# Rapport de projet

Table des matières

[Rapport de projet 1](#_Toc468866078)

[Présentation du groupe 2](#_Toc468866079)

[Reformulation du besoin 3](#_Toc468866080)

[Planification 3](#_Toc468866081)

[Organisation globale du programme 3](#_Toc468866082)

# Présentation du groupe

Bah y a moi et Mandel.

# Reformulation du besoin

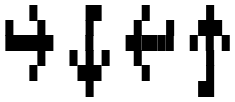
Il faut créer un écran de veille pour la console linux.

Le programme eXiaSaver doit être un exécutable qui lancera un ‘’jolie termSaver’’. Il sera lancer depuis une ligne de commande d’un terminale GNU/Linux.

Il peut être de 3 types différents : Type statique *(On affichera le contenu d’un fichier)*, type dynamique *(On affiche une information, qui se mettra à jour sans intervention de l’utilisateur)* ou de type interactif *(Apres un affichage initial, l’écran de veille demande une action de l’utilisateur, qui modifiera l’affichage)*.

**Pour le type statique,** l’affichage doit être centré. Pour se débloquer, il faut une action du clavier.

**Pour le type dynamique,** l’information doit être centré. L’information sera l’heure courante dans le format HH:MM:SS. En bas de la console se trouvera le message « Cet écran sera actualiser dans quelque seconde ». Toute les secondes on ajoutera un point à cette ligne. Le temps d’actualisation doit être paramétrable. Au bout du temps, l’horloge est actualisée et les points « remis à 0 ». Il faut que crtl+C enlève l’écran de veille.

**Pour le type interactif,** il faut représenter un avion sur l’écran :

Il a y donc 4 positions de l’avion *(6\*6)*. Il faut charger

les positions au début du programme. L’avion se déplacera selon

les inputs de l’utilisateur *(Z pour haut, S pour bas*

*Q pour gauche et D pour droite)*. S’il n’y a pas d’input, l’avion ira dans

la direction du dernier input. On peut aussi déplacer l’avion avec les flèches

et les touches du clavier numérique. L’espace aérien est la taille de la console moins la dernière ligne *(Pour la ligne de saisie) (80\*23)*. Lorsque l’avion atteint le bout de l’écran, il réapparaitra de l’autre cote de l’écran. Il faut pouvoir voir le bout de l’avion d’un côté alors que la queue est visible de l’autre cote. Quand l’avion tourne, il tourne sur un point central *(le point 3\*3 par exemple)*. On peut sortir de l’écran de veille en appuyant sur une touche prédéfinie *(E ou X par exemple)*.

Le choix d’un écran de veille doit être automatique sans l’interaction de l’utilisateur. Pour faire cela, **il faut un autre exécutable qui fera « lanceur » d’eXiaSaver.**

Il faut que cet exécutable affiche l’historique qui permettra de faire des statistiques sur les types d’écrans lancés. Il faudra lancer le programme avec l’argument « -stats ».

**Réfléchir avec attention la structure de données !!**

**C’est la chose la plus IMPORTANTE donc à choisir avec soin.**

Elle sera demandée 2 jours après le début du projet.

Pour chaque image utilisée dans les écrans de veille (image statique, tous les chiffres de 0 à 9, les avions), vous devez construire un fichier PBM ASCII avec la syntaxe suivante à respecter scrupuleusement :

Le nombre magique du format :

Il dépend du format et de la variante (binaire ou ASCII).

Un caractère d'espacement (espace, tabulation, nouvelle ligne)

La largeur de l'image (codée en caractères ASCII)

Un caractère d'espacement

La hauteur de l'image (codée en caractères ASCII)

Un caractère d'espacement

Les données binaires de l'image :

L'image est codée ligne par ligne en partant du haut

Chaque ligne est codée de gauche à droite

Toutes les lignes commençant par # sont ignorées.

Un pixel noir sera code avec un 1 et un pixel blanc avec un 0.

La lecture d’un fichier PBM doit être partagée par les 3 exécutables.

La taille des chiffres pour l’heure est de 5\*3 pour tous les chiffres.

Le séparateur de HH et MM et de MM et SS est le :

# Planification

# Organisation globale du programme