

Ultimade

ULTIMADE

Cahier des charges Version 1.2

Ultimaade Team ultimaade@gmail.com
Eugène NGONTANG ngonta_e@epitech.net

Résumé : *Ultimaade est un logiciel libre et multiplateforme, de création d'histoires et de support multimédia.*

La principale fonction du logiciel est l'édition de romans visuels (visual novels en anglais), type particulier de jeux vidéos très répandu au Japon et peu connu en occident.

Ultimaade permettra également de faire des présentations/montages type vidéo ou texte, offrant ainsi aux enseignants par exemple, de créer du contenu pédagogique selon leurs domaines de compétence.

Dans un sens plus large le logiciel est destinée à tout type d'utilisateur, désirant créer une application multimédia autonome et multiplateforme : technicien ou non, enfant ou adulte

Ce document constitue la deuxième version du cahier des charges du projet, rédigée pour prendre en compte les changements issus de la première analyse.

Il décrit la manière dont l'équipe prendra en charge l'ensemble des phases du projet (analyse, conception, développement, documentation), les paramétrages, la reprise des données et le support aux utilisateurs du logiciel.

Nous aborderons dans un premier temps le contexte général du projet et les parties impliquées. Ensuite nous présenterons le projet d'un point de vue fonctionnel, puis nous décrirons les prestations attendues de l'équipe de développement, et nous terminerons par les modalités de réalisation du projet.

Table des matières

I	GENERALITES	2
I.1	Contexte	2
I.2	Présentation générale d'EPITECH	2
I.2.1	Rappel sur l'EIP	2
I.2.2	La mission d'EPITECH	2
II	LE GROUPE ULTIMAADE	3
II.1	Le groupe	3
II.2	Partenaires	3
II.3	Relations entre l'équipe projet et EPITECH	3
III	PRESENTATION DU PROJET	4
III.1	La solution à développer	4
III.1.1	Module de dessin	5
III.1.2	Module d'animation	6
III.1.3	Module de gestion de la scène	6
III.1.4	Module de montage	7
III.1.5	Module de gestion d'évènements	8
III.1.6	Module de création de support multimédia/Export	9
III.1.7	Le moteur	10
III.1.8	L'IHM	10
III.2	Contraintes	11
III.2.1	Environnement technique	11
III.2.2	Méthode de développement de la solution	11
IV	PRESTATIONS ATTENDUES	14
IV.1	Approche	14
IV.2	Modalités	15
IV.2.1	Modalités de réalisation et livraison des prestations	15
IV.2.2	Modalités de réception des prestations	16
IV.3	Gantt associé	16

Chapitre I

GENERALITES

I.1 Contexte

Suite à l'évolution rapide des technologies, EPITECH membre du groupe IONIS, a entrepris de former des futurs experts en technologies de l'information.

Dans cette optique l'école a mis sur pied une stratégie consistant pour les étudiants de chaque promotion, en la conception et la réalisation d'un projet innovant, et répondant à une problématique concrète de l'heure. Ce concept s'appelle EIP (Epitech Innovative Project).

Pour les étudiants, l'EIP réalisé représente le projet de fin d'étude, et conditionne le diplôme d'expert décerné par l'école.

C'est dans ce contexte que le groupe ULTIMA ADE s'est réuni autour du projet qui fait l'objet de ce cahier des charges.

I.2 Présentation générale d'EPITECH

I.2.1 Rappel sur l'EIP

Comme mentionné ci-dessus, EIP est un concept EPITECH. Le but est la valorisation des compétences acquises, par la création et la réalisation d'un projet novateur, utile et qui sera utilisé.

I.2.2 La mission d'EPITECH

EPITECH s'est donné pour vocation de produire des experts et développeurs avérés, capables d'apporter des réponses concrètes et efficaces aux problèmes actuels et à venir, qui mélangent considérations matérielles et logicielles.

Les étudiants sont régulièrement appelés à rendre des projets qu'ils réalisent suivant des descriptions spécifiques et précises, et dans les délais impartis.

Chapitre II

LE GROUPE ULTIMA ADE

II.1 Le groupe

ULTIMA ADE est un groupe d'étudiants d'EPITECH de la promotion 2013, qui s'est réuni dans le cadre de l'EIP, autour d'une idée : celle de développer un logiciel libre de création multimédia.

II.2 Partenaires

Pour la réalisation de son projet EIP, le groupe a sollicité et obtenu le partenariat de l'entreprise Ryxeo Sarl, groupe d'experts en logiciels libres basé à Bordeaux.

Le groupe Ryxeo joue un rôle stratégique dans le projet, car le développement du logiciel part d'un projet existant et hébergé chez Ryxeo, dans la série de projets Abulédu à [cette adresse](#).

Nous disposons ainsi d'une forge logicielle hébergée chez Ryxeo Sarl, contenant la plupart des outils nécessaires à la réalisation du projet.

II.3 Relations entre l'équipe projet et EPITECH

Dans le contexte actuel, la proposition de l'idée par le groupe à l'équipe pédagogique de l'école peut être vue comme une réponse à un appel à propositions.

En tant que soumissionnaire à l'appel à propositions, le groupe ULTIMA ADE doit mettre en œuvre les moyens pour assurer l'ensemble des prestations demandées.

Dans un but de compréhension, ces prestations sont décrites par lot dans le présent document.

La direction EIP d'EPITECH sera assignée à la maîtrise d'ouvrage.

De même, le groupe ULTIMA ADE devra désigner une personne chargée du pilotage et du suivi du projet ainsi qu'un assistant.

En cas d'escalade sur incidents, le prestataire(ULTIMA ADE) devra désigner la ou les personnes responsables avec leurs titres respectifs.

Chapitre III

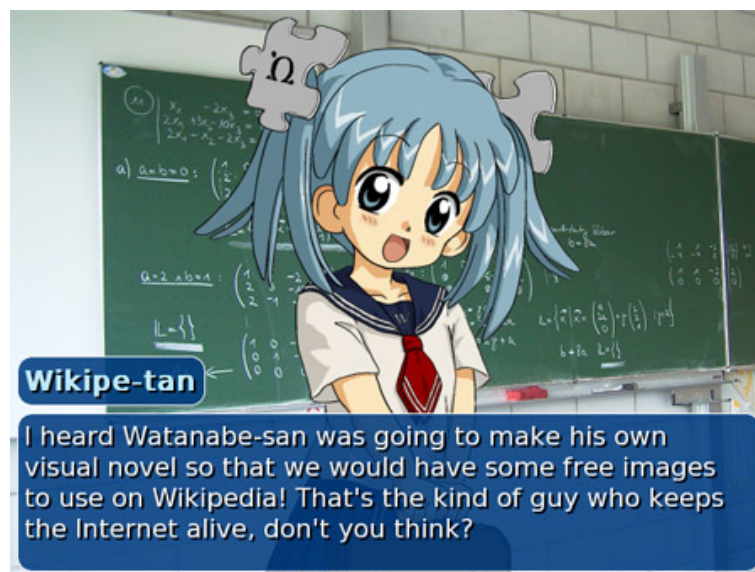
PRESENTATION DU PROJET

ULTIMA ADE propose une solution qui viendra combler un grand manque dans le domaine du logiciel libre : celui de l'utilisation pédagogique du multimédia.

Le logiciel permettra à des enseignants de créer des applications pédagogiques selon leurs domaines d'expertise.

Il permettra également à un utilisateur technicien ou non, de créer une application multimédia autonome, dans un environnement ouvert et libre.

Enfin les enfants pourront éditer des histoires dont ils sont eux même des héros.



L'objectif de l'outil à développer est d'avoir une perception orientée « usage » et non « technique ».

La technique étant au service du monde éducatif et non l'inverse.

Par ailleurs l'application aura pour principale caractéristique d'être intuitive et facile d'utilisation.

III.1 La solution à développer

Le logiciel Ultimaade doit être open source et multi plate-forme.

Il disposera des modules suivants, qui seront conçus comme des éditeurs/lecteurs pour la phase de développement :

- **Dessin**, qui permettra aux utilisateurs désirant le faire directement dans le logiciel, de dessiner rapidement leurs objets
- **Animation**, grâce auquel les objets dessinés ou importés depuis un autre système seront animés
- **Gestion de la scène**, permettant de préparer la scène pour le rendu des animations
- **Montages/Interactions**, qui permettra en enchainant différentes scènes, de créer des films/clips
- **Gestion d'évènements**, permettant à l'utilisateur de définir et paramétrer les interactions dans son application
- **Création de support multimédia/Export**, permettant à l'utilisateur de créer un CD-ROM, exporter un projet sur le web
- **Ordonnancement**, module de gestion de l'ensemble du système. C'est le moteur du programme, constituant sa base de fonctionnement
- **Lecture multimédia(player)**, module permettant de lire et de contrôler la lecture d'un projet ultimaadee.
Ce module sera livré comme plugin (bibliothèque), et comme application à part entière (standalone).
Ce qui permettra aux utilisateurs de lire rapidement un projet conçu avec ultimaade sans installation complète du logiciel.

III.1.1 Module de dessin

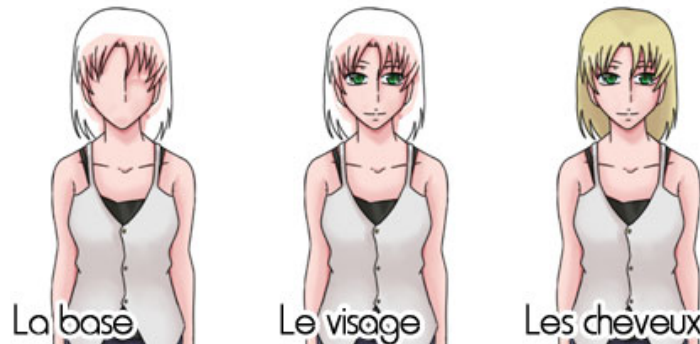
Ce module implémentera un éditeur d'objets (sprites), et permettra :

- l'insertion d'objets de départ



- les objets peuvent avoir été créés dans un logiciel de dessin ou directement sous Ultimaade.
Comme mentionné dans le document d'étude technique détaillée du projet, la solution proposera une bibliothèque d'objets(sprites) de base, utilisables dans ce module.
- objets graphiques. Exemple : .bmp, .gif, .jpg, .png....

- objets vectoriels. Exemple : .eps, .ps.....
- la création et modification des sprites



- créer des dessins, éditer des objets(sprites), et y appliquer des calques
- changer l'apparence de l'objet (forme, couleur de fond, contour, etc)
- modifier les caractéristiques de l'objet/sprite telle que la taille(longueur, largeur), la forme, etc

III.1.2 Module d'animation

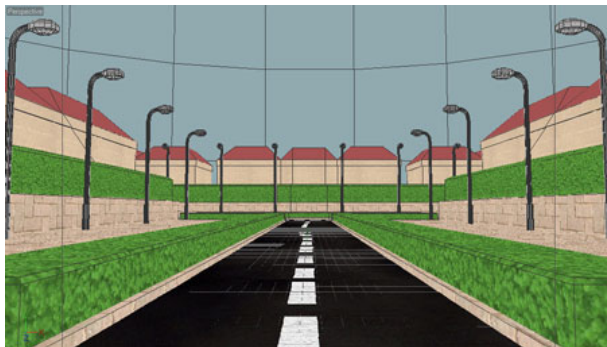
Il est question dans ce module de développer pour l'utilisateur, des commandes menu simples.

Ces commandes lui permettront de créer des interfaces expressives, et du contenu interactif sans aucune programmation. L'utilisateur pourra :

- organiser une suite d'image pour en faire une animation
- appliquer des transformations telles que la transparence, la rotation sur l'objet tout au long de son animation
- ajuster le temps de transition entre chaque image
- choisir le type de transition entre chaque image (fondu, direct, ...)
- exporter l'animation vers un autre format (exmple : gif)

III.1.3 Module de gestion de la scène

Ce module fera abstraction d'un éditeur de « monde » (on parlera de plateau ou scène), dans lequel les animations et interactions seront visibles.



Il permettra à l'utilisateur de :

- insérer des objets dans un décor(background)
- créer un point de départ pour chaque objet utilisé dans son projet
- définir les zones sensibles du plateau et de créer le décor pour un film
- définir et paramétrer la vue qu'il souhaite de son monde, dans le cas d'une application 3D (fonctionnalité optionnelle pour les premières versions du logiciel)
- transformer un objet (Objet bouton, objet graphique,...) et lui permettre de choisir son sens d'orientation dans la scène

III.1.4 Module de montage

L'utilisateur pourra grâce à ce module produire un clip en combinant plusieurs scènes qui s'enchaînent en temps réel.

L'accent sera mis sur l'aspect temps réel dans le développement de ce module, afin de garantir au développeur la possibilité de travailler en toute confiance.

Ceci permettra de produire des contenus fiables pour les utilisateurs de l'application finale.

Le film pourra être également réalisé à partir des vidéos importées et introduites dans la scène.

La solution devra donc supporter les formats de vidéo les plus connus.

L'utilisateur pourra grâce à ce module :

- importer des vidéos et audios
- ajouter une frise chronologique(TimeLine) à la vidéo et l'audio dans le montage
- arrêter/suspendre une vidéo ou un son en cours d'exécution
- créer des effets visuels (par exemple la rotation d'un texte)



FIGURE III.1 – Scène 1

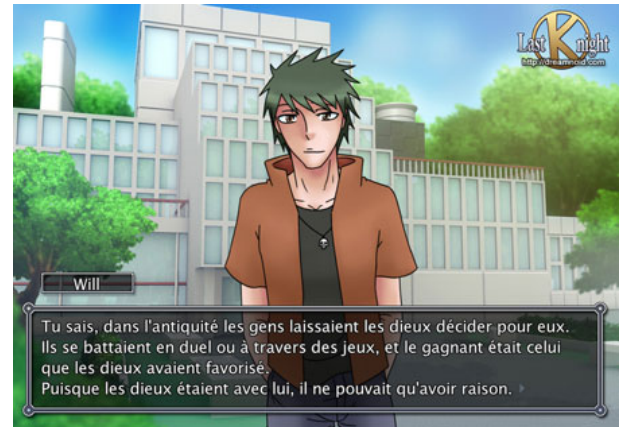


FIGURE III.2 – Scène 2.

- créer des effets de transition entre différentes vidéos, ainsi que des effets plus spéciaux comme rendre la vidéo miroir
- exporter son film sous différents formats (par exemple : .avi, .mkv)

III.1.5 Module de gestion d'évènements

Ce module implémentera un éditeur d'évènements + frise chronologique (Timeline), que l'on pourra définir dans un menu prévu à cet effet.

Il permettra à l'utilisateur de :

- paramétrer la frise chronologique pour choisir sa durée, sa vitesse (fps = image par seconde)
- créer et modifier des objets
- définir des « zones sensibles » sur lesquelles l'objet peut réagir

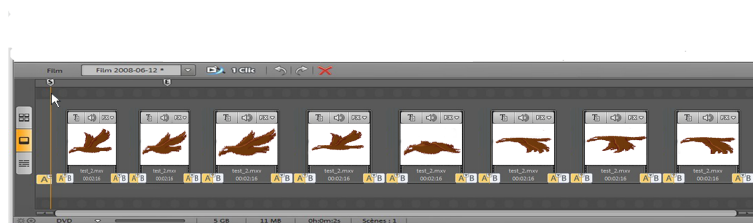


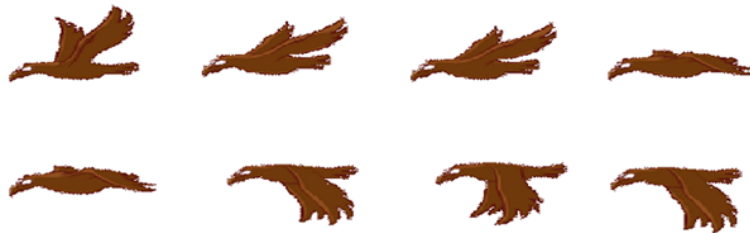
FIGURE III.3 – Timeline

- utiliser la frise de temps (Timeline) pour créer une scène différente pour chaque image si l'utilisateur le souhaite
- créer des calques sur la frise chronologique

- créer un évènement qui sera déclenché dès que le programme arrive à un instant T, comme créer un mouvement d'un point A à un point B, une transformation d'objet (exemple : passer d'un carré à un rond).
L'utilisateur pourra se déplacer sur la frise chronologique et la modifier dans l'interface utilisateur du logiciel
- définir la manière dont seront gérés les évènements enclenchés lors de l'utilisation de son application
- faire évoluer une histoire ou un film dans le temps.

Exemple : Pour un oiseau dont toutes les images composant son mouvement ont été créées par un logiciel de dessin, on pourra par exemple :

- importer huit images PNG correspondant à ses différents états,
- ajouter différents bruits et diverses autres caractéristiques,
- puis animer les objets(sprites) dans la scène.



Pour un objet/personnage se déplaçant d'un point A à un point B, on devra pouvoir enchaîner les objets(sprites) un par un, pour donner un effet de déplacement temps réel : ajout d'un bruit + diverses autres caractéristiques.

III.1.6 Module de création de support multimédia/Export

C'est ce module qui définit le côté multimédia de la solution à mettre en œuvre. L'utilisateur pourra :

- intégrer des vidéos, des effets sonores, et des contenus dynamiques et des illustrations statiques.
On doit pouvoir également ajouter des interactions permettant de contrôler la lecture des vidéos et illustrations. Cela se fera via le mode player de l'IHM Ultimaade
- obtenir rapidement le résultat souhaité à l'aide de fonctions et outils de mise en forme intégrés à la solution
- définir et pré visualiser l'aspect et le comportement des données dynamiques, sans se connecter à une base de données

- tester une scène du projet ou le projet entier grâce à un sous-module « test » à intégrer au présent module
- produire un CD-ROM à partir d'une application précédemment conçue
- exporter un projet sur le web sous un format de fichier exécutable en ligne dans n'importe quel navigateur, conforme aux normes de la W3C.
L'exportation ne devra pas requérir de plugin particulier sur le système hôte. Le fichier exporté devra également être exécutable en dehors d'un navigateur.
Sur la base de ces prérequis un bon choix pour le format de fichier d'export web serait le HTML5, qui se définit aujourd'hui comme le format de référence.

III.1.7 Le moteur

C'est dans ce module que seront définies et implémentées toutes les fonctions permettant à un utilisateur du logiciel, de communiquer avec son ordinateur.

Il implémentera donc tous les moyens et outils nécessaires à cette fin.

Ces outils devront avoir été définis dans une interface spécialement programmée pour la solution.

Il s'agit de l'Interface Homme - Machine(IHM) du logiciel, et les fonctions définies dans cette dernière, le seront sur la base des fonctionnalités acquittées par chacun des modules précédemment cités.

Ainsi tous les modules développés pour le logiciel seront utilisés par le moteur sans dépendre de ce dernier.

Le moteur quant à lui implémentera une interface dans laquelle seront définies toutes les relations entre lui et les autres modules.

III.1.8 L'IHM

Elle devra être déconnectée de l'implémentation réelle des mécanismes contrôlés, et devra couvrir les paradigmes suivants :

- de la métaphore, permettant de mimer le comportement de l'interface sur celui d'un objet de la vie courante que l'utilisateur maîtrise, tel que le clavier ou la souris
- idiomatique, permettant de définir des comportements stéréotypés sur l'interface, pas nécessairement calqués sur les objets de la vie réelle.
Ces comportements seront ainsi cohérents et simples à apprendre pour l'utilisateur. Ce qui permettra aux utilisateurs n'ayant pas la maîtrise de l'outil informatique, une prise en main rapide du logiciel.

L'IHM devra définir un environnement graphique permettant aux utilisateurs d'accéder et d'utiliser les fonctionnalités du logiciel, à partir d'un écran et de visualiser la réalisation de leurs projets.

Une option dans ce modèle sera de permettre la personnalisation de l'interface par l'utilisateur, grâce à une fonctionnalité de prototypage d'interface utilisateur.

Elle devra également permettre à l'utilisateur de démarrer l'application en deux mode :

- editor : dans ce mode le développeur/utilisateur peut créer un nouveau projet ou travailler sur un projet existant.
C'est le mode par défaut du programme
- player : ici l'utilisateur peut lire une application multimédia créée avec Ultimaade ou une autre solution.
Pour une meilleure flexibilité, ces deux modes devront être accessibles à partir d'un même menu.

III.2 Contraintes

III.2.1 Environnement technique

La solution doit fonctionner sur une machine disposant de :

- 1 ou 2 processeurs dual core,
- 2GB de mémoire vive devraient suffire pour un fonctionnement optimal du logiciel
Si une capacité mémoire supérieure à cette limite est requise, ceci devra faire l'objet d'une justification par l'équipe de développement
- 500Mo d'espace disque dur devraient suffire pour l'installation de l'application et des fichiers de configuration.
Ce paramètre peut être revu au cours de l'évolution du projet
- la solution doit être portable, sous licence GPL, et fonctionner sur les systèmes Linux/UNIX, Windows et Mac.
- un gestionnaire de version sera déployé pour les développeurs et le partage du code.

III.2.2 Méthode de développement de la solution

Le développement de la solution devra se faire de manière itérative et chaque itération à partir de la deuxième, devra inclure tous les modules.

Ci-dessous une proposition d'itérations :

- production d'une maquette IHM pour l'utilisateur final de la solution
- mise en place d'un MCD (Modèle Conceptuel de Données)
- conception de la solution :
 - définition des données (toutes les informations nécessaires au développement des modules de la solution)
 - modélisation des traitements (un ou plusieurs diagrammes de séquences)

- modélisation des objets nécessaires à la réalisation du projet (un ou plusieurs diagrammes de classes)
- modélisation du processus interactif de l'ensemble du système (un ou plusieurs diagrammes d'activités)
- description détaillées des données nécessaires à la réalisation du projet

- définition et documentation d'une API pour les plugins du logiciel
- mise en œuvre des fonctionnalités IHM basiques (interface principale) : analyse, conception, développement, tests unitaires
- mise en œuvre du cœur/moteur, et implémentation des interactions avec l'interface graphique : analyse, conception, développement, tests unitaires
- production et développement d'une IHM pour le module de gestion scènes : analyse, conception, développement, tests unitaires
- production et développement d'une IHM pour le module de montage : analyse, conception, développement, tests unitaires
- mise en œuvre des fonctionnalités de gestion de la scène (Module de gestion de la scène) : analyse, conception, développement, tests unitaires
- mise en œuvre des fonctionnalités de montage (module de montage) : analyse, conception, développement, tests unitaires
- intégration des frises de temps (timeline) et compteurs : analyse, conception, développement, tests unitaires
- développement d'un premier lecteur ultimaade (player) basique, capable d'enchaîner les scènes d'un montage de la première à la dernière, en respectant les frises chronologiques associées, et de lire du texte simple : analyse, conception, développement, tests unitaires
- production du premier prototype du logiciel (première version release) incluant l'enchaînement des scènes statiques : premiers assemblage, intégration et mise en production (Mise en Prod)
- développement du module de gestion d'animation avec l'IHM associée : analyse, conception, développement, tests unitaires
- développement du module de gestion d'événements avec l'IHM associée : analyse, conception, développement, tests unitaires
- développements supplémentaires sur le player (gestions des compteurs, des différents formats audios et vidéos, et d'autres types de média en vigueur) : analyse, conception, développement, tests unitaires
- production du deuxième prototype du logiciel (deuxième version release) incluant

les nouveaux développements :

assemblage, intégration et mise en production(Mise en Prod)

- mise en œuvre d'un environnement permettant d'effectuer des premiers dessins, d'importer des objets(sprite) et d'effectuer des actions basiques : analyse, conception, développement, tests unitaires
- développement du module de dessin complet, avec l'IHM associée : analyse, conception, développement, tests unitaires
- production du troisième prototype du logiciel(troisième version release) incluant les nouveaux développements : assemblage, intégration et mise en production(Mise en Prod)
- mise en œuvre du module de création de support multimédia : analyse, conception, développement, tests unitaires
- mise en œuvre du module d'exportation (export web principalement) : analyse, conception, développement, tests unitaires
- intégration du module de création de support multimédia au moteur : analyse, conception, développement, tests unitaires
- production du quatrième prototype du logiciel (première version stable) incluant les nouveaux développements : assemblage, intégration et mise en production (Mise en Prod)
- tests d'usines globaux et de vérification d'aptitude au bon fonctionnement (VABF)
- mise en œuvre des fonctionnalités de gestion de la scène en 3D : analyse, conception, développement et tests unitaires

Chapitre IV

PRESTATIONS ATTENDUES

IV.1 Approche

Le projet Ultimaade est découpé en cinq phases ou parties (Cf voir schéma cycle de développement ci-dessous).

- PHASE N°1 : gestion des scènes statiques + montages + première version player
- PHASE N°2 : gestion des animations + évènements + deuxième version player
- PHASE N°3 : gestion des dessins
- PHASE N°4 : création de support multimédia et exportation
- PHASE N°5 : gestion de la 3D

Chaque phase sera réalisée suivant un planning détaillée et débouchera sur une version du logiciel.

La mise en oeuvre de chaque phase(version) s'appuiera sur un découpage en lots suivant une approche fonctionnelle et un cycle homogène, comme le montre la figure ci-dessous :

- LOT 1 : MODELISATION
- LOT 2 : DEVELOPPEMENT
- LOT 3 : HOMOLOGATION
- LOT 4 : EXPLOITATION

La maîtrise d'ouvrage est réalisée par le la direction EIP EPITECH.

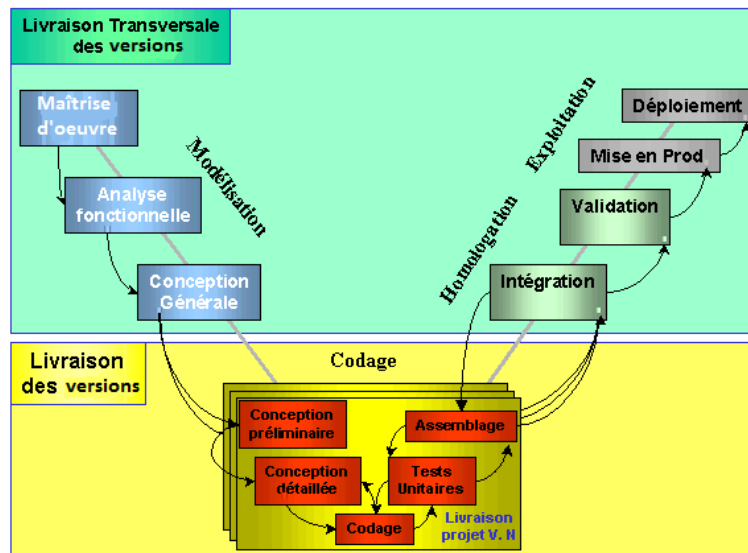


FIGURE IV.1 – Cycle de développement du projet

Les flèches entre les différentes étapes de ce cycle de développement illustrent les opérations de promotion/rétrogradation au sein de la même version du logiciel, d'un ensemble de livrables : code, spécifications textuelles, ressources diverses - images, fichiers de configuration, etc.

IV.2 Modalités

IV.2.1 Modalités de réalisation et livraison des prestations

La livraison et la mise en œuvre des prestations pour la première version du logiciel (PHASE N°1) se fera suivant le planning ci-dessous :

- Analyse fonctionnelle détaillée avant le 31/01/2012 ;
- Spécifications avant le 04/02/2012 ;
- Conception architecturale avant le 31/03/2012 ;
- Conception détaillée avant le 07/04/2012 ;
- Développement premier prototype avant le 04/07/2012 ;
- Tests unitaires avant le 07/08/2012 ;
- Tests d'intégration avant le 21/08/2012 ;
- Tests de validation avant le 31/08/2012 ;
- Formation et support avant le 05/09/2012 ;
- Recette le 15/09/2012.

En tout état de cause, la vérification d'aptitude au bon fonctionnement pour la première version du logiciel doit être prononcée avant le 15/09/2012.

IV.2.2 Modalités de réception des prestations

La réception des produits et services, sujets du présent projet fera l'objet d'une VABF (vérification d'aptitude au bon fonctionnement), et d'une VSR, (vérification de service régulier), par la maîtrise d'ouvrage.

La VABF est destinée à vérifier la conformité de l'ensemble des produits et services fournis par les développeurs, conformément aux stipulations du cahier des charges.

Dans l'hypothèse où la procédure de VABF donnerait lieu à d'éventuelles réserves, l'équipe projet devra procéder aux corrections requises et à la livraison d'une nouvelle version des fournitures. Ceci dans des délais compatibles avec le calendrier.

IV.3 Gantt associé

L'équipe projet établira un planning d'exécution des différentes phases du projet à remettre à la maîtrise d'ouvrage.

Une proposition de planning des réalisations pour la première version du logiciel est consultable sur [ce lien](#).