

Projet commun Java avancé, Réseau et Génie logiciel

M1 Miage – 2016

Création d'un jeu de société

Description du sujet

Ce projet consiste à réaliser un programme Java permettant à plusieurs joueurs de jouer à un jeu de société sur leurs ordinateurs respectifs (donc en réseau). Vous pourrez choisir votre jeu de société dans la liste suivante (les règles sont disponibles sur <http://jeuxstrategie.free.fr> et sur www.ludism.fr) :

- Colons de Katane (http://jeuxstrategie.free.fr/Siedler_presentation.php)
- Tigre et Euphrate (http://jeuxstrategie.free.fr/Euphrat_presentation.php)
- Santiago (http://jeuxstrategieter.free.fr/Santiago_presentation.php)
- Acquire (http://jeuxstrategie.free.fr/Acquire_presentation.php)

Pour la réalisation de ce projet, les restrictions suivantes sont imposées: au plus deux groupes pour un jeu de société donné et 5 personnes par groupe. Si plus de deux groupes choisissent le même sujet, la règle appliquée est celle du « premier arrivé, premier servi ».

Objectifs

Les objectifs principaux du projet sont les suivants:

- travailler à plusieurs (groupes de 5) sur le développement de l'application,
- réussir à livrer un produit fonctionnel dans les temps,
- avoir le code le plus maintenable possible (couplage/cohésion, SRP, patterns,...),
- utiliser les différentes notions vues dans les cours de programmation Java et de réseau.

Consignes

Vous devrez rendre une archive pour le 6 mai 2016 (délai de rigueur). Cette archive doit inclure :

- un rapport au format pdf ne devant pas excéder 10 pages expliquant vos choix, les difficultés rencontrées (vous pouvez inclure une javadoc en annexe),
- les sources,
- les tests unitaires,
- les backlogs précis de chaque sprint,
- les tâches, l'estimation, et leurs développeurs (2 maxi sur une tâche),
- les burndown charts à l'issue de chaque sprint,
- la version fonctionnelle obtenue à l'issue de chaque sprint incluant la version exécutable finale obtenue lors du dernier sprint (elle devra être mise en évidence).

Vous aurez donc une archive dont la structure ressemblera à ceci:

- Sources version finale
- Tests
- Sprint 1: backlog, tâches, estimations, burndown chart, version livrée à l'issue du sprint (sources +exécutable)
- Sprint 2: backlog, tâches, estimations, burndown chart, version livrée à l'issue du sprint (sources +exécutable)
- Sprint 3: ...

Une soutenance du projet aura lieu la semaine du 19 mai 2016 (date précisée ultérieurement). Vous disposerez de 30 minutes pour présenter votre projet. Le temps imparti devra être réparti entre :

10 minutes de présentation du sujet et des choix d'implémentation (slides)

10 minutes de démonstration logicielle

10 minutes de questions du jury

Ce travail devra être réalisé par groupe de **cinq personnes**. Envoyez un mail à olivier.perrin@loria.fr afin de valider votre groupe et votre sujet.

Évaluation

Ce travail donnera lieu à trois évaluations distinctes (une pour le cours de Réseau, une pour le cours de Java avancé et une pour le cours de Génie logiciel). Voici les différents points qui serviront à l'évaluation de votre travail:

1. réponse aux besoins exprimés: compréhension du jeu, de ses mécanismes, des besoins du client,
2. conception:
 - modélisation adéquate,
 - cohésion des composants,
 - couplage des composants,
 - adaptabilité des composants.
3. réalisation
 - utilisation de concepts avancés vus en Java (threads, sérialisation, interfaces, ...) et réseau,
 - tests unitaires, résultats des tests,
 - cohésion des composants,
 - couplage des composants,
 - lisibilité/simplicité du code,
 - structuration du projet.
4. utilisation optimisée d'un SCM (Git) et de la méthode Scrum,
5. la clarté et la pertinence du rapport,
6. la qualité de la soutenance orale.

Recommandations

Bien que ludique et intéressant , ce projet est dense et nécessite une **quantité de travail relativement importante**. Il vous est donc conseillé de le commencer le plus tôt possible et de ne pas sous-estimer sa difficulté ! Vous disposez d'environ 9 semaines pour réaliser ce projet (date de recette fixée au 6 mai 2016). Vous devez penser ces 9 semaines comme 4 à 5 sprints Scrum. Pour chacun de ces sprints, vous devrez donc déterminer le backlog, créer la liste des tâches, vous engager sur une ou plusieurs tâches, faire un *burndown chart* à l'issue de chaque séance, et repartir sur un nouveau sprint. Pour chaque tâche, vous devrez réaliser les tests unitaires vous permettant de valider les composants, et de refactoriser si besoin en toute « sécurité ».