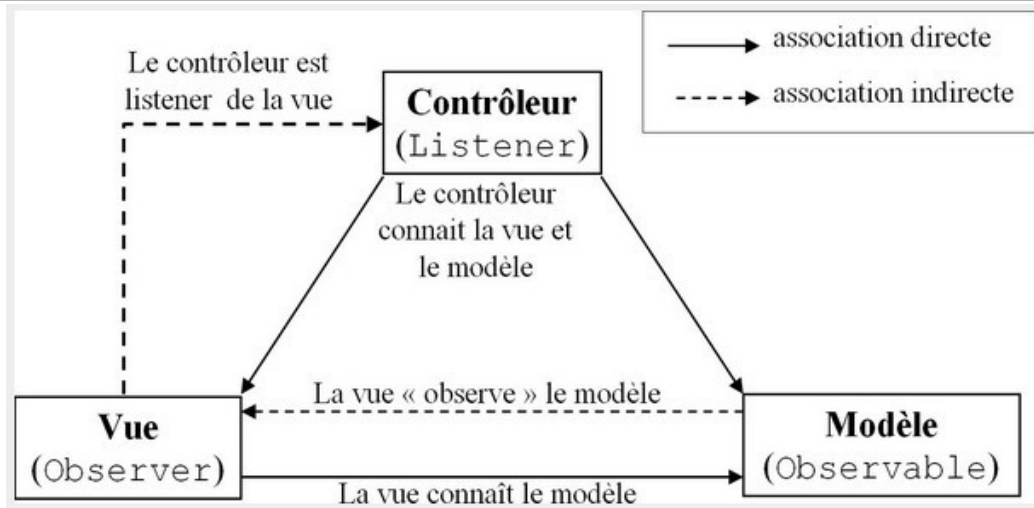


L2 INFO 2021-2022
POGL
Rapport de Projet Île-Interdite



Afin de réaliser ce projet, nous avons suivi la méthodologie du cours, en utilisant un pattern MVC (Modèle-Vue-Contrôleur).

La vue contient la description de l'interface graphique avec ses différents composants.

Le modèle contient tous les attributs et méthodes qui doivent participer à la tâche principale à accomplir (C'est le «cœur du projet»).

Enfin **le contrôleur** contient l'ensemble des "listeners", qui, lorsque des événements parviennent via la vue, prévient alors directement le modèle mis en place.

Organisation du travail

Concernant "l'architecture" du projet, nous nous sommes inspirés de l'exemple du jeu de la vie (Conway.java).

Nous avons donc travaillé à 2 l'ensemble afin d'être efficace, nous avons commencé par répondre aux questions qui nous permettaient de nous guider sur le projet en écrivant les fonctions et méthodes sur papier (nous avons essayé d'écrire le plus possible des fonctions d'affichage sur console au cas où nous arriverons pas à avoir un affichage graphique),

Sonia a commencé à traiter les actions spéciales quand Samy s'est occupé des rôles de de chaque personnages, nous avons travaillé la simulation du paquet de cartes ensemble.

Enfin pour la partie "Échange de clés", nous l'avons faites lors de la partie 3, car elle nécessitait des éléments de celle-ci (comme le fait de récupérer un artefact etc...). Cette organisation nous a permis de venir à bout du projet, et nous a permis de faire les tests unitaires de façon coordonnée.

Tests unitaires JUnit5 Java

Nous avons utilisé la bibliothèque JUnit5 de Java, permettant de faire cela. Ces tests unitaires avaient 2 avantages, le premier étant celui de tester des portions de code lorsque nous avançons au fur et à mesure du projet, et donc de vérifier si nos fonctions et méthodes étaient correctes. Le second avantage était que cela nous permettait de tester que chaque portion de code remplissait bien sa fonction et renvoyait ce qu'il fallait. En effet chaque portions de code doit remplir correctement sa fonction pour que l'ensemble du projet soit fonctionnel et fiable.

Nous avons réalisés les tests des fonctions les plus pertinentes selon nous, comme le test qui vérifie qu'un artefact a bien été récupéré, qu'une clé a bien été donnée, ou alors le test qui vérifie qu'une case n'est pas submergée en testant le niveau de l'eau sur cette même case etc...

Difficultés rencontrées lors du projet

Le début de ce projet était assez facile, en effet beaucoup de concepts que nous avons vu en TP durant le semestre ont été revus et réutilisés. (comme la gestion d'affichage de grilles vu dans le TP1 avec les N-Reines, les lambdas expressions pour représenter les interfaces fonctionnelles etc...). Les 3 premières parties du projet étaient assez bien guidés et ne nous ont pas vraiment posé problème. En revanche la partie 4 était un peu délicate dans le sens où il fallait faire attention à bien tout coordonner et mettre en place les différents éléments qui étaient demandés. Enfin ce qui nous a le plus posé problème et pris du temps est l'affichage graphique, nous avons utilisés la librairie Swing de Java et tous ses éléments JPanel, JFrame, JButton etc.. consacrés à l'affichage graphique. Nous avons, en effet réussi à obtenir une fenêtre de jeu (en plus de l'affichage sur console qui permettait de rentrer les coordonnées de chaque joueurs avec leur rôle ainsi que les différentes commandes de déplacement et d'actions à faire).

Intérêt pour le projet & Conclusion

Ce projet fut très intéressant, en effet il nous a permis de travailler tout ce qui a été étudié durant ce semestre, en reprenant les notions les plus importantes, comme la gestion des exceptions ou alors l'affichage graphique. De plus cela était enrichissant car nous avons mis en place les tests unitaires qui sont complètement nouveau ce semestre et très pratique pour la suite, car cela permet de tester des petites portions de code et de vérifier si le code est opérationnel en temps réel. Ce projet nous a permis de relier la partie Programmation Objet au Génie Logiciel, avec la réalisation du schéma de base UML(Diagramme de classes), qui nous permet d'avoir une vue d'ensemble sur le projet avec les classes et interfaces qui sont utilisés et notamment les fonctions et méthodes avec leurs attributs.

Enfin, cela nous a permis de nous initier à l'architecture MVC (Modele-Vue-Contrôleur), qui est très intéressante dans l'organisation et la réalisation d'interfaces graphiques pour des applications ou alors des jeux comme dans notre cas ici.