

Rétro-Ingénierie Maintenance et Evolution (Mini-projet LMU)

Objectifs :

- Etre confronté à du code existant peu documenté ;
- Effectuer un travail de ré-ingénierie pour repackager du code et faire évoluer sa fonction/utilisation

Pour ce travail vous êtes autorisés/encouragés à travailler en groupes de 2 ou de 3 personnes. Il y aura un seul rendu par groupe.

1. Récupérer le logiciel LMU (source et bytecode) : <http://www.i3s.unice.fr/~hogie/lmu/>
2. Utiliser ce logiciel ou un autre (ex : VisualParadigm) pour analyser les classes principales de ce logiciel
3. Fournir une *feature* Eclipse (avec un ensemble de plugins) qui propose une intégration de LMU dans Eclipse
4. Fournir un rapport qui explique le travail réalisé

La feature fournie devra proposer au minimum

- Un **menu popup** Eclipse pour lancer la génération d'un modèle UML à partir du code sélectionné. Le menu doit s'activer uniquement sur :
 - Un projet Java ;
 - Un package ;
 - Un fichier source Java ;
 - Un fichier .jar Java.
- Une solution à définir pour appliquer les différents paramètres de LMU :
 - Format d'image généré ;
 - [opt] Répertoire de génération ;
 - [opt] Cacher certains éléments parmi ceux sélectionnés (sous-package, expression régulière).
- La *feature* fournie doit contenir **UNIQUEMENT** le code nécessaire à la fonction demandée, en particulier, pas l'interface graphique LMUGui initiale, ni les bibliothèques qui ne sont pas utilisées.

Le rapport de maximum 10 pages doit :

- Identifier les membres du groupe et leur rôle dans la réalisation du projet ;
- Décrire, de façon synthétique et abstraite, l'architecture logicielle de LMU avant modifications : principaux « morceaux », packages et classes importantes ;¹
- Lister les solutions logicielles retenues dans LMU qui empêchent ou ralentissent une intégration immédiate ;
- Lister les classes qui ne respectent un des cinq principes SOLID énoncés en cours et qui ont dû être modifiées pour réaliser le reconditionnement demandé ;
- Décrire, de façon synthétique la nouvelle architecture logicielle proposée, en insistant sur les différences avec la version initiale.

¹ Vous êtes encouragés à vous aider de VisualParadigm (ou LMU lui-même) mais il faut faire une description synthétique, pas simplement donné un diagramme illisible.

Modalités de rendu :

- Date : avant le 15 janvier à minuit
- Envoyer un email à Frederic.Mallet@unice.fr
- Objet de l'email : [RIMEL] LMU - Nom1, Nom2, Nom3
- Contenu : à joindre à l'email directement ou à déposer sur un site de dépôt de fichiers/code
 - le rapport au format PDF
 - Une archive .zip contenant les sources des projets Eclipse réalisés (Feature et Plugins) ;
 - Une archive .zip contenant les plugins (.jar ou .class) à déployer.

Note : Les plugins déployés doivent s'exécuter dans un Environnement **Eclipse Modeling Mars** sans extension, sous Windows ou sous Linux. Il est donc vivement conseillé de tester le déploiement attentivement dans le dossier « dropins » d'un Eclipse sans extension.

Barème :

Le barème est donné à titre indicatif pour organiser et répartir l'effort du groupe.

- Suivre des consignes et modalités de rendu [5%]
- Déploiement sans intervention [10%]
- Rapport [40%]
 - Description de l'architecture existante [10%]
 - Analyse des limitations et actions [15%]
 - Nouvelle architecture proposée [15%]
- Code [45%]
 - Intégration dans Eclipse [10%]
 - Fonctionnalité [10%]
 - Organisation en plugins/feature [5%]
 - Ratio Code déployé/code mort [10%]
 - Mise en œuvre des principes SOLID [10%]