

>~/ATILLA

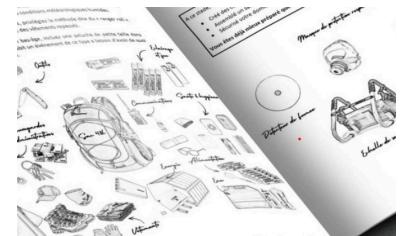
Sujet Hackathon Junior 2025

(2ème édition)

“Ce qui se passe à condorcet, reste à condorcet”

Thomas H. (2024)

“Ce matin en vous réveillant vous ne vous attendiez pas à ne rien reconnaître autour de vous, pris de panique vous fouillez votre téléphone. Messages, mails, rien ne semble sortir de l’ordinaire. Vous vous apprêtez à verrouiller votre téléphone quand sur l’écran principal vous constatez la présence d’une icône d’application que vous ne reconnaissiez pas. Curieux, vous l’ouvrez. Dès la première ligne, vous vous sentez à la fois rassuré et anxieux, vous venez de lire : **Application de survie.**”



1 - Le cadre :

Ainsi, vous l'aurez compris, d'ici vendredi soir vous devrez avoir codé une application qui devra permettre à n'importe qui de survivre dans son nouvel environnement. Entre autres il vous faudra faire :



- Un descriptif du fonctionnement du nouvel environnement.
- Une boutique d'achat en ligne.
- Une carte.
- Des notifications en cas de danger (alerte météo extrême, ...).
- Un panel de conseils divers. (optionnel)
- Un onglet paramètres. (optionnel)

Avant tout, au cours de la première heure, toutes les équipes devront définir le thème de ce nouvel environnement. Cela peut être un monde héroïque fantaisie - CY-tech - ou même un monde post-apocalyptique. Votre seule limite : votre *imagination*. **Puis le donner le plus vite possible à un membre du staff.**

Attention! Chaque thème doit être unique, de cette manière si un même thème est choisi par deux équipes différentes, la règle du “premier arrivé, premier servi” s’appliquera.

Rappel :

- Seuls les développeurs sont autorisés à coder.
- Les slides devront être réalisées par les manageurs, et présentées au jury par les développeurs.
- Cette année les manageurs ont également un rendu à fournir (voir section manager).

Have fun (^▽^)/♪



2 - développeurs :

2 - 1 Le backend :

L'architecture de votre application devra suivre le paradigme : "développement orienté objet". Bien que rien ne soit obligatoire, nous vous recommandons d'utiliser le python. Il s'agit du langage le plus facile à prendre en main, pour plus de détails, n'hésitez pas à demander à votre manager (un cours de programmation sur le tas 😊) !

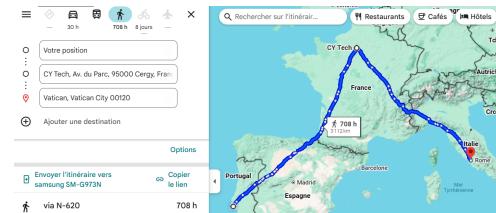
Comme annoncé plus haut, certaines fonctionnalités obligatoires sont demandées :

- **Un descriptif du fonctionnement du nouvel environnement** : Ici rien de très compliqué, cela peut être juste un texte mis en forme, ou plus si vous avez des idées.
- **Une boutique d'achat en ligne** : À cet endroit, vous devez pouvoir faire un panier d'achat, cela implique donc la gestion d'une monnaie virtuelle ou équivalent.

Attention : Ici vous devez coder un algorithme de tri qui peut ordonner les achats possibles d'au moins deux manières (par alphabet et par prix par exemple), **le tri par insertion et le tri par sélection sont interdits.**



- **Une carte** : C'est la partie la plus complexe. Vous devez avoir une vingtaine de lieux différents n'étant pas tous interconnectés et votre position géographique. Vos lieux doivent tous se trouver à des distances différentes, et ils doivent pouvoir se regrouper en plusieurs catégories à votre convenance. Parmi les fonctionnalités attendues, vous devez pouvoir rechercher un lieu précis et définir l'itinéraire le plus rapide en fonction de différents moyens de transport : le train et la voiture. Il doit y avoir des routes uniquement avec des voitures, et d'autres uniquement avec des trains. On partira du principe que les voitures roulent tout le temps à la même vitesse, pareil pour les trains (qui roulent eux plus vite que les voitures). La dernière fonctionnalité est de pouvoir forcer l'itinéraire à passer par un troisième lieu intermédiaire.



Attention : Cette partie est complètement nouvelle pour vous développeur, mais il s'agit de définir un graphe et de construire un algorithme qui permet de définir le meilleur itinéraire possible (selon le temps ou la distance par exemple).

- **Des notifications en cas de danger** : Vous devez faire en sorte qu'à n'importe quel moment dans l'application, vous puissiez avoir une notification avec un événement

qui peut être source de danger (par exemple : inondation, séisme, radiation, monstres...).

Les deux dernières fonctionnalités sont optionnelles, mais vous pouvez les tenter si vous avez déjà correctement implémenté les 4 précédentes :

- **Un panel de conseils divers** : Un onglet principal de conseils, qui regroupe plusieurs catégories (par exemple : nutrition, maladie). Chaque catégorie s'ouvre sur une liste de conseils détaillés et adaptés au thème choisi.
- **Un onglet de paramétrage** : Ici, c'est à vous de voir ce que vous voulez ou pouvez paramétrier : luminosité, langue, taille du texte...

Dans cette édition du Hackathon Junior, la dimension algorithmique occupe une place centrale. Il est donc essentiel de mettre en œuvre des structures de données adaptées aux besoins de l'application, afin de garantir une complexité optimale et des performances efficaces.

Pour rappel, il est attendu que l'application fonctionne sans bugs, ni crashes. Le jury préférera une application avec une fonctionnalité manquante à une autre qui ne puisse pas être correctement exécutée à cause de bugs. Assurez-vous aussi que l'application peut tourner sans connexion à internet.

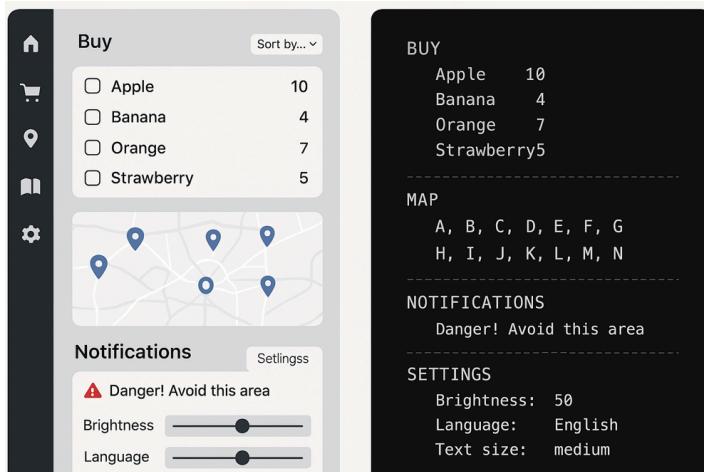
Le projet devra être déposé sur le repository de l'équipe Teams du Hackathon Junior, au sein de l'équipe qui vous sera attribuée.

Votre projet doit donc avoir un README expliquant comment tourne l'application : librairies, frameworks, comment exécuter votre programme, étapes de mise en place...

2 - 2 Le frontend

Dans cette édition nous souhaitons que vous puissiez travailler sur une interface graphique, elle représente ainsi 25% des points du projet. Cependant, les membres du jury n'attendent pas une interface graphique dite "belle". Néanmoins, elle devra être en accord avec le thème choisi. Aucun point ne porte sur l'esthétique, il suffit de présenter une interface intuitive et simple.

Exemple (non représentatif) :



Ainsi, vous devez simplement utiliser une bibliothèque graphique pour réaliser les affichages et rendus de l'application. Si vous n'avez pas le temps de développer l'interface graphique pour l'ensemble des fonctionnalités, vous pouvez proposer deux versions : l'une avec les fonctionnalités intégrées dans une interface graphique, et l'autre avec le reste des fonctionnalités accessibles via un terminal.

3 - Managers :

Dans cette édition pour votre compétition de manager, il vous est demandé de fournir un rapport et le powerpoint de l'oral. L'ensemble de ces rendus comptent pour $\frac{3}{4}$ de vos notes.

3 - 1 Le rapport :

Pour donner une idée de votre capacité de manager, nous vous demandons un rapport synthétique et efficace sur l'ensemble du hackathon junior qui doit être en corrélation avec la production des développeurs.

Ainsi il doit contenir à minima ces 5 parties :

1 - Présentation :

Dans cette partie vous devez présenter l'univers de votre guide de survie. Vous devez ensuite justifier votre choix de langue de bibliothèque et de framework (si utilisé). Enfin dans cette partie, il serait idéal de faire un bilan des compétences techniques (acquises ou utilisées) des développeurs de votre équipe liées à ce hackathon.

2 - Calendriers :

Dans tout projet, on se doit de le fragmenter le projet en tâches, de les répartir, et d'établir des deadlines. Vous devez fournir ici deux calendriers distincts que vous aurez fait pendant le Hackathon :

- Un calendrier prévisionnel présentant la répartition des tâches et les échéances envisagées, que vous aurez fourni au début du Hackathon à vos développeurs pour les aider à s'organiser.
- Un calendrier de suivi permettant d'évaluer l'état d'avancement des tâches (en retard, dans les temps, ou en avance).

3 - Schéma technique :

Dans ce projet vous devrez sans doute dessiner des schémas UML. Faites-en 3 de type différents et mettez-les dans votre rendu (correspondent évidemment à votre application).

4 - Fonctionnalités et algorithmes à implémenter :

Pour chaque fonctionnalité, vous présenterez son contexte d'utilisation en rapport avec votre thème ainsi que les choix techniques et conceptuels que vous avez effectués lors de son implémentation.

Attention : Pour les deux algorithmes demandés, vous devrez justifier votre choix (complexités, rapidité, structure de données, ...).

5 - Formations :

Vous devrez simplement mettre les notions que vous auriez pu expliquer, n'hésitez pas à mettre en annexe tous supports que vous auriez dû créer pour aider les développeurs.

Si vous voulez y ajouter un concept important non listé ci-dessus, cela pourra vous ajouter des points manquants sur certaines parties, voire vous ajouter des points bonus.

3 - 2 Le powerpoint :

Comme l'année dernière les managers devront fournir les slides pour les oraux de leurs développeurs. Il n'y a aucune contrainte sur le nombre ou le contenu, ils doivent cependant respecter des principes d'efficacité, de pertinence, et de lisibilité.

4 - Bonus :

Notez que les bonus ne peuvent vous donner que des points en plus! S'ils ne sont pas présents, vous ne serez pas pénalisé.

Depuis le début, nous nous plaçons dans la perspective d'un utilisateur. Il serait sympathique, d'y implémenter une méthode d'authentification pour plusieurs utilisateurs y compris un accès de gestionnaire pour vos collègues qui gèreront l'application après vous.

Un gestionnaire pourra modifier des éléments et de design de l'application, tout en empêchant les joueurs normaux d'y avoir accès.

Bon courage à toutes et à tous!



annexe 1 :

pour aider aux algorithmes : [visualising data structures and algorithms through animation - VisuAlgo](https://visualisingdatastructures.com/)

pour le front :

->

<https://www.bing.com/ck/a?!&&p=e7b12ca0302d9826b58b7277db4b68f0153c398e3bd72ae5e83004d88c90d988JmltdHM9MTc2NDcyMDAwMA&ptn=3&ver=2&hsh=4&fcid=14455040-fb52-65bc-2398-46ecfa35642a&psq=pygame+&u=a1aHR0cHM6Ly93d3cucHlnYW1lLm9yZy9kb2NzLw>

-> ma recommandation : <https://doc.qt.io/qtforpython-6/>

annexe 2 :

rappelle sur les UML :

<https://www.bing.com/ck/a?!&&p=ad8616da8b1c5ef0cfab002d5647d6b1b61b8397cbd96495395e59c2c04b8570JmltdHM9MTc2NDcyMDAwMA&ptn=3&ver=2&hsh=4&fclid=14455040-fb52-65bc-2398-46ecfa35642a&psq=different+diagramme+uml+&u=a1aHR0cHM6Ly9naXRtaW5kLmNvbS9mc90eXBlc1kaWFncmFtbWVzLXVtbC5odG1sIzp-OnRleHQ9UG91ciUyMHZvdXMIMjBkb25uZXIIMjB1bmUIMjB2dWUIMjBkJUyJTgwJTk5ZW5zZW1ibGUIMjByYXBpZGUIMkMIMjB2b2ljaSxsZXIMjBjb21wb3NhbnRzJTIwc29udCUyMGMIQzMIQTJibCVDMyVBOXMIMjBwb3VyJTIwZm9ybWVyJTIwbGUIMjBzeXN0JUMzJUE4bWUu>