

IUT GO !

L'objectif de ce projet est de concevoir collaborativement les briques d'un réseau social basé sur la géolocalisation. Les PokeStud (qui n'aiment pas être enfermés) veulent pouvoir se déplacer à leur guise mais pouvoir retrouver leurs amis à volonté.

Principes

Chaque groupe travaille en équipes de 3 à 4 étudiants dirigées par un chef de projet transverse aux équipes. Le chef de projet a pour rôle d'identifier les points communs entre les équipes, d'aider les équipes à trouver des solutions et de préparer le travail d'intégration.

Le projet se décompose en 2 grandes phases : 1) Création des briques de bases en sous-groupes; 2) Intégration par le développement de fonctionnalités par sous-équipes au choix des étudiants.

Des codes vous seront donnés, vous devez les utiliser pour vous aider à vous intégrer au mieux :

- les codes relatifs à [OpenStreetMap](#) : ce module vous permettra de visualiser les éléments intéressants sur une carte.
- le module "[Member](#)" (correspondant au cas d'étude "Qui est-ce?" ci-dessous) qui vous donne des briques de base pour vous aider à développer vos propres modules intégrant du JSON, des tests unitaires, MAVEN. Le code source et les tests associés à ce module sont disponibles [ici](#) ...

Vos codes doivent :

1. être réalisés en utilisant un dépôt GIT (Vous serez individuellement évalués sur vos commit);
2. correspondre à des modules Maven;
3. respecter les règles de séparation de préoccupations et la qualité de vos codes est un critère d'évaluation.

Chaque module est fourni avec :

1. des tests unitaires;
2. une classe de test dédiée et qui prend une forme différente en fonction des modules.

Les fonctionnalités attendues présentent à la fois de la **valeur** par rapport à notre application cible et une **complexité** de développement. Pour vous aider, des indices vous sont données à titre indicatif, ils peuvent être modifiés lors de l'avancement du projet. Ils sont liés à la notation selon le schéma suivant.

Valeurs :

- ++ signifie très forte valeur : ne pas l'avoir fait **perdre 4pts**
- + signifie utile : ne pas l'avoir fait **perdre 2 pts**
- = signifie tant pis si on ne l'a pas

Complexité :

- \$\$ fonctionnalité difficile : **+ 4 pts**
- \$ fonctionnalité normale : **+ 2 pts**
- - fonctionnalité facile : **+ 1 pt**

Au travers des “Histoires utilisateurs (User Stories/US)”, vous devrez préciser les fonctionnalités en gérant les cas particuliers (e.g. personne non enregistrée, erreurs de coordonnées). De même les problèmes de propriétés non fonctionnelles comme le nombre d'enregistrements, les temps d'affichage doivent être exprimés sous forme de contraintes et limites. Evidemment une fonctionnalité mal construite ne vous permet d'avoir tous les points et peut vous en faire perdre.

Les étudiants peuvent demander à ajouter des fonctionnalités/histoires. Dans ce cas, il faut une validation par le “Product Owner” i.e. l'enseignant.

Première partie : Briques de base

Qui est-ce ?

Objectif : identifier de manière unique les membres du réseau.

1. Enregistrer un membre : Nom de la personne, prénom, (identifiant)
2. Récupérer un membre à partir de son identifiant
3. Enregistrer les membres dans un fichier
4. Lire un fichier contenant des membres

Ce module vous est donné: <http://sparks-vm26.i3s.unice.fr/repository/iut-public/>

1) Marauder's trace

Objectif : tracer les déplacements des PokeStud.

1. (++, \$) Enregistrer la position de X : Jour, Heure, positionGPS;
2. (+, \$) Où était X à telle Date ? On renvoie la position au plus près de la date;
3. (++, \$) Quelle est la dernière position de X ? GPS + Date
4. (=, \$) Donne-moi la suite des positions de X entre Date1 et Date2?
5. (=\$, \$) Où est X habituellement le lundi, avec quel pourcentage de probabilité ?
6. (+,\$\$) Qui sont les membres en position Y à la distance près de Z ?
7. (+,\$) Mémoriser les positions enregistrées depuis Date1
8. (+,-) Récupérer les positions enregistrées dans un fichier donné

(++, -) Voir : Console de saisie

Bilan :

- vous pouvez perdre $3*4pts + 4*2points = 20$ points;
- vous avez un potentiel de $2*1 + 5*2 + 2*4 = 20$ points;

2) Face Notebook

Objectif : Prendre en compte l'individualité de chacun

1. (++, -) Enregistrer la formation suivie par X : (Info1, Infocom2,)
2. (+, -) Quelle est la formation suivie par X ?
3. (=, -) Qui est inscrit avec telle formation ?
4. (+, -) Enregistrer l'humeur de X : happy, sad, ...
5. (=, -) Quelle est l'humeur de X ?
6. (+, -) Enregistrer le surnom de X
7. (=, -) Quel est le surnom de X ?
8. (++, -) Enregistrer l'adresse (coordonnées GPS) de X.
9. (+, -) Quelle est l'adresse de X ?
10. (++, \$\$) Qui habitent à une distance inférieure à Y km de X ?
11. (=, \$\$) Qui habite le plus près de X ?
12. (=, \$) A quelle distance X habite de Y ?

(++, -) Voir : Console de saisie

Bilan :

- vous pouvez perdre $3*4pts + 4*2points = 20$ points;
- vous avez un potentiel de $10*1 + 1*2 + 2*4 = 20$ points;

3) Linked In my world

Objectif : Enregistrer et retrouver des liens entre des personnes : ami, collègue, famille,

1. (++, -) Enregistrer un lien entre X et Y en précisant son type
2. (++, \$) Quels sont les liens directs entre X et Y ?
3. (+, -) Existe-il un lien de type Z entre X et Y ?
4. (+, \$) Existe-il une relation entre X et Z (il suffit qu'il existe des relations entre X et W et entre W et Z) ?
5. (-, \$\$) Quelle est la relation entre X et Z (e.g. parenté avec W et ami entre W et Z)
6. (-, \$) Existe-il une relation de parenté entre X et Z (e.g. parenté avec W et parenté entre W et Z)
7. (+, \$\$) Quel est le chemin le plus court entre X et Z ? (e.g. parenté avec W et parenté entre W et Z)
8. (-, \$\$) Permettre d'ajouter un type de relation par exemple *conflit*

(++, -) Voir : Console de saisie

Bilan :

- vous pouvez perdre $3*4pts + 4*2points = 20$ points;
- vous avez un potentiel de $3*1 + 3*2 + 3*4 = 21$ points;

4) Tchatche'Car

Objectif : Faciliter le co-voiturage.

1. (+, \$) Enregistrer un voyage : conducteur X, Depart (GPS), Arrivee (GPS), NombreDePassagers possibles, Jour et heure de départ
2. (+, \$) Obtenir des informations sur un voyage : conducteur X, Depart (GPS), Arrivee (GPS), NombreDePassagers possibles, Date de départ, nombre de passagers enregistrés
3. (=, \$) Quels voyages sont prévus à Date donnée à X heures près?
4. (+, -) Quels voyages sont prévus au départ de D ?
5. (+, -) Quels voyages sont prévus avec une arrivée en A ?
6. (+, \$) Enregistrer un passager sur un voyage donné.
7. (+, \$) Quels sont les passagers sur un voyage donné ?
8. (=, -) Quels sont les voyages faits par X ?
9. (=, -) Quels sont les voyages faits par X en tant que conducteur ?
10. (=, \$) Quels sont les voyages faits par X en tant que passager ?
11. (=, \$) Enregistrer les voyages dans un fichier.
12. (=, -) Charger les voyages non encore réalisés qui se trouvent dans un fichier.

(+, -,) voir : Console

Bilan :

- vous pouvez perdre $3*4pts + 4*2points = 20$ points;
- vous avez un potentiel de $6*1 + 7*2 = 20$ points;

5) Place Advisor

Objectif : Favoriser l'organisation de voyages

1. (+, -) Enregistrer un point d'intérêt : nom, description, coordonnées GPS
2. (+, -/\$) Incrémenter ou décrémenter un coefficient d'intérêt associé à un point d'intérêt (j'aime, je n'aime pas)
3. (=, -) Obtenir le coefficient d'intérêt associé à un point d'intérêt
4. (+, -) Associer un coût de nuité à un point d'intérêt
5. (+, \$) Calculer le coût d'un "itinéraire"¹ (somme des coûts des points d'intérêts qui composent l'itinéraire)
6. (=, \$) Calculer des itinéraires en fonction d'un budget²
 - a. (=, \$\$) En maximisant l'intérêt
 - b. (=, \$\$) En forçant la présence d'un point d'intérêt donné
 - c. (=, \$) En évitant la présence d'un point d'intérêt donné
7. (+, -) Mémoriser les points d'intérêts dans un fichier
8. (=, -) Charger les points d'intérêts qui se trouvent dans un fichier

¹ itinéraire = une liste de points d'intérêts pour l'instant

² On prendra en compte le cout des trajets et des distances lors de l'intégration

(++, -) voir : Console

Bilan :

- vous pouvez perdre $4*4pts + 2*2points = 20$ points;
- vous avez un potentiel d'au moins $7*1 + 3*2 + 2*4 = 21$ points;

6) Alice's MAP

Objectif : Visualiser sur une carte des points donnés. Ce sous-projet est central. Sa réussite va impliquer toute la suite du projet. Contrairement aux autres sous-projets, il repose sur l'utilisation du module fourni.

Voici les fonctionnalités que vous devez ajouter :

- (++, -) Positionner un point (Marker) à partir de coordonnées GPS et d'un libellé
- (=, \$) Saisir le point dynamiquement
- (++, -) Visualiser un chemin (ligne) entre deux markers donnés
- (=, \$) Saisir les points dynamiquement
- (++, \$/\$\$) Visualiser une description associée à un marker
- (=, \$) Modifier une description associée à un marker
- (+, \$/\$\$) Visualiser une description associée à un chemin
- (=, -) Modifier une description associée à un chemin
- (+, \$) Prévoir une zone d'affichage d'informations
- (=, ?) Interroger un Marker pour obtenir des informations

(++, \$/\$\$) voir : une démonstration basée sur des boutons éventuellement prédéfinis.

Bilan :

- vous pouvez perdre $4*4pts + 2*2points = 20$ points;
- vous avez un potentiel d'au moins $2*1 + 7*2 = 16$ points; ((difficile à déterminer. L'évaluation de la complexité des fonctionnalités additionnelles sera réalisée en direct).

Partie Suivante : Intégration :

Voici quelques exemples, mais il appartient à chaque groupe de choisir ses "histoires".

1. Visualiser une offre de covoiturage sur une carte
2. Vérifier que la personne qui offre un covoiturage est bien au point de départ /arrivée
3. Vérifier que le passager est bien au point de départ/arrivée
4. Visualiser les personnes autour de moi à une distance donnée
5. Rechercher qq qui a telles caractéristiques autour de moi : formation,

humeur, ...

6. Rechercher qq avec qui je suis ami/parent/... autour de moi
7. Alimenter les reseau avec les covoitureurs en ajoutant un lien "co-voyageur"
 - a. A la demande : en passant un co-voiturage
 - b. Automatiquement à chaque déclaration d'un co-voiturage
8. Visualiser la position de tous ses amis etc
9. A chaque saisie de l'humeur on enregistre la position
10. Visualiser la maison des membres
11. Visualiser les points d'intérêts sur la carte

TIPS :

1. Les coordonnées doivent être exprimées en coordonnées GPS, aller voir au moins l'interface ICoordinate qui vous est donnée dans le module OpenStreetView.
2. Le calcul de la distance entre deux données GPS : <http://www.movable-type.co.uk/scripts/latlong.html>