

Sujets pour le projet Python 2022-2023

Travail à faire au plus tard pour le 6 mars :

- Lire les sujets (page 3 à 9),
- Choisir 3 sujets par ordre de préférence,
- Indiquer sur ENT vos 3 sujets par ordre de préférence. Les binômes seront formés par votre enseignant•e de façon à contenter au maximum vos demandes.

Répartition des séances

Le projet informatique se déroule de mars à mai. Plusieurs séances (avec ou sans votre enseignant•e) sont prévues. À la fin du projet, une soutenance sera planifiée pour que vous puissiez présenter votre travail.

Le projet est réalisé en binôme.

Vous ne devez pas partager votre code avec d'autres binômes (sanctions pour le binôme qui donne le code et pour le binôme qui reçoit).

I. 1^{ère} séance : Spécifications et conception détaillée

Ce premier document est à déposer sur ENT au plus tard à la fin de la 1^{ère} séance.

Pour les spécifications, il devra comporter :

- Le titre du sujet choisi,
- Les objectifs du projet : les sujets proposés sont ouverts, expliquez en quelques lignes ce que vous allez réaliser. Vous devrez également choisir si vous souhaitez utiliser des fichiers ou une base de données (à créer vous même), si vous souhaitez un affichage dans le terminal ou une interface graphique.
- Les fonctionnalités détaillées,
- Les IHM (les menus, écrans, ...).

Pour la partie conception détaillée, elle devra comporter :

- La liste des classes utiles pour le projet,
- La liste des fichiers de données ou des tables utiles pour le projet (stockage des données),
- La liste des fonctions utiles pour réaliser le projet (objectif, entrées, sorties) regroupées par nature de traitement,
- Le planning prévisionnel du projet (qui fait quoi et quand ?).

II. Jusqu'à la fin du projet :

Codage et tests du projet.

Remarque : Deux de ces séances seront réalisées en autonomie.

III. Avant la soutenance (la date vous sera communiquée) : Rapport final

À déposer sur ENT en .pdf.

Le contenu indicatif peut être :

- Un sommaire,
- Une introduction,
- Rappel des spécifications et de la conception détaillée,
- Méthode générale de résolution,
- Détail éventuel des parties difficiles ou originales,
- Problèmes rencontrés et solutions apportées,
- Les écarts par rapport au cahier des charges,
- Bilan : ce que le projet vous a apporté (les +, les -),
- Conclusion : Discussion des résultats, prolongements possibles, améliorations, ...

Le code **commenté** sera également à déposer au format numérique sur ENT. Il devra être compressé (format .zip), et devra contenir tout ce qui est nécessaire pour que votre enseignant.e puisse **l'exécuter**.

La qualité du code sera prise en compte dans l'évaluation.

Sujet 1 : Le Théâtre des Arts

Projet tous niveaux avec fichiers ou base de données.

Écrire en Python un programme de gestion pour le *Théâtre des Arts* :

- Les spectacles (titre, durée, type, etc...)
- Les acteurs présents dans les spectacles (nom, prénom, etc...)
- La liste des représentations pour chaque spectacle (date et heure de début)

Le logiciel permettra de gérer ces différentes ressources (ajout, modification, suppression, ...). Il pourra également proposer des services, comme par exemple :

- Liste des spectacles pour un mois donné,
- Lister des spectacles dans lesquels est présent un acteur donné,
- Lister chronologiquement toutes les dates des représentations d'un spectacle donné,
- Etc...



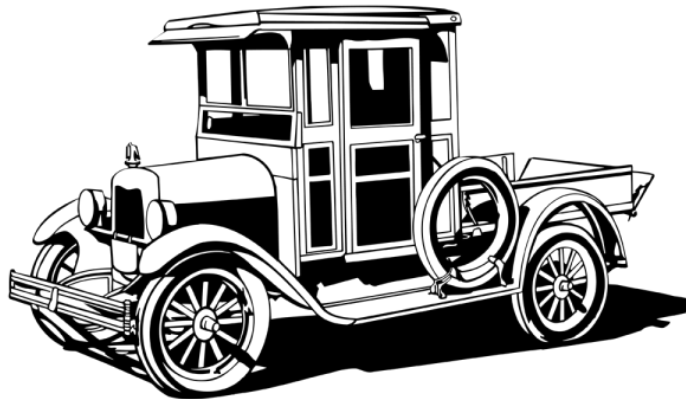
Sujet 2 : Auto-école

Projet tous niveaux avec fichiers ou base de données.

Écrire en Python un programme permettant de gérer une auto-école. Ainsi il sera possible de gérer des informations telles que : les élèves, les moniteurs, les voitures, etc.

Votre programme permettra de vérifier quel utilisateur se connecte. Et offrira des options différentes :

- Un élève pourra accéder à ses informations personnelles, son nombre d'heures de conduite, la date de sa prochaine leçon, etc...
- Un moniteur a la possibilité de voir les informations de ses élèves, et peut également ajouter une leçon à un élève en lui associant une voiture, etc...
- Un administrateur a la possibilité d'ajouter, modifier, supprimer toutes les informations sur les élèves, les moniteurs et les voitures.
- Etc...



Sujet 3 : Agence de voyage

Projet tous niveaux avec fichiers ou base de données.

Écrire en Python un programme permettant de gérer une agence de voyages. Le programme devra permettre de gérer des informations telles que :

- Les moyens de transport (type, volume utile, prix, etc.),
- Les clients (nom, prénom, adresse, etc.)
- Des destinations (pays, ville(s), ...).

Les fonctionnalités que doit proposer votre programme sont les suivantes :

- Le personnel de l'agence a la possibilité d'ajouter toutes les informations nécessaires,
- Un client a la possibilité de consulter la liste des voyages. S'il est connecté il a la possibilité de consulter l'historique de ses voyages,
- Etc.

Le sujet est vaste, à vous de décider si vous souhaitez spécialiser votre agence (weekend ou semaine, national ou international, un seul lieu ou tour, etc...)



Sujet 4 : Armada

Projet de difficulté moyenne à élevée. À faire avec une base de données non fournies.

Écrire en Python un programme permettant la gestion de l'Armada en juin prochain. Voici un échantillon des informations à gérer :

- Les visiteurs (nom, prénom, etc.),
- Les bateaux (nom, type, année, nationalité, dimensions, etc.),
- Les marins (nom, prénom, nationalité, etc.),
- Les événements (cérémonie d'ouverture, soirées, ...)
- Etc.

Les fonctionnalités attendues sont les suivantes :

- Les visiteurs peuvent acheter des billets, éventuellement avec un tarif spécifique. Ils peuvent participer à un événement, ou juste visiter,
- Les organisateurs peuvent attribuer des emplacements aux différents bateaux,
- Trouver la localisation d'un bateau. Par exemple à partir de son nom, ou de son type, ou de sa nationalité,
- Les organisateurs doivent pouvoir retrouver facilement à quel bateau appartient un marin,
- Etc.



Sujet 5 : Labyrinthe

Projet un peu difficile. Une version graphique demandant une autoformation doit être réalisée avec le module *pygame*.

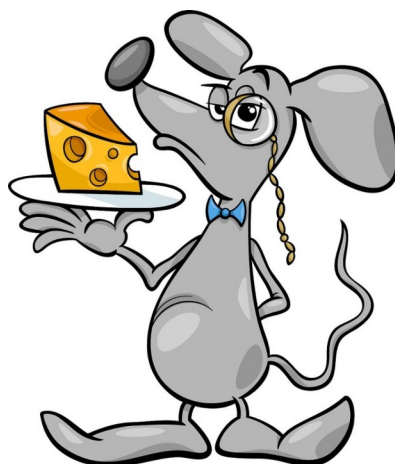
Le but de ce projet est de réaliser un programme permettant de résoudre un problème de labyrinthe. L'ordinateur devra résoudre seul le problème.

1. Votre programme doit permettre d'ouvrir un fichier texte, par exemple de cette forme :

```
*****
*      S      *
***  *      *  *
*      *      *
* *  *      *
* *  *      *
* *  *      *
* *  *      *
* *  *      *
* *  *      *
*****
```

- Les symboles contenus dans ce fichier sont les suivants :
- Un mur dans le labyrinthe est matérialisé par le signe multiplié (*),
- Une zone sans mur est représentée par un espace (),
- Le S représente la position de départ de la souris,
- Le F représente la position du fromage.

2. Réaliser un programme Python permettant de visualiser les déplacements automatiques de la souris dans le labyrinthe, jusqu'à ce qu'elle trouve le morceau de fromage.



Sujet 6 : Le jeu « *Velonimo* »

Projet assez difficile. Une version graphique demandant une autoformation est obligatoire. La réalisation sera faite avec le module *pygame*.

Écrire un programme Python permettant à un (ou plusieurs joueurs) de jouer au jeu *Velonimo*¹. Il vous est demandé entre autres les points suivants :

- Pour pouvoir jouer, un joueur doit avoir un compte,
- Le nombre de victoires et de défaites doivent être conservés pour chaque joueur,
- Un joueur doit pouvoir consulter ses statistiques de victoires et défaites,
- Le programme doit permettre de jouer contre des joueurs de type ordinateur (et/ou humain).

Il est également possible de prévoir d'autres fonctionnalités :

- Le jeu en réseau par exemple,
- Plusieurs niveaux de jeu pour les joueurs de type ordinateur,
- Etc.



1 Une vidéo de démonstration est disponible à cette adresse : <https://www.youtube.com/watch?v=TzYbyxliI4M>

Sujet 7 : Le jeu « *Tokaido* »

Projet difficile. Une version graphique demandant une autoformation est obligatoire. La réalisation sera faite avec le module *pygame*.

Écrire un programme Python permettant à un (ou plusieurs joueurs) de jouer au jeu *Tokaido*². Il vous est demandé entre autres les points suivants :

- Pour pouvoir jouer, un joueur doit avoir un compte,
- Le nombre de victoires et de défaites doivent être conservés pour chaque joueur,
- Un joueur doit pouvoir consulter ses statistiques de victoires et défaites,
- Le programme doit permettre de jouer contre des joueurs de type ordinateur et/ou humain.

Il est également possible de prévoir d'autres fonctionnalités :

- Le jeu en réseau par exemple,
- Plusieurs niveaux de jeu pour les joueurs de type ordinateur,
- Etc.



² Une vidéo de démonstration est disponible à cette adresse : <https://www.youtube.com/watch?v=FKoLeTaZkxs>