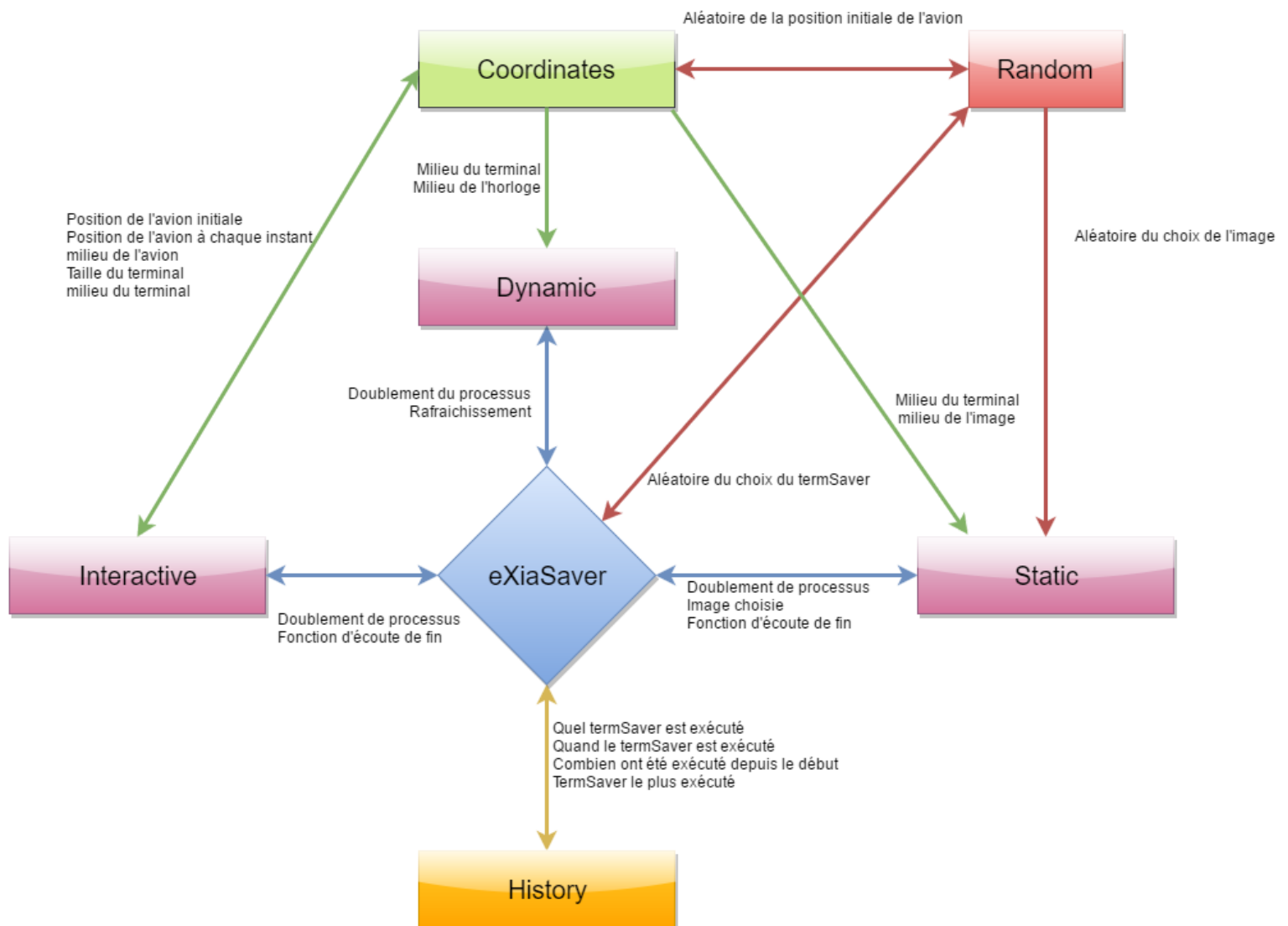


Première partie : ANALYSE

1. Dessinez l'architecture logiciel – comment avez-vous compris le projet ?



COMMENT AVEZ-VOUS COMPRIS LE SUJET ?

Le projet qui nous a été donné porte sur la création d'un écran de veille : l'eXiaSaver.

Cet écran de veille devra afficher trois types de termSaver affichés aléatoirement :

- TermSaver statique : ce mode va afficher sur l'écran de veille un simple dessin fait à partir de 0 et de 1 (0=pixel blanc, 1=pixel noir). Il y aura 7 dessins tirés au sort, un seul sera affiché et centré sur le Shell.
- TermSaver dynamique : ce mode va afficher une horloge digitale (centrée) qui se mettra à jour toutes les x secondes.
- TermSaver interactif : ce mode affichera un avion (dessiné en 0 ou 1, d'une taille de 6x6) qui s'affichera aléatoirement sur l'écran (qui fait 80x24). L'avion pourra être déplacé grâce à l'appui de touches spécifiques (définie dans le manuel d'utilisation) dans les 4 directions possibles (haut, bas, gauche, droite). Il devra aussi prendre en compte les limites de l'écran, c'est-à-dire lorsque le dessin dépassera les bordures, il devra apparaître de l'autre côté de l'écran.

A la suite de ces tâches effectuées, un historique contenant des informations comme l'heure à laquelle l'eXiaSaver a été lancé sera disponible en tapant la commande suivante : exiasaver -stats

ARCHITECTURE LOGICIEL :

L'utilisation du losange bleu au centre nous permet de mieux cerner qu'il s'agit de notre logiciel, l'exécutable.

History en jaune car c'est un fichier à part, il s'agit en effet d'un fichier texte. C'est le seul à être de cette nature dans notre programme. Il va être stocké sur l'ordinateur même.

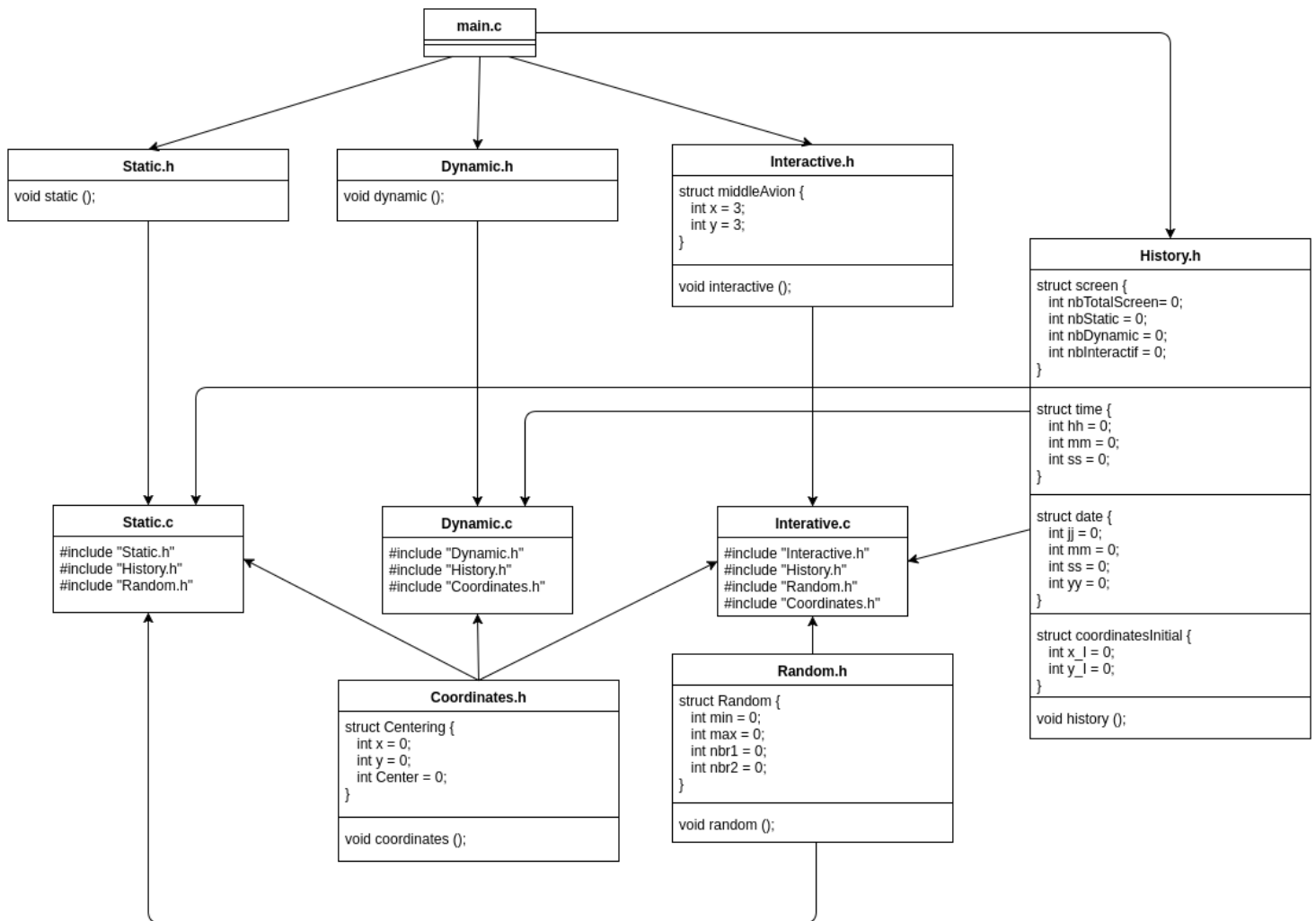
Les 3 modules termSaver sont de même couleur. Ils agissent directement à la suite du lancement de l'exécutable.

Ces 3 modules font appel à des struct contenant des variables.



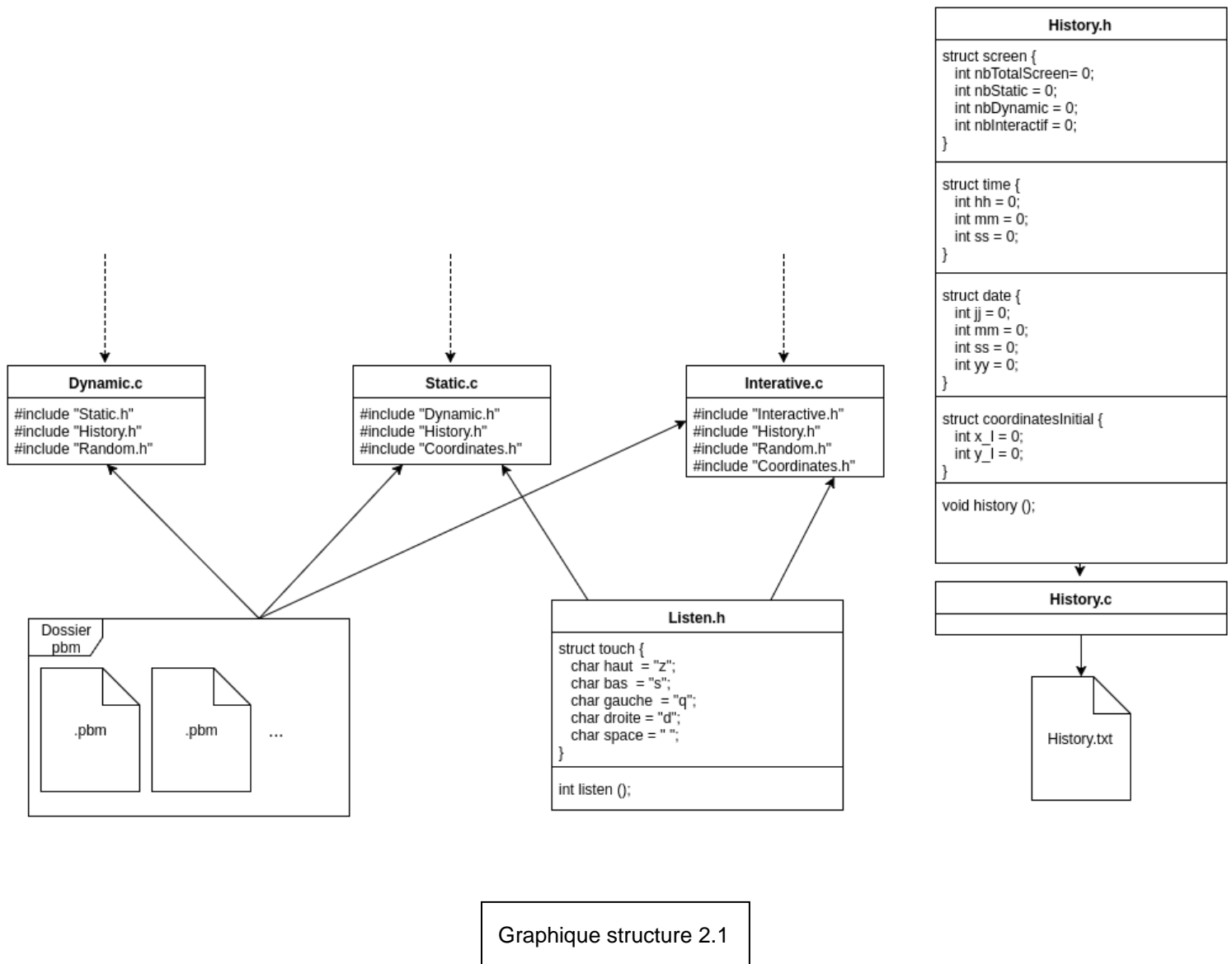
Deuxième partie : ANALYSE DE DONNEES ET STRUCTURES

1. *Représentation graphique de toutes les structures nécessaires. Sans oublier les liens avec les fichiers externes lus ou écrits.*



Graphique structure 2.0



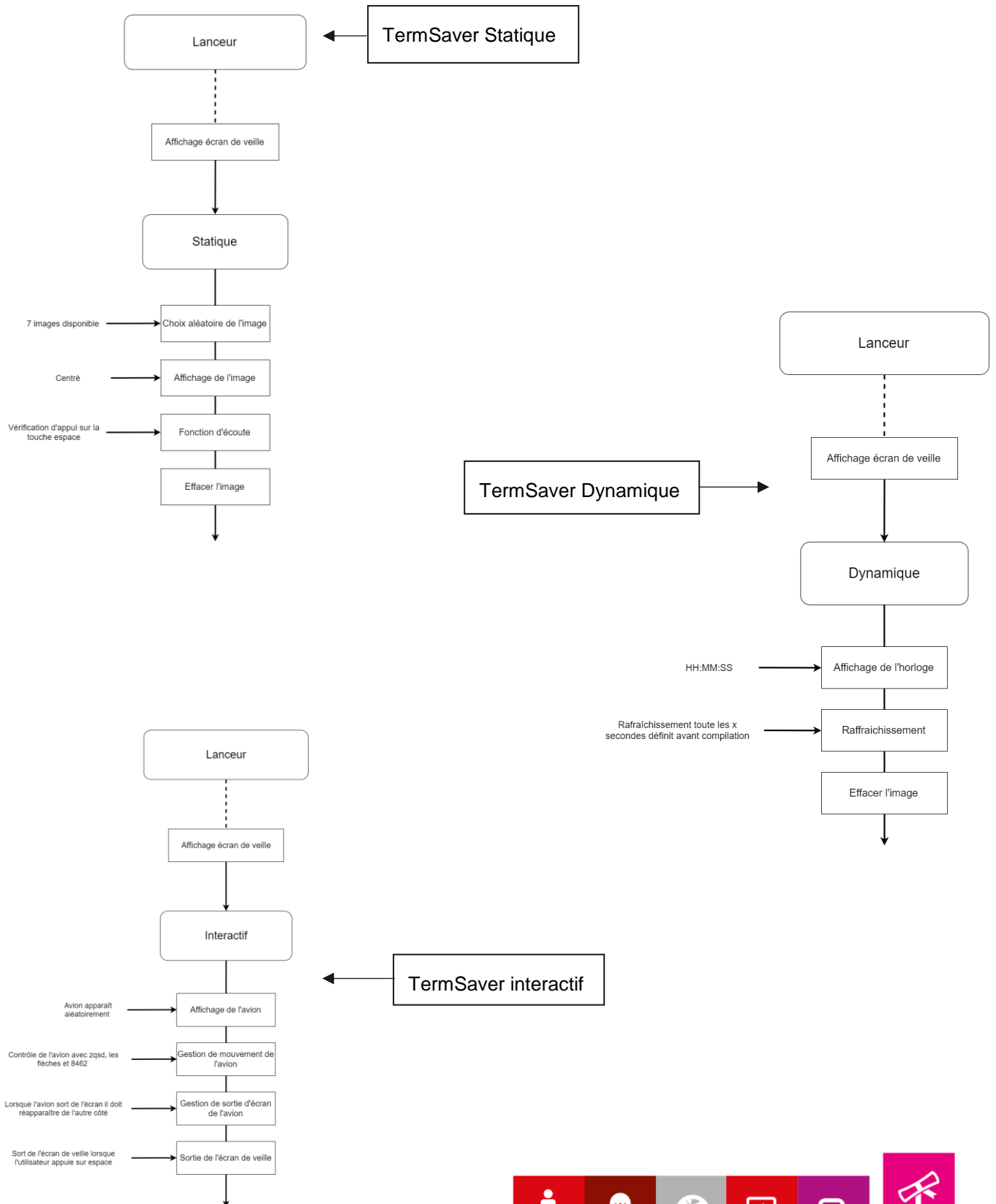


Ce graphique représente l'utilisation des fichiers écrits (historique) et lus (.pbm) dans le programme.

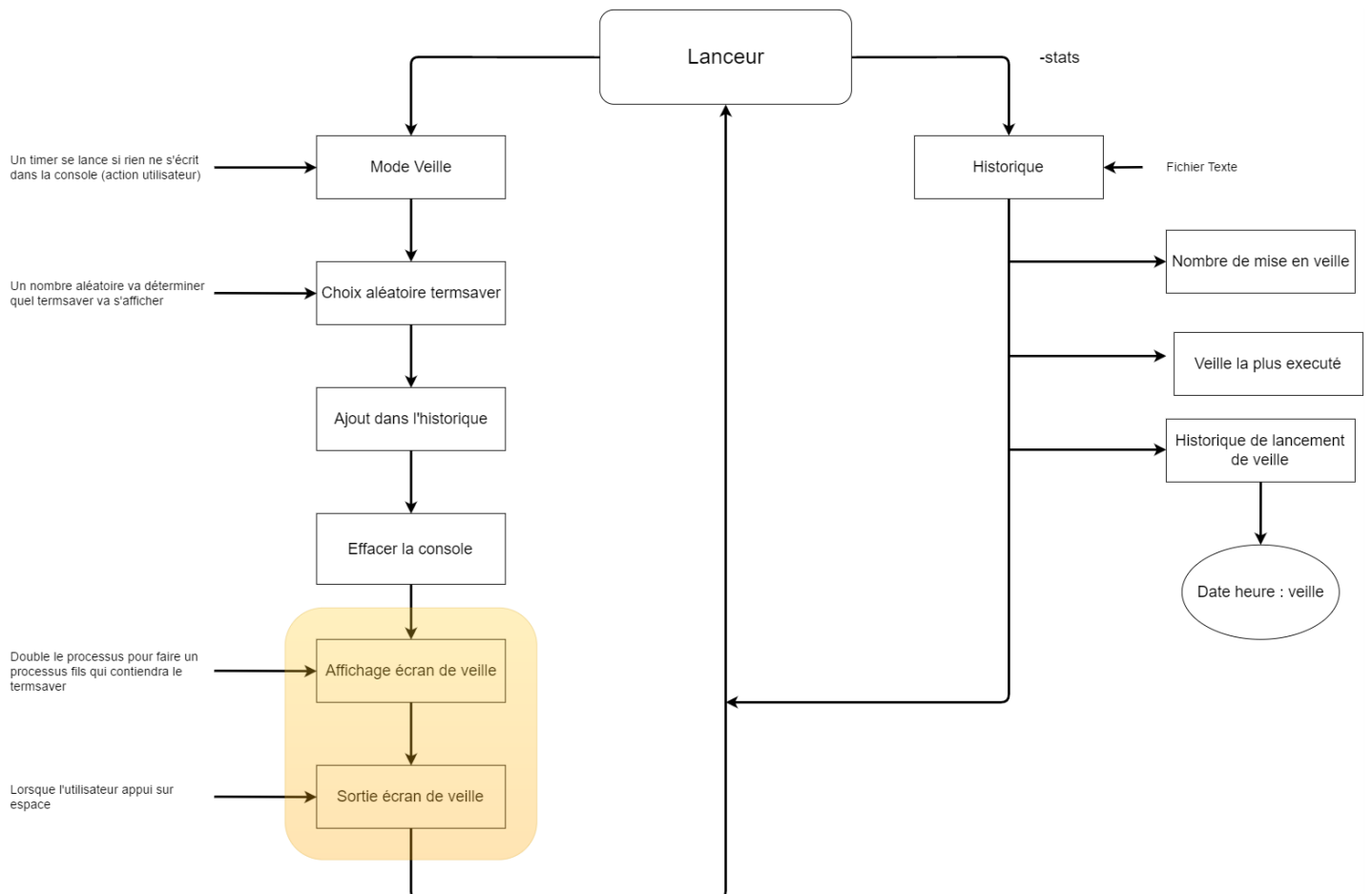


Troisième partie : MODULARISATION & WORKFLOW DE FONCTIONS

1. Description graphique de chaque module (lanceur exiaSaver et les 3 termSaver) - logigramme ou workflow



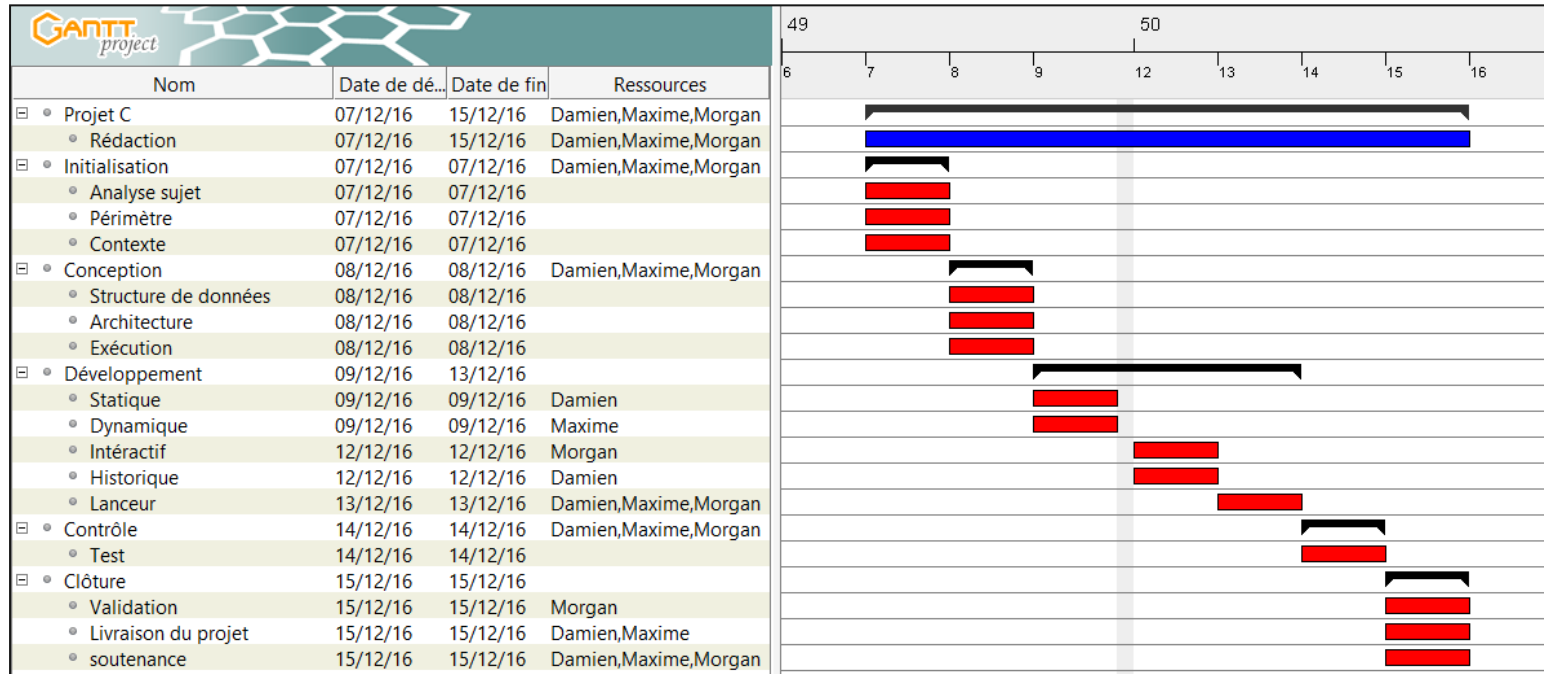
2. Prototypes de l'ensemble des fonctions du projet (faites-le le plus « graphique » possible)



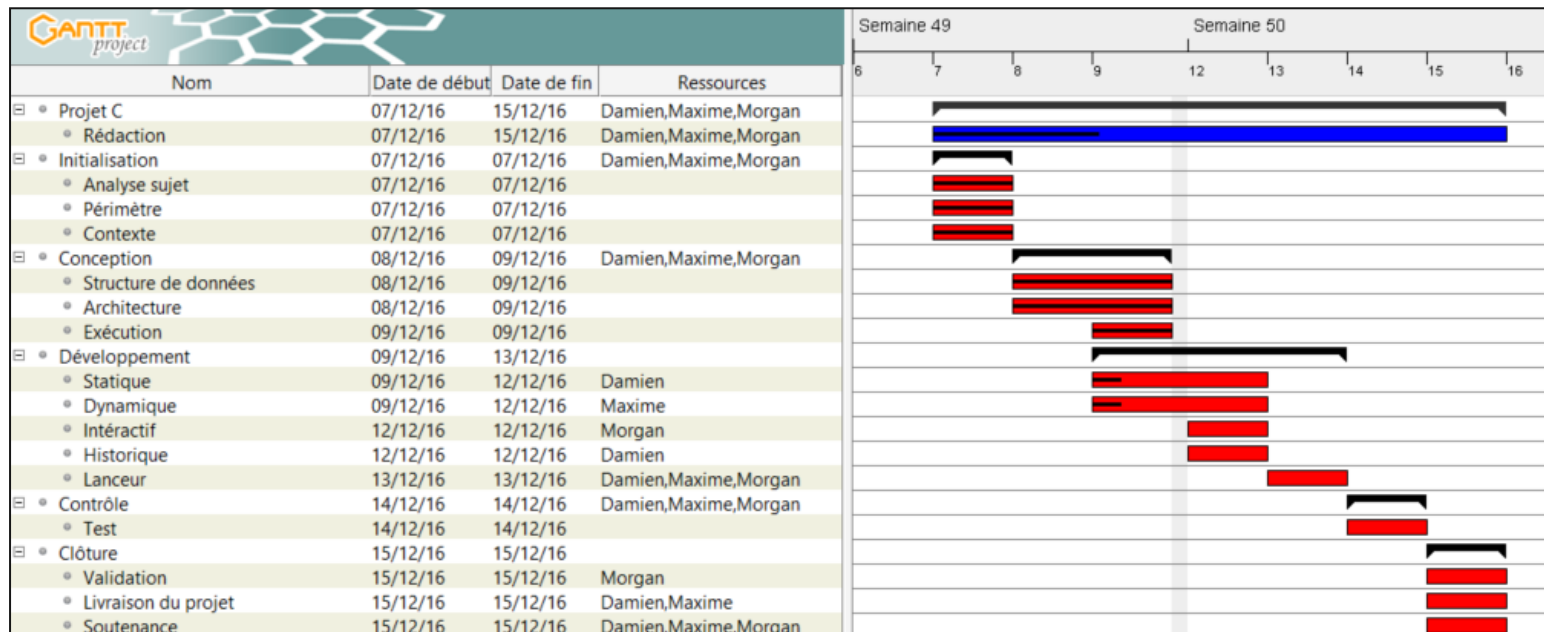
PROJET – PROGRAMMATION SYSTEME

FEUILLE D'AVANCEMENT DU PROJET

Quatrième partie : REPARTITION DES TACHES



Planning prévisionnel



Planning réel 09/12/2016 14h44



Sur le planning réel, la marge de progression des tâches correspond au trait noir qui traverse de gauche à droite le rectangle rouge et bleu.

Vous pouvez donc constater que nous avons une demi-journée de retard, car j'avais estimé que la partie initialisation serait terminée hier. Hors nous l'avons terminée en tout début d'après-midi. Ce temps perdu sera à rattraper durant le week-end, en travail personnel, sur la partie développement.

