

ExiaSaver

Rapport Projet

**ARNOL Maxence - DESCAMPS Anthony - BOOGAERTS Paul - LELEUX Charles**

Table des matières

[I. Contexte 2](#_Toc469530533)

[II. Qu’est-ce qu’un ScreenSaver ? 2](#_Toc469530539)

[III. Les 3 types de ScreenSaver demandés : 3](#_Toc469530545)

[IV. Organisation 3](#_Toc469530552)

[A) Début du projet 3](#_Toc469530553)

[B) Travail partagé 4](#_Toc469530558)

[V. Planning 4](#_Toc469530565)

[VI. Résultat 6](#_Toc469530568)

[VII. Contraintes 7](#_Toc469530569)

[VIII. Bilan 7](#_Toc469530574)

[A) Groupe 7](#_Toc469530575)

[B) Individuel 7](#_Toc469530577)

1. **Contexte**

Le contexte de ce projet est de pouvoir créer plusieurs écrans de veille :

* Un écran de veille statique
* Un écran de veille dynamique
* Un écran de veille interactif

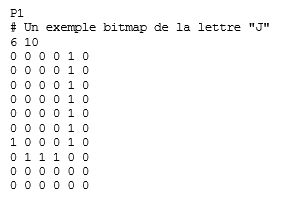
Pour permettre de lancer ces différents écrans de veille, nous avons dû créer un lanceur qui permet de choisir aléatoirement le type d’écran de veille puis d’exécuter un des types. Suite à cela, le type d’écran choisi exécutera sa fonction respective

1. **Qu’est-ce qu’un ScreenSaver ?**

Un screensaver est un écran qui s’affiche pendant un temps d’activité interrompu, on l’appelle l’écran de veille. Ces types d’écran sont, dans notre cas composé avec des fichiers PBM (sans couleurs) ou PPM (avec couleurs).

Un fichier en format PBM est un fichier réalisé en binaire, ce qui veut dire qu’un dessin va être fait avec les chiffres 1 et 0. Le chiffre 1 représente un carré tandis que 0 représente un carré blanc. Il a un nombre magique de « P1 ». Une ligne ne doit pas dépasser 70 caractères.

Voici un exemple d’un fichier PBM avec une image représentée en binaire :



Alors qu’un fichier en format PPM est un fichier codé en ASCII, il permet donc de réaliser des images en couleur. Chaque pixel qui constitue l’image est codé par trois valeur : le rouge, le vert et le bleu. Par exemple, si on veut coder un pixel en couleur rouge, il va falloir lui donner comme valeur 255 0 0. De ce fait, on va utiliser les 3 couleurs primaires pour créer les autres couleurs.

1. **Les 3 types de ScreenSaver demandés :**

**Le type statique** :

* Ce type d’écran de veille consiste à mettre une image sans mouvement, c’est que l’on appelle statique. Un des fichiers PBM sera pris aléatoirement pendant l’exécution du programme et sera afficher à l’écran.

**Le type dynamique** :

* Ce type d’écran permet d’afficher l’heure dans un délai de temps choisi par l’utilisateur, une phrase sera affichée en montrant le nombre de seconde qu’il faut attendre avant le rafraichissement de l’écran, puis l’heure sera affichée de nouveau. Ce type de ScreenSaver se met à jour automatiquement.

**Le type interactif** :

* Ce type d’écran permet de diriger un avion grâce à des touches du clavier prédéfini comme les flèches ou encore Z/Q/S/D. Cet écran de veille attendra de recevoir un « ordre » par l’utilisateur avant de pouvoir changer de commande. Ce type de ScreenSaver se met également à jour automatiquement.

1. **Organisation**
2. Début du projet

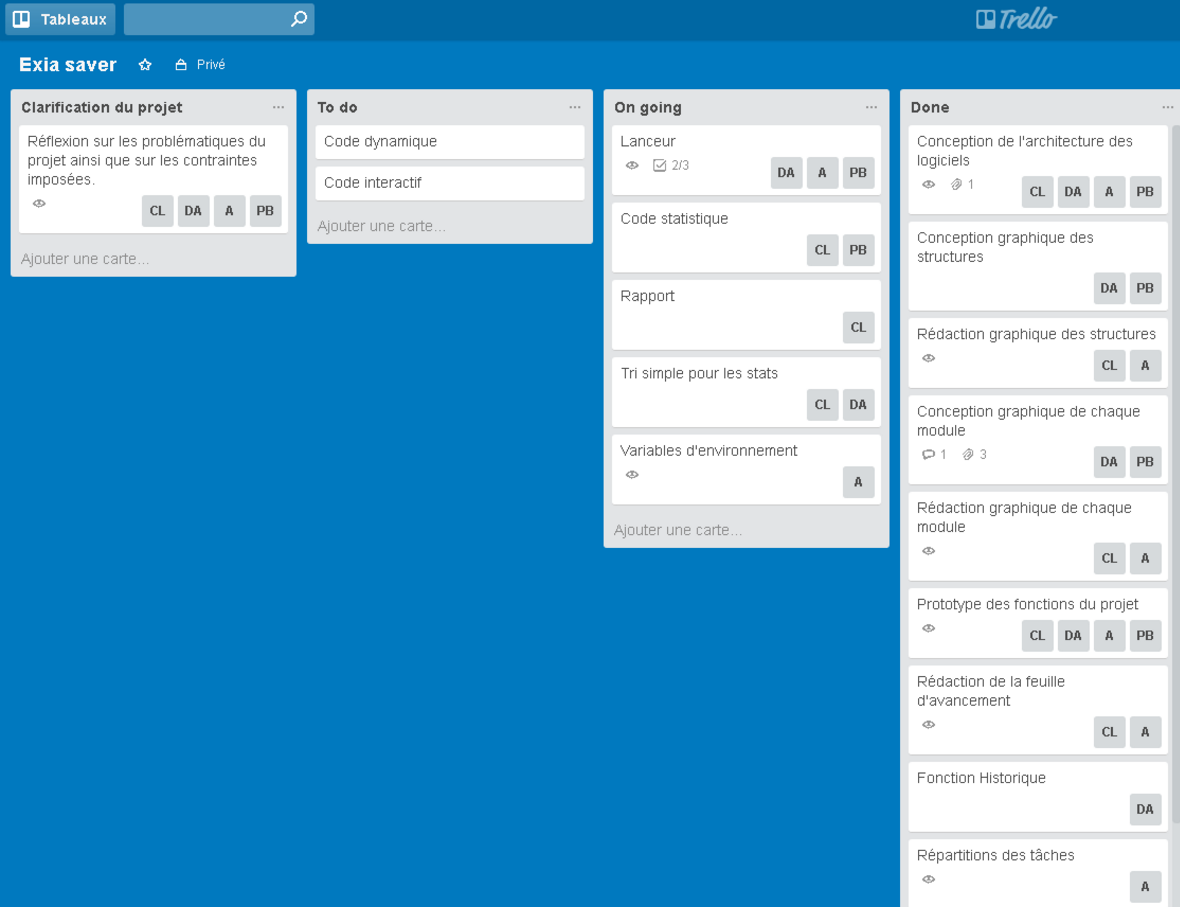
* ­Dès le début de ce projet, nous avons de suite lut l’intégralité du projet demandé puis regardé le fichier Excel fournit pour voir les points importants à remplir lors de la conception.
* Nous avons réalisé plusieurs documents résumant toutes les choses importantes à utiliser que nous avons mis sur un écran pouvant être vu par tout le groupe. Ceci nous a permis de na pas oublier toutes les choses importantes à réaliser.
* Nous avons dû remplir une fiche d’avancement pour pouvoir avancer sur le projet et nous aider à avoir une piste de « départ ». Une fois cette fiche remplie de logigramme et autre éléments demandés, nous l’avons envoyé le vendredi 9 décembre 2016.
* Nous avons créé un GitHub pour déposer toutes nos avancés afin de pouvoir assembler petit à petit notre code. GitHub nous a aussi servi pour déposer tous les livrables qui été demandés le jeudi 15 décembre.

1. Travail partagé

* Après avoir réalisé les étapes précédentes, nous nous sommes répartie différentes tâches, chacun a eu quelque chose à réaliser pour permettre de faire avancer le projet.
* Nous avons diversifié les tâches pour permettre au groupe de pouvoir réaliser des choses différentes et non de toujours faire la même chose.
* Pour permettre d’avancer avec un rythme constant, nous avons travaillé chacun de notre côté puis nous avons fait de petit temps de regroupement pour mettre en commun nos avancements et également pour expliquer à chacun nos codes et autres.
* Des groupes se sont formés pour permettre de construire les codes nécessaires au fonctionnement des différents types d’écran de veille et permettre d’avancer en ayant des résultats plus rapidement.
* L’application « trello » nous a fortement aidé pour savoir où le groupe en était sur le projet, en effet, cette application permet de savoir ce qu’il nous reste à faire, ce qui a été effectué et nous pouvons mettre aussi tout ce que nous avons réalisé afin que tout le monde puisse voir, prendre, continuer et améliorer le contenu déposé.

1. **Planning**

Voici comment nous arrivions à nous organiser pendant tout le long du projet. Nous avons tout défini ce qu’il fallait effectuer dans la section « To do ». Ensuite, nous avons fait un petit planning pour savoir quel jour nous allons les réaliser. Lorsque nous commençons à créer nous passons la case de « to do » à « on going ». Dans cette section, nous pouvons mettre ce que nous avons créé pour que les autres personnes du groupe puissent y avoir accès. Lorsque nous avons fini ce que nous devons créer, nous passons de la case « on going » à « done ». Dans cette section se trouve ce que nous avons fait et terminer. Cette application est très pratique pour se retrouver puisque ce projet contenait beaucoup de choses à réaliser.



1. **Résultat**

Suite à notre projet, les résultats que nous avons réalisés sont :

* Un lanceur qui choisit aléatoirement un type d’écran de veille, puis qui le lance avec les paramètres respectifs à celui-ci, il choisit aléatoirement l’une des 5 images du type statique si celui-ci est choisi par le premier random. Il contient aussi un menu -stat avec plusieurs choix et un historique.

2 écrans de veille fonctionnel :

**Ecran de veille statique :**

* Contient 5 images de taille différente.
* Les images sont centrées dans le terminal
* Ils sont chargés correctement
* Ils sont choisis aléatoirement

**Ecran de veille intéractif :**

* L’avion est centré dans le terminal au lancement du programme
* L’avion à une position aléatoire lors du lancement du programme (position haut, bas, gauche, droite).
* L’avion peut être commandé par l’utilisateur grâce au touche « Z » (haut), « Q » (gauche), « S » (bas), « D » (droite).
* Quand l’avion arrive à la limite de l’écran du terminal, celui-ci réapparaît de l’autre côté.

1. **Contraintes**

* Lors de la réalisation de ce projet, nous avons rencontré plusieurs difficultés qui nous ont un peu retardé dans l’avancement de celui-ci, notamment au niveau du code, et de la fiche d’avancement car c’était la première fois que l’on devait en remplir une.
* De plus, le projet était assez conséquent, de ce fait le niveau de travail à fournir est important en sachant que c’est notre premier « gros » projet.
* Nous avons perdu beaucoup de temps pour créer les fonctions nécessaires dans les types d’écrans car il y avait quelques erreurs qui nous empêchaient de continuer d’avancer.
* Le lanceur contenait beaucoup de spécifications techniques, nous avons eu beaucoup de difficulté sur certaines d’entre elle comme le random qui choisit une des 5 images pour l’écran statique.

1. **Bilan**
2. Groupe

Ce projet était notre premier « gros » projet, il a pu nous apprendre beaucoup de chose. Tout d’abord, le fait de bien commencer en clarifiant tout le sujet et définir tout ce qu’il faut faire. Mais aussi de bien se répartir le travail afin de ne pas être débordé. Nous avons pu mettre en action ce que nous avons appris dans les Prosit. Nous sommes contents d’avoir fait ce projet même si un peu plus de temps n’aurait pas été de refus.

1. Individuel

Paul : ce projet m’a permis d’appliquer un ensemble de connaissance, et d’apprendre le fonctionnement de la programmation en équipe.

Anthony : Ce projet a pu conclure tout ce que j’ai appris dans les Prosits. Il m’a aussi fait découvrir le travail en groupe et comment se répartir les tâches équitablement afin de ne pas être débordé. Je suis très content d’avoir fait ce projet car j’ai travaillé avec un groupe motivé et voulant réussir à tout prix ce projet.

Maxence : Ce projet m’a permis de gérer une équipe et de travailler avec des contraintes (techniques et temporelles). De plus j’ai pu revoir et appliqué ce que j’avais appris en prosit. Grâce à ce projet je suis maintenant plus à l’aise avec Linux.

Charles : Pour ma part, j’ai vraiment aimé réaliser ce projet avec mon groupe car c’était quelque chose d’assez conséquent à faire mais la répartition des tâches a été effectuer rapidement, ce qui nous a permis de démarrer ce projet assez vite. Le groupe dans lequel j’étais été motivé et avait l’envie de réussir.