

TD 14 - Projet Zeldiablo - Fonctionnalités (2h)

Ce TP est le début du projet qui se poursuivra en autonomie fin de semestre.

Le projet se fera par groupe de 4 et utilisera git pour coordonner le travail. Un seul *repository* sera utilisé et partagé par groupe.



Objectif pédagogique

A l'issue de ce TP, vous devriez savoir

- structurer correctement un dépôt git (séparation du projet et des documents) ;
- proposer des fonctionnalités simples correctement présentées.

1 Démarche incrémentale à suivre

La démarche suivie dans le cadre du projet sera une démarche incrémentale consistant à ajouter progressivement des fonctionnalités au projet.

Attention :

Cette démarche se fera en plusieurs itérations. Une itération consiste à

- décider des fonctionnalités (deux max) à ajouter à cette version (10 min);
- expliciter les critères de validation (10 min) ;
- réfléchir collectivement à une conception (1h);
 - ☐ pour chaque fonctionnalité, dessiner un ou plusieurs diagrammes de séquence ;
 - ☐ faire des diagrammes de séquence pour les cas très particuliers ;
 - ☐ mettre à jour votre diagramme de classe en conséquence.
- se répartir le travail de manière équitable à partir de la conception (10 min);
- écrire les tests à valider et réaliser le code de l'application(1h30 min);
- valider grâce aux tests les nouvelles fonctionnalités par rapport aux critères exprimés et vérifier que le reste de votre application fonctionne encore correctement (30 min).

Une fois que cette version et ses fonctionnalités sont validées, on peut passer à la suivante.

Attention :

Le cœur de cette démarche est de ne considérer qu'une ou deux fonctionnalités à la fois. Ne vous concentrez que sur une fonctionnalité et ne passez à la suite que lorsque cette fonctionnalité sera validée.

2 Mise en place du dépôt (10 min)



Question 1

Faites un groupe de 4, créez un dépôt git nommé `2017_C00_Zeldiablo_logins` où *logins* représente l'ensemble des logins du groupe. Partagez le dépôt entre vous et avec votre enseignant.

Votre dépôt doit être structuré de la manière suivante:

- un répertoire "*projet_zeldiablo*" destiné à contenir votre projet eclipse (cf TP suivant) ;
- un répertoire "*documents*" destiné aux documents de conception de chaque version;
- un répertoire "*documents\version_1*" destiné à contenir vos fonctionnalités, diagrammes de classe et de séquence de la version 1;
- un répertoire "*documents\version_n*" pour chaque conception.



Question 2

Ajouter à la racine du dépôt un fichier `README.txt` contenant les noms des membres de votre groupe.



Question 3

Ajouter à la racine du dépôt un fichier `.gitignore` permettant de filtrer les fichiers du dépôt. Parmi ceux-ci, vous penserez à filtrer :

- les fichiers liés à Eclipse ;
- les fichiers `.class` ;
- les fichiers liés au système d'exploitation utilisé (ex `Thumbs.db`).

A l'issue de cette étape, votre projet doit être structuré selon la figure 1.

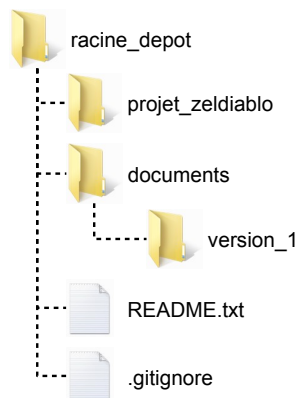


Figure 1: Structuration du dépôt

3 Analyse (1h)

Dans ce premier TP, il s'agit de réfléchir aux classes de votre jeu et à la manière de concevoir votre application.

Attention :

L'objectif de cette séance est de produire **un premier document de conception** décrivant ce que vous envisagez de faire dans la version 1 de l'application. Ce document contient l'ensemble des éléments à produire dans ce TP (diagrammes de classe, liste des tâches, ...)

Attention :

Vous ne coderez l'application que lorsque votre conception et vos diagrammes de classe et de séquence auront été présentés à votre enseignant.

3.1 Classes et interfaces (20min)

A l'aide du texte descriptif ci-dessous, faites rapidement la liste des classes qui vous paraissent intéressantes. Ne rentrez pas dans le détail, ce premier diagramme servira à avoir une vision d'ensemble à raffiner dans les prochaines itérations.

Description du problème Dans ce jeu, le joueur déplace un aventurier dans un labyrinthe de dimensions connues. L'aventurier contrôlé par le personnage se déplace case par case.

Le labyrinthe est constitué de cases qui peuvent être soit vides soit des murs. Certaines de ces cases sont donc traversables (comme les cases vides) d'autres non (comme les murs) ce qui restreint les mouvements de l'aventurier.

Les déplacements possibles pour le héros sont les différentes directions cardinales (N, S, E, O). On suppose que le NORD est situé vers le Haut de la carte. Le jeu reçoit une commande en entrée et évolue en fonction de l'action choisie.

Le labyrinthe contient une entrée où le personnage sera situé à la création du jeu.



Question 4

Faites un diagramme de classe élémentaire (pas d'attributs ni de méthodes) présentant ces classes. Ajoutez ce diagramme dans le répertoire *"documents\version_1"* de votre dépôt.

3.2 Fonctionnalités du problème (20 min)

Pour rappel, une fonctionnalité correspond à un comportement possible de l'application. Elle inclut un descriptif textuel et des **critères de validation** qui serviront pour déterminer les tests unitaires.

A l'aide d'un brainstorm et suite à ce descriptif, faites ressortir les différentes fonctionnalités de votre application en les isolant une par une. Vous pouvez d'abord réfléchir chacun de votre côté puis mettre en commun les fonctionnalités.

Attention :

Trouvez les fonctionnalités les plus simples possibles. N'hésitez pas à séparer une fonctionnalité en deux plus petites si c'est possible (séparer l'affichage des personnages du calcul de son déplacement - il s'agit de deux fonctionnalités différentes).



Question 5

Lister rapidement les fonctionnalités auxquelles vous pensez. Stocker l'ensemble dans un fichier texte (une ligne par fonctionnalité) dans la racine du répertoire **documents**. Faites valider par votre enseignant.

3.3 Sélection des Fonctionnalités version 1 (20 min)

Prenez le temps de regarder sur chaque fonctionnalité si

- cette fonctionnalité est complexe à réaliser ;
- cette fonctionnalité est prioritaire ;
- cette fonctionnalité nécessite d'autres fonctionnalités.

Décider ensuite la liste des fonctionnalités à réaliser sur lesquelles vous vous **engagez** pour le prochain TP de deux heures.

Choisissez les deux/trois fonctionnalités les plus simples. Ne surchargez pas ce qu'il y a à faire et gardez à l'esprit qu'il faudra écrire la conception correspondante, les tests et que l'application devra être fonctionnelle après deux heures.

Ce fichier texte doit avoir la forme suivante :

```
* Fonctionnalités version 1

- fonctionnalité 1 : nom
- descriptif texte : [...]
- critères de validation :
  - critère 1 [...]
  - critère 2 [...]

- fonctionnalité 2 : nom
- descriptif texte : [...]
- critères de validation :
  - critère 1 [...]
  - critère 2 [...]

[....]
```

Question 6

Écrire dans un fichier texte `fonctionnalites_v1.txt` la liste des fonctionnalités à traiter dans votre première version. Chaque fonctionnalité doit posséder un nom, un descriptif rapide et des critères de validation. Déposez ce fichier dans le répertoire `documents\version1`.

A l'issue du TP, votre dépôt doit être structuré selon la figure 2.

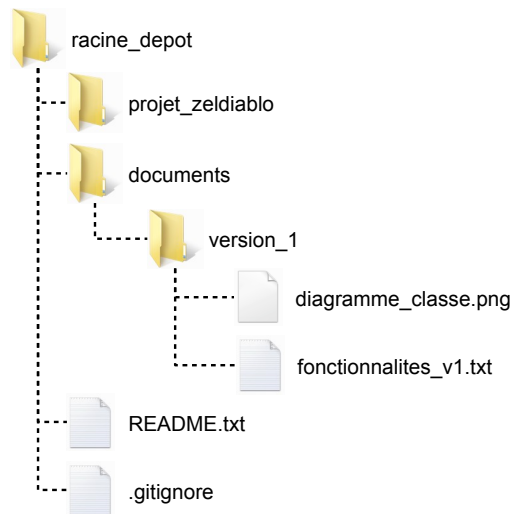


Figure 2: Structuration du dépôt