

Rapport d'amélioration du FluProject

Analyse des trois projets

Projets	Forces	Faiblesses
Benjamin	Architecture simple Utilisation de l'héritage	Pas de pattern MVC Pas de configuration Pas d'organisation en package
Anaïs	Bonne architecture Respect du modèle MVC Utilisation du pattern State Utilisation de l'héritage	Configuration trop complexe Présence d'instanceof
Didier	Bonne architecture Respect du modèle MVC Utilisation du pattern State Utilisation de l'héritage Gestion de la configuration en JSON Projet complet	Code un peu complexe Présence d'instanceof

Comparaison

Tous les programmes sont fonctionnels, cependant on ne choisira pas celui de Benjamin car il présente une mauvaise structure et impliquerait de tout revoir. Ensuite, entre le projet d'Anaïs et celui de Didier qui présente une architecture semblable au niveau des patterns, on gardera celui de Didier. En effet, il présente les mêmes forces que les deux autres projets et possède l'avantage de la configuration en plus. Par ailleurs, il ne présente que très peu de faiblesses. Il est donc naturel qu'il serve de base pour faire un meilleur projet.

Stratégie d'amélioration

Notre stratégie est donc la suivante :

- Nous nous servons du projet de Didier comme base
- Nous améliorerons ses quelques faiblesses c'est-à-dire la présence des instanceof
- Et nous ajouterons des tests unitaires pour consolider le projet

Répartition du travail

Nous avons corrigés les instanceof à trois et nous nous sommes répartis les JUnits.