Projet de Génie Logiciel
Rapport de Planification
Année Académique 2014-2015

Groupe numéro: **7**
Membres du groupe: **DELGRANGE FLORENT, LECOQ ALEXIS, LEMPEREUR MARTIN**

BAC 2 INFO
Faculté des Sciences, Université de Mons
martin.lempereur@hotmail.be {florent.delgrange — alexis.lecoq}@student.umons.ac.be

October 14, 2014

Ce rapport de planification est rendu dans le cadre du cours "Projet de Génie Logiciel" (dispensé par le Prof. *Tom Mens* en année académique 2014-2015). Le but de ce rapport est d'expliquer quels sont les objectifs du projet, quelles sont les différentes tâches à réaliser et planifier ces différentes tâches.

1 Introduction

1.1 Objectifs

Le travail s'inscrit dans le cadre du projet de Génie logiciel.
Le travail consiste à concevoir un jeu de rôle dont l'unvivers est celui de l'Unviversité.
Nous conceverons donc une application java en décomposant la réalisation en plusieurs étapes clés.

1.2 Exigences fonctionnelles

Nous devons implémenter au moins une quête principale et secondaire de l'année d'étude BAC2 et un framework permettant au joueur de réaliser les différentes quêtes implémentées.

1.3 Exigences non-fonctionnelles

L'application doit être executable sur les systèmes exploitation Linux, Mac et Windows.
Le framework pourra supporter l'ajout de quêtes, de caractèristiques et même de classes réalisées par d'autres personnes ayant respecté l'interface commune.

1.4 Contraintes de temps

Les dates importantes sont :
Mardi 14 octobre 2014 : date de remise pour la phase 1 : Rapport de planification et cahier des charges.
Dimanche 30 novembre 2014 : date de remise pour la phase 2 (rapport de modélisation et maquette de l'interface utilisateur) .
Fin de la semaine du 10 ou début de la semaine du 17 décembre 2014 : réunions d'inspection de modélisation.
Jeudi 3 avril 2014 : date de remise de la phase 3 : Implémentation.
Les dates pour la remise de la phase 4 et de la défense orale et des tests d'acceptation sont encore á determiner.
Remarque : du lundi 22 décembre au vendredi 2 janvier, le blocus commence, il sera donc extrêmement difficile d'être productif pendant cette période.

1.5 Contraintes de budget

Aucun achat n'est prévu. De plus, tous les logiciels indispensables á la réalisation du projet sont soit gratuits, soit disponibles avec une clé académique sur moodle.

2 Ressources

2.1 Les ressources humaines (personnel)

nom	rôle	durée
Florent Delgrange	développeur	environ 8 mois
Martin Lempereur	développeur	environ 8 mois
Alexis Lecoq	développeur	environ 8 mois
Tom Mens	Titulaire du projet	
Mathieu Goeminne	Enseignant	
Romuald Deshayes	Assistant	

Table 1. Ressources humaines.

2.2 Les ressources logicielles

Les logiciels utilisés dans le cadre du projet sont :
Gantt Project (gratuit), utilisé pour réaliser les diagrammes de Gantt et de PERT. http://www.ganttproject.biz/
Visual Paradigm (liscence académique), utilisé pour réaliser les diagrammes lors de la phase 2 (modélisation ; diagramme de cas d'utilisation, de classe, d'état, de séquence,...) http://www.visual-paradigm.com/
IntelliJ (liscence académique), environnement de développement. https://www.jetbrains.com/idea/
Eclipse (gratuit), environnement de développement. https://www.eclipse.org/
Git (gratuit), logiciel de gestion de versions. http://git-scm.com/
Maven (gratuit), gestion et automatisation de production des projets logiciels Java. http://maven.apache.org/
Vim (gratuit), éditeur de texte. http://www.vim.org/.
Kile (gratuit), éditeur de texte pour les documents crits en Latex. http://kile.sourceforge.net/
JUnit (gratuit), bien qu'il soit intégré par défaut dans Eclipse et intelliJ, il est utile de spécifier que nous utilisons ce framework pour réaliser nos tests unitaires. http://junit.org/
LibGDX (gratuit), framework utilisé pour concevoir l'interface graphique. http://libgdx.badlogicgames.com/
Tiled Map Editor (gratuit), outil ayant pour objectif de crer des cartes oú le joueur pourra se déplacer (interface graphique). http://www.mapeditor.org/

2.3 Les ressources matérielles

Nom	Systeme d'exploitation	Memoire	Processeur	Carte graphique	Disque dur
Lempereur Martin	Linux	2 Go	Intel Core 2 Quad CPU Q6600 @ 2.40GHz 4		300 Go
Delgrange Florent Pc 1	Windows 8.1 pro	8 Go 1600 MHz DDR3	Intel Core i5 3470 CPU @3.20 GHZ	His Radeon R9 280X	1To
Delgrange Florent Pc 2	Mac OS X 10.9.5	4 Go 1067 MHz DDR3	2.53 GHz Intel Core 2 Duo	NVIDIA GeForce 9400M 256 Mo	
Lecoq Alexis	Smoby	VTech	2Ko		

Table 2. Ressources materielles

3 Analyse des risques

3.1 Identification des risques

Risque	Probabilité	Sévérité
1 : Perte de code	basse	sérieuse
2 : Livrable final non terminé ou incomplet.	modérée	sérieuse
3 : Code déffectueux	haute	sérieuse
4 : Problème de compatibilité.	modérée	sérieuse
5 : Abandon de la part d'un des collaborateurs	basse	sérieuse
6 : Code incomplet	modérée	sérieuse
7 : Une des personnes est absente ou malade.	haute	tolérable
8 : Incomprehension d'un problème	haute	tolérable
9 : Problèmes d'horraire ou de communication	haute	tolérable

Table 3. Analyse des risques.

3.2 Gestion des risques

Risque	Comment eviter	Comment vérifier	Resoudre ou reduire le risque
1	Utilisation d'un gestionnaire de versions	Comparer notre code avec le code d'un autre collaborateur	Rcuperer le code d'un autre collaborateur
2	Vérifier avant l'envoie si le livrable est complet	Relire lénoncé et regarder si tout apparait dans le livrable	Si le délai n'est pas passé envoyer une nouvelle version complète
3	Effectuer réguliérement des tests unitaires	Essayer de compiler ou bien tester l'application	Repérer le code déffectueux avec des tests unitaires
4	Tester le code sur plusieurs environnement	Demander a un collaborateur qui utilise un autre environnement de tester le code	Repérer les endroits non compatibles et les rendre plus génériques
5	Connaitre le travail effectuer par chaque membre et savoir comment bien le répartir	Envoi d'un email au membre	Répartir le travail du membre qui a abandonné entre les autres membres
6	Rédiger une bonne javadoc et comparer la doc avec l'ennoncé	Tester toute les focntionnalités demandées dans l'énoncé	Si le délai n'est pas passé envoyer une nouvelle version complète du code
7	Connaitre le travail effectuer par chaque membre et savoir comment bien le répartir	Envoi d'un email au membre	Répartir le travail du membre qui est absent entre les autres membres
8	Ameliorer son niveau d'anglais	Essayer de comprendre la documentation qur internet	Faire une demande d'aide sur un forum francophone
9	Reunion et organisation au sein du groupe pour fixer des horraires	Demander a toutes les personnes d'être présentes	Repport â une date ultèrieure

Table 4. Gestions des risques.

4 Répartition du travail

4.1 Work Breakdown Structure

Nom	Date de début	Date de fin	Resposable
Phase 2 : conception et modélis...	14/10/14	30/11/14	Alexis, Florent, Martin
Maquette de l'interface utilisa...	14/10/14	25/11/14	Alexis, Florent, Martin
Diagramme de cas d'utilisati...	14/10/14	21/10/14	Alexis, Florent, Martin
Diagramme de classe	14/10/14	04/11/14	Alexis, Florent, Martin
Diagramme d'état	05/11/14	14/11/14	Alexis, Florent, Martin
Diagramme de séquence	15/11/14	25/11/14	Alexis, Florent, Martin
Rapport de modélisation	26/11/14	30/11/14	Alexis, Florent, Martin
Phase 3 : Implémentation	17/12/14	03/04/15	Alexis, Florent, Martin
Implémentation	17/12/14	17/03/15	Alexis, Florent, Martin
Test unitaires + correction de...	17/12/14	24/03/15	Alexis, Florent, Martin
Javadoc	17/12/14	24/03/15	Alexis, Florent, Martin
Manuel d'utilisation	18/03/15	03/04/15	Alexis, Florent, Martin
Rapport d'implémentation	25/03/15	03/04/15	Alexis, Florent, Martin
Phase 4 : Intégration	06/04/15	29/05/15	Alexis, Florent, Martin

Fig. 1. Tableau des tâches.

Remarque : la répartition du travail ainsi que les dates ne sont pas définitives.

4.2 Etapes clés

Date	étape clé	Livrables
14 octobre 2014	Date de remise pour la phase 1	Rapport de planification et cahier des charges
30 novembre 2014	Date de remise pour la phase 2	Rapport de modélisation et maquette de l'interface utilisateur.
10 au 17 décembre 2014	Réunions d'inspection de modélisation	
3 avril 2014	Date de remise de la phase 3	Implémentation
?	Date de remise de la phase 4	Intégration
?	Défense orale, tests d'acceptation par les enseignants	

Table 5. Tableau d'étapes clés.

5 Ordonnancement

5.1 Diagramme GANTT

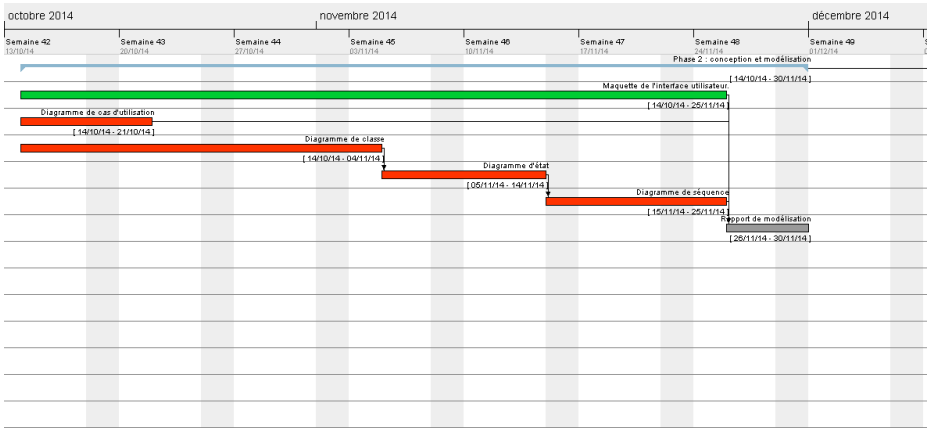

Fig. 2. Diagramme Gantt de la phase 1.

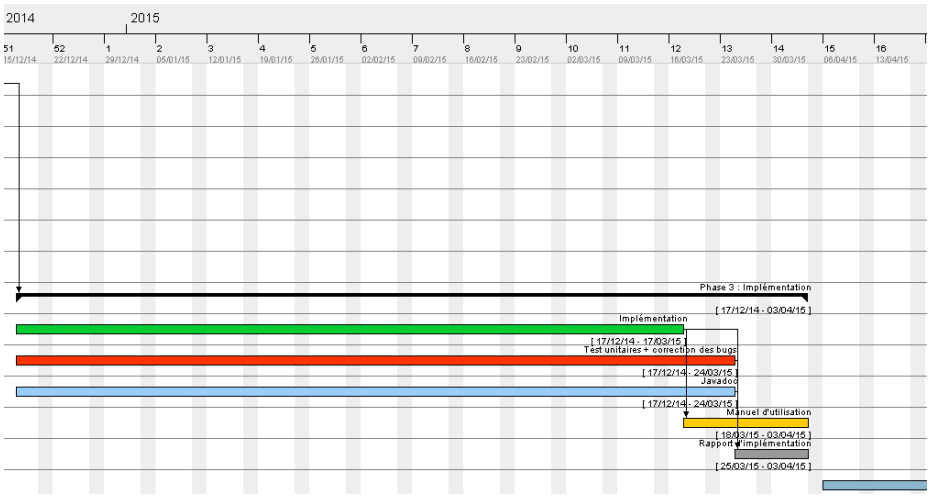

Fig. 3. Diagramme Gantt de la phase 2.

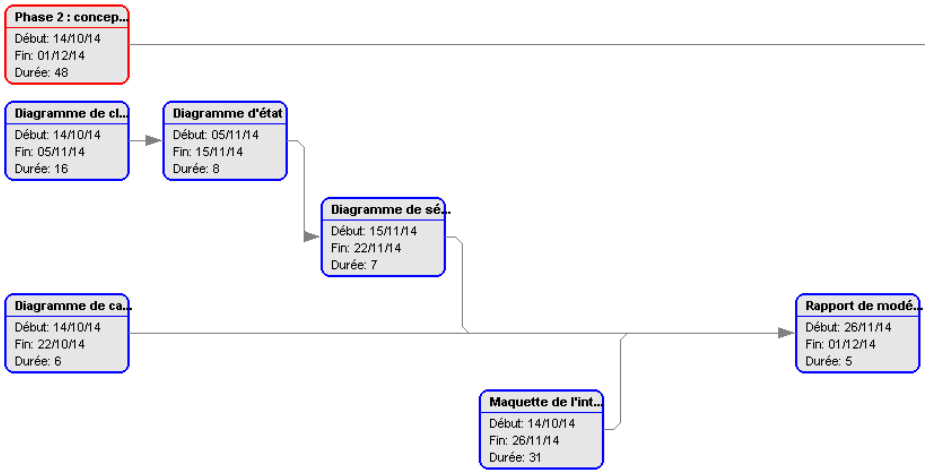

Fig. 4. Diagramme PERT de la phase 1.

5.2 Diagramme PERT

5.3 Analyse de l'ordonnancement

5.4 Surveillance

En cours de projet pour vérifier si un retard peut arriver, nous allons comparer le pourcentage de la tâche terminé par rapport au pourcentage de temps écoulé.

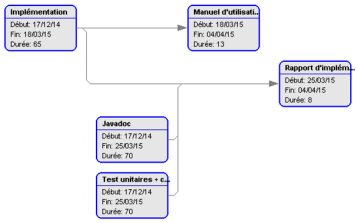

Fig. 5. Diagramme PERT de la phase 2.

Tâche	Durée (jours)	Effort	Earliest Start	Latest Start	Slack Time	Free Float
Diagramme de classes	16	Eleve	14/10/14	14/10/14	0	0
Diagramme de cas d'utilisation	6	Moyen	14/10/14	24/11/14	41	41
Diagramme d'tats	8	Moyen	05/11/14	05/11/14	0	0
Diagramme de squences	7	Moyen	15/11/14	15/11/14	0	0
Maquette de l'interface	31	Eleve	14/10/14	14/10/14	0	0
Rapport de modlisation	5	Moyen	26/11/14	26/11/14	0	0
Implementation	65	Eleve	17/12/14	17/12/14	0	0
Javadoc	70	Faible	17/12/14	17/12/14	0	0
Tests unitaires	70	Eleve	17/12/14	17/12/14	0	0
Manuel d'utilisation	13	Faible	18/03/15	18/03/15	0	0
Rapport d'implementation	8	Moyen	25/03/15	25/03/15	0	0

Table 6. Informations temporelles importantes sur les tâches.

On pourra ainsi rattraper notre retard au augmentant notre temps de travail sur cette tâche par exemple.
Pour éviter tout retard nous respecterons les dates de début de tâche.

En ce qui concerne les risques, nous essayerons au maximum d'application la gestion des risques décrite dans le tableau de gestions des risques. (Table : 4, page : 5)