle MVC très important)

ce modèle sert à séparer 3 métiers, chacun travaille sur son domaine

si on veut changer par exemple JavaFX pour un autre comme Swing, on a juste à changer le contrôleur

j'ai pas trop compris mais appeler le contrôleur je sais pas où (classe Factory, singleton, tuto javafx sur la multivue qui parle du modèle du singleton)

bon pas besoin finalement de changer ça marche bien comme ça