Inria Spirals

Test technique

Ce test technique vise à évaluer vos compétences en matière de génie logiciel. Il vous est demandé de nous partager, dès que possible, un dépôt GitHub qui contiendra votre réponse à ce test. Outre l’exactitude et la qualité du code développé, nous évaluerons également les tests livrés avec le code, le bon usage du dépôt GitHub et le respect de bonnes pratiques de développement.

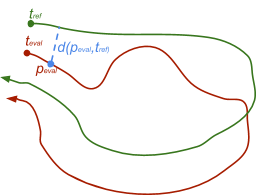
Pour tout complément d’information, ne pas hésitez à nous contacter: [spirals-recrute@inria.fr](mailto:spirals-recrute@inria.fr)

1. Comparaison de trajectoires

La première partie de ce test consiste à développer un algorithme de comparaison entre 2 trajectoires fournies en paramètre dans le format [GeoJSON](http://geojson.org/) (une trajectoire est définie comme une séquence de points, repérés par une longitude et d’une latitude).

En particulier, nous nous intéressons ici à estimer l’erreur entre une trajectoire de référence *tref* (*ground truth*) et autre trajectoire à évaluer *teval*. L’algorithme consiste donc, pour chaque point *peval* composant la trajectoire évaluée, à calculer la distance d la plus courte entre le point en question et la trajectoire de référence, d(*peval*,*tref*). Le résultat d’une estimation d’erreur consiste en une séquence de distance *e(teval*,*tref*) = [*d1*,*d2*...*dn*] pour les *n* points qui composent la trajectoire *teval*.

Un bonus sera accordé pour les réponses tenant compte de la direction de la trajectoire (cf. Figure 1) et les réponses qui utilisent la bibliothèque [Turf](http://turfjs.org/) en JavaScript ou [Spatial4J](https://projects.eclipse.org/projects/locationtech.spatial4j) en Java.

  
**Figure 1**. Illustration de la comparaison de trajectoires.

N’hésitez pas à faire la démonstration de toutes les techniques et outils de génie logiciel dont vous avez connaissance.

1. Définition d’une API REST

La seconde partie de ce test technique consiste à intégrer l’algorithme de comparaison ci-dessus dans un serveur web en proposant une API REST. En particulier, nous vous demandons de proposer une API web conforme au [style architectural REST](http://opikanoba.org/tr/fielding/rest/) en utilisant le *framework* [Eclipse Vert.x](https://vertx.io/) dans le langage qui vous convient le mieux.

Un bonus sera accordé si vous mettez à disposition le binaire de votre solution sous la forme d’un conteneur Docker optimisé et hébergé sur [Docker hub](https://hub.docker.com/). Le fichier Dockerfile associé doit être présent sur votre dépôt GitHub.