

Instructions pour le rendu du projet.

Le respect des consignes ci-dessous sera pris en compte dans l'évaluation du projet.

Archive

Le projet doit être rendu dans une archive `nombinome1_nombinome2.tar.gz` contenant un répertoire `nombinome1_nombinome2` pour faciliter la correction. L'archive doit avoir l'architecture suivante :

```
nombinome1_nombinome2/  
+ src/                // sources du projet (rappel : en anglais)  
+ lib/                // librairies externes (jdom.jar, ...)  
+ xml/                // DTD et exemples de fichiers XML  
+ properties/         // Les fichiers de propriétés  
+ rapport/            // rapport + document en anglais  
+ build.xml            // fichier de configuration ant
```

Le fichier de configuration `ant` doit comporter les cibles suivantes :

- **compile** : compile le projet et crée un répertoire `classes/` contenant les fichiers `.class`
- **build** : crée une archive jar exécutable `game.jar` du projet et la place dans un répertoire `build/`
- **doc** : génère la javadoc du projet et la place dans un répertoire `docs/`
- **clean** : supprime les répertoires “inutiles” comme `classes/`, `build/` et `docs/`.

Il ne faut évidemment pas fournir les répertoires `build/`, `classes/` et `docs/` dans votre archive `.tar.gz` puisqu'ils peuvent être générés automatiquement grâce à `ant`.

Vous devez rendre avec votre projet au moins 2 fichiers XML générant des jeux différents et votre projet doit pouvoir fonctionner au moins dans 2 langues “fournies” également.

L'archive doit être exclusivement rendue via l'interface PROF, date limite le 15 mars, 12h (sinon elle sera refusée et donc non corrigée). Vérifiez bien avant cette date limite que vous êtes bien enregistrés pour votre groupe en PL dans cette interface.

Rapport

Le rapport doit être écrit à destination de quelqu'un sachant programmer mais ne connaissant pas le sujet du projet. Il doit donc contenir toutes les informations nécessaires à une éventuelle poursuite du projet.

Il vous faut donc :

- présenter le projet de manière générale (cahier des charges).
- dire, le cas échéant, quelle partie du cahier des charges n'est pas respectée et décrire, le cas échéant, ce qui a été ajouté au cahier des charges et consigner des originalités qui pourraient différencier votre projet d'un autre,
- écrire un manuel expliquant comment utiliser votre jeu (guide de l'utilisateur final).
- donner des diagrammes UML **commentés** présentant les classes de votre projet et citer les problèmes de conception principaux (selon vous) que vous avez rencontrés lors de votre projet et décrivez pour un ou deux d'entre eux comment vous les avez gérés,
- critiquer votre projet suivant la logique ouvert/fermé : que permet de faire facilement votre code ? Quelles libertés laisse-t-il pour faire des extensions et quelles sont alors les contraintes à respecter (nommage, interfaces à implémenter, etc.) ?
- présenter **précisément** ce qu'il faut faire pour étendre le projet : ajout de personnages, de chose, évolution du menu (nouvelles capacités), de types de terrain, etc.
Cette partie peut prendre la forme d'un tutoriel.
- expliquer votre utilisation des propriétés et plus généralement l'externalisation des paramètres de l'application,
- une conclusion bilan de votre travail sur votre projet.

Le rapport doit être rendu au format PDF.

Rappel : Vous aurez une modification à réaliser entre le 15 et le 22 mars et il y aura une soutenance individuelle (durée 5 à 10mn) portant sur la version avec extension la semaine du 22 mars. Le projet devra fonctionner dans votre environnement personnel.