



Update README.md
Olivier Festor authored in 34 minutes

ec33ad81

README.md 7.8 KB

Projet PPII Semestre 2 · 2022-2023

Groupe

À l'instar du projet du premier semestre, les groupes librement sont constitués de 4 élèves. Les groupes doivent être remontés à Olivier.Festor@telecomnancy.eu **au plus tard le 24 mars 2023**. Comme au premier semestre, le groupe désigne une/un responsable de projet qui envoie le courriel de constitution du groupe avec TOUS les membres en copie. Dès réception et validation de la constitution de groupe, un projet sur la plateforme GitLab l'école est créée et le projet peut démarrer.

Quelques liens utiles

- le serveur Etalab <https://www.data.gouv.fr/fr/datasets/fichier-consolide-des-bornes-de-recharge-pour-vehicules-electriques/> contenant les données ouvertes sur les stations de recharge de véhicule disponibles sur le territoire français.
- un site contenant des données sur l'autonomie et la consommation de tous les véhicules électriques [Electric Vehicle Database :](https://ev-database.org/cheatsheet/range-electric-car) <https://ev-database.org/cheatsheet/range-electric-car>.
- la [page du cours de C](#) sur Arche.

Contexte

La Commission européenne a récemment voté l'interdiction de mise sur le marché de véhicules à moteurs thermiques à l'horizon 2023. Dans cette trajectoire, le développement des alternatives se poursuit et à date, les véhicules électriques ont le vent en poupe. Afin de permettre de les alimenter, des réseaux de stations de recharge se développent sur l'ensemble du territoire.

Le but de ce projet est de fournir aux usagers (conducteurs de véhicules électriques, mais également autorités de régulation et acteurs économiques), un ensemble de fonctions qui les aident dans le déploiement et le dimensionnement d'un réseau de recharge de véhicules adapté aux besoins.

Sujet

Le sujet comprend deux étapes complémentaires.

Première étape

La première consiste à construire en langage "C" un programme qui permet :

- de proposer à tout usager souhaitant se rendre d'un point A à un point B du territoire, un parcours de charge de son véhicule électrique (identifier les stations qui lui permettent d'attendre sa destination). Chaque véhicule possède des caractéristiques différentes. Vous prendrez un modèle simple : chaque véhicule à une capacité et une consommation (les données sont extractibles de la [Electric Vehicle Database](#)). On supposera qu'il existe un chemin direct entre toutes les stations de recharge.
- d'enrichir la fonction précédente avec des paramètres supplémentaires, par exemple, ne jamais laisser le véhicule en dessous d'une charge de 30%, ou si votre modèle le permet, borner les temps de recharge (l'usager ne veut jamais rester plus de 20 minutes en charge)

Afin de construire les fonctions ci-dessus, vous identifierez les structures de données qui permettent de modéliser le problème et réaliserez un état de l'art des algorithmes adaptés à la résolution du problème.

Seconde étape

La seconde étape vise à proposer sur la base des structures précédentes, un mode "simulation" à destination des investisseurs et autorités, simulateur qui va évaluer la charge du réseau en fonction d'un ensemble d'usagers fictifs (ou pas) dont le parcours est connu. Concrètement, il vous est demandé de construire en langage "C" un module de simulation qui gère un ensemble d'usagers (leur nombre, leur parcours, et les caractéristiques de leurs véhicules sont un paramètre du simulateur) et qui, sur la base des bornes de recharges calculées par les fonctions de la première partie du projet, va à chaque étape (le temps étant discrétisé par pas de 10 minutes), calculer le taux de charge des bornes du territoire, identifier celles sur lesquelles des files d'attente se créent et, si le temps le permet, proposer des chemins alternatifs aux véhicules pour éviter les files d'attente.

Toujours pour des raisons de simplicité, on pourra supposer que tous les véhicules roulent à la même vitesse sur le territoire entre deux bornes.

Les plus téméraires des groupes, pourront réaliser une fonction de visualisation des parcours proposés et de la simulation.

Rendus

Comme pour le projet du premier semestre, le rendu se fera sur le gitlab de TELECOM Nancy. Il doit comprendre les éléments suivants :

- Gestion de projets : compte-rendus de réunions + éléments de planification, de suivi analytique et d'analyse post-mortem du projet ;
- Rapport : décrivant les choix motivés des structures de données et des algorithmes (sur la base d'un état de l'art détaillé), les fonctions réalisées et leur analyse (complexité mémoire, temps), les fonctions réalisées et les tests mis en œuvre ;
- Guide utilisateur : décrivant l'installation du programme et les modalités de son utilisation ;
- Code : le code doit être compilable par un appel à la commande `make` .

Dates importantes

La soutenance du projet est prévue le 31 mai 2023. Les rapports et programmes devront être déposés sur les dépôts git des groupes au plus tard le **mercredi 24 mai 2023 à 20h00 CEST**.

Fraude, tricherie et plagiat

Ne trichez pas ! Ne copiez pas ! Ne plagiez pas ! Si vous le faites, vous serez lourdement sanctionnés. Nous ne ferons pas de distinction entre copieur et copié. Vous n'avez pas de (bonnes) raisons de copier. De même, vous ne devez pas utiliser de solution clé en main trouvée sur internet.

Par tricher, nous entendons notamment : — Rendre le travail d'un collègue en y apposant votre nom ; — Obtenir un code, une solution par un moteur de recherche (ou une IA) et la rendre sous votre nom ; — Récupérer du code et ne changer que les noms de variables et fonctions ou leur ordre avant de les présenter sous votre nom — Autoriser consciemment ou inconsciemment un collègue à s'approprier votre travail personnel. Assurez-vous notamment que votre projet et ses différentes copies locales ne soient lisibles que par vous et les membres de votre groupe.

Nous encourageons les séances de *brainstorming* et de discussion entre les élèves sur le projet. C’est une démarche naturelle et saine comme vous la rencontrerez dans votre vie professionnelle. Si les réflexions communes sont fortement recommandées, vous ne pouvez rendre que du code et des documents écrits par vous-même. Vous indiquerez notamment dans votre rapport toutes vos sources (comme les sites internet que vous auriez consultés), en indiquant brièvement ce que vous en avez retenu. Il est quasi certain que nous détectons les tricheries. En effet, les rapports et les codes sont systématiquement soumis à des outils de détection de plagiat et de copie. Il existe notamment des outils de détection de manipulation de code extraordinaire mis à disposition par l’Université de Stanford, tels que MOSS (<https://theory.stanford.edu/~aiken/moss/>) ou compare50 (<https://cs50.readthedocs.io/projects/compare50/>). De plus, chacun a son propre style de programmation et personne ne développe la même chose de la même manière.

Puisqu'il s'agit d'un projet réalisé dans le cadre de cours avancés de programmation, nous nous attendons à ce que vous soyez capable d'apprendre à déboguer des programmes par vous-même. Par exemple, demander à un autre élève de regarder directement votre code et de donner des suggestions d'amélioration commence à devenir un peu délicat au niveau éthique.

Si vous rencontrez des difficultés pour terminer une tâche, veuillez contacter l'un de vos enseignants afin que nous puissions vous aider. Nous préférons de loin passer du temps à vous aider plutôt que de traiter des cas de fraudes.