

RAPPORT PROJET POO

SOMON BASTIAN

ALEXANDRE DYLAN

GROUPE N°123

[HTTPS://GITHUB.COM/BASTIANSMN/COCHONOU-ADVENTURES](https://github.com/Bastiansmn/cochonou-adventures)

Préparé par SOMON Bastian et ALEXANDRE Dylan,

Réalisé à l'aide de l'Université Paris Diderot.

SOMMAIRE DU RAPPORT

Voici les différents points du rapport qui expliquent la globalité de l'expérience

1 / PRINCIPE DU JEU

Explications générales du fonctionnement du jeu

2 / INTERFACES

Fonctionnement du jeu lors de l'utilisation de ce dernier avec un terminal ou depuis une fenêtre.

3 / PROBLEMES RENCONTRES

Les épreuves que nous avons dû surmonter en trouvant des solutions, ou un nouveau procédé afin d'arriver au résultat souhaité, qui a pour objectif de favoriser une structure logique et claire.

4 / TOUCHE PERSONNELLE

Ce que nous avons apporté au jeu

5 / AMELIORATIONS ENVISAGEABLES

Idées avec les premiers raisonnements

6 / RETOUR PERSONNEL

Ce que nous avons tiré de ce projet

PROJET POO PARIS-DIDEROT

Nous avons dû réaliser ce projet de fin d'année pour notre enseignement en Programmation Orientée Objet qui souhaitait que nous fassions un jeu s'inspirant de l'application mobile Pet Rescue Saga.

En suivant les consignes à respecter et en ajoutant notre petit touche personnelle, nous avons pu mener à bien notre création.

Nous tenons à remercier les professeurs Cristina Sirangelo, Benjamin Lipp et Constantin Enea qui nous ont grandement guidé dans ce projet.

PRINCIPE DU JEU

COCHONOU-ADVENTURES inspiré par les créateurs de Candy Crush Saga et de Farm Heroes Saga !

Un casse-tête animalier inégalé !
Associe au moins deux blocs de même couleur pour terminer le niveau et sauver les animaux des ignobles voleurs ! Les déplacements sont limités, planifie-les donc avec soin. Tes capacités en résolution de casse-têtes vont être mises à l'épreuve pendant des heures d'amusement !

EXECUTEZ LA COMMANDE :

"JOUER <PSEUDONYME> --<INTERFACE>"

EN CHOISISANT COMME INTERFACE SOIT TEXTUAL (-T), GRAPHICAL (-G), OU ENCORE LE BOT (-B) (CF README)

- Pleins de niveaux plus complexes les uns que les autres !
- Bonus permettant de vous débloquer de situations très difficiles, ou alors nécessaires dans la conception du niveau auquel vous êtes confronté !
- Tableau des scores de votre progression avec le score obtenu, le nombre d'animaux sauvés...
- Possibilité de rejouer un niveau déjà aventure.
- Sauvegarde de votre progression afin de reprendre là où vous vous êtes arrêté.
- Testez tous les niveaux à l'aide d'un bot qui tentera de résoudre les casse-têtes à différentes vitesses !



STRUCTURE DU JEU

COCHONOU-ADVENTURES

JEU
PLATEAU
NIVEAU
GRILLE
ANIMAL/BLOC
BONUS
TERMINAL
GRAPHIQUE
MODELE/VUE

- Harmonise le niveau la grille et l'état du jeu ainsi que son évolution afin d'afficher le jeu et tout son environnement.
- Gère la globalité des niveaux et la progression du joueur sur cet ensemble de niveaux (nommé ici, plateau).
- Permet l'interaction entre le plateau et la grille afin de savoir où en est le joueur et si il peut jouer à certains niveaux et peut être pas d'autres.
- Génère le tableau du jeu où le joueur se retrouve confronté afin de le comprendre et réussir le niveau et de progresser sur le plateau (donc gagner en niveau). Il récupère le csv constitué de la reproduction du tableau où le joueur se situe et pouvoir afficher le pattern de départ du niveau.
- Ces types d'objet permettent de caractériser quels sont les objets acteurs et d'autres spectateurs. Ils permettent de pouvoir caractériser l'action à effectuer sur la grille.
- Ils peuvent être bonifiant mais aussi parfois nécessaire afin d'arriver à la fin du niveau !
- Lien entre l'utilisateur et le jeu : il est incarné en joueur (constitué d'un score, de vie...) présent dans un jeu lui-même paramétré par un plateau composé de niveaux qui pointent vers une grille. Ceci incarne le modèle.
La vue dans le terminal se présentera sous forme d'affichage déplacement gauche droite ou en grille avec des colonnes et des lignes étiquetées : le joueur entrera les coordonnées du blocs qu'il souhaite actionner, par exemple. Il y aura aussi des déplacements sur le plateau pour voir l'intégralité des niveaux disponibles mais aussi où nous en sommes.
- C'est la même chose que le terminal mais sous forme graphique et donc avec une fenêtre constituée de composants, eux-mêmes constitués de composants... Et les interactions se feront au travers de l'affichage et de l'état du jeu
- C'est l'interaction entre le modèle cité ci-dessus et les 2 interfaces traitées précédemment.

INTERFACE : TERMINAL

Ici, nous n'allons pas uniquement aborder l'aspect de l'interface graphique mais aussi celui du modèle plus en détails dans l'objectif de clarifier tous les points cruciaux de notre projet et de nos prises de décisions.

L'interface terminal a été le premier élément qui nous a permis de construire correctement notre modèle, alors il nous semble évident de joindre ces deux aspects ensemble.

Nous avons décidé qu'il serait judicieux de voir ce jeu sous forme d'un plateau lui-même constitué d'une liste de niveaux donc chaque élément de cette liste incarne une grille.

Chaque grille est constituée de plusieurs type de cases qui se répartissent en deux grandes familles : les Blocs et les Animaux. Parmi ces familles, il y a évidemment d'autres sous classes plus spécifiques incarnant des éléments plus caractéristiques : par exemple, dans les blocs certains sont déplaçables d'autres, non; certains sont des blocs de couleurs, d'autres sont des blocs obstacle... Ou encore, des cases de type Animal qui possèdent des particularités afin d'incarner par exemple, un chien, un panda...

La partie bonus a été implémentée à la suite de cette première structure. Nous avons implémenté 5 bonus.

Pour faciliter la création de niveau, nous avons opté pour un parseur de CSV. L'idée est de lire un fichier csv et entrer les valeurs correspondantes dans un tableau. Nous avons ajouter un autre type de parseur qui prend les valeurs du CSVOBJECT pour les transformer en Container[]. Ainsi on a une librairie créer par nous même qui s'adapte à notre code.

L'interface terminal se calque évidemment sur le modèle. Il fonctionne comme une suite de questions/saisies entre le terminal et l'utilisateur afin qu'il puisse intéragir dans le jeu (se déplacer sur le plateau, demander de l'aide, avoir des explications sur les règles du jeu...). Lorsque l'utilisateur est confronté à un niveau, il lui suffit de saisir les coordonnées ou alors le bonus qu'il souhaite invoquer et la position à laquelle il veut l'utiliser. Le joueur peut évidemment quitter le niveau en cours pour revenir au plateau et lancer un autre niveau si il le souhaite.



INTERFACE : GRAPHIQUE



L'interface graphique se caractérise par un objet qui lie les interactions de l'utilisateur et les actions sur le tableau qui est bien différent de celui du terminal : la fenêtre. Cette dernière est notre racine du fonctionnement de notre interface graphique. A partir de cette fenêtre, nous avons différent composant liés entre eux, mais tout ceci avec une mise en page (layout) particulière : sous forme de cartes (cardlayout). Cette structure nous semblait fondamentale afin de respecter les règles de la Programmation Orientée Objet. Cela nous a permis de découper cette vue avec toutes les classes nécessaires afin de définir de la meilleure façon celle-ci. Nous nous étions tout simplement adapté au modèle avec quelques améliorations de sorte à l'adapté aussi pour l'interface graphique. Cela nous a permis aussi de nous confronter à quelques cas particuliers de sorte à traiter toutes les éventualités en fonction des choix d'interactions de l'utilisateur. Nous avons constaté qu'il était possible d'intégrer du HTML dans les composants de notre fenêtre afin de rendre l'interface plus agréable et instinctive pour l'utilisateur. Nous avons également implémenté un plateau qui fonctionnerait même avec une très grande quantité de niveaux et que le joueur puisse y accéder dans sa totalité. Nous affichons les erreurs éventuelles que l'utilisateur peut rencontrer dans des messageDialog qui sont des méthodes de JOptionPane, c'est d'ailleurs de cette même manière que nous présentons les règles du jeu. Afin de garder un style assez épuré nous avons aussi implémenté des affichages sous forme de ToolTips comme par exemple pour le tableau des scores lorsque l'utilisateur a lancé un niveau. Pour l'utilisation des bonus, nous avons pris la décision que l'utilisateur devait préalablement sélectionner le bonus désiré puis de cliquer là où l'utilisateur souhaite l'utiliser.

PROBLEMES RENCONTRES

Ce projet semblait assez accessible au premier abord mais de nombreuses connaissances pouvaient manquer à notre savoir ce qui nous a fait buter à plusieurs reprises sur diverses points que nous allons exposer :

- Nous nous sommes dès le départ réfléchi par quel procédé allons nous passer de sorte à traiter l'affichage de nos niveaux. Nous avons décidé d'utiliser les CSV de sorte à recourir à toutes nos connaissances acquises au cours de l'année (TP de parsing...) mais aussi offrir une opportunité de traiter les niveaux de manières pertinentes mais aussi en cas d'optimisation du jeu ainsi que des ajouts de fonctionnalité à ce dernier serait plus pratique à implémenter avec ce genre de procédure sous le format CSV.
- La structure du modèle a été revue plusieurs fois afin d'obtenir la conception la plus claire et la plus fonctionnelle possible afin de couvrir au maximum le cahier des charges que nous nous étions fixés.
- Au début de la conception de l'interface graphique, un premier problème s'est exposé à nous... Nous devions réfléchir de notre organisation afin d'afficher les différents états du jeu que nous avons bien structuré avec l'approche pour l'interface terminal. Nous avons obtenu pour les cardlayout même si nous nous étions jamais confronté à ceci auparavant lors des divers TP, ou même d'une approche personnelle. Alors au travers de la documentation java et des diverses informations récupérées au contact d'une immense liste de forum.
- Nous avons réussi à instaurer ce système de sorte à ce que nous possédions un menu général constitué d'un menu, d'une partie et d'autres composants utilitaires (règlements, affichage des éléments clés du jeu type score, vie...)
- La multitude de cas à traiter a été une tâche très laborieuse afin que l'utilisateur puisse accéder à toutes les fonctionnalités utiles à son égard et sans erreur; mais nous avons su nous réunir ensemble afin de mobiliser toutes nos connaissances et de solutionner ceci le plus efficacement possible.
- Cependant le contexte que nous vivons tous à compliquer pas mal les choses et a rendu le travail en tout point plus difficile à réaliser. Avec beaucoup de recul, je constate que le bien-être psychologique est l'un des piliers permettant la réussite d'un individu dans ce qu'il entreprend.





AMELIORATIONS ENVISAGEABLES DU PROJET

DESIGN UPDGRADE

LEVEL MAKER

QUETES

- L'amélioration du design reposera sur une animation et des graphismes plus poussés pour que cela soit plus agréable aux yeux des utilisateurs et permette une customisation de l'interface ! Il serait envisageable de rajouter une boutique avec des points accumulés grâce à notre avancé sur le parcours, afin d'obtenir ces avantages !
- Le concept de LEVEL MAKER inspiré de Mario consisterait à laisser la possibilité aux joueurs de réaliser leur propre niveau et qu'il soit disponible à tous dans le cas où ce dernier ait pu finir au moins une fois son niveau.
- Le principe de quêtes ouvrirait sur beaucoup de nouvelles fonctionnalités car il octroierait à l'utilisateur des médailles en fonction de diverses paramètres. (score effectué, temps nécessaire afin de finir le niveau...)

RETOUR PERSONNEL



Cette expérience nous a apporté beaucoup sur un point de vue personnel : nous avons su nous organiser de sorte à ce que nous structurions le plus efficacement nos idées.

Nous avons dû nous familiariser avec de nouvelles techniques, de nouvelles connaissances, de nouveaux outils informatiques... Et cela a été une réelle source d'apprentissage !

Ce travail nous a permis aussi de mener à bien un projet en ayant pour objectif de remplir un cahier des charges tout en y apportant notre touche personnelle.

On a pu donc voir au travers de cette aventure un avant gout de la vie professionnelle. (avec l'utilisation de la méthode scrum et l'utilisation de GitHub pour centraliser le travail de chacun).

Nous avons énormément partagé avec d'autres camarades afin que nous progressions encore plus vite et que nous puissions aller plus loin.

Nous avons aussi pris des initiatives personnelles en nous renseignant sur des forums, sur les manuels explicatifs, la documentation java...

Nous pensons que cette expérience était autant enrichissante que difficile à vivre en cette période compliquée.

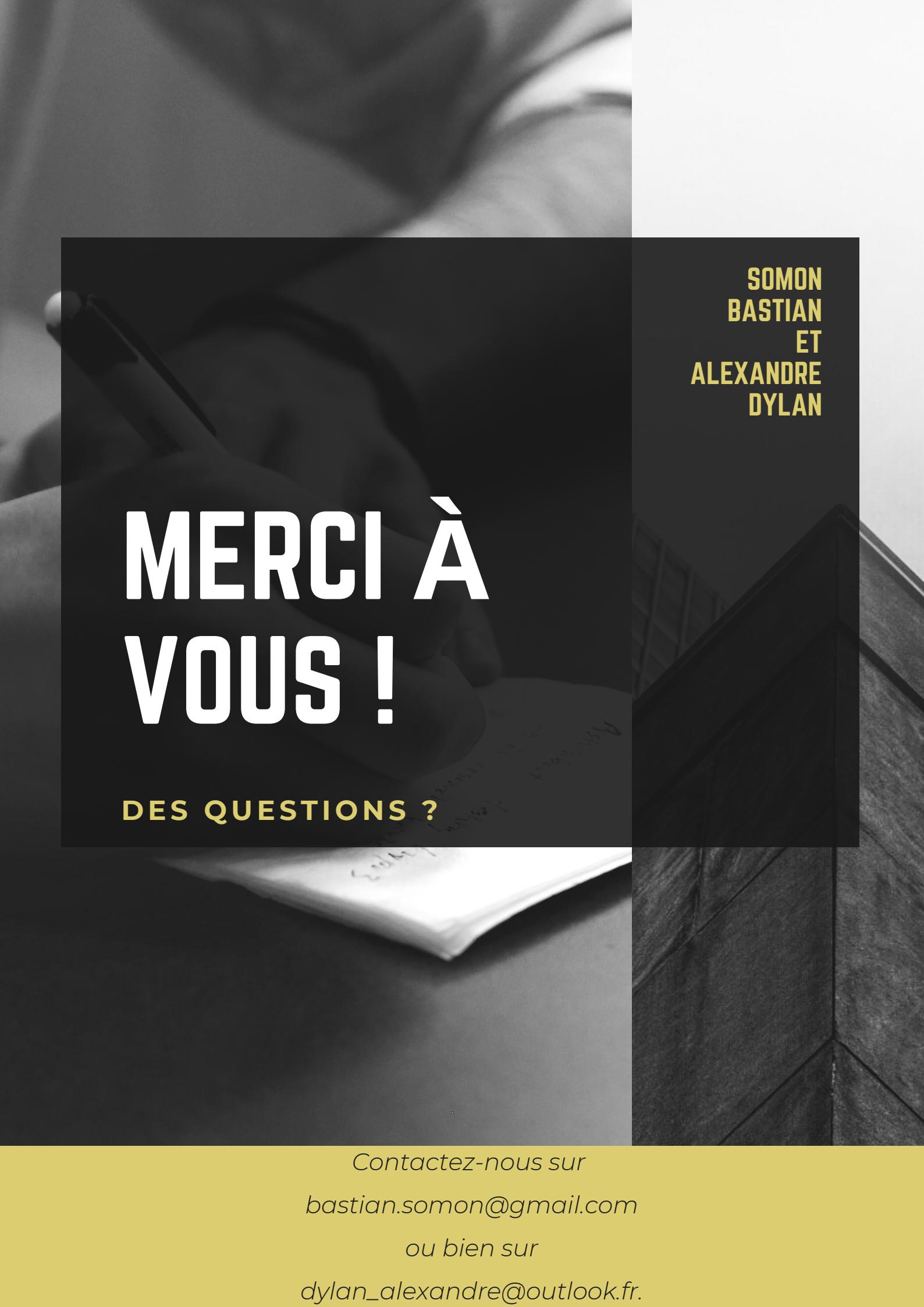
Mais le point positif que les deux ont en commun (heureusement, c'est le seul), c'est que nous sortons grandi.

Nous sommes assez satisfaits de notre travail, même si nous aurions voulu le développer d'avantage; cependant le temps nous manquait malheureusement...

Nous avons honnêtement apprécié de travailler !

Nous avons constaté qu'il y a énormément d'éléments à prendre en compte lorsque nous souhaitons structurer notre projet; il nécessite d'établir des plans solides et cohérents afin que le code mis en lumière ressorte avec facilité et clarté.

Nous espérons que notre travail vous plaira !



SOMON
BASTIAN
ET
ALEXANDRE
DYLAN

MERCI À
VOUS !

DES QUESTIONS ?

Contactez-nous sur
bastian.somon@gmail.com
ou bien sur
[dylan_alexandre@outlook.fr.](mailto:dylan_alexandre@outlook.fr)