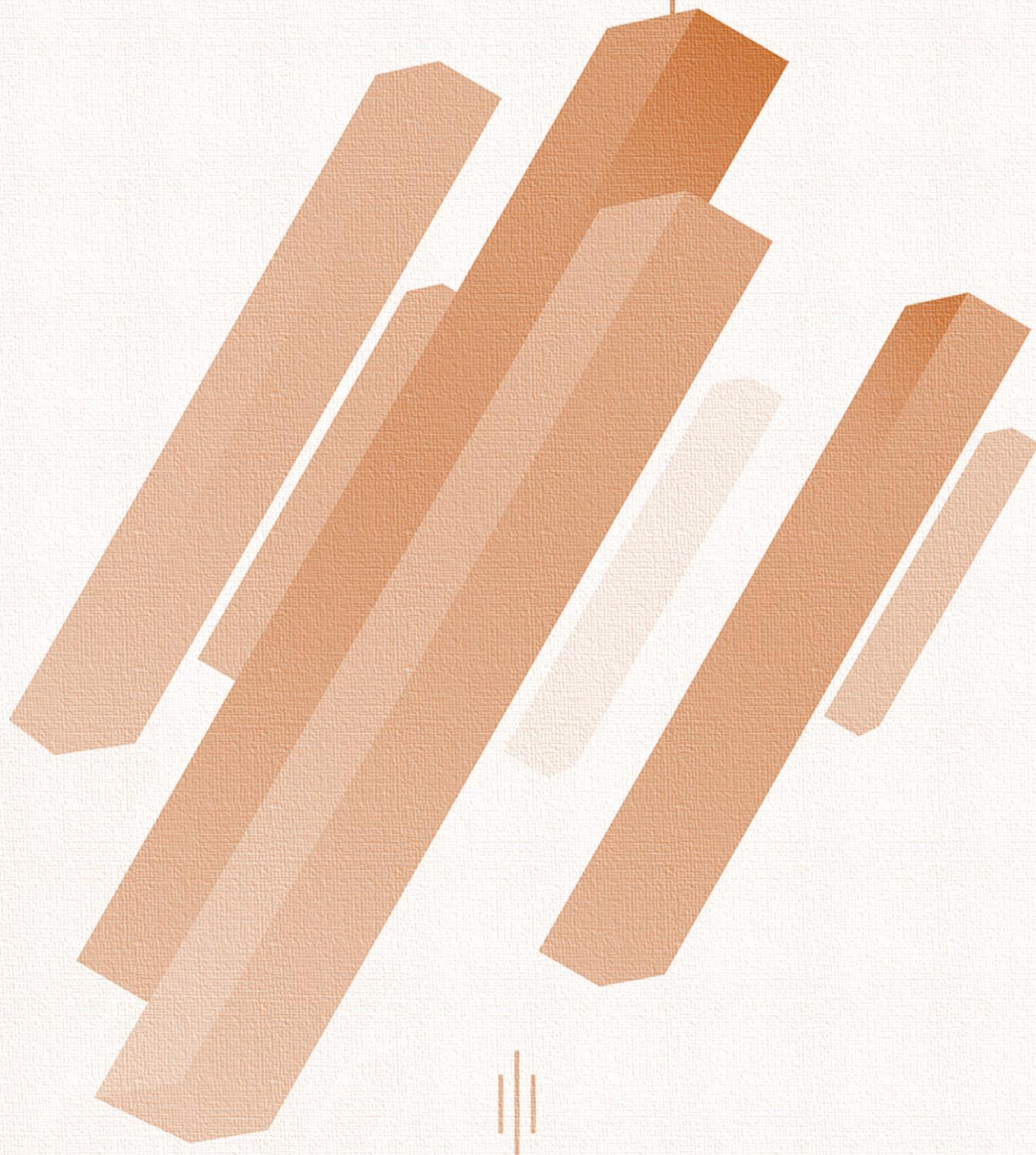


PROJET SCREEN SAVER

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



Sommaire :

ANALYSE DU CONTEXTE	3
Analyse.....	3
Organisation.....	4
Interprétation	6
Conception Du Projet	10
Introduction	10
Les Contraintes	10
Bilan	11
Bilan de groupe	11
Bilans personnels	11

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 possédant 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 réapparaître 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 devons 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	X	X						
Réalisation du code			X	X	X			
Réalisation des PBMs		X			X			
Réalisation du Rapport			X			X	X	
Réalisation de Feuille d'avancement		X	X					

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	X			X				
Réalisation du code		X	X	X	X			
Réalisation des PBMs			X		X			
Réalisation du Rapport						X	X	

Réalisation de Feuille d'avancement		X						
-------------------------------------	--	---	--	--	--	--	--	--

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	X			X				
Réalisation du code		X	X	X	X			
Réalisation des PBMs			X	X				
Réalisation du Rapport						X	X	
Réalisation de Feuille d'avancement		X						

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	X			X				
Réalisation du code			X	X	X			
Réalisation des PBMs		X			X			
Réalisation du Rapport						X	X	
Réalisation de Feuille d'avancement	X	X						

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

[illegible]

Interprétation

TERMSAVER STATIQUE :

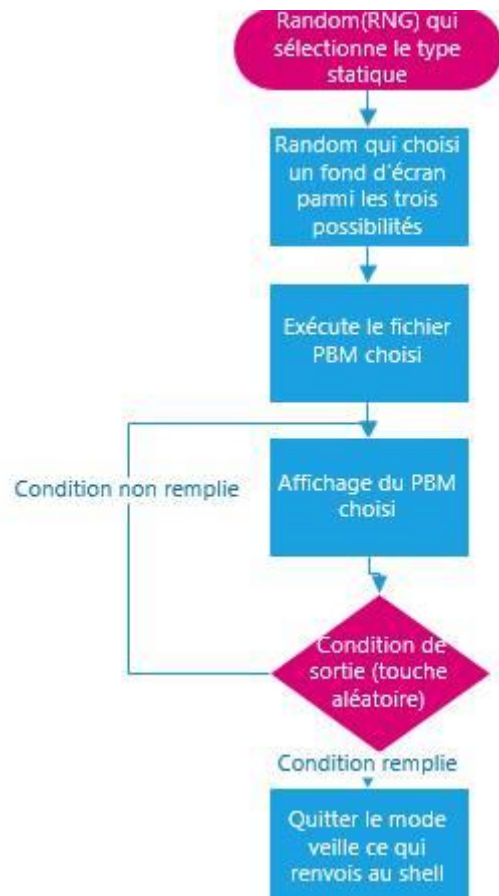
Afficher le contenu d'un fichier (dessin et format prédéfini)

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)

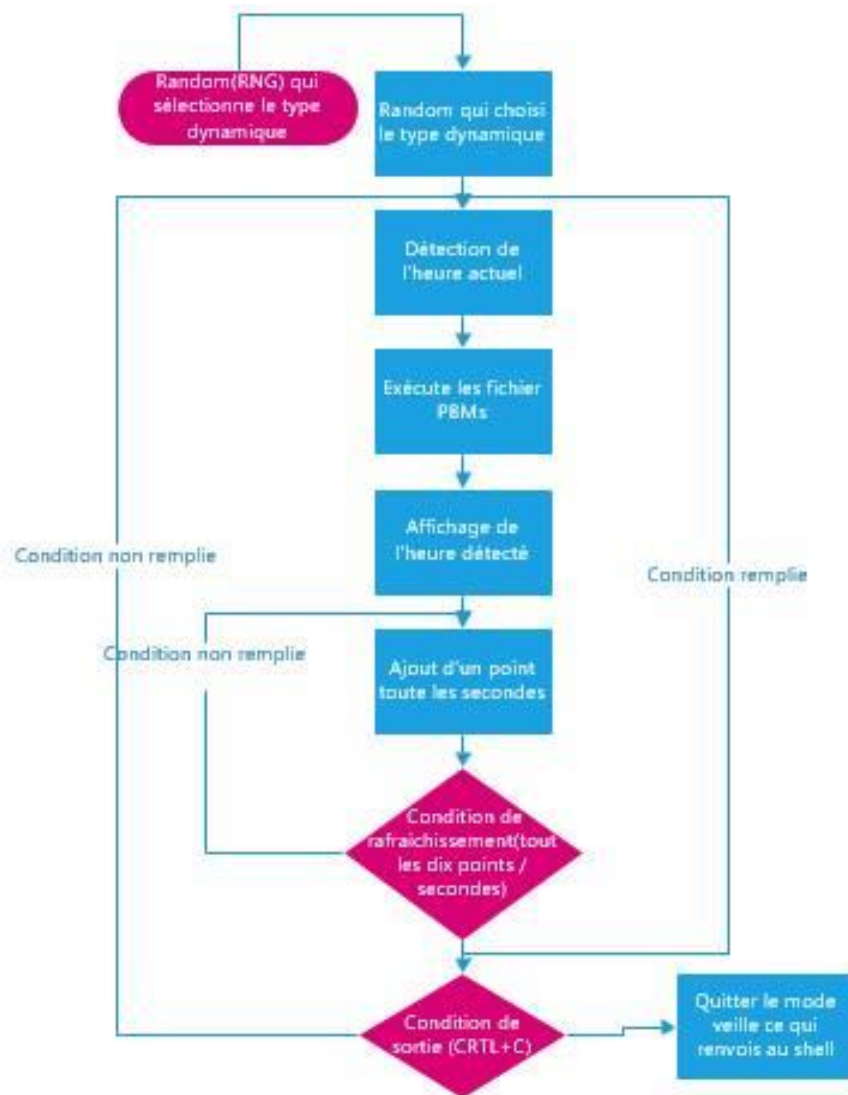
But : afficher l'heure actuelle -> "HH :MM : SS » -> HH (de 0 à 23) MM/SS (de 0 à 59)

Faire apparaître 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)

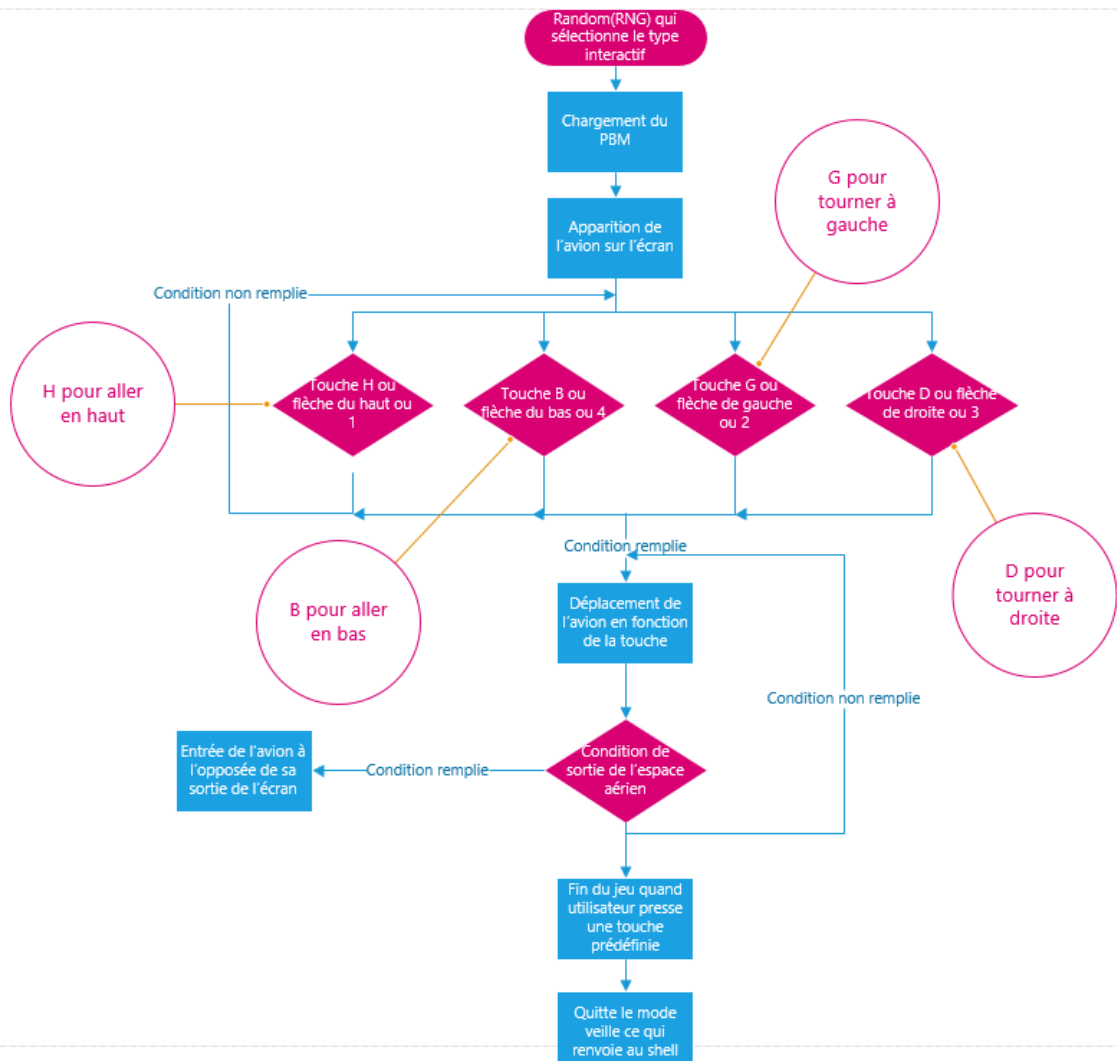
Condition : si l'avion dépasse l'extrémité -> sort par le côté opposé progressivement

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 ECRAN DE VEILLE DE FAÇON AUTOMATIQUE <- CREER UN exécutable "LANCEUR".

PISTES DE SOLUTION :

<u>STATIQUE</u>	<u>DYNAMIQUE</u>	<u>INTERACTIF</u>
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.

Bilan

Bilan de groupe

Dans l'ensemble, le projet était long mais intéressant.

De nombreux problèmes ont cependant été rencontrés durant le projet car le temps imparti nous a paru court. Cependant, le travail fourni par l'ensemble des personnes du groupe était plus que raisonnable, les tâches étaient bien équilibrées par membre.

Bilans personnels

Maxence.K : j'ai trouvé le projet intéressant, mais le temps que l'on a eu était très court à mes yeux puisque la difficulté de l'épreuve m'a paru assez élevée bien que ce projet m'a semblé réalisable. De mon point de vu, l'ensemble du groupe s'est donné à fond et l'organisation était bien faite, bien que nous ayons eu quelques difficultés tout au long de ce projet. Les points positifs sont que tout le monde a essayé de toucher à toutes les tâches possibles et que nos connaissances en C se sont accrues durant ce projet.

Maxime : Ce projet, malgré les contraintes et la restriction de temps, était très intéressant sur le point développement car il m'a permis de mettre en applications les commandes et fonction apprises lors de nos workshop/profit. Il m'a également permis de mieux comprendre le langage C de façon plus précise.

Florian : J'ai trouvé que nous avons eu un manque de temps conséquent. Ce projet était plus difficile que le premier. J'ai perdu pas mal de temps en étant bloqué sur certains points, et nous avons eu beaucoup de mal à trouver une solution. Ce projet m'a quand même permis d'exercer le langage c et d'avoir travaillé sur Linux.

Dimitri : pour ce second projet, le sujet était plutôt attractif et sympathique dans la lecture du sujet et dans sa réalisation. Pour ma part, lors de la réalisation nous avons eu plusieurs problèmes qui n'étaient pas prévus comme des erreurs alors que le code fonctionné la veille, le temps aussi a été un petit problème . Dans l'ensemble le groupe était de niveau équitable j'ai bien aimé travailler avec eux, il y a eu une bonne entente mais peut-être un manque de communication de temps en temps