**Gras** : important

**Rouge** : mot très important ou ambigüe

**Orange**: interprétation

**Bleu** : Réponse d’interprétation (donc moins fiable)

**Vert** : interrogation sur la tournure de la phrase

(Laps de temps qui ne pouvait être modifié qu'en **recompilant** le programme !).

**Erreur à ne pas reproduire ?**

termSaver -> 3 types :

- statique

Afficher à l’écran le **contenu d’un fichier (tout type de contenu)**

**« Adapter à n’importe quelle image »**

**Centré** sur la **taille** de la console

**Au centre ou étirer ?**

Dans **la plupart des cas**, la taille de la console est de 24 lignes de 80 caractères par ligne

**Voir les autres cas ?**

rend la main au shell quand l’utilisateur **clique sur n’importe quelle touche du clavier** **(évènement ?)**.

-dynamique

Il s’agit d’afficher une information et que, de façon automatique, cette information se mette à jour **sans l’intervention de l’utilisateur**.

**On ne peut pas définir et taux de rafraichissement ? N**

Au bout de N secondes (N doit être paramétrable), réafficher la nouvelle heure courante

-Interactif

Type Interactif : après un affichage initial, l’écran de veille **demande une action à l’utilisateur le préciser par un affichage ?** . En fonction de cette action (clavier), l’affichage doit être modifié. C’est le même principe qu’un jeu interactif mais dans un termSaver. L’idée est de représenter un avion sur la console et le faire voler. L’avion sera représenté en **4 positions** de la façon suivante (tailles 5x6 ou 6x5). Pour simplifier, il est possible de prendre une seule dimension de 6x6. Il est **demandé de stocker ces 4 positions**, chacune dans un fichier **PBM** et de les charger **au démarrage** de l’écran de veille :

L’avion avance selon les commandes **de** l’utilisateur. Pour **piloter** **?** l’avion, vous devez taper une des lettres **H**(aut), **B**(as), **D**(roit) ou **G**(auche) **puis taper Entrée**. Si l’utilisateur ne rentre **aucun caractère** **entrée ?** ou s’il se **trompe de caractère** **sortie,plusieurs ?** lors de la saisie, l’avion avance avec la dernière commande donnée **si c’est la première ?**. Il est possible de piloter l’avion également avec n’importe quelles **autres touches** **lesquelles ?** (flèches, chiffres 1234, …) L’avion vole dans un « espace aérien » de la taille de la console (80x23 car on gardera la dernière ligne pour la **saisie des commandes quelles commandes ?**). Quand il arrive aux extrêmes de l’espace aérien, il doit ressortir du côté opposé. L’avion devra apparaître petit à petit **automatique/d’un coup ?** , et non d’un seul bloc. Ainsi lorsque l’avion passe sur le bord horizontal ou vertical de l’écran, le nez de l’avion doit être visible de l’autre côté, tandis que l’arrière est encore visible Quand l’avion tourne et change de direction ou de sens **déplacement 2/3 en temps ?**, **combien de casse par déplacement ?** il pivote sur son point central (le point (3,3) peut tout à fait convenir). **La position initiale sera passée en paramètre de l’écran** ? . La direction et le sens du vol doivent être choisis par le programme de façon totalement aléatoire à chaque démarrage de l’écran. Plus de précisions seront données dans les spécifications techniques.

Prévoir une sortie de l’écran de veille à tout moment en tapant un caractère bien prédéterminé (par exemple, ‘q’ , ‘x’, …) **afficher ?**

Le choix d’un des écrans de veille doit se faire de façon **automatique** **aléatoire ?** et sans l’interaction de l’utilisateur. Pour cela, il est demandé de créer **un autre exécutable** qui fera la tâche « **lanceur** » : eXiaSaver **?** . Voir spécifications techniques pour plus de détails. eXiaSaver doit proposer également une fonctionnalité de historique qui permettra de faire de statistiques sur le **nombre de types d’écrans lancés**, sur les **dates de lancement** triées par ordre **chronologique** ou l’**inverse**. Cette fonctionnalité sera accessible en lançant **le même exécutable** **le lanceur d’exiasaver ou exiasaver?** avec le paramètre « -stats ». Voir spécifications techniques pour plus de détails sur cette fonctionnalité.

Prévoir une sortie de l’écran de veille à **tout moment** en tapant **un** caractère bien prédéterminé (par exemple, ‘q’ , ‘x’, …)

**Sans entrée ?**

L'avion réagit quand on tape sur une des 4 touches de commande

**Sans entrée ?**

les structures de données, le découpage du code et les affichages de chacune des demandes et tout se passera très bien.

**Prévoir la taille des images ?**

Horaires soutenances 13h30 14h

3 Partie écran de veilles

1 programme principal

1 historique et statistiques

1 livrable

Historique erreurs mauvaises images optionnel

Le programme doit pouvoir s’adapter aux images

Afficher une possibilité de modifier N

Afficher une possibilité de modifier