

Antoine de ROQUEMAUREL
Dario OLIBET
Groupe 1.1

Projet de programmation

Partie Java

Avant-propos

Ce dossier comporte l'organisation et les étapes de développement d'un projet d'Arènes en Java.

Il à été conçu par Antoine de ROQUEMAUREL et Dario OLIBET dans le cadre du module *Projet de programmation* de la L3 Informatique de l'université Toulouse III – Paul Sabatier.

Contenu de l'archive du projet

L'archive que vous avez reçus était organisée comme ceci :

rapport.pdf Le présent rapport que vous êtes en train de lire

V1/ Contient le dossier avec la première version de l'application : notre propre arènes avec les règles que nous avons implémentées.

V1/doc/ Contient la documentation de la première version du projet. Celle-ci à été générée à l'aide de *Doxygen*, elle est disponible en HTML ou en PDF (générée avec \LaTeX).

L'utilisation de *Doxygen* contrairement à *Javadoc* permet d'avoir une génération en PDF, mais également d'avoir l'apparition des diagrammes de classes.

V2/ Contient le dossier avec la version contenant notre strategie pour l'arène commune distante.

Le projet à été développé en Java6.

Table des matières

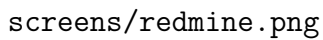
1	Organisation du travail d'équipe	5
1.1	Un outil de gestion de projet : Redmine	6
1.2	Un logiciel de versionnement : Git	7
1.3	Répartition des tâches	7
2	Première version	8
2.1	Nos règles de jeu	8
2.2	Conception	8
3	Seconde Version	9
3.1	Stratégie	9

1

Organisation du travail d'équipe

Pour ce projet, nous étions deux à travailler dessus, ainsi nous avons utilisé plusieurs techniques afin de se coordonner et de limiter les problèmes. Ceci n'est pas notre premier projet ensemble, notre travail en fut simplifié.

1.1 Un outil de gestion de projet : Redmine



screens/redmine.png

FIGURE 1.1 – Affichage des demandes dans Redmine

Pour le projet, nous avons utiliser *Redmine*, une plateforme web de gestion de projet (Cf figure 1.1). Elle nous a permis de simplifier le travail, et de ne rien oublier.

En effet, nous pouvons créer des tâches, signaler qu'elles sont en cours/terminés/en tests, leur donner des dates limites, les affecter à une personne etc. . . Ainsi lorsque l'un de nous commençait une tâche, il le signalait sur le *redmine*, ce qui permettait de tenir au courant son binôme de ses actions et de l'avancée du projet.

1.2 Un logiciel de versionnement : Git

Afin de limiter les problèmes du travail collaboratif, nous avons utilisé un logiciel de versionnement Git. Il a deux intérêts, tout d'abord, nous pouvons travailler à deux en parallèle sur le projet sans se soucier de fusionner notre travail¹.

D'autre part, tous les logs étant enregistrés, nous pouvons savoir qui a fait quoi et quel jour, cela permet de voir également l'avancée du projet.

Enfin, toutes les modifications sont stockées sur le serveur, ainsi en cas de problème, il est très facile de revenir à la version précédente ou même de comparer deux versions afin de voir les changements et de comprendre rapidement pourquoi une fonctionnalité a régressé.

1.3 Répartition des tâches

1. À condition de ne pas travailler sur deux lignes de code identiques

2

Première version

2.1 Nos règles de jeu

2.2 Conception

3

Seconde Version

3.1 Stratégie