Projet

Objectifs

- Concevoir et développer un projet complet selon la logique orientée-objet
- Savoir utiliser les diagrammes utiles pour concevoir et documenter le projet
- Comprendre les relations entre COO et POO

Sujet

Vous avez à réaliser le développement (analyse des besoins, conception niveau abstrait puis niveau physique, et codage), le plus complet possible, selon la logique de l'orienté-objet. La modélisation sera réalisée en UML et la programmation en Java.

Vous avez **2 choix de sujet possible** pour votre étude :

- **Un Donjon-RPG (Role-Playing-Game) en mode textuel**. Le joueur incarne un personnage qui doit parcourir un donjon sous forme de labyrinthe pour réussir sa quête. Chaque pièce du donjon est décrite (porte à gauche et à droite, coffre au milieu, monstre devant le coffre...) et l'utilisateur fait un choix (utiliser, attaquer, aller à, ramasser...) en fonction du contexte de la pièce, des armes et objets de son inventaire... Quelques exemples en vidéo : Warior Quest, Console Game, Dongeon Escape.

- **« Qui veut gagner son DUT » proposé en projet tutoré.** Ce sujet peut être l'occasion d'explorer des fonctionnalités et de développer une partie du code que vous aurez à produire dans le cadre du projet tutoré. Notez toutefois, que le projet COO/POO et le projet tuteuré seront cependant notés séparément et vous pouvez donc vous mettre en binôme avec une personne qui n'est pas dans votre groupe de projet tutoré.

Organisation

- Le projet peut être fait **seul ou en binôme**.
- Le projet s'étend sur 6 séances de TP de 3h **: 3 séances en COO et 3 séances en POO**. Bien qu'il soit recommandé de faire des diagrammes pendant vos séances COO et la programmation pendant les séances POO pour profiter de l'aide des chargés de TP, vous être libre d'avancer vos projets comme vous le souhaitez.

Livrables et évaluation

Vous aurez un livrable documentaire commun (le rapport) et un livrable logiciel (le code complet – sources et binaires).

Le rapport sera toutefois à déposer à 2 endroits afin de faciliter les évaluations séparés selon les objectifs de chacun des 2 modules COO et POO. Vous obtiendrez donc 2 notes d'examen pratique distinctes.

Le rapport (document PDF) contiendra :

- une introduction rappelant
  - le sujet choisi,
  - les personnes impliquées
  - le partage des tâches et l'organisation
- une section *analyse des besoins*
- une section *conception niveau abstraite*
- une section *conception niveau physique* qui décrit le code
- une section *codage* avec des bouts de code Java mettant en avant les classes et algos les plus importants

Le ZIP du produit contiendra :

- un fichier JAR exécutable de votre programme ;
- un ZIP du projet Eclipse complet contenant les sources.

Les sections suivantes détaillent les modalités et précisions propres à chaque module.

Partie COO

Le rapport sera à déposer via l'espace-cours dédié au module M2104, ceci au plus tard pour le **vendredi 25 avril 2014**. Il y aura des pénalités de retard.

Suivez la méthode présentée dans le dernier chapitre du cours et revue en TDm afin de fournir un ensemble de diagrammes complet et cohérent. Il est notamment important de fournir des diagrammes qui couvrent les 3 vues complémentaire d'un système : structure, comportement, fonctionnalité.

Dans le rapport, vous présenterez l'ensemble de vos diagrammes UML en expliquant de manière détaillée ce qu'ils modélisent et éventuellement les choix de modélisation que vous avez suivis. Vous pouvez utiliser Modelio ou tout autre outil pour la réalisation des diagrammes. Votre rapport doit contenir toutes les informations nécessaires pour qu'une autre équipe de développeur puissent facilement reprendre votre travail.

Le résultat final sera calculé en prenant en compte les critères suivants (non exhaustif) :

- Qualité du rapport (présentation, qualité de la représentation des diagrammes, etc.)
- Pertinence des diagrammes proposés

- Validité des diagrammes (modélisation « valide » vis-à-vis de la syntaxe et de la sémantique du langage UML)
- Qualité de la modélisation réalisée
- Bonne articulation des diagrammes

Partie POO

Le rapport et le ZIP du produit seront à déposer via un ZIP global (le troisième si vous comptez bien) via l'espace-cours dédié au module M2103, ceci au plus tard pour le **vendredi 25 avril 2014**. Il y aura des pénalités de retard.

Concernant la partie « codage » du rapport il est attendu des **extraits de code commentés** des parties les plus significatives (du point de vue de la programmation objet) de la solution : focus sur les classes abstraites / interfaces, héritage / implémentation, redéfinition, surclassement, etc.).

Il vous faudra mettre en avant votre bonne compréhension et application des principes de bases de la POO à travers votre code.

L'évaluation s'appuiera sur cette section du rapport mais également sur celle de conception physique afin de mieux comprendre votre approche objet et s'assurer de la cohérence conception <=> codage. Les sources seront également analysées. La version exécutable permettra de s'assurer du fonctionnement de votre programme.

Il n'est pas nécessaire que le code produit couvre 100% des modèles de conception envisagés. L'évaluation de la partie POO portera sur les aspects suivants :

- qualité de présentation (forme) de la section codage du rapport ;
- pertinence et argumentation des extraits de code présentés (fond) ;
- exécutabilité du JAR ;
- qualité du code (respect conventions de nommage, sources commentées – l'utilisation de Javadoc serait un +) ;
- qualité de la POO globale présente dans le code ;
- qualité des algorithmes et codes ne rentrant pas le champ de la POO ;
- les fonctionnalités produites.

Il est possible que les personnes impliquées dans un même binôme n'aient pas la même note si une implication non homogène est détectée lors des séances encadrées. La réutilisation totale comme partielle de diagrammes/code entre groupes sera sanctionnée. Toutefois, il n'y pas d'inconvénient à ce que les différents groupes échangent oralement sur leurs intentions et techniques de COO/POO.