

Warhammer 40K : Army Factory

De la conception à la réalisation d'une
application de gestion de créations d'armées
dans l'univers de Warhammer 40000

Alexis CORNET, Nicolas APERCE

Le 14 décembre 2016

Warhammer 40K : Army Factory est un projet débuté dans le cadre du cours « 8INF957 Programmation objet avancée » dispensé au cours du semestre d'automne 2016 à l'Université du Québec à Chicoutimi (UQAC).

Ce projet a ainsi été pensé, conçu et réalisé par Alexis CORNET et Nicolas APERCE, tous deux étudiants à l'UQAC.

CORA24129403, alexis.cornet1@uqac.ca

APEN05049401, Nicolas.aperce1@uqac.ca

Abstract

Dans le but d'apprendre la programmation en langage C++, deux étudiants de l'Université du Québec à Chicoutimi ont décidé de réaliser une application de gestion des armées pour le jeu de figurines Warhammer 40K. Ils ont pour ce faire, décidé d'y penser et de l'imaginer sur feuille avant de la concevoir via l'emploi du langage UML, puis par la suite, en la développant via le C++.

Mots Clef

Projet informatique, Études, Warhammer 40K, Programmation Orientée Objet, C++, Conception, UML

Table des matières

Abstract.....	2
Mots Clef.....	2
1. Introduction.....	3
2. La préparation aujourd'hui.....	4
3. Matériel et méthode	6
3.1. Matériels	6
3.2. Méthodes	6
3.2.1. Conception.....	6
3.2.2. Réalisation prévue.....	8
4. L'application.....	9
4.1. Préambule.....	9
4.2. Objectif	9
4.3. Atteint.....	10
4.4. Fonctionnement	10
Pistes d'améliorations	13
Table des légendes.....	14
Références.....	14

1. Introduction

Warhammer 40000 (ou Warhammer 40K) est, à la base, un jeu de figurines existant depuis 1987 réalisé et produit par la société Games Workshop.

Univers et jeu de plus en plus connus des joueurs et collectionneurs de jeux de figurines et/ou de plateaux, il met en opposition un minimum de deux joueurs autour d'un terrain, disposant chacun d'une armée de figurines, qu'il aura préalablement assemblées et peintes.

Depuis sa création, le jeu a fait l'objet de très nombreuses modifications et mises à jour. À ce jour, les règles du jeu sont définies dans un livre dit « de règles universelles » de près de 200 pages, auxquelles il faut ajouter des suppléments obligatoires d'une de centaine de pages chacun, appelés « codex » ou « livres d'armées », définissant les règles de l'une des nombreuses factions existantes. Bien que cet univers soit en constante expansion, aucune application n'existe actuellement dans le but d'aider ou d'assister le joueur à la conception et à la création d'une armée avant de réaliser une partie.

Par conséquent, l'application « Warhammer 40K : Army Factory », constituant le cœur de cet article, est conçue dans ce but. Le joueur peut désormais penser, créer, maintenir et optimiser son armée via cette application.

Le document qui suit explique les phases de conception et de réalisation d'un projet visant à concevoir une telle application, réalisée par deux étudiants à l'Université du Québec à Chicoutimi. Bien que cette application ne soit, ni complète, ni achevée, la phase de conception ayant permis de procéder à sa réalisation restera, par la suite, similaire.

2. La préparation aujourd'hui

À Warhammer 40K, la « force » d'une armée de figurines est définie par une valeur en points, fixée avant la partie par les joueurs.

Pour pouvoir participer à une partie, un joueur doit :

- Effectuer des recherches dans le codex de sa faction par rapport aux figurines qu'il possède
- Recenser les différentes figurines et unités qu'il souhaite aligner dans son armée
- Calculer les points de la création de chacune de ces unités en fonctions des figurines et équipements la composant, et ce, dépendant des différentes règles respectées
- S'assurer que la valeur totale de son armée soit égale à la valeur fixée pour la partie, plus ou moins une dizaine de points.

Le tout en respectant un certain nombre de règles et conditions, parfois universelles, parfois dépendantes de la faction, voire de l'unité.

Une armée est ainsi composée d'unités de figurines. Chaque figurine possède un profil de caractéristiques reflétant ses aptitudes au combat, et a accès, via son unité, à un panel d'équipements et de règles spéciales visant à améliorer son potentiel guerrier.

La plupart du temps, un joueur joue pour le plaisir. Mais dans le cas de compétitions, officielles ou non, s'il souhaite optimiser son armée, le joueur doit calculer l'intégralité de ses points avant d'effectuer un changement et gérer ses figurines, procéder à des schémas d'armées et à la réalisation de calculs statistiques.

Tout ceci étant fastidieux, et les calculs pouvant s'avérer complexes en fonction des corps d'armées et des règles choisies, il est aisé pour le joueur de se tromper dans ses calculs, et, par conséquent, dans la création même de son armée.

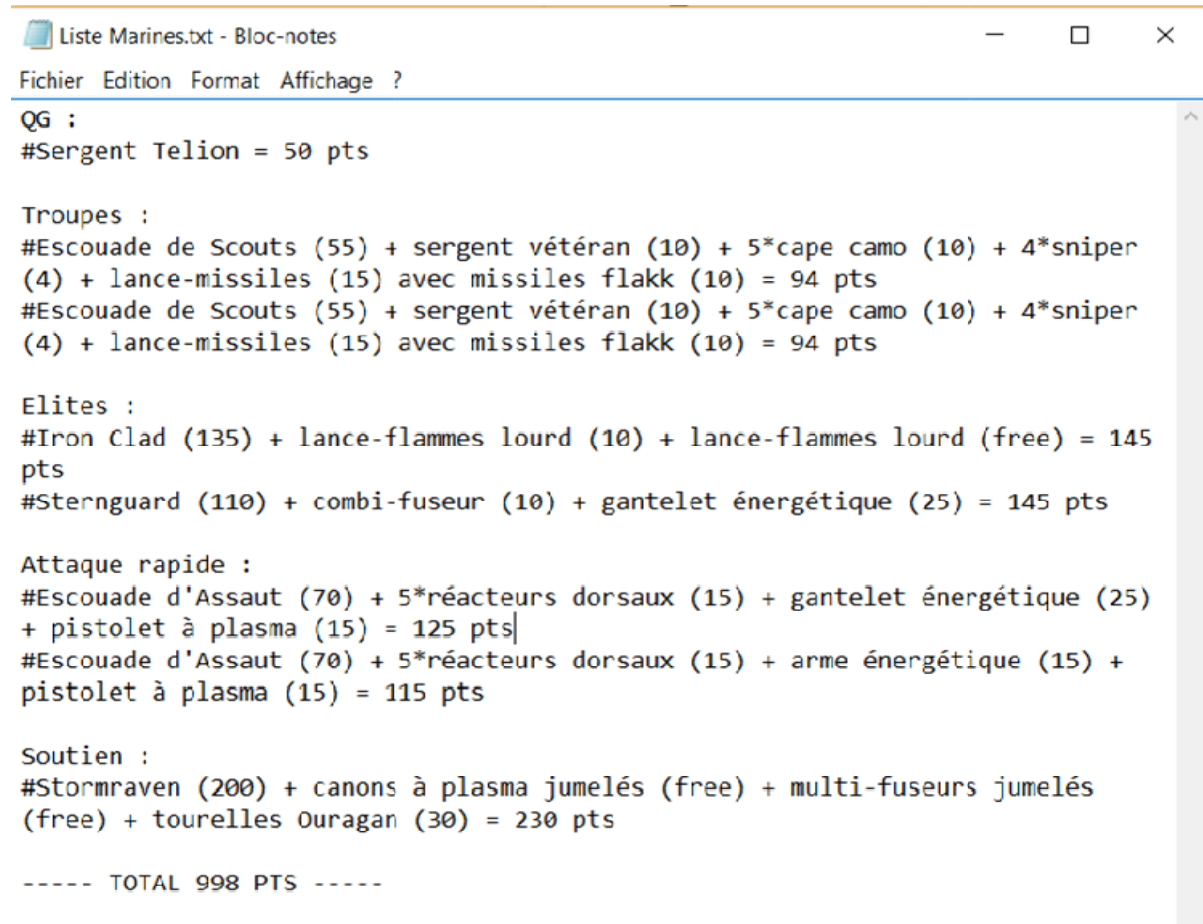
La figure 1 ci-dessous consiste en un exemple de création d'une armée, réalisée avec le bloc-notes Windows, pour une partie établie à 1000 points.

On y retrouve des rôles tels que « QG », « Troupes », « Elites », « Attaque rapide » et « Soutien » ; et on remarque que cette armée fait 998 points.

L'armée dispose des unités « Sargent Telion », « Escouade de Scouts », « Iron Clad », « Sternguard », « Escouade d'Assaut », « Stormraven ».

On aperçoit de plus les figurines composant ces différentes unités, ainsi que les équipements leur étant attachés.

De la sorte, si le joueur souhaite désormais améliorer et ainsi optimiser son armée pour la rendre plus compétitive, il se doit de trouver par quel moyen (changement de figurine, d'équipement, etc.), tout en effectuant un nouveau calcul de points pour chaque changement envisagé.



```
Liste Marines.txt - Bloc-notes
Fichier Edition Format Affichage ?

QG :
#Sergent Telion = 50 pts

Troupes :
#Escouade de Scouts (55) + sergent vétéran (10) + 5*cape camo (10) + 4*sniper
(4) + lance-missiles (15) avec missiles flakk (10) = 94 pts
#Escouade de Scouts (55) + sergent vétéran (10) + 5*cape camo (10) + 4*sniper
(4) + lance-missiles (15) avec missiles flakk (10) = 94 pts

Elites :
#Iron Clad (135) + lance-flammes lourd (10) + lance-flammes lourd (free) = 145
pts
#Sternguard (110) + combi-fuseur (10) + gantelet énergétique (25) = 145 pts

Attaque rapide :
#Escouade d'Assaut (70) + 5*réacteurs dorsaux (15) + gantelet énergétique (25)
+ pistolet à plasma (15) = 125 pts
#Escouade d'Assaut (70) + 5*réacteurs dorsaux (15) + arme énergétique (15) +
pistolet à plasma (15) = 115 pts

Soutien :
#Stormraven (200) + canons à plasma jumelés (free) + multi-fuseurs jumelés
(free) + tourelles Ouragan (30) = 230 pts

----- TOTAL 998 PTS -----
```

Figure 1 : Exemple de création d'une armée

3. Matériel et méthode

3.1. Matériels

L'application appelée « Warhammer 40K : Army Factory », parfois appelée WH40KAF par la suite, a été conçue et réalisée à l'aide de logiciels entièrement gratuits.

Pour ce faire, les technologies employées sont :

- L'IDE « Microsoft Visual Studio Community » ainsi que le langage C++ pour la programmation de l'application, ainsi que le respect du paradigme orienté objet ;
- Le logiciel UMLet et le langage UML pour la partie conception de l'application
- L'application « Git Hub » pour le stockage et partage d'éléments entre les différents contributeurs au projet
- Un serveur Apache pour la gestion et l'hébergement de la base de données

3.2. Méthodes

3.2.1. Conception

« Warhammer 40K : Army Factory » a d'abord été pensée sur feuille de papier afin de regrouper l'ensemble des idées et informations que l'application doit nécessairement contenir.

L'application a été découpée en une architecture 3-tiers dont les 3 couches sont les suivantes :

- DAL pour « Data Access Layer »
- BLL pour « Business Logic Layer »
- IHM pour « Interface Homme Machine »

La conception du projet fut définie en employant le langage UML via le logiciel open source UMLet. Deux types de diagrammes UML furent réalisés.

Un diagramme de package (tel présenté sur la figure 2 ci-dessous) a été conçu, mettant en évidence les dépendances architecturales des 3 couches.

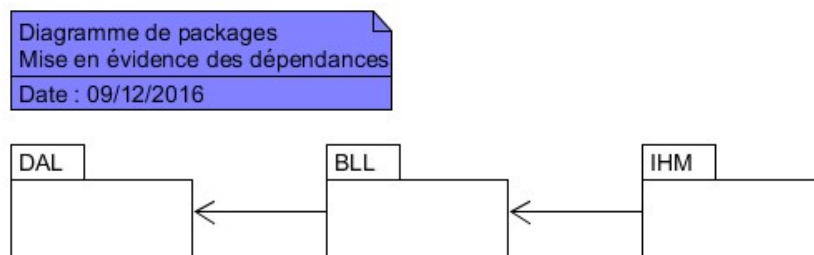


Figure 2 : Diagramme de packages et dépendances

On remarque sur la figure 2 ci-dessus, que la couche IHM dépend de la couche BLL, elle-même dépendante de la couche DAL.
La couche DAL sert à la connexion avec la base de données.
La couche BLL permet la réalisation d'opérations métiers sur les différents éléments mis en base.
La couche IHM s'assure de l'affichage de l'application.

Des diagrammes de classes ont été dessinés dans le but de montrer les interactions des dites classes au sein d'une même couche.

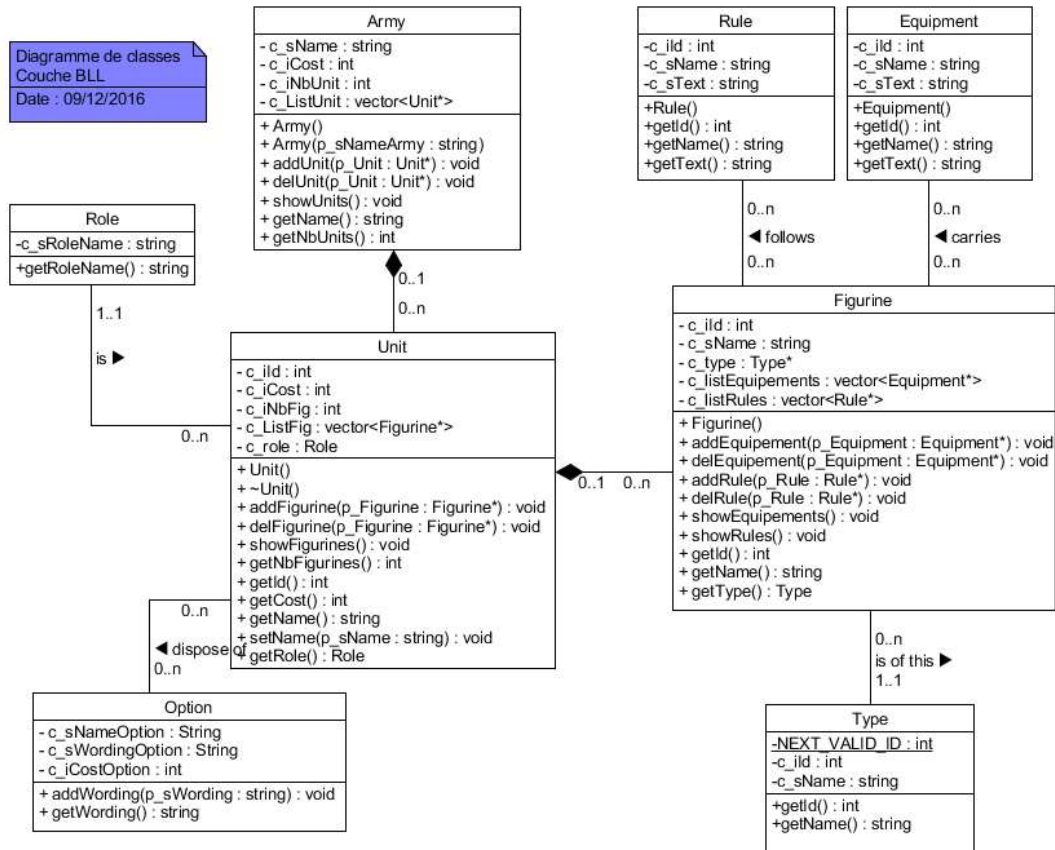


Figure 3 : Diagramme de classes de la couche BLL

La figure 3 ci-dessus représente le diagramme de classes correspondant aux interactions qui s'effectuent dans la couche BLL. L'armée est composée d'unités, elles-mêmes composées de figurines. L'unité respecte un rôle et dispose d'options. La figurine est d'un certain type, porte des équipements et suit certaines règles.

3.2.2. Réalisation prévue

Afin de réaliser l'application WH40KAF, les principes du paradigme de la programmation orientée objet furent respectés. On retrouve ainsi les principes classiques comme le polymorphisme (au travers de l'héritage), la réutilisabilité ou la séparation des responsabilités (une classe appelée « Figurine » ne traitera donc que des dites figurines).

L'application est une application dite Windows Form réalisée avec l'IDE « Microsoft Visual Studio Community ». Il s'agit donc d'un projet de type « CLR », et employant le langage Visual C++ basé sur le Framework .NET4.5 développé par Microsoft.

L'application est connectée à une base de données MySQL réalisée via PHPMyAdmin et gérée par un serveur Apache.

La couche DAL telle réalisée définit les méthodes de connexion et de déconnexion à la base de données.

La couche BLL permet la récupération et la modification des éléments mis en base, et sert de « base des opérations » pour toute interaction entre ce que voit l'utilisateur et la base de données.

La couche IHM dispose quant à elle seulement de fichiers de type fenêtres ou vues, appelées respectivement « Windows Form » et « CLR User Control » sur l'IDE. Ces fichiers correspondent ainsi à la partie visible par l'utilisateur.

4. L'application

4.1. Préambule

Reprenant la partie « La préparation aujourd'hui » telle définie précédemment, notre application a pour objectif de simplifier cette préparation.

L'application se doit par conséquent d'éviter à l'utilisateur d'effectuer ses recherches et calculs de façon continue. Il continuera d'effectuer ses recherches de figurines afin de savoir laquelle ajouter à son armée, mais n'aura nullement nécessité à vérifier l'intégralité des interactions possibles entre la dite figurine et les équipements ou règles auxquels la lier, ni même besoin de calculer le nouveau coût de son armée, le tout devant s'effectuer automatiquement sur l'application.

4.2. Objectif

Ayant pour objectif d'aider et d'assister l'utilisateur dans la création et l'optimisation d'armées jouables, l'application se doit de permettre à l'utilisateur de réaliser l'ensemble des tâches nécessaires à la création de son armée.

Autrement dit, l'utilisateur cherche dans la base de données de l'application, la figurine qu'il souhaite ajouter à son unité. De là, l'application lui fait connaître, en le lui affichant, le coût de cette figurine, puis le coût de l'unité contenant cette figurine.

Lorsque l'utilisateur décide de modifier son unité, par le biais d'un ajout ou suppression, modification d'une figurine, l'application affiche le nouveau coût de l'unité.

Les équipements disposant de coûts respectifs dépendamment des armées jouées, l'application doit également afficher à l'utilisateur le nouveau coût de l'unité et de l'armée lorsque celui-ci décide de la modifier par le biais d'un ajout, suppression, ou modification d'un équipement.

L'application propose à l'utilisateur d'enregistrer les unités correspondant à ses choix, ainsi que l'enregistrement et la validation de son armée après y avoir inclus les différentes unités la composant.

Parallèlement à ce travail d'édition, l'application devrait également permettre d'effectuer des calculs statistiques sur les capacités des unités de l'armée créée.

L'ensemble des données de l'application (figurines, unités, armées créées, équipement, etc.) sont présentes en une base de données MySQL hébergée sur un serveur Apache.

4.3. Atteint

La conception UML de l'application montrait que cette dernière devait suivre une architecture 3-Tiers, cependant. Cependant, pour des raisons techniques, cette architecture n'a pas pu être respectée.

En effet, il était initialement prévu de réaliser une « solution » dans l'IDE « Microsoft Visual Studio », qui contiendrait 3 projets « DAL », « BLL », « IHM », et où les références DLL à chacun de ces projets permettraient le respect de la dite architecture. Néanmoins, un problème certainement dû à la configuration de l'IDE ou des différents projets a conduit à revoir ce schéma, et un seul projet fut finalement réalisé. Ce projet est ainsi séparé en 3 « namespaces » différents, représentant chacun une partie de l'architecture ; on retrouve ainsi un namespace « DAL », un namespace « BLL » et un namespace « IHM ».

L'accès à la base de données a été effectué comme prévu, et cette dernière est hébergée sur un serveur Apache.

4.4. Fonctionnement

Lorsque l'utilisateur démarre l'application « Warhammer 40K : Army Factory », celle-ci se connecte à sa base de données « test_army_factory.sql » et affiche l'écran d'accueil. L'utilisateur n'a nullement connaissance de l'établissement de connexion, et ne voit que la fenêtre d'accueil telle présentée sur la figure 4 présente ci-dessous.



Figure 4 : Écran d'accueil de l'application

L'utilisateur a dès lors la liberté de :

- Créer une armée ;
- Voir (et modifier) une armée ;
- Voir les figurines ;
- Vérifier les options propres à l'application ;
- Revenir à cet écran lorsqu'il n'est plus dessus ;
- Quitter l'application

On retrouve sur les figures 5 et 6 suivantes, l'écran de création d'une armée ainsi que la possibilité d'y ajouter des unités

WarHammer40KArmyFactory

Accueil Créer une armée Visualiser ses armées Visualiser ses figurines Options À propos

Créer son armée

Nom de l'armée :

Nombre de points :

	Nom de l'unité	Rôle	Coût	Nombre d'unités	Modifier l'unité
*					

Ajouter une unité Enregistrer l'armée Annuler

Figure 5 : Écran de création d'une armée

L'écran représenté par la figure 5 ci-dessus propose à l'utilisateur de choisir un nom pour son armée. Ce même écran lui propose également d'ajouter une unité (cf. figure 6 ci-dessous), ainsi que de valider, ou bien au contraire, d'annuler la création de la dite armée.

L'application affiche à l'utilisateur, l'ensemble des unités qu'il aura préalablement définies et qu'il souhaite incorporer dans son armée, ainsi que la totalité des points relative à ces unités.

Lorsque l'utilisateur appuie sur « Enregistrer l'armée », cette dernière est automatiquement enregistrée dans la base de données ; tandis que lorsqu'il clique sur « Annuler », cette fenêtre se vide.

La fenêtre présente sur la figure 6 ci-dessous propose à l'utilisateur de sélectionner, via une liste déroulante, les modèles de base possibles pour une unité. L'utilisateur saisi de plus un nom pour son unité, et sélectionne son rôle dans l'armée. La liste déroulante d'équipements affiche à l'utilisateur les équipements indiqués par celui-ci.

Afin de remplir cette liste, l'utilisateur appuie sur « Ajouter un équipement » qui ouvre une nouvelle fenêtre indiquant les informations à renseigner.

Il procède de même pour ajouter une nouvelle figurine en appuyant sur « Ajouter une figurine ».



BtnOk

Ajouter une unité

Modèle d'unité :

Équipements :

Nom de l'unité :

Rôle dans l'armée :

Nombre de points :

Taille de l'unité :

Ajouter une figurine Ajouter un équipement Valider

Figure 6 : Écran d'ajout d'une unité

Pistes d'améliorations

Ayant rencontré divers problèmes techniques comme par exemple l'apprentissage du Framework nécessaire à la réalisation d'un projet CLR, l'application n'est aujourd'hui pas fonctionnelle.

Ayant désormais connaissance, et du langage C++, et du Framework .NET4.5, et de certaines parties du Framework CLR, il devrait désormais être faisable (sans davantage de difficultés apparentes) de réaliser les différentes fonctionnalités prévues, qui ne sont pour l'instant pas implémentées dans la couche IHM. En effet, ces actions ou opérations sont réalisées dans la couche BLL, cependant leurs liens avec la couche IHM restent à l'heure actuelle inachevés.

En allouant une vingtaine d'heures supplémentaires, il serait possible de mettre en place ces différentes fonctionnalités.

De plus, lors de la prise de décision de réaliser une telle application, l'idée de réaliser des calculs statistiques a été mise en avant. Ces calculs auraient pour objectif de permettre au joueur d'optimiser son armée. Cette implémentation pourrait ainsi correspondre à une autre release.

Table des légendes

Figure 1 : Exemple de création d'une armée	5
Figure 2 : Diagramme de packages et dépendances	6
Figure 3 : Diagramme de classes de la couche BLL	7
Figure 4 : Écran d'accueil de l'application	10
Figure 5 : Écran de création d'une armée	11
Figure 6 : Écran d'ajout d'une unité	12

Références