

**PROJET SCREEN SAVER**

* **Libessart Dimitri**
* **Fritsch Florian**
* **Kruger Maxence**
* **Hollebecq Maxime**

Sommaire :

[ANALYSE DU CONTEXTE 3](#_Toc469555535)

[Analyse 3](#_Toc469555536)

[Organisation 4](#_Toc469555537)

[Interprétation 6](#_Toc469555538)

[Conception Du Projet 10](#_Toc469555539)

[Introduction 10](#_Toc469555540)

[Les Contraintes 10](#_Toc469555541)

ANALYSE DU CONTEXTE

* Objectifs :
* Créer des screens saver
* Créer un exécutable d’un programme appelé « Un joli term saver » lancé depuis une ligne de commande d’un terminal GNU/LINUX.
* L’idéal est de pouvoir faire un term saver dynamique et un autre interactif
* appel d’offre : Effectuer un term saver (ecran de veille)
* Statique
* Dynamique
* Interactif

## Analyse

Dans ce second projet, nous devons réaliser un fond d’écran sous un environnement Linux. Ce dernier possède plusieurs spécifications dont les trois principales sont les types :

* Statique
* Dynamique
* Interactif

**Pour le statique :** Le but est d’afficher un écran dans lequel on a un format prédéfini et qui représentera un dessin possèdent une taille similaire à la console. Lorsque l’utilisateur appuie sur une touche quelconque du clavier, on doit pouvoir déverrouiller l’écran de veille, ce qui permet de rendre la main au Shell.

**Pour le dynamique :** Ce second type doit afficher une information de manière automatique sans aucune intervention venant de l’utilisateur. Cette information est l’heure courante qui doit être centrée dans la console (Format : HH : MM : SS). Pour l’actualisation de l’heure un message devra apparaître disant : « *Cet écran sera actualisé dans quelque secondes* ». Pour symboliser les secondes qui s’écoulent, il sera imposé d’afficher des points par seconde sans effacer ou modifier quoi que ce soit. Nous devons donc paramétrer un nombre « N » de seconde pour rafraîchir l’heure. Contrairement au type statique, pour rendre la main au Shell, il suffira d’effectuer un « Ctrl + C » qui tuera le processus sur Linux.

**Pour l’interactif :** L’écran devra demander une action du clavier par l’utilisateur**.** L’avion sera représenté sous 4 positions différentes, elle-même stockées dans un fichier PBM pour chaque position. L’avion sera commandé par l’utilisateur par le biais des touches « H(Haut), B(Bas), G(Gauche), D(Droite) et ensuite appuyer sur Entrée ». Dans le cas où, l’utilisateur se trompe de touche, la dernière commande sera ré exécutés. Il sera aussi possible de diriger l’avion avec les touches 1,2 ou 3. Cet avion volera dans la console et lorsqu’il quitte l’espace aérien, ce dernier doit réapparaitre de l’autre côté de l’écran. Lorsque l’on change de direction, celui-ci va pivoter sur son point central, le programme devra placer aléatoirement l’avion dans une certaine position. Pour quitter cet écran de veille nous devrons taper un caractère pré défini.

## Organisation

Dans un premier temps, nous avons organisé un plan de travail sur la semaine qui nous était impartie :

|  |  |
| --- | --- |
| **Nom : Kruger Maxence** | **Rôle principal :** **Chef de projet** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **7/12** | **8/12** | **9/12** | **12/12** | **13/12** | **14/12** | **15/12** | **16/12** |
| Réalisation du plan de travail |  |  |  |  |  |  |  |  |
| Réalisation du code |  |  |  |  |  |  |  |  |
| Réalisation des PBMs |  |  |  |  |  |  |  |  |
| Réalisation du Rapport |  |  |  |  |  |  |  |  |
| Réalisation de Feuille d’avancement |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nom : Fritsch Florian** | **Rôle principal : Apprenti développeur** |

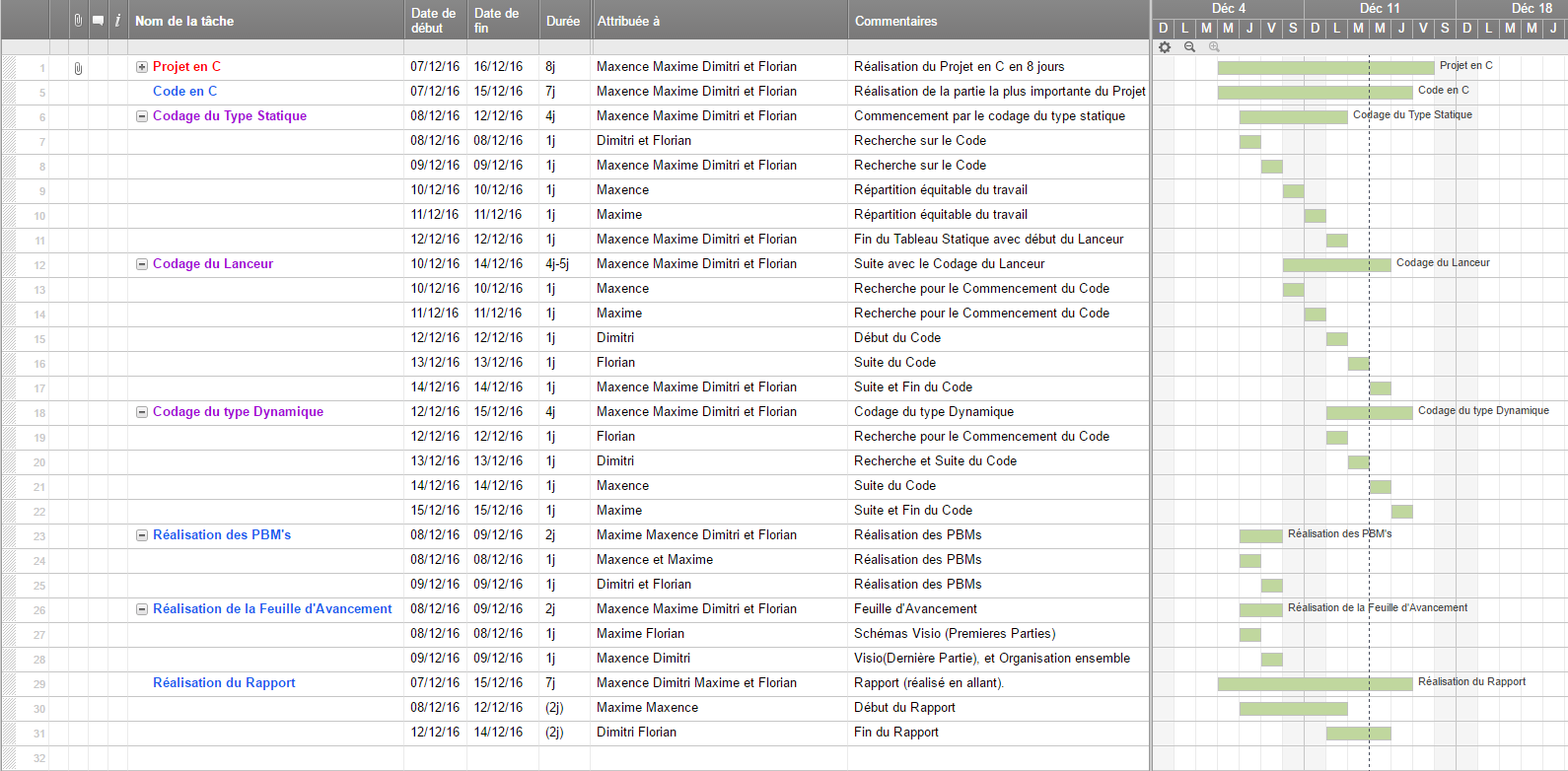
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **7/12** | **8/12** | **9/12** | **12/12** | **13/12** | **14/12** | **15/12** | **16/12** |
| Réalisation du plan de travail |  |  |  |  |  |  |  |  |
| Réalisation du code |  |  |  |  |  |  |  |  |
| Réalisation des PBMs |  |  |  |  |  |  |  |  |
| Réalisation du Rapport |  |  |  |  |  |  |  |  |
| Réalisation de Feuille d’avancement |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nom :** **Libessart Dimitri** | **Rôle principal : Apprenti développeur** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **7/12** | **8/12** | **9/12** | **12/12** | **13/12** | **14/12** | **15/12** | **16/12** |
| Réalisation du plan de travail |  |  |  |  |  |  |  |  |
| Réalisation du code |  |  |  |  |  |  |  |  |
| Réalisation des PBMs |  |  |  |  |  |  |  |  |
| Réalisation du Rapport |  |  |  |  |  |  |  |  |
| Réalisation de Feuille d’avancement |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nom : Hollebecq Maxime** | **Rôle principal : Apprenti développeur** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **7/12** | **8/12** | **9/12** | **12/12** | **13/12** | **14/12** | **15/12** | **16/12** |
| Réalisation du plan de travail |  |  |  |  |  |  |  |  |
| Réalisation du code |  |  |  |  |  |  |  |  |
| Réalisation des PBMs |  |  |  |  |  |  |  |  |
| Réalisation du Rapport |  |  |  |  |  |  |  |  |
| Réalisation de Feuille d’avancement |  |  |  |  |  |  |  |  |

Finalement, nous avons opté pour un plan, qui y ressemble mais qui comporte quelques différences et plus de détails :

## Interprétation

TermSaver statique :

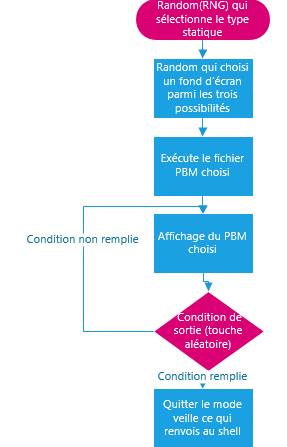
Afficher le contenu d'un fichier (dessin et format prédéfini)

Affichage centré sur la taille de la console (24X80)

Grille millimétrée par obligation

Présenter des tests important

Ecran de veille permanent sauf si l'utilisateur utilise n'importe quelle touche du clavier



TermSaver dynamique :

Afficher une info

MAJ de l'info automatiquement (pas d'intervention de l'utilisateur)

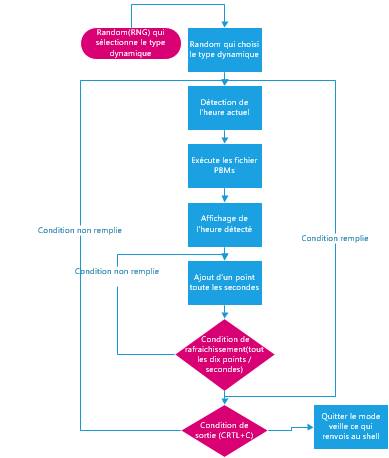
But : afficher l'heure actuelle -> "HH :MM : SS » -> HH (de 0 à 23) MM/SS (de 0 à 59)

Faire apparaitre un message en bas de la console :"cet écran sera actualisé dans quelques secondes"

Afficher un point "." toute les secondes sur la même LIGNE. -> sans effacer l'heure (heure différente du compteur) -> pas de ligne supplémentaire

Réafficher la nouvelle heure courante au bout de « n » secondes (N -> paramétrable)

Rendre la main au Shell si on tue le processus (CTRL+C)



TermSaver interactif :

Affichage initial -> termsaver demande une action (clavier) à l'utilisateur

Action -> change l'affichage

Représenter un avion et le faire voler

Avion -> 4 positions (taille 5x6, 6x5) ou 6x6 pour simplifier (carré)

Stocker les positions dans un fichier PBM (fichiers graphiques) <- à charger au démarrage de l'écran de veille

Avion se déplace selon les commandes utilisateurs (HBDG, FLECHES, chiffres 1234)

Espace aérien = 80X23 (dernière ligne pour la saisie de commande)

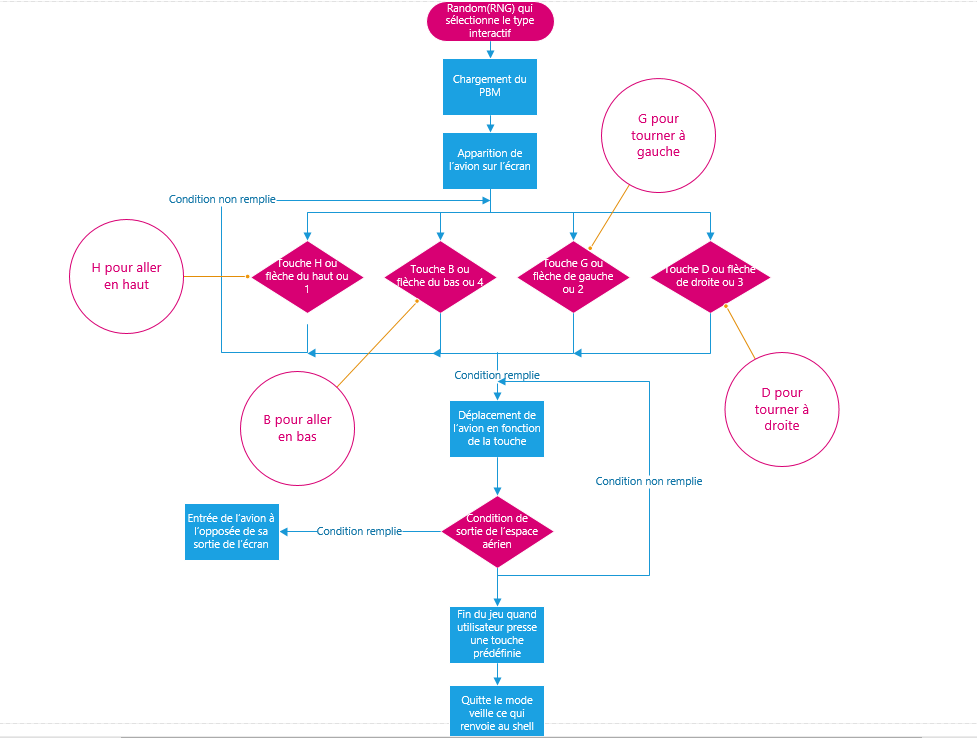
Condition : si l'avion dépasse l'extrémité -> sort par le côté opposé progressivement

L'avion pivote sur le point central (3,3)

Position initiale : paramètre de l'écran

Direction / sens -> aléatoire à chaque démarrage.

Sortie de l'écran -> caractère prédéterminé



Écran de veille :

Choix d'un écran de veille de façon automatique <- créer un exécutable "lanceur*".*

Pistes de solution :

|  |  |  |
| --- | --- | --- |
| **Statique** | **Dynamique** | **Interactif** |
| **1 fichier par dessin (PBM)**  **Fichier contenu dans le répertoire "veille"**  **Code : random et execution**  **Commande de sortie (touche clavier)** | **1 PBM par chiffre 0 à 9 + « : »**  **Code :**  **Les points (LIMITES. DELAY.) -> boucle tant que + message en bas**  **Faire des appel (fichiers) au PBM dans le code et centrée**  **Actualiser l'heure tous les N point**  **"" retour au shell" (automatique lorsqu'on ctrl+c ???)** | **Dimension avion : 6x6**  **Taille de la map : 80x23-> PBM ?**  **4 PBM : 1 par orientation**  **Changement de PBM selon la touche utilisée -> If**  **Limite : si dépassée retour à l'opposée**  **Coordonnée avec PBM, on délimite l'avion (6x6) et on attribue +1/-1 au coordonnée** |

Conception Du Projet

## Introduction

Après l’analyse du sujet, Maxime, Dimitri, Florian et Maxence êtes lancés dans la réalisation du projet. Chaque travail a été réparti équitablement par personne. De ce fait, chaque personne a touché à peu près à toutes les tâches possibles :

* Dimitri s’est occupé particulièrement du code du lanceur et m’a accompagné dans la tâche de la feuille d’avancement.
* Florian s’est occupé du code du type statique et du type dynamique accompagné de Maxime de Maxence
* Maxime s’est occupé de la réalisation des PBMs, du code de l’historique et du type statique
* Maxence, chef de projet, s’est occupé du code du statique, de la feuille d’avancement et particulièrement dans la réalisation de ce rapport de projet.

## Les Contraintes

Dans l’ensemble le projet est intéressant, cependant beaucoup de contraintes ont été un frein dans l’avancement de ce projet :

* L’historique a posé un problème au début de la réalisation, et était assez long à réaliser.
* Les variables d’environnements ont posé problème puisqu’elles doivent être appelés directement et donc, il ne faut pas les appeler par le biais d’un chemin d’accès de type « /home/user/EXIASAVER\_HOME/Statique1.pbm ».
* La contrainte du « Fork » était difficile à interposer dans le programme, il a fallu s’y adapter pour notre part.
* Cette avant dernière contrainte n’est pas des moindres, puisqu’il s’agit du centrage de l’image dans la console puisqu’on dispose de peu d’information sur le net et il a fallu réfléchir pendant quelque temps (jours) pour comprendre comment réaliser cela.
* Enfin, la dernière contrainte est sans doute celle du temps, puisqu’il a été difficile de réaliser ce projet dans les temps car beaucoup de recherches ont dû être nécessaires pour mener à bien ce projet.