MANUEL D’UTILISATION DU LOGICIEL

EXIASAVER

1. Contexte

MATRIX International Company est une entreprise chargée de créer des écrans de veilles pour ordinateur sous Windows. Pour aider les ingénieurs développant très souvent dans un terminal, il nous est demandé de leur créer sensiblement la même chose tout en s’adaptant aux contraintes imposées par ce type d’affichage.

1. Principe

Nous devons réaliser un exécutable lançant aux hasard trois types d’écrans de veilles différents avec chacun leurs spécificités depuis une commande écrite dans le terminal Linux. Ces trois écrans doivent affichés trois éléments précis ; le premier, une image fixe centré dans la console ; le deuxième, l’heure du système qui se rafraichira toutes les n-secondes et enfin le dernier, un avion pouvant être dirigé par l’utilisateur. Il faudra également que ce lanceur puisse lancer un fichier statistique si l’on rentre un paramètre précis en même temps que la commande.

1. Besoin

Il va donc y avoir un gros travail d’adaptation, il va falloir penser tout notre programme comme étant exécutable sous une distribution Linux. Nous serons donc également dépendant de la résolution ainsi que de la taille de la console dans laquelle nous exécuterons notre écran de veille. Il faudra maitriser l’utilisation de fichiers PBM, car c’est ce type d’image qu’il nous est demandé d’afficher. Pour finir il est nécessaire de mettre en place un système de sauvegarde des données correspondant au contexte d’exécution des écrans lancés.

1. Architecture

Il a donc fallu s’organiser du mieux possible pour réussir à retranscrire toutes les fonctionnalités demandées par un tel programme, suite à quoi nous avons créé un logigramme permettant de visualiser toutes les actions exprimées de façon claire pour pouvoir nous repérer lors de la création de documents plus techniques.

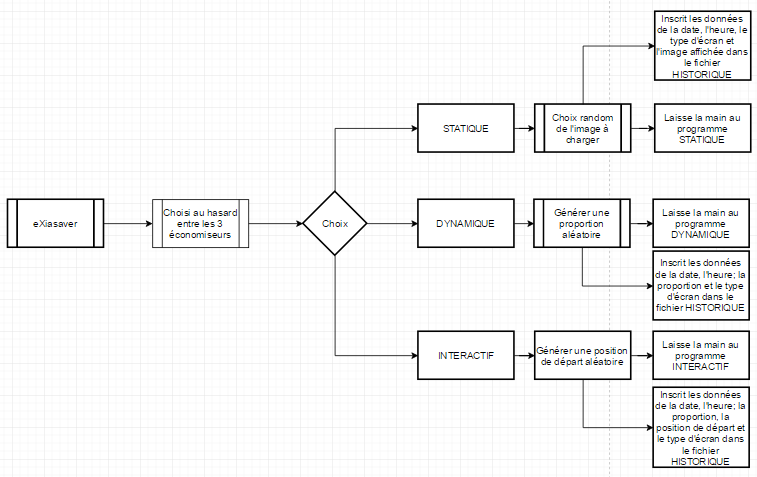


Figure Logigramme général

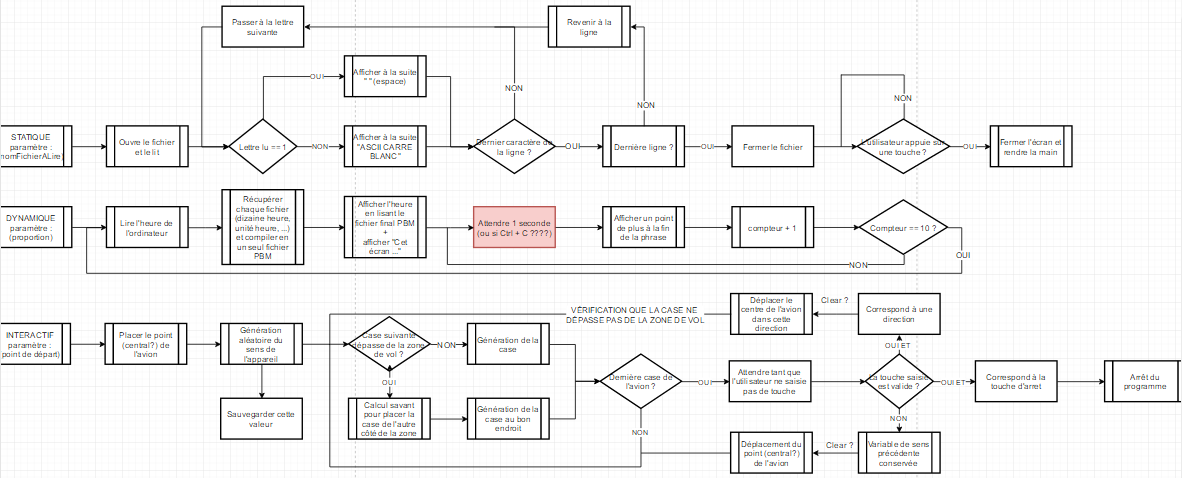


Figure Logigramme des fonctions

On voit ici de façon précise toutes les fonctionnalités et attentes du projet. L’utilisation du logigramme nous permet de mettre en avant des zones de doutes comme la case rouge de la figure 2. Ceci nous signale ainsi que c’est une fonction que nous ne maitrisons pas du tout lors la pré-production et qu’il va donc falloir faire des recherches approfondis la concernant.

1. Fonctionnement

Il était nécessaire d’expliciter plus en détails toutes les fonctions, s’y attarder de manière poussée et savoir les exprimer d’un point de vue plus technique, cela afin de simplifier la création du code un peu plus tard. Pour cela nous avons fait également un logigramme mais il est par conséquent moins digeste que celui montré plus haut. Nous nous sommes servis de Trello, afin de pouvoir découper chaque fonction et en attribuer la réalisation à l’un des membres du groupe.

1. Outils

Il a fallu utiliser des outils que nous ne maitrisions pas, notamment pour simplifier les méthodes organisationnelles, tels que Trello, GitHub, VirtualBox, Word et Code :: Block.

1. Conclusion

Au final nous avons réussi à avoir un programme fonctionnel malgré quelques écarts, surtout sur la partie interactive. Nos retours sont positifs, malgré le fait que le projet ait été trouvé relativement dur, et par certains moments nous a plongé dans une atmosphère de tension et de frustration plutôt que dans une ambiance agréable de réalisation de projet. Ce projet nous servira donc à être plus efficace pour les prochains et nous changerons probablement un peu notre méthode de travail bien que nous ayons bien été guidé par notre «tuteur» de 5ème année.