POLYTECH PARIS-SUD

Projet Java - Graphique - IHM
Spécialité Informatique, 3ème année 2021 - 2022

« OBIS 3D »

1. Contexte

OBIS (Ocean Biodiversity Information System) est un centre mondial d'échange de données et d'informations en libre accès sur la biodiversité marine pour la science, la conservation et le développement durable. Plus de 20 nœuds OBIS dans le monde relient 500 institutions de 56 pays. Collectivement, ils ont fourni plus de 45 millions d'observations de près de 120 000 espèces marines, des bactéries aux baleines, de la surface à 10 900 mètres de profondeur, et des tropiques aux pôles. Il est possible de rechercher de manière transparente par nom d'espèce, zone géographique, profondeur, temps, paramètres environnementaux...

Le but du projet est de réaliser une application permettant de visualiser ces observations sur un globe terrestre en 3D. Cette application devra permettre de visualiser pour une espèce donnée le nombre d'observations par zone géographique, d'abord à partir d'un fichier, puis en faisant des requêtes à cette base de données.

Vous pouvez utiliser l'interface Swagger pour comprendre cette API et la tester: https://api.obis.org/

1. Déroulement du projet

L'application sera réalisée en **Java**, tandis que l'interface graphique utilisera **JavaFx** pour la partie 2D et la partie 3D. Un tutoriel vous sera proposé lors de la première séance de projet afin de vous initier à l'utilisation de la 3D en **JavaFx**.

Le projet se compose en deux parties principales :

- une partie applicative qui permettra d'accéder aux données facilement,
- une partie interface graphique qui permettra à l'utilisateur de visualiser les informations et de contrôler l'application.

Le travail est à réaliser en binôme et il est donc important de s'assurer que ces deux parties soient le plus indépendantes possible l'une de l'autre pour qu'une fois l'API de la partie applicative définie, vous puissiez travailler en parallèle.

Vous pouvez tester votre partie applicative à l'aide des tests JUnit (http://junit.org/).

Par ailleurs, un accent particulier sera mis sur la conception logicielle tout au long du projet :

- Pour la partie applicative, il vous est demandé de rendre un **diagramme UML** modélisant votre application dès la 2ème séance.
- Pour la partie interface graphique, il vous est demandé de rendre un **prototype papier** de quelques pages expliquant le fonctionnement de votre interface utilisateur avant de passer à l'implémentation.

L'environnement de développement conseillé est **Eclipse** mais vous pouvez utiliser celui avec lequel vous êtes le plus familier. L'utilisation d'un outil de gestion de versions (Git de préférence) est fortement conseillé pour faciliter le partage entre binômes.

2. Partie applicative

L'application «OBIS 3D» doit être capable de :

- Charger les données json en provenance d'un fichier ou directement depuis la base de données du projet OBIS et de présenter les résultats de la façon la plus efficace possible pour la partie interface graphique
- Permettre d'effectuer différentes requêtes:
  - Récupérer le nombre de signalement par région pour un nom d'espèce passé en paramètre
    - Ex: Dauphins pour une précision Geohash de 3 :
      - https://api.obis.org/v3/occurrence/grid/3?scientificname=Delphinidae
    - Ex: Requins pour une précision GeoHash de 3:
      - https://api.obis.org/v3/occurrence/grid/3?scientificname=Selachii
  - Récupérer le nombre de signalement par région pour un nom d'espèce et entre deux dates passées en paramètre
  - Récupérer les détails d'enregistrement pour un nom d'espèce passé en paramètre et un GeoHash
    - Ex: Raies Mantas pour le GeoHash "spd":
      - https://api.obis.org/v3/occurrence?scientificname=Manta%20birostris&geometry=spd
  - Récupérer les 20 premiers noms scientifiques d'espèce commençant par une chaîne de caractères passée en paramètre
    - Ex: Commençant par un "a":
      - https://api.obis.org/v3/taxon/complete/verbose/a

Les fonctionnalités demandées doivent être effectuées en fonction de l'ordre de priorité indiqué ci-dessous :

Fonctionnalités	**Niveau de priorité**
Charger un fichier Json local avec le nombre de signalements pour une espèce par zone et mettre à disposition les coordonnées de chaque zone ainsi que le nombre d'occurrences. Il sera utile de facilement pouvoir accéder aux nombres minimal et maximal d'occurrences pour pouvoir s'en servir par la suite pour créer les légendes des visualisations.	1
Accéder aux mêmes informations que ci-dessus à partir en passant le nom scientifique de l'espèce en paramètres et le degré de précision GeoHash et en faisant la requête adéquate de manière asynchrone.	1
Rajouter la possibilité de faire la même chose que ci dessus pour un intervalle de temps	1
Charger le nombre de signalements par zone par intervalle de temps en passant en paramètres: • nom scientifique • précision GeoHash • date de départ • durée d'intervalle de temps • nombre d'intervalles Ces données serviront à afficher l'évolution du nombre d'occurrences d'une espèce. Les requêtes devront se faire une par une de manière asynchrone.	2

Convertir une coordonnée GPS en GeoHash	2
Faire une requête pour un GeoHash et un nom scientifique (qui peut être vide) passés en paramètre pour récupérer les détails des différentes observations: ● “scientificName” ● “order” ● “superclass” ● “recordedBy” ● “species”	2
Récupérer les premiers noms des espèces commençant par une chaîne de caractères passée en paramètre	3

Toutes ces fonctionnalités devront être testées avec une classe de test JUnit.

3. Interface graphique

La partie interface graphique doit permettre à l’utilisateur de pouvoir visualiser le nombre d'observations par zone, de les voir évoluer par pas de cinq années.
Plus précisément, l’interface graphique devra implémenter les fonctionnalités suivantes :

Fonctionnalités	**Niveau de priorité**
Afficher un globe en 3D et permettre à l’utilisateur de tourner autour grâce à la souris.	1
Au démarrage l’application doit afficher les zones comportant des signalements d’une espèce à partir d’un fichier inclus. Ces zones doivent être de différentes couleurs en fonction du nombre de signalements. La légende doit comporter un nombre fixe de couleurs (par exemple 8 comme sur le site https://mapper.obis.org/) et indiquer le nombre minimal et le nombre maximal d’occurrences pour chaque couleur.	1
De la même façon, afficher les occurrences d’une espèce en passant le nom scientifique exact saisi par l’utilisateur. Si le nom n’est pas reconnu, l’utilisateur doit en être averti.	1
Ajouter la possibilité de saisir deux années pour n’afficher que le nombre d'occurrences d’une espèce entre ces deux années.	1
Afficher l’évolution du nombre de signalements d’une espèce entre deux années à l’aide d’une interface permettant de démarrer, mettre en pause et stopper (avec un pas fixe de 5 ans, pour ne pas faire trop de requêtes sur le serveur).	2
Au lieu de saisir un nom d’espèce, l’utilisateur pourra cliquer sur le globe pour faire apparaître la liste des espèces relevées sur ce GeoHash. Un clic dans cette liste démarrera la requête comme si l’utilisateur avait saisi ce nom d’espèce.	2
Afficher la liste des signalements lorsqu’un utilisateur clique sur une zone comportant un nombre d’occurrences positif. Les informations affichées par signalement doivent être: ● “scientificName” ● “order” ● “superclass” ● “recordedBy” ● “species”	2

Ajouter une liste de noms scientifiques qui s'affichera au fur et à mesure que l'utilisateur saisit des caractères dans le champ de saisie	3
Afficher le nombre d'occurrences sous forme d'histogramme 3D, dont la hauteur et la couleur sera fonction de ce nombre	3

Rappel : vous devez d'abord réaliser un prototype papier expliquant le fonctionnement de votre interface graphique avant de passer à son implémentation.

1. Evaluation

L'évaluation du projet Java - Graphique - IHM portera sur le **code du projet** à rendre au plus tard le **vendredi 24 juin 2022**.

2. Contacts

Groupe 1 : Arthur FAGES – arthur.fages@ universite-paris-saclay.fr
Groupe 2 : Tifanie BOUCHARA - tifanie.bouchara@universite-paris-saclay.fr