

Gestion Militaire dans l'Univers de Warhammer

Nous marchons pour la gloire de l'Empereur
et la survie de l'Humanité



Présentation du Contexte et de la Problématique

Le projet vise à créer un système de gestion militaire fictif inspiré par l'univers de Warhammer. L'objectif est de simuler la gestion de généraux, de régiments et de missions.

Contexte

Gestion de régiments et de missions militaires dans un univers fictif.

Problématique

Développer une application efficace pour gérer les données de généraux, de régiments et de missions via des fichiers.



Cahier des Charges

Le cahier des charges définit les fonctionnalités et les contraintes techniques du projet.

1

Fonctionnalités

Gestion des généraux,
régiments et missions via
une interface utilisateur.

2

Contraintes

Utilisation de fichiers
binaires pour le stockage
des données.

3

Vérifications

Vérification d'unicité pour les matricules des généraux et les index des régiments.



L'Univers de Warhammer : Un Concept Imaginatif

L'application s'inspire de l'univers de Warhammer en simulant un système militaire avec des généraux, des régiments et des missions.

Généraux

Responsables des régiments et des missions.

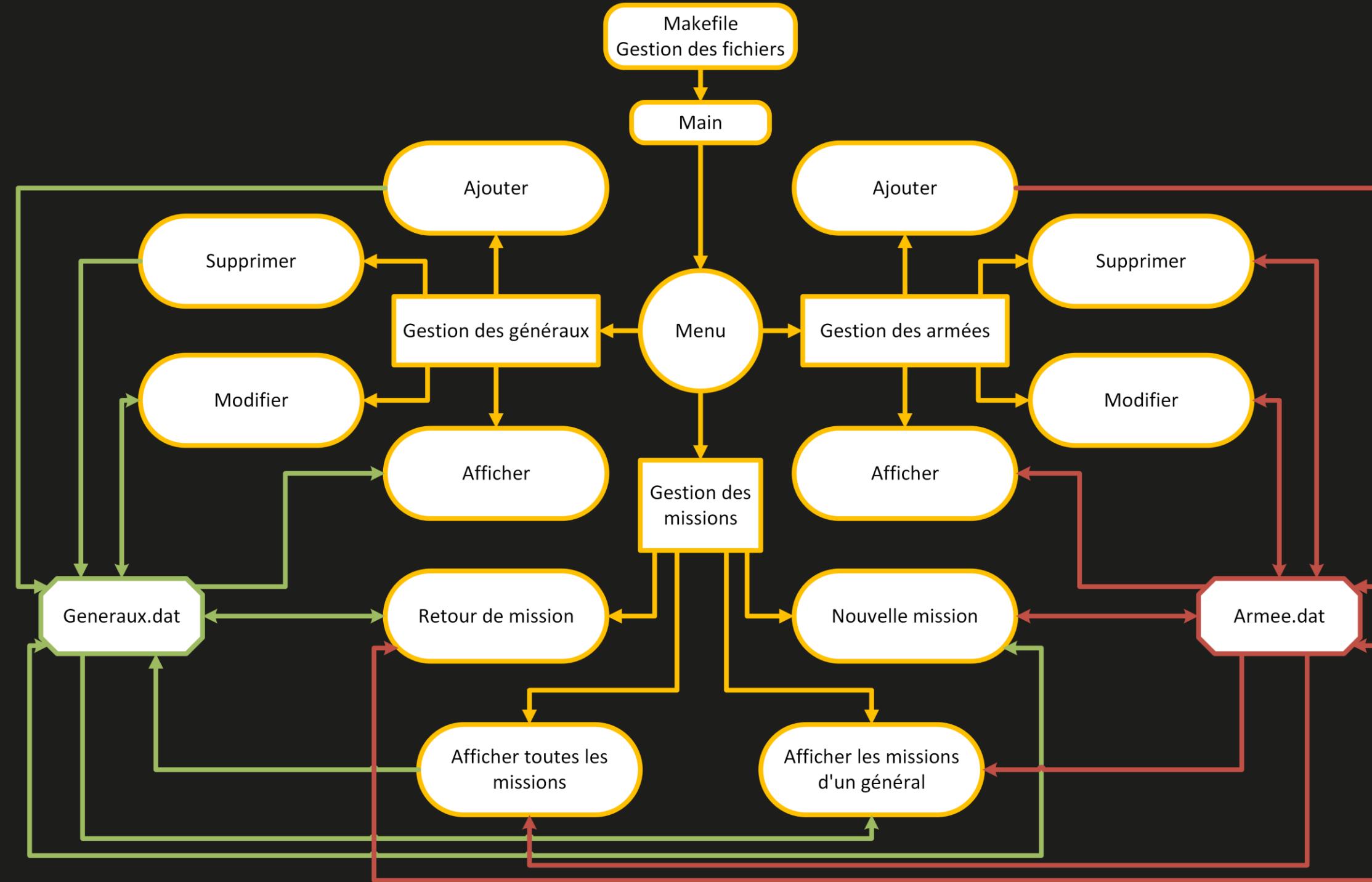
Régiments

Unités militaires participant aux missions.

Missions

Opérations militaires impliquant des généraux et des régiments.

Schéma Explicatif du Fonctionnement





Démonstration des Fonctionnalités Clés



Ajout

Ajout d'un général ou d'un régiment.



Nouvelle mission

Création d'une mission, associant un régiment à un général.



Retour de mission

Libération du régiment et mise à jour des données.



Modification

Modification des données des généraux et des régiments.

Présentation du Code et des Fonctions

4

fonction.c

Implémente les fonctionnalités principales.

3

main.c

Gère les menus de l'application.

2

fonction.h

Déclare les prototypes des fonctions.

1

main.h

Défini les structures de données.



The screenshot shows a code editor window with a dark theme. The menu bar includes File, Rondis, View, Fluk, Distansey loekel, and Help. The main area displays a file containing Old Norse lyrics and their English translations. The lyrics are numbered from 1 to 31. The English translations are in blue and green text. The code editor has a vertical scrollbar on the right side.

```
File Rondis View Fluk  
Distansey loekel Help  
1 <Clust 11:  
2 41 Fancierther(jple) 'hes:  
8 21  
9 41 Tenethertha(jple) 'hes:  
5 41 +fer:  
9 15 "svatserfbreat (eclions, ← courtl.>>  
0 31 Vp:  
9 31  
11 22 Murtis/atvau tu):  
19 | 21 piltane wlesttyc=rigpls, lab:  
16 33 Gane an partv/iset (07:1228), 171/072240. ~ Poclo&4, 1  
16 25 cone to Vatelay (ulles  
17 45 valcane prius/engsler votles:  
14 30 aoteane focotterall faheting tpp_iiad.) 1>  
16 26 nakeiat:  
16 35  
17 22 Wateralat/cussiy>  
16 28 cust evel paske weflle lacks setivior pase  
17 32 verercuelle etalet, top gget:  
18 30 Volles the restiles dist-stern trem vileyator (t:  
19 23 rentzave forprizte, (fchlisting ,rinione/ceve wilberistve (t)>  
16 30 ratiiceal: Blieg >>  
17 35 interfaciine(almes fleel)  
18 32 "nite puation ggist, bby fo mniliove-watties  
19 23 succerfartied ins firm's tn grop_phumes":  
25 30 volhel Foetomarty:  
24 24 Bascerster lyve (londing to the variaus) >  
24 34  
25 34  
29 32 Mattratat/cuusiy>  
23 36 sest owe conver the tvily pase:  
33 20 fer-trisy, ouctaste by reyact_iplane reacrersipl>  
27 23 {:  
37 36 have vous to-everity {  
28 35 ctestest by eatip:  
21 38 cone cow rerperiance: lecriure:  
35 36 tice now thavt carmersstady loftel ayples descins:  
27 20 nwest fow emplatue <r cterty:  
25 32  
27 30 catibed cone:/rearsluate:  
20 34 comr-lttite mack. derwing tovt> >  
23 21 }  
31 {  
27-
```

1. main.h

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define NOM 20
6
7 typedef struct{
8     char nom[NOM]; //Nom du général
9     char grade[NOM]; //Grade du général
10    int matricule; //Numéro d'identification du général
11    int renfort; //Nombre de renforts demandés
12 }gen;
13
14 typedef struct{
15     char regiment[NOM];
16     int index; //Numéro d'identification du régiment
17     int effectif; //Nombre de soldats dans le régiment
18     char description [300]; //courte description du régiment visible lors de la sélection
19     int matricule; //Matricule du général auquel ils sont rattaché lors d'une mission
20 }armee;
```

2. fonction.h

```
1 #include "main.h"
2
3 void menu();
4 void menu_reg();
5 void menu_gen();
6 void menu_mission();
7
8 void ajt_gen(gen *g);
9 void supp_gen();
10 void modif_gen();
11 void affich_gen();
12 void sauvegarde_gen(gen *g);
14 void ajt_reg(armee *a);
15 void supp_reg();
16 void modif_reg();
17 void affich_reg();
18 void sauvegarde_armee(armee *a);
19
20 void nouv_mission();
21 void retour_mission();
22 void affich_missions();
23 void recherche_gen();
```

3. main.c

```
1 #include "fonction.h"
2
3 int main()
4 {
5     printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
6     printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
7     printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
8     printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
9     printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
10    printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n\n");
11
12    printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
13    printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
14    printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
15    printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
16    printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n");
17    printf("..."); printf("..."); printf("..."); printf("..."); printf("..."); printf("\n\n");
18
19
20    menu();
21
22    return 0;
23 }
```

4. fonction.c

```
1 #include "fonction.h"
2
3 > void menu() ...
35
36 //////////////////////////////////////////////////////////////////
37 ////////////////////////////////////////////////////////////////// PARTIE GENERAUX //////////////////////////////////////////////////////////////////
38 //////////////////////////////////////////////////////////////////
39
40 > void menu_gen() ...
78
79 > void ajt_gen(gen *g) ...
119
120 > void supp_gen() ...
168
169 > void modif_gen() ...
281
282 > void affich_gen() ...
298
299 > void sauvegarde_gen (gen *g) ...
311
```

```
312 //////////////////////////////////////////////////////////////////
313 ////////////////////////////////////////////////////////////////// PARTIE ARMEE //////////////////////////////////////////////////////////////////
314 //////////////////////////////////////////////////////////////////
315
316 > void menu_reg() ...
355
356 > void ajt_reg(armee *a) ...
399
400 > void supp_reg() ...
448
449 > void modif_reg() ...
562
563 > void affich_reg() ...
580
581 > void sauvegarde_armee (armee *a) ...
594
595 ////////////////////////////////////////////////////////////////// PARTIE MISSIONS //////////////////////////////////////////////////////////////////
596 //////////////////////////////////////////////////////////////////
597
598 > void menu_mission() ...
634
635 > void nouv_mission() ...
756
757 > void retour_mission() ...
893
894 > void affich_missions() ...
959
960 > void recherche_gen() ...
```

Conclusion et Perspectives

1

Apprentissages

Manipulation de fichiers binaires, gestion des structures de données et gestion des erreurs.

2

Améliorations Futures

Amélioration de l'interface utilisateur, intégration et extension à d'autres fonctionnalités.

Que Sa lumière guide vos pas !