Projet - Traitement de données

E.N.S.A.I - 1A

2019-2020

1 Avant propos

, Le Projet "Traitement de données" s'adresse aux étudiants en 1ère année à l'Ensai. Chaque groupe projet sera constitué de 3 étudiants pour travailler sur ce projet. Il permet tout d'abord d'approfondir et de mettre en pratique les connaissances acquises en informatiques en général et plus particulièrement en `Python`. Vous devez mettre en avant le côté *Programmation Orientée Objet* du langage Python. L'architecture de vos codes doit être bien structurée, bien documentée.

Pour chaque trinôme, quatre séances encadrées sont consacrées à l'avancement et au suivi du projet. Attention, pour chaque séance de 3h, un encadrant devra s'occuper de plusieurs trinômes en même temps. Une part du travail lors de ces séances se fera donc en autonomie. Des créneaux et salles vous sont aussi réservées pour vous permettre d'avancer en autonomie en dehors des séances encadrées.

La création d'un programme informatique est souvent structurée en 4 phases :

1. Spécification des besoins : analyser et reformuler le problème posé afin de rédiger le cahier des charges.
2. Conception globale : définir l'architecture globale du programme via une analyse descendante par exemple.
3. Conception détaillée : décrire la structure interne des composants utilisés dans la conception globale.
4. Implémentation : transcrire notre conception détaillé dans un langage de programmation, `Python` ici.

En coordination avec l'encadrant, les étudiants doivent organiser le projet en sous-tâches réparties chronologiquement et/ou par sous-groupes d'étudiants. Chaque séance débute par un point d'avancement de chacune des sous-tâches et, si nécessaire, par une mise à jour des tâches. **Lors de la première séance encadrée**, vous validerez avec votre encadrant les spécifications de votre programme, sa conception et l'organisation prévisionnelle des tâches. Vous devez donc **venir à cette séance avec un *(pseudo-)cahier des charges*** illustrant le fruit de votre réflexion concernant ces sujets.

Vous trouverez sur *Moodle* la base de données que vous devrez traiter. Le rapport doit être rédigé en LaTeX. Un template a aussi été déposé sur Moodle, vous pouvez l'utiliser librement. **La date limitée pour rendre le rapport ainsi que le code est fixée au 13 mai 2020 à 23h55**.

Les soutenances seront programmées les 19 et 20 mai 2020. Une présentation avec démonstration est demandée. Vous aurez 30 minutes : **20 minutes de présentation et démonstration, 10 minutes de questions**. Les attendus pour le cahier des charges, le rapport, le code et la soutenance sont détaillés dans la table 1.

Vous êtes libre d'organiser votre travail en groupe et notamment le partage de fichier :

- Vous pouvez tout simplement travailler ensemble via l'espace partagé de l'Ensai.
- vous pouvez utiliser un système de gestion de versions tel que Git. Vous pouvez par exemple déposer votre projet sur Github `https://guides.github.com/activities/hello-world` ou Bitbucket `https://bitbucket.org/product` puis utiliser le client Sourcetree `https://www.sourcetreeapp.com` pour synchroniser les outils sur votre machine personnelle. Il est fortement conseillé de rendre votre projet privé afin que les sources ne soient pas visibles. Addressez-vous à votre encadrant pour davantage de renseignements.
- Vous pouvez aussi partager vos travaux via Dropbox `https://www.dropbox.com/`. Soyez toutefois prudent sur la gestion des conflits lorsque plusieurs personnes travaillent sur le même fichier.
- Un **forum** est à votre disponibilité pour échanger entre groupe vos problèmes et les solutions possibles `https://foad-moodle.ensai.fr/mod/forum/view.php?id=1757&forceview=1`
- ...

Cahier des charges ($\leq$ 5 pages) **A rendre pour la première séance**	— Rédiger de façon claire et propre. — Décrire "la maquette" de votre application : que doit-il se passer lors de l'exécution ? — Effectuer un diagramme d'état de votre application — Décrire les fonctionnalités que vous coderez.
suivi personnel ($\leq$ 2 pages) **A rendre pour la quatrième séance**	chaque étudiant explique : — son rôle dans le groupe — ses tâches réalisées — ses tâches en cours de réalisation
Code	— Votre programme doit fonctionner. — Vos codes doivent être cohérent avec ce que vous décrivez dans le rapport. — La quantité et la qualité des fonctionnalités sont aussi un point important pour la validation. — Le code doit utiliser les "menus", c-à-d, une liste des choix d'action possibles(voir l'exemple sur *Moodle*). — Le code doit utiliser l'héritage dans la partie "gestion de données". — Le code doit être découpé en sous fichiers. — Effectuer des tests unitaires sur un des fichiers que vous choisissez. — Choisissez un module que vous commenterez de façon détaillée pour générer une documentation automatique. — Question personnelle
Rapport ($\leq$ 25 pages)	— Rédiger de façon claire et propre. Afin de rédiger un document latex, vous pouvez utiliser le logiciel **TeXstudio** ou des outils en ligne `overleaf.com` — Énoncer les fonctionnalités *implémentées* vis-à-vis du cahier des charges. Dans le cas contraire, expliquer pourquoi. — Rédiger un chapitre entier pour expliquer votre gestion des données(c-à-d, comment importer/stocker/exporter les données, type de données, les opérations faites sur les données comme l'ajout, la suppression, etc.). — Présenter l'architecture de votre application et des classes. Il est indispensable d'utiliser des outils d'UML(diagramme de classes pour la deuxième séance). Pour cela, vous pouvez utiliser des outils en ligne comme `https://www.draw.io/` — Les différents diagrammes ainsi que les images doivent être **claires** et **lisibles**. — L'architecture dans votre rapport doit être cohérent avec ce que vous codez. — Annexe : la documentation générée automatique à partir des commentaires dans le code.
Soutenance	— Une démonstration de votre application. — Une explication de l'architecture de votre code. Vous n'avez que 20mn, donc vous pouvez vous focaliser sur un point précis. — Les pistes d'amélioration (architecture, gestion de projet, ...)

TABLE 1 – Les attendus

2 Données

Dans ce projet, on travaille sur une base de données issue de la publication du site "The World Factbook" qui est une publication officielle de 2017 de la CIA détaillant chaque pays du monde, des points de vue de ces informations

1. Introduction : Une introduction historique rapide du pays
2. Geography : information géographiques comme la superficie, le climat, etc.
3. People and Society :informations autour de la population et les communautés
4. Government :
5. Economy :
6. Energy :
7. Communications :
8. Transportation :
9. Military and Security :
10. Transnational Issues :

La base de données est stockée dans le fichier "factbook-country-profiles.json". Pour pouvoir stocker les données sous forme d'une liste de dictionnaire en `Python`, vous pouvez utiliser le code suivant :

```
import json
filename="factbook-country-profiles.json"
with open(directory_data + filename) as json_file:
    donnees = json.load(json_file)
```

où `directory_data` est le chemin du répertoire contenant le fichier "factbook-country-profiles.json".

À partir de cette base de données, on souhaite effectuer des analyses que l'on décrit dans la suite.
Pour simplifier les choses, on traite que les informations précisées dans le tableau 2 Pour accéder

N°	information	accès en fichier `JSON`
1	Nom de pays	`['Government']['Country name']['conventional short form']['text']`
2	Superficie	`['Geography']['Area']['total']['text']`
3	Population	`['People and Society']['Population']['text']`
4	Croissance démographique	`['People and Society']['Population growth rate']['text']`
5	Inflation	`['Economy']['Inflation rate (consumer prices)']['text']`
6	Dette	`['Economy']['Debt - external']['text']`
7	Taux de chômage	`['Economy']['Unemployment rate']['text']`
8	Taux de dépenses en santé	`['People and Society']['']Health expenditures'']['text']`
9	Taux de dépenses en éducation	`['People and Society']['Education expenditures']['text']`
10	Taux de dépenses militaires	`['Military and Security']['Military expenditures']['text']`
11	cinq classes d'âge	`['People and Society']['Age structure']`

TABLE 2 – Informations des pays

à une information concernant un pays, on utilise l'indice du pays dans notre liste `donnees` avec le code d'accès à cette information. Par exemple, pour savoir la population de la France on écrit : `donnees[82]['People and Society']['Population']['text']` car 82 est l'indice de la France

dans notre base de données brute.

Le mot **critère** utilisé dans ce projet signifie une des informations citées dans le tableau 2 à l'excepte de "Nom de pays" et "cinq classes d'âge". Pour certain pays, il y a des **critères** manquants dans notre base de données et pour cela, ces pays ne seront pas pris en compte dans le projet.

3 Acteurs des données

Un *consultant* peut consulter un pays et afficher ses informations.

Un *Géographe* se connecte avec son pseudo et un mot de passe et une fois qu'il s'est identifié il a le droite en outre d'un consultant d'ajouter/modifier un pays.

Un *Data Scientist* est celui qui est responsable de l'affichage graphique ainsi que de la résumé des informations utiles concernant les pays.

Un administrateur est celui qui gère les comptes d'employeurs (*Géographe* ou *Data Scientist*), il a le droit de valider/supprimer le compte d'un employeur ainsi de supprimer un pays.
Un fois le programme est lancé, il demande à l'utilisateur son statut puis il lui propose les tâches qu'il a le droit de faire.
Le tableau suivant détaille toutes les tâches possibles en fonction du statut du consultant :

	Tâche	Consultant	Géographe	Data Scientist	Admin
1	Afficher un pays	✓	✓	✓	✓
2	Proposer une correction d'une info	✓	×	✓	×
3	Accepter/Refuser cette proposition	×	✓	×	✓
4	Ajouter/modifier un pays	×	✓	×	✓
5	Supprimer d'un pays	×	×	×	✓
6	Connexion/Déconnexion	×	✓	✓	✓
7	Création/supprimer un compte d'employeur (Géographe/Data Scientist)	×	×	×	✓
8	Résumé d'informations (voir sec.5)	×	×	✓	✓
9	Représentation graphique (voir sec.6)	×	×	✓	✓
10	Fonctionnalité avancée (bonus) (voir sec.7)	×	×	✓	✓

Table 3 – Liste des tâches

4 Affichage

L'affichage d'un pays nécessite au moins l'affichage des informations notées dans le tableau 2.

5 Résumé d'informations

1. Afficher les informations citées dans le tableau 2 concernant des certains pays.
2. Les n premiers/derniers pays dont un **critère** est le plus important.
3. Les pays dont un **critère** dépasse (ou ne dépasse pas) un certain seuil.
4. Le tableau des classes d'âge pour certains pays (10 pays au maximum), comme dans l'exemple :

	0-14	15-24	25-54	55-64	65 $\geq$
France	18.59%	11.8%	38.04%	12.44%	19.12%
Germany	12.83%	10.22%	40.96%	14.23%	21.76%
Spain	15.43%	9.56%	45.24%	11.91%	17.85%

5. Chaque membre du groupe propose une question dont il propose une solution

6 Représentation graphique

1. Diagramme en barres d'un certain **critère**
2. Les Box-plots correspondant aux répartitions des valeurs des 5 classes d'âge pour tous les pays.

7 Fonctionnalité avancée (bonus)

Existe-t-il des profils de pays en fonction de ses **critères** ? Trouver un moyen ou un algorithme pour répondre à cette question.