



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à [info@trophees-nsi.fr](mailto:info@trophees-nsi.fr).

**NOM DU PROJET :**

## > PRÉSENTATION GÉNÉRALE :

- Idée et objectifs
- Origines et intérêts du projet
- (...)

*Pour commencer, ce projet est une reconstitution d'un GPS des villes de France. Ce projet s'inscrit dans l'ère du temps avec une volonté de vitesse pour éviter de perdre du temps et donc de prendre le chemin le plus court d'un point à un autre, c'est ce qui nous a poussés à choisir ce sujet. De plus, d'un point de vue plus "scolaire", cela nous permet de traiter plusieurs thématiques du programme de Terminale comme les bases de données, la complexité en temps et en mémoire ou même le fait de résoudre des problèmes qui nous semblent impossible à résoudre sans essayer. Enfin, pour notre développement personnel avec l'autonomie, le travail en collaboration ou encore sur la gestion du temps. Notre objectif était de faire un GPS des villes de France rapide et efficace.*

## > ORGANISATION DU TRAVAIL :

- Présentation de l'équipe (prénom de chaque membre et rôle dans le projet)
- Répartition des tâches
- Organisation du travail (répartition par petits groupes, fréquence de réunions, travail en dehors de l'établissement scolaire, outils/logiciels utilisés pour la communication et le partage du code, etc.)

*Notre équipe se compose de 3 personnes : Valentin Dumond, Juliette Guillemain et Nils Viollet, tous en classe de Terminale au lycée Paul Émile Victor à Champagnole. Nous nous sommes répartie le travail de manière à ce que les partie algorithmiques soit pour Nils Viollet, l'organisation et la gestion des bases de données par Juliette Guillemain et la partie graphique et "Front-end" par Valentin Dumond. Nous avons utilisé le langage Python pour l'algorithmique et les bases de données et les langages du Web (HTML, CSS et JavaScript) pour la partie graphique. Pour nous partager le travail, nous avons décidé en fonction de ce qui nous intéressait. Malgré le fait que 2 d'entres nous soit dans la même chambre d'internat, nous avons utilisé Git-Hub pour nous passer le code et SnapChat pour discuter sur des points techniques. Nous travaillions environs 4 heures par semaines sur ce projet ce qui a été suffisant pour remplir la plupart de nos objectifs.*

## LES ÉTAPES DU PROJET :

- Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)

Au début, nous avons l'idée de faire un GPS des villes du Jura, mais nous avons très vite agrandi notre vision en élargissant à toute la France. Cela décidé, nous avons dû trouver une base de données des villes française ce qui, après recherche nous a mené à 2 bases différentes : une pour avoir tout les voisins d'une ville et une autre pour avoir les coordonnées des villes avec leurs population. Nous avons ensuite implémenté l'algorithme de Dijkstra avec une file par priorité (un dériver de la file que nous avons vu en classe) ce qui nous a permis de résoudre un problème : comment calculer le chemin d'une ville à une autre. Mais rapidement nous avons trouvé la vitesse trop lente. Nous avons donc implémenter une amélioration, l'algorithme A\* ainsi que la structure de donnée "Fibonacci Heap" (sans la décroissance des clés). Pendant ce temps d'optimisation nous nous sommes aussi demandé comment avoir un rendu visuel satisfaisant car utiliser la console n'est pas très parlant quand on veut montrer un chemin. Nous nous sommes donc orienté dans le module Python « Folium » qui permet d'afficher une carte du monde dans un navigateur internet ce qui nous a poussé à utiliser un autre module : Flask. Ce module permet de créer et de gérer des pages internet avec Python.

## > FONCTIONNEMENT ET OPÉRATIONNALITÉ :

- Avancement du projet (ce qui est terminé, en cours de réalisation, reste à faire)
- Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet
- Difficultés rencontrées et solutions apportées

Dans notre projet, nous avons fait ce que nous voulions, c'est à dire un GPS fonctionnel avec une interface facile d'utilisation, une rapidité satisfaisant et un grand nombre de ville. Cela n'a évidemment pas été sans péripéties. Pour trouver les éventuelles bugs nous nous posions en nous demandant quel étaient les cas spéciaux et en faisant des tests. Malheureusement nous ne sommes pas infailibles... Mais à chaque découverte d'un bug nous l'avons identifier (en le répétant plusieurs fois) et l'avons réglé. Pour donner des exemples de bugs que nous avons rencontré il y aurait la sauvegarde impossible des pages internet que nous générions avec Folium et donc l'impossibilité d'afficher la carte avec le parcours calculer... ou encore des problèmes de jointure des bases de données avec des données inexistantes, qui disparaissait ou qui se modifiaient. La solution a été de vérifier l'intégrité de la base après la jointure et de supprimer les éléments y faisant défaut. Enfin nous avons eu des problèmes d'implémentation dû à la complexité des algorithmes ce qui c'est évidemment réglé sans réel problèmes.

## > OUVERTURE :

- Idées d'améliorations (nouvelles fonctionnalités)
- Stratégie de diffusion pour toucher un large public (faites preuve d'originalité !)
- Analyse critique du résultat (Si c'était à refaire, que changeriez-vous dans votre organisation, les fonctionnalités du projet et les choix techniques ?)

*Il y a assez facilement plusieurs possibilités d'amélioration de notre projet avec des plus grandes bases de données (une de l'Europe ou encore du monde). On pourrait aussi prendre en compte le réseau routier et non seulement les positions des différentes villes et donc d'éviter un voyage à vol d'oiseau. En stratégie de diffusion, on pourrait penser à mettre notre page web en ligne et à utiliser un serveur. On pourrait penser à un transfert vers une application mobile et une version simplifiée et qui aide à guider les personnes qui ont plus de difficultés avec les nouvelles technologies. Une commande vocale est aussi une possibilité pour les personnes qui conduisent. Si le projet était à refaire avant de se lancer dans le projet, savoir sur quoi il faut travailler pour chacun. Ce qui nous éviterait de devoir recommencer plusieurs fois la même chose car ça ne va pas. C'est ce qui est arrivé du côté de la base de données ou on a commencé à la trier en SQL mais qui ne marcherait pas avec la partie graphique.*

# DOCUMENTATION

- *Spécifications fonctionnelles (guide d'utilisation, déroulé des étapes d'exécution, description des fonctionnalités et des paramètres)*
- *Spécifications techniques (architecture, langages et bibliothèques utilisés, matériel, choix techniques, format de stockage des données, etc)*
- *Illustrations, captures d'écran, etc*

Pour notre projet nous avons utilisé certain module, 3 d'entre eux ne sont pas installé par Python automatiquement :

- folium
- flask
- math
- csv

*Pour lancer notre projet, rendez vous dans le dossier de projet et lancez le script python 'main.py', ensuite celui-ci vous donnera acces à une adresse internet local. Tout ce trouve sur cette page internet et est plutot intuitif. Si vous ne trouvez pas voici comment lancer une recherche, pour cela appuyez sur le bouton 'Recherche' en haut à droite et, après que la fenêtre ce soit ouverte entrez le noms des 2 villes dont vous voulez le trajet. Pour finir appuyez sur le bouton 'Recherche' de la fenêtre et regardez.*

Pour ce projet nous avons utilisé le langage Python avec les modules folium et flask pour afficher une carte et pouvoir avoir acces à des données de la page web. Evidement nous avons aussi utilisé les langages du web (html, css, javascript) pour faire une page internet interactive. La base de donnée est enregistré comme un fichier csv. L'algorithmique derriere est l'algorithme A\* et les structures de données sont une Fibonnacci Heap et un arbre binaire de recherche pour que la vitesse de traitement soit la plus rapide possible.