LES TROPHÉES NSI

Édition 2023

DOSSIER DE CANDIDATURE
PRÉSENTATION DU PROJET

LIFE SCORE

notation de communes françaises

NOM DU PROJET : LIFE SCORE

> *PRÉSENTATION GÉNÉRALE :*

• Idée et objectifs

- *Le projet LifeScore a pour but de permettre à ses utilisateurs de pouvoir noter des communes de France en fonction de préférences qu'ils auront fournis. En plus d'une note, ceux-ci pourront regarder un détail de celle-ci (par type) mais aussi voir la note des communes aux alentours afin de se donner un avis sur la zone.*

• Origines et intérêts du projet

- *Nathan et Frédéric avaient déjà travaillé sur un projet qui permettait de donner la météo d'une ville sur demande. L'idée nous est d'abord venue de se servir de données afin de trouver des terrains susceptibles d'accueillir des éoliennes en réfléchissant avec notre professeur Mr Ballouki, ce qui pourrait être intéressant pour des investisseurs. Cependant, nous avons plutôt fixé l'analyse de villes au centre du projet, ce qui pourrait profiter au plus grand nombre.*

> *ORGANISATION DU TRAVAIL :*

- ***Nathan Bosy*** *: Gestion des données, calculs et comptabilité*
- ***Raphaël Farenc*** *: Interface graphique et interprétation des résultats*
- ***Thor Naughton*** *: Calcul des coefficients et API*
- ***Frédéric Marquet*** *: Recherches pour la base de données*

• Répartition des tâches

- *Pour la répartition des tâches, chacun avait son objectif, sa "quête principale". Mais, cela ne nous a pas empêché de nous aider mutuellement et si un de nous avait un problème avec une partie du code, un ou deux autres membres du projet s'y mettaient pour réfléchir à une solution.*

• Organisation du travail :

- *Nous nous sommes servis de la plateforme https://github.com afin de pouvoir chacun travailler de son côté et ce, à n'importe quel moment, et n'importe où.*
- *Lors de certains moments avant des présentations en classe (notamment pendant les vacances et les week ends), nous nous sommes retrouvés en appel visio et avons pu coder au même moment sans soucis de commits grâce à l'assistance de LiveShare disponible sur Visual Studio code (application de codage principale).*
- *Sinon, lorsque nous ne faisions pas cela, nous nous servions d'un groupe instagram où on partageait nos problèmes et idées et où on signalait quel(s) fichier(s) nous souhaitions "prendre" pendant une durée déterminée. A part la période qui précédait les épreuves de spécialités (environ 2 semaines avant), nous travaillons quotidiennement sur le code.*

LES ÉTAPES DU PROJET :

Entre chaque point, une présentation du projet en classe a été effectuée afin de récupérer un avis général et des conseils d'améliorations.

- *Nous avons commencé le travail en séparant le programme en 3 fichiers :*
  - *LifeSCORE.py (code principal) qui exécute toutes les fonctions et implémente l'interface graphique. Dans celui-ci, le questionnaire a été une priorité initiale*
  - *Classes.py Dans lequel nous mettons la classe principale (Donnees) accompagnée de ses méthodes (entre autres une qui vérifie que la ville existe, qu'elle soit en France ou encore sa note en fonction d'établissements sportifs).*
  - *Update.py Qui permet notamment de mettre à jour les fichiers csv.*
  - *calcul_coefficients.py Qui permettait d'ajouter les coefficients à la note*

Une base de données en .json a d'ailleurs été mise en place pour gérer l'arrivée des csv. Ceux-ci étaient difficilement téléchargeable à cause de plusieurs problèmes mais l'étape suivante s'est concentré justement sur ce problème.

- *Ensuite, nous avons amélioré chaque fichiers :*
  - *LifeSCORE.py est passé de Tkinter à CustomTkinter (visuellement plus agréable) avec la résolution de certains bug sur les boutons*
  - *Classes.py Les premières notes (basées uniquement sur le sport marchaient désormais très bien*
  - *Update.py Le téléchargement était désormais et plus rapide, et moins buggé*
  - *calcul_coefficients.py Un système plus précis pour le calcul des notes de météo*

- *Puis, nous avons réalisés d'autres changements aussi importants :*
  - *Ajout d'un style et une sauvegarde de paramètres grâce à la fenêtre de paramètres*
  - *Possibilité d'arrondissements dans les grandes villes*
  - *Au lieu de créer une nouvelle fenêtre à chaque changement de page, on reste désormais sur la même en modifiant seulement les widgets présents*
  - *Ajout de l'installation des bibliothèques automatiques*
  - *Animation de la note et polices d'écritures*
  - *Interface de téléchargement*
- *S'approchant de la fin du projet, nous avons ajouté :*
  - *Une carte montrant la ville*
  - *Plus de données prises en compte dans la notation(csv avec et sans code INSEE)*
  - *Un nouveau logo*
  - *Une notation de la météo (ajoute des données)*
  - *Les avantages et inconvénients de la ville*
  - *Une compatibilité avec Ubuntu 22*
  - *La note des 10 communes les plus proches de celle qu'on a cherchée*
  - *Amélioration du système de téléchargement avec un .zip*
- *Enfin, nous avons réorganisé le code en effectuant des modifications sur les nos de fichiers et de dossiers :*
  - *classes.py -> notation.py*
  - *update.py -> mise_a_jour.py*
  - *calcul_coefficients.py a été supprimé et son code a été rajouté dans notation.py*
  - *Le dossier "systeme" stocke les données obligatoires au lancement de LifeSCORE*
  - *Le dossier "donnees" stocke les données telles que le cache de l'appli et les préférences*

> *FONCTIONNEMENT ET OPÉRATIONNALITÉ :*

• Avancement du projet

Ce que nous avons réussi à implémenter tel qu'on le concevait :

- *Une application utilisant un module similaire à TKinter, CTKinter qui utilise des widgets de TKinter*
- *Une téléchargement/vérification de csv et leur analyse*
- *Une notation de villes*
- *Une prise en charge des arrondissements des villes qui en possèdent*
- *Un widget pour faire apparaître les avantages et les inconvénients d'une ville*
- *Une utilisation d'API pour des données météorologiques et une carte intéractive*
- *L'affichage des 10 communes les plus proches grâce à l'algorithme (k plus proches voisins)*

Ce que nous sommes encore en train d'essayer de terminer :

- *Essayer d'intégrer un nouveau "type" de csv, ceux-ci sont téléchargeables dans des fichiers .zip et on essaie de les télécharger puis de seulement extraire le fichier du zip qui nous intéresse pour supprimer le reste*

Ce que nous n'avons pas pu "terminer":

- *Ajouter plus de csv dans la base de données (plus on en trouve, plus les données se raréfient)*
- *Gérer le problème de villes homonymes (qui portent rigoureusement le même nom) qui empêche d'avoir une note correcte (on ne sait pas quelle ville parmis celles qui portent le même nom va être analysée*

• Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet

- *Nous avons proposé à notre groupe de spécialité de tester le code et de nous faire des retours sur tout ce qui pouvait les gêner. On a ensuite fait tester à des membres de notre famille et cela nous a ouvert les yeux sur des détails que nous avions omis (problèmes d'affichage, mise en page à améliorer et prise en main)*
- *Ensuite, Thor et Nathan se sont chargé respectivement de la partie qui télécharge les modules nécessaires et des fichiers csv (maintenant avec un fichier .zip compressé) ce qui permet, à condition que la version de Python soit assez récente (3.10 ou plus), de lancer aisément le programme même pour une personne qui n'est pas beaucoup familière avec le langage Python.*
- *Pour ce qui est du programme en lui-même, nous avons voulu souligner le désir de rester compréhensibles et ainsi nous avons formaté les noms de variable et agencé les fonctions de telles sortes à ce que la lecture et la prise en main de celui-ci ne pose aucun problème. Nous avons enfin enrichi chaque fichier python de commentaires pertinents expliquant la plupart des opérations qu'une personne en première par exemple pourrait mal comprendre.*

• Difficultés rencontrées et solutions apportées

- *Des problèmes sur l'interface graphique et certains widgets(états, placements,...). Cela a été résolu par une analyse poussée de l'interface graphique CTK*
- *Un problème avec les arrondissements et les codes Insee. Nous l'avons réglé avec plusieurs conditions*
- *Les caractères spéciaux, accents et déterminants empêchaient l'analyse de toutes les villes. Pour y remédier, ce qui est entré passe par un formatage des accents et des majuscules afin de comparer les bons str*
- *Nous avons eu un problème avec une variable globale "n" utilisée à deux endroits en même temps(cela créait une inégalité dans les réponses et ce que le code récupérait). Un déplacement de celle là a permi de résoudre l'erreur*
- *Nous avons dû régler un problème de notes non exactes dans les 10 villes les plus proches car celles-ci ne prenaient pas en compte les notes de météo. Cela nous a permis de découvrir les "variables statiques" de classes*
- *Nous avons dû faire appel à nos connaissances en mathématiques afin de trouver une fonction qui retourne une taille de police pour afficher les villes*
- *La fonction avantages_inconvénients est passé par plusieurs changements de variables afin de prendre en compte plusieurs scénarios*
- *Définir la taille des fenêtres était complexe, tantôt on travaillait sur un écran 1280x960, tantôt on travaillait avec un écran en 1024x768 pixels. Nous avons trouvé par tâtons des valeurs minimales pertinentes*

> OUVERTURE :

• Idées d'améliorations (nouvelles fonctionnalités)

- *Notre problème des villes homonymes peut faire l'objet d'une étude plus poussée (notamment en utilisant un système similaire à celui des arrondissements en posant par exemple le département de la ville a côté de son nom.*
- *Ensuite, faire croître la base de données ne serait pas de trop.*
- *Finalement, on pourrait rajouter l'idée qu'un stagiaire de notre professeur a proposé : Afficher les 10 meilleures villes aux alentours et non pas les 10 plus proche (cela n'a pas pu être réalisé à cause du nombre important de calculs à faire)*

• Stratégie de diffusion pour toucher un large public

- *Pour l'instant, seul le bouche à oreille (dans notre lycée comme dans nos domiciles) a permis à plusieurs de tester l'application. Le fait que python ne soit pas installé par défaut sur les systèmes d'exploitations est un léger frein à la diffusion du projet mais nous envisageons de parler de notre projet à certains officiels (de mairie par exemple) pour qu'ils puissent le tester et pourquoi pas essayer d'améliorer leurs communes*

• Analyse critique du résultat

- *L'organisation nous a semblé pertinente, chacun avait plus de facilité dans le domaine auquel il était associé. Ce que nous aurions pu changer dans notre projet serait la manière d'aborder les fichiers csv : au lancement de celui-là, nous n'avions pas encore eu le cours sur les bases de données et les recherches avec SQL et nous avons décidé de partir sur des analyses avec le module pandas ce qui nous coûtait en temps pendant les analyses. Une association entre les requêtes SQL et les fichiers CSV aurait peut être permis d'accélérer le processus d'analyse des fichiers.*