

# Projet de Génie Logiciel

Sujet proposé par

Pierre Gérard

pierre.gerard@iutv.univ-paris13.fr

*DUT Informatique, semestre 4 décalé*

*IUT de Villetaneuse*

*Université de Paris 13*



## Résumé

Ce projet a pour principal objectif l'application de techniques de GL dans la réalisation d'un projet concret et réel. Une attention toute particulière sera donc portée au processus de conception et de développement. Il est difficile, dans le cadre de l'IUT, de proposer un projet si gros que l'absence de méthode mènerait inévitablement à un échec. Étant données les contraintes de temps, nous ne pouvons que proposer un projet relativement modeste et qui pourrait peut-être être mené à bien sans méthode bien définie. Néanmoins, dans un but pédagogique, il vous est demandé de bien vouloir jouer le jeu de l'organisation plutôt que du produit fini à tout prix : il s'agit d'un projet de GL et pas de programmation. En tout état de cause, la méthode tiendra une part très importante dans l'évaluation finale. Un projet pas tout à fait achevé mais développé selon les règles de l'art sera mieux évalué qu'un projet complètement fonctionnel mais bricolé.

## 1 Organisation du projet

### 1.1 Méthode

Vous devrez former des équipes de 5 personnes. Le projet s'organise en 6 séances de 4 heures. A l'issue de chacune de ces séances, vous devrez rendre votre travail à l'enseignant, ainsi que la dernière version des documents de suivi. Au début de chaque séance, il vous est recommandé de vous réunir pour vous accorder quant au travail à réaliser.

**Séance 1** Mise en place des groupes de projet, lecture de l'énoncé, rédaction de questions à l'enseignant qui jouera le rôle du client, rédaction d'un rapport d'interview récapitulant les réponses apportées, rédaction du cahier des charges, production du WBS dans sa première version, estimation des coûts, attribution des responsabilités au sein de l'équipe.

**Séance 2** Finalisation du WBS, estimation des coûts, modèle de processus et transformation en diagramme PERT, planification et affectation des ressources humaines aux différentes tâches.

**Séances 3,4,5,6** Le travail de chacun durant ces séances est défini par la planification effectuée. Chaque tâche doit donner lieu à un artefact qui sera rendu à la fin de la séance. Les éventuels retards ou avances devront être commentés en marge de l'analyse de la valeur acquise mise à jour.

**Soutenance** A l'issue des six semaines de travail, une soutenance est organisée pour la validation du logiciel. Une démonstration sera effectuée et chacun se verra interroger individuellement. Les deux premières séances sont primordiales parce qu'elles vous permettront d'organiser le travail pour que chaque membre du groupe ait quelque chose à faire et que le projet soit terminé dans les temps.

Si la planification effectuée s'était avérée irréaliste et que l'avancement réel en était très éloigné, il vous est demandé de prendre du recul et d'analyser les causes du dysfonctionnement. On peut s'attendre à ce que les choses ne fonctionnent pas comme prévu et le contraire serait même surprenant ou suspect. Le travail d'une équipe qui saurait analyser des dysfonctionnement et proposer des solutions serait bien mieux apprécié que celui d'une équipe qui chercherait à dissimuler d'éventuelles déconvenues derrière un produit fini, même spectaculaire.

## 1.2 Spécifications

Il vous appartiendra de rédiger le Cahier des Charges et de définir vous-même les besoins détaillés. Ainsi, vous pourrez moduler vos ambitions en fonction de ce dont vous vous sentez capables. Il vaut mieux s'engager sur un projet modeste et le mener à terme plutôt que d'échouer dans un projet ambitieux. Ceci dit, de deux projets finis, on préférera souvent le plus ambitieux quand même.

Si vos ambitions initiales s'étaient avérées trop limitées et si vous terminiez plus tôt que prévu, libre à vous de définir un nouveau cahier des charges, même sommaire, pour porter des améliorations au logiciel dans sa première version. Cette première version représentera alors l'existant et vous vous placerez dans l'optique d'une maintenance d'extension, pour permettre détendre les fonctionnalités du logiciel.

Vous êtes également invités à penser le processus de développement de votre projet de manière à pouvoir produire des versions intermédiaires exécutable. Ainsi, même dans le cas où vous auriez été trop optimistes au départ, vous serez en mesure de montrer un produit fonctionnel, même partiellement achevé.

## 1.3 Documentation

Différents documents devront être produits au cours du cycle de vie du développement du logiciel. Les documents suivants sont des exemples qui devront être produits dans tous les cas.

**Cahier des charges** Un document annexe donne un squelette de Cahier des Charges

**WBS** Le « Work Breakdown Structure » découpe le projet en tâches organisées hiérarchiquement

**Estimation des coûts** Le temps de travail nécessaire pour chaque tâche du WBS est évalué

**Modèle de processus** Ce modèle définit les tâches et les artefacts qui seront produits. On l'utilisera pour en déduire un diagramme de PERT

**Diagramme PERT** Ce diagramme organise les tâches définies dans de WBS

**Feuille de suivi** Ce document récapitule l'avancement dans chacune des tâches : qui est intervenu sur la tâche en question et quelle quantité de travail il a fourni

**Analyse de la valeur acquise** Ce document reprend les informations des documents précédents pour analyser la progression du projet.

**Spécifications UML** Spécification des besoins en UML

**Conception UML** Conception du logiciel en UML

**Implantation** Code source du logiciel

**Manuel utilisateur** Manuel utilisateur du logiciel

**Cas et rapports de test** Les éléments du logiciel devront être testés et des tests de non régression possibles. Les rapports de tests consignent les résultats des tests effectués. Selon les différentes tâches auxquelles vous vous livrez, d'autres documents pourront vous paraître pertinents, aussi cette liste n'est-elle en aucun cas exhaustive.

## 2 Application à développer

### 2.1 Présentation du projet

Il vous est demandé de développer un jeu de type « MMORPG » au tour par tour. Parmi les exemples célèbres de jeux de ce type, on peut citer « Mountyhall » (<http://www.mountyhall.com>) ou encore « Les Souterrains de Delain » (<http://www.jdr-delain.net>). MMORPG signifie « Massively Multi-Player Online Role Playing Game » :

- « Massively Multi-Player » parce que le jeu doit être multi-joueurs et doit permettre à un grand nombre de personnes de jouer simultanément
- « Role Playing Game » parce que chaque joueur incarne un personnage livrant bataille dans un monde peuplé de monstres et d'autres personnages.
- « Online » parce que le jeu doit se dérouler en ligne au moyen d'une connexion IP. L'état du monde est en permanence stocké sur un serveur et les clients (joueurs) doivent s'y connecter pour pouvoir jouer.

Le choix des technologies employées est laissé à votre appréciation mais on pourrait envisager :

- Une base de données MySQL pour stocker en permanence l'état du monde ;
- Des requêtes PHP pour que les clients puissent interagir avec le monde via leur navigateur.

### 2.2 Règles du jeu

Les règles du jeu suivantes sont données à titre indicatif pour vous permettre de mieux démarrer. Elles sont incomplètes et vous pouvez les adapter à vos besoins et vos envies.

#### 2.2.1 Caractéristiques des PJ

On distingue deux types de personnages : les PJ (personnages joueurs) et les PNJ (personnages non joueurs, souvent des monstres). Les PJ sont définis par un certain nombre de caractéristiques :

**Force** pour la puissance physique

**Adresse** qui représente la capacité du personnage à mouvoir son corps comme il l'entend

**Résistance** qui représente le point jusqu'auquel le corps du personnage peut résister aux agressions extérieures

Chacune de ces caractéristiques se voit attribuer un niveau mesuré en degrés. Ces niveaux sont comparés pour opposer certaines caractéristiques à d'autres. Par exemple, pour déterminer si un PJ peut en toucher un autre lorsqu'il attaque, on comparera leur niveau d'adresse.

La création de PJ par les joueurs doit être personnalisée. Chaque personnage se voit attribuer 18 degrés à répartir entre les trois caractéristiques.

#### 2.2.2 Tirages aléatoires

Pour mettre un peu d'incertitude dans la résolution des actions, on introduit un facteur aléatoire en transformant les degrés dans les caractéristiques niveaux exprimés en dés (D). Chaque dé donne lieu à un tirage aléatoire entre 1 et 6. Par exemple, un tirage de  $2D+2$  est en fait la somme de deux nombres aléatoires tirés chacun entre 1 et 6, auquel on ajoute 2. On a 1 dé pour 3 degrés et les degrés restants sont des unités additionnelles. Par exemple, une caractéristique avec 7 degrés donne lieu à un tirage de  $2D+1$  et une caractéristique avec 14 degrés donne lieu à un tirage de  $4D+2$ .

Les objets portés par les PJ leur permettent d'accroître certaines de leurs capacités. Par exemple, une arme permettra de faire plus de dégâts et une armure permettra de mieux encaisser les coups. Ces objets donnent donc lieu à certains bonus lorsqu'ils sont portés.

### 2.2.3 Capacités

Les caractéristiques et les bonus apportés par les objets permettent de calculer des niveaux de capacités comme :

**L'initiative** (adresse - bonus vêtements) qui représente la capacité des personnages à attaquer avant les autres

**L'attaque** (adresse) qui représente l'habileté du personnage à en toucher un autre quand il l'attaque

**L'esquive** (adresse) qui représente l'habileté du personnage à éviter les coups dont il est la cible

**La défense** (résistance + bonus vêtements) qui représente la capacité du personnage à supporter les dommages physiques

**Les dégâts** (force + bonus armes) qui représentent la puissance des coups portés par le personnage.

Les PJN ont aussi ces capacités. Pour les monstres, elles sont définies directement et ne sont pas calculées (les monstres n'ont pas de caractéristiques et ne portent pas d'objets) mais pour d'éventuels autres types de PNJ plus « évolués » on pourra considérer des caractéristiques et des bonus liés aux objets.

### 2.2.4 Points d'action

Pour entreprendre une action, un joueur doit dépenser des points d'action (PA). Ces PA permettent aux joueurs d'agir : tant qu'il en a, le joueur peut dépenser des points d'action à tout moment en choisissant une action parmi :

**Attaquer** (coût 3PA)

**Se déplacer** d'une case (coût 2PA)

**Utiliser un objet** (coût variable selon l'objet)

Dès qu'une action est entreprise, son coût est déduit du nombre de PA restants au joueur. Une action ne peut pas être entreprise si le joueur ne dispose pas de suffisamment de PA.

Les PA se regagnent par période de 12 heures, selon un barème qui dépend de l'initiative du PJ : un personnage avec beaucoup d'initiative regagne plus de PA par tranche de 12 heures qu'un autre plus pataud. A la fin d'une période, la moitié des PA (arrondie à l'inférieur) restants sont conservés pour la période suivante.

L'heure de la fin d'une période de 12 heures est appelée DLA (date limite d'action). La première DLA correspond à la date de création d'un joueur à laquelle on ajoute 12 heures. La seconde intervient encore 12 heures plus tard. A tout moment, il est possible à un joueur de reculer sa DLA de manière à synchroniser autrement ses cycles de 12 heures.

### 2.2.5 Utilisation d'objets

Le résultat de l'utilisation d'un objet dépend de l'objet utilisé. Dans le cas d'une arme ou d'un vêtement, cette action prend la forme d'un équipement / déséquipement de l'objet. Dans le cas d'un autre objet comme une potion, par exemple, il s'agira d'appliquer les effets de l'objet et de le supprimer. Une potion de soins permettra par exemple au PJ de regagner un nombre défini de niveaux de blessures. Une potion molotov permettra d'infliger des dégâts à tous les personnages sur les cases alentours.

### 2.2.6 Combats

En cas d'attaque, il faut désigner un personnage cible qui doit se situer sur une case adjacente. La cible tire un score d'esquive et l'attaquant un score d'attaque. Si le score d'attaque est supérieur au score d'esquive, la cible est touchée et on procède à la détermination des blessures.

Pour ce faire, on procède de manière similaire : le personnage touché tire un score de défense et l'attaquant tire un score de dégâts en utilisant les capacités adéquates. Si le score de défense est supérieur au score d'attaque, le personnage s'en tire indemne. Dans le cas contraire, il est blessé et son niveau de blessures dépend de la différence entre les deux scores.

Chaque personnage dispose de 6 niveaux de blessure :

- 1 Blessures superficielles
- 2 Légèrement blessé
- 3 Blessé
- 4 Gravement blessé
- 5 Inconscient (dans ce cas, le personnage ne peut plus jouer : il doit attendre de recevoir une potion de soins ou de récupérer naturellement)
- 6 Mort (dans ce cas, le personnage est retiré de la partie, toutes ses possessions tombent au sol et le joueur ne joue plus)

Dès qu'un personnage est blessé, un niveau de blessure est perdu par tranche de 3 points de différence entre le score de dégâts et le score de défense. Les potions de soin permettent de regagner des niveaux de blessure selon la puissance de la blessure. Il faut attendre 2 cycles pour qu'un niveau de blessure soit regagné naturellement.

### 2.2.7 PNJ

Le comportement et les capacités des PNJ restent à définir. Si un PNJ meurt et qu'il possède des objets, ceux-ci restent au sol sur la case de sa mort. Dans le cas d'un monstre, un nombre aléatoire d'objets tirés au hasard apparaissent au sol.

### 2.2.8 Expérience

Chaque fois qu'un PJ réussit une attaque, il gagne 1 point d'expérience (PX). Il en gagne plus dès qu'il parvient à tuer un adversaire. Son total de PX est comparé à celui de son ennemi et un ratio est calculé entre les deux (les monstres et PNJs ont un nb de PX fixe et déterminé par les concepteurs du jeu). Si un PJ A tue un personnage B, que A a 50 PX et que B en a 100, le ratio est alors de 2. De ce ratio dépend le nombre de PX gagnés par l'attaquant victorieux.

Ces PX peuvent être à tout moment dépensés par les joueurs pour améliorer les caractéristiques de leur personnage. Une caractéristique avec un haut niveau sera bien plus difficile à faire progresser encore qu'une caractéristique avec un niveau assez faible.

La manière de calculer les PX attribués et le coût des améliorations sont laissées à votre convenance.

## 2.3 Fonctionnalités attendues

L'application développée devra permettre au moins ...

- Aux administrateurs de créer un monde
- Aux administrateurs de créer de nouveaux types de monstres
- Aux administrateurs d'administrer les comptes des abonnés
- Aux joueurs de s'inscrire
- Aux joueurs de choisir leurs actions
- Aux joueurs de gérer leur inventaire (sac à dos, mains droite et gauche, éléments d'armure...)
- Aux joueurs de s'envoyer des messages

On pourra en outre imaginer toute fonctionnalité secondaire que l'on estimera souhaitable.