

RAPPORT TECHNIQUE  
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
DANS LE CADRE DU COURS  
LOG792: PROJET DE FIN D'ÉTUDES EN GÉNIE LOGICIEL

**BRIGANDINE REBIRTH**

FRANÇOIS TAILLEFER  
TAIF22118406

DÉPARTEMENT DE GÉNIE LOGICIEL ET DES TI

**Professeur superviseur**

**Alain April**

MONTRÉAL, 20 AVRIL 2010  
HIVER 2010



## REMERCIEMENTS

- Hearty Robin et Atlus pour avoir développé et publié le jeu original.
- L'administrateur de <http://www.forsena.org/>, de son surnom *Dryst*, pour avoir créé la communauté où s'est déroulée la conversation qui est à l'origine de l'inspiration pour le projet.
- L'ÉTS et Alain April pour l'organisation et le support fourni dans ce cours.
- Mes coéquipiers dans d'autres cours de la session (Faviann Di Tullio, Mathieu Fleurant-Doré et Jonathan Tremblay) pour leur compréhension face à mon horaire chargé.
- Philippe Breault pour ses conseils par rapport au démarrage et à l'orientation du projet.

## **BRIGANDINE REBIRTH – DE L'IDÉE À LA CONCEPTION**

**FRANÇOIS TAILLEFER  
TAIF22118406**

### **RÉSUMÉ**

L'idée de ce projet est de se baser sur le jeu de Playstation Brigandine, publié en 1998, et de permettre de reprendre le principe du jeu et de l'appliquer à d'autres histoires et événements.

Le projet se sépare en deux applications, une qui permet de construire de telles campagnes et une autre qui permet de charger ces campagnes et jouer.

Le sous-projet lié directement au présent cours et au présent rapport concerne une première itération pour le projet, au cours de laquelle le but premier a été de monter la base du projet. Au moment où ce cours s'achève, le prochain est rendu à l'étape de conception.

## TABLE DES MATIÈRES

	Page
Introduction.....	VI
CHAPITRE 1 : Analyse des besoins – Constructeur de campagne.....	7
Objectifs.....	7
Décisions.....	7
Méthode.....	8
Réalisations.....	9
CHAPITRE 2 : Analyse des besoins – application de jeu.....	10
Objectifs.....	10
Décisions.....	10
Méthode.....	10
Réalisations.....	11
CHAPITRE 3 : Élaboration des exigences – constructeur de campagne.....	12
Objectifs.....	12
Décisions.....	12
Méthode.....	13
Réalisations.....	14
CHAPITRE 4 : Élaboration des exigences – application de jeu.....	14
Objectifs.....	14
Décisions.....	14
Méthode.....	14
Réalisations.....	15
CHAPITRE 5 : Conception – constructeur de campagne.....	16
Objectifs.....	16
Décisions.....	16
Méthode.....	16
Réalisations.....	17
Conclusion.....	18
Recommandations.....	19
Liste de références.....	20
Bibliographie.....	21
Annexe I: Document de vision.....	22
Annexe II: Document SRS.....	23
Annexe III: Document de conception.....	24



## **LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES**

SRS: Software Requirements Specification: Modèle de document proposé par RUP qui présente les exigences d'un logiciel.

RUP: Ratified Unified Process: Processus de développement logiciel itératif et incrémental créé par une division de IBM.

## Introduction

Par sa nature, l'élaboration de ce projet se sépare par deux dimensions différentes. Il y a tout d'abord la dimension classique, une séparation selon la tâche accomplie. Dans le cadre de ce projet, cette dimension mène à trois catégories: l'analyse des besoins, l'élaboration des exigences et la conception.

Mais il y a également la deuxième dimension de séparation: l'application elle-même. En effet, comme le projet se sépare en deux applications dont les buts et utilisateurs diffèrent assez fortement, le résultat net est que l'approche pour accomplir une même tâche peut différer de manière significative.

Ce rapport est donc séparé par tâche et par application, même si les artefacts produits avant la conception regroupent le résultat obtenu pour les deux applications.



## **CHAPITRE 1: ANALYSE DES BESOINS – CONSTRUCTEUR DE CAMPAGNE**

### **Objectifs**

Les buts pour cette étape se séparent en deux. Tout d'abord, il fallait faire une analyse préliminaire des données utilisées par le jeu et déterminer lesquelles devraient être modifiables. Ensuite, il fallait déterminer les autres caractéristiques importantes de l'application.

### **Décisions**

Il a été décidé de représenter l'analyse des besoins à l'aide du document de vision tel que défini par RUP. Ce format a été choisi à cause de sa simplicité de compréhension, étant donné que son but est de permettre à tous les intervenants d'avoir un aperçu du projet. Ceci permettra d'expliquer plus facilement le projet à de nouveaux collaborateurs. Étant donnée l'ampleur du projet, ceci est primordial à moyen et long terme.

Les caractéristiques relatives à ce chapitre sont les caractéristiques C1-01 à C1-25.

Une autre décision majeure a été prise avant de se lancer, au sujet des graphiques. Puisque le développement d'un engin graphique n'apporterait probablement pas beaucoup de valeur ajoutée face à l'intégration d'un engin existant, il a été décidé de ne pas s'attarder aux graphiques à ce moment-ci. L'hypothèse a été émise qu'un engin acceptable pourrait être trouvé et intégré.

De plus, il a été établi que les graphiques se devaient d'être rudimentaires. La raison principale est que si on veut permettre un bon degré de liberté aux développeurs de campagne, alors ils doivent fournir la représentation graphique de ce qu'ils créent. L'idée d'un bel engin graphique 3d qui permet de présenter les animations d'habiletés, comme dans le jeu original, paraît bien jusqu'à ce qu'on réalise que ceci demande simplement trop de travail de la part de l'utilisateur de l'application et risque de ne jamais servir. Si Brigandine avait eu la

même base d'utilisateurs que, disons, Final Fantasy VII, alors la décision aurait pu être différente.

## **Méthode**

Pour pouvoir analyser les besoins correctement, il était primordial d'avoir une certaine compréhension des utilisateurs potentiels. Pour ce faire, les membres de forsena.org ont été sondés, dans la discussion suivante: <http://forsena.org/brigboard2/viewtopic.php?f=7&t=963>.

Comme le créateur du projet fréquente cette communauté depuis quelque temps, les utilisateurs potentiels ont aussi été observés par les autres discussions présentes, même si celles-ci n'étaient pas ciblées directement vers le projet.

Un point majeur qui est ressorti est que les utilisateurs n'ont pas tous une bonne maîtrise de l'anglais

Les principales autres observations qui ont été faites sont les suivantes:

- Il y a un intérêt marqué pour la création de nouveaux monstres.
- La capacité d'étendre l'histoire a une grande importance
- Les utilisateurs potentiels n'ont pas nécessairement de connaissances en programmation.

L'étape suivante était d'identifier grossièrement les éléments qui devaient être modifiables. Des heures d'expérience à expérimenter avec le jeu ont contribué à simplifier cette tâche. L'approche utilisée a été de noter les actions entreprises par un joueur au cours du jeu et de prendre en note tous les éléments impliqués dans l'action ou encore la prise de décision. Les caractéristiques C1-01 à C1-20 ainsi que C1-24 et C1-25 proviennent de cette réflexion.

D'autres caractéristiques auraient pu être créées, mais il y avait l'obligation de tracer une ligne quelque part. En clair, il fallait déterminer jusqu'à quel niveau on devait rendre une campagne modifiable. La décision prise a été de conserver la mécanique de combat. Par mécanique de combat, on entend:

- Les attributs et leur contribution lors des combats
- Les formules pour calculer le dégât ou la chance de base de frapper. Ceci inclut les coefficients des interactions élémentales.
- La limitation de 3 Knights par combat par nation
- La limitation de 6 monstres par Knight

Cette décision a été prise dans le but de conserver l'esprit de Brigandine. Même s'il est vrai que certaines de ces décisions peuvent potentiellement simplifier la conception, il est possible d'imaginer comment ces contraintes auraient pu être éliminées s'il avait été jugé préférable de le faire.

Les caractéristiques C1-21 à C1-23 proviennent d'une analyse de comment les campagnes devraient être gérées. En observant la quantité de caractéristiques et l'importance des interdépendances, il est devenu évident qu'il deviendrait vite impossible pour un utilisateur, surtout un n'ayant pas de connaissances en conception et programmation, d'apporter des modifications à une campagne sans briser des liens entre les éléments. C'est pourquoi les caractéristiques de validation et d'exportation ont vu le jour.

Finalement, puisque le document de vision doit aider à expliquer le projet, une grande quantité d'efforts a été mise dans l'élaboration de définitions des termes relatifs à Brigandine.

## **Réalisations**

Tel qu'indiqué précédemment, le résultat obtenu par cette étape est consolidé dans le document de vision. Les caractéristiques pertinentes sont celles associées à l'application de personnalisation, qui est un autre nom pour le constructeur de campagne. Ce dernier est présenté à l'annexe I.

## **CHAPITRE 2: ANALYSE DES BESOINS – APPLICATION DE JEU**

### **Objectifs**

Au contraire du chapitre 1, cette partie de l'analyse des besoins comportait moins de variables et d'impondérables, et avait une portée beaucoup plus rigide définie. L'objectif était d'obtenir une représentation de tout ce que l'application de jeu devrait faire – donc, tout ce que Brigandine fait déjà.

### **Décisions**

Suite à une décision prise précédemment, une décision consciente d'exclure une caractéristique du jeu a été prise lors de cette étape. Cette caractéristique est que dans Brigandine, il est possible de faire jouer une animation 3d montrant une unité utilisant une habileté contre une autre, ainsi que la contre-attaque de cette unité (lorsqu'applicable). Le même principe est possible lors de l'invocation d'un monstre. Pour les raisons décrites à l'étape 1, ceci n'était pas considéré envisageable pour les utilisateurs.

### **Méthode**

Pour éviter les oublis, l'analyse a été faite au moyen de deux approches différentes:

1. Passer à travers chacune des caractéristiques déterminées pour le constructeur de campagne et identifier les scénarios dans lesquels l'élément visé par la caractéristique jouerait un rôle.
2. Passer à travers chacun des menus et chacune des situations rencontrées dans le jeu et identifier tout ce que le jeu fait ou permet à l'utilisateur de faire.

## **Réalisations**

Le résultat obtenu est une liste de 26 caractéristiques, C2-01 à C2-26. Comme à l'étape précédente, les résultats obtenus ont été compilés dans le document de vision, présenté à l'annexe I.

## **CHAPITRE 3: ÉLABORATION DES EXIGENCES – CONSTRUCTEUR DE CAMPAGNE**

### **Objectifs**

L'objectif de cette étape était d'obtenir les exigences fonctionnelles à partir des caractéristiques du constructeur de campagne. À un niveau de priorité un peu moins élevé pour l'instant, un autre but était d'obtenir des exigences non fonctionnelles.

### **Décisions**

Le choix a été fait de représenter l'élaboration des exigences avec le modèle SRS de RUP. Ce modèle se marie bien avec le document de vision. De plus, ce document est produit plusieurs fois au courant des cours du baccalauréat et il était important d'éviter les risques par rapport à la documentation, considérant les autres risques du projet (principalement, sa taille).

Le document SRS est bâti avec un modèle de qualité FNFC en tête, mais il aurait été possible de l'adapter à d'autres modèles comme ISO 9126 ou le modèle de McCall. La décision de conserver le modèle FNFC (Fonctionnel, Non Fonctionnel, Contraintes) provient surtout de la réalité que les autres modèles sont souvent plus adaptés à un logiciel commercial ou développé en entreprise. Le modèle FNFC est simple, facile à comprendre, et applicable dans une multitude d'environnements de développement.

Il a été décidé d'apporter une déviation au modèle SRS standard. Celui-ci propose l'emploi de cas d'utilisation. Après analyse et réflexion, il a été déterminé que l'élaboration de cas d'utilisation serait un mauvais emploi du temps considérant que les caractéristiques et exigences du constructeur de campagne reviennent essentiellement à assigner des valeurs et éléments à d'autres éléments. Pour l'application de jeu, il est encore plus évident que tous les cas d'utilisation sont déjà représentés dans le jeu original, et les observer dans le jeu offre une représentation beaucoup plus claire qu'une description papier.

Pour l'élaboration des exigences, il a été décidé de rencontrer les possibilités non seulement du Brigandine original (Legend of Forsena) mais également les possibilités ajoutées par Brigandine: Grand Edition, la version retravaillée publiée en 2000 au Japon seulement.

## **Méthode**

Une pratique intéressante appliquée pour cerner toutes les exigences fonctionnelles d'une caractéristique a été d'identifier toute relation avec d'autres caractéristiques, et d'identifier les règles qui bornent cette relation. Néanmoins, il a fallu toutefois observer aussi la caractéristique par elle-même et identifier tous les attributs et comportements qui la définissent. Le résultat est environ une centaine d'exigences fonctionnelles.

Une autre manière d'élaborer les exigences a été de retourner voir la discussion sur [forsena.org](http://forsena.org), pour identifier les nouvelles possibilités ajoutées par Brigandine: Grand Edition. Plusieurs membres de cette communauté ont joué à la version japonaise du jeu, traduite en partie par un projet indépendant.

Il va de soi que le retour en arrière a été fréquemment utilisé puisqu'il est pratiquement impossible de tout cerner du premier coup; ceci explique pourquoi les exigences fonctionnelles n'ont pas toujours des valeurs qui se suivent; celles qui sont en désordre ont été ajoutées ou déplacées plus tard.

Quelques exigences non fonctionnelles ont été développées. Les plus importantes sont au point de vue de l'utilisabilité et de la maintenabilité.

## **Réalisations**

Les exigences identifiées et décrites sont consolidées dans le document SRS. Comme le document de vision, celui-ci contient également les exigences pour l'application de jeu. Le document SRS est présent à l'annexe II.

## **CHAPITRE 4: ÉLABORATION DES EXIGENCES – APPLICATION DE JEU**

### **Objectifs**

Le but de cette étape est d'ajouter des détails et précisions aux caractéristiques identifiées à l'étape détaillée dans le chapitre 2.

### **Décisions**

Les décisions prises pour cette étape sont des décisions dont la portée était les deux applications; ainsi, elles sont décrites au chapitre précédent.

### **Méthode**

La méthode employée a été d'utiliser le jeu et, pour chaque caractéristique, noter les détails pertinents relatifs à l'emploi de cette caractéristique. À cause de la nature des caractéristiques de l'application de jeu, le nombre d'exigences fonctionnelles dégagées par caractéristique a été beaucoup moins élevé que pour le constructeur de campagne.

Une attention un peu plus grande a été portée sur les exigences de performance. En effet, ces exigences portent sur des actions qui se répètent beaucoup plus souvent que celles du constructeur de campagne; les exigences non fonctionnelles de cette catégorie sont donc les plus importantes pour l'application de jeu.



## **Réalisations**

Comme pour le constructeur de campagne, les exigences fonctionnelles et non fonctionnelles identifiées sont consolidées dans le document SRS, présenté à l'annexe II.

## **CHAPITRE 5: CONCEPTION – CONSTRUCTEUR DE CAMPAGNE**

### **Objectifs**

Le but de cette étape est de représenter les éléments du "dictionnaire de données" par une conception orientée objet. Par dictionnaire de données, on entend le recueil de données et variables qui sont personnalisées par le développeur de campagne et utilisées par l'application de jeu.

### **Décisions**

Les diagrammes de classes sont représentés par la spécification UML. Il a été choisi parce qu'il est le langage de modélisation utilisé *de facto* à l'ÉTS et n'empêchait pas de représenter les éléments de la conception.

Pour éviter d'ajouter du bruit inutile, il a été décidé que plusieurs méthodes (particulièrement les getters et setters sur des attributs) seraient omises des diagrammes.

Il a aussi été décidé de faire un certain effort pour aligner la conception vers l'utilisation des classes par l'application de combat, mais la conception de cette application reste encore à faire. Il paraissait ridicule d'employer des classes différentes pour les deux applications puisque les mêmes concepts sont représentés dans chacune, donc les particularités qui ne servent pas pour le constructeur de campagne n'y seront simplement pas utilisées plutôt que d'être exclues.

### **Méthode**

Un diagramme de dépendances a été développé très tôt dans le processus. Celui-ci a servi à identifier par où commencer. Les éléments n'ayant aucune dépendance ont été identifiés en premier mais jugés trop petits pour être représentés seuls. Ils ont donc été considérés comme

"traités". La conception a ensuite pu commencer, en imposant comme règle qu'on ne pouvait traiter un élément que s'il dépendait seulement d'éléments traités. Ceci a permis de commencer au bas du diagramme de dépendances et de monter graduellement jusqu'à avoir la fait conception pour l'entier du dictionnaire de données.

Pour assurer la traçabilité et pour éviter les oublis, une explication a été ajoutée pour chaque exigence fonctionnelle pour indiquer où elle était traitée. Les points plus particuliers et donc moins instinctifs ont aussi été l'objet d'une explication ajoutée après le diagramme, dans le but que le document de conception (comme la documentation du projet en elle-même) soit clair et compréhensible par lui-même.

### **Réalisations**

Le résultat obtenu par ces efforts est le document de conception, qui contient 12 diagrammes de classes. Ce dernier est présenté à l'annexe III.

## Conclusion

Par rapport aux prévisions initiales, qui planifiaient qu'au moment présent la conception serait terminée et l'élaboration de l'interface utilisateur serait en cours, il est clair que le travail s'est fait à un rythme moins élevé qu'attendu. Par contre, le plus important objectif de cette itération était que le développement puisse commencer lorsqu'elle se terminerait. Bien qu'il reste de la conception à faire, l'objectif est néanmoins atteint puisque le développement pour le constructeur de campagne peut certainement s'amorcer dès maintenant.

Le point le plus important à retirer de cette itération est que la partie la moins intéressante selon plusieurs développeurs est maintenant faite et le projet est maintenant libre de prendre son envol. Le travail qui reste à faire est néanmoins monumental et il faudra maintenant décider comment procéder.

## **Recommandations**

Même si le développement peut probablement commencer, je recommanderais de se pencher sur la conception de l'application de jeu tout d'abord. Les décisions prises lors de cette tâche risquent d'avoir un impact sur la conception du constructeur de campagne et risqueraient d'entraîner une certaine reprise du travail de développement, si déjà entamé.

Avant d'aller trop loin dans le projet, il faudrait aussi se pencher sur le problème des algorithmes de prise de décision. Les algorithmes pour les tours Organize et Attack ne devraient pas poser un problème trop grand, mais l'algorithme de prise de décision par l'ordinateur lors d'un combat est de loin le plus grand risque dans le projet à ce jour. Cet algorithme devra déplacer jusqu'à une vingtaine d'unités, choisir pour chacune parmi souvent une vingtaines de hex accessibles et une dizaine d'habiletés, tout en ayant une contrainte de temps très serrée. De plus, il sera virtuellement impossible de jouer sans celle-ci, à moins d'implémenter un mode multi-joueurs et d'avoir autant de joueurs que de nations, ce qui est loin d'être pratique et ne justifie pas tout l'effort à mettre dans le projet.

## Liste de références

*Hearty Robin. Brigandine: Legend of Forsena: Atlas; 1998*

## **Bibliographie**

*Leffingwell, D. et Widrig, D. (2008). Managing Software Requirements (2<sup>nd</sup> edition): Addison-Wesley*

*Larman, C. (2005). Applying UML and Patterns (3<sup>rd</sup> edition): Prentice Hall*

## **ANNEXE I: DOCUMENT DE VISION**

Le document de vision présente les caractéristiques et besoins associés au projet Brigandine Rebirth, ce qui englobe les deux applications. Il inclut également une vue d'ensemble du projet, ainsi que sa portée et ses buts. Le document de vision est contenu dans le fichier Ftaillefer\_LOG792\_DocumentVision.doc .



## **ANNEXE II: DOCUMENT SRS**

Le document SRS présente principalement les exigences fonctionnelles et non fonctionnelles du constructeur de campagne et de l'application de jeu. Il présente aussi les principes directeurs à garder en tête plus tard, lors de l'élaboration de l'interface utilisateur. Le document SRS est contenu dans le fichier Ftaillefer\_LOG792\_SRS.doc .

### **ANNEXE III: DOCUMENT DE CONCEPTION**

Le document de conception présente le rôle de chaque élément du dictionnaire de données à l'intérieur d'un diagramme de classes. Il présente aussi un diagramme des dépendances qui permet de voir rapidement les interactions entre les différents éléments du système. Le document est contenu dans le fichier Ftaillefer\_LOG792\_DocumentConception.doc .