M1 ISEN - Cahier des charges - Projet Python avancé

Cyril Barrelet

January 2026

1 Critères d'évaluation

- Qualité du code (structure, POO, lisibilité, PEP 8, etc.) - 25 % de la note
- Qualité du README et documentation - 20 % de la note
- Fonctionnalité du projet - 25 % de la note
- Originalité / complexité maîtrisée - 10 % de la note
- Utilisation pertinente des outils vus en cours - 10 % de la note
- Travail en équipe (Git) - 10 % de la note

Note individuelle :

La note de chaque membre d'un même groupe pourra être légèrement modulée en fonction de sa participation réelle au projet.
Chaque groupe dispose d'un nombre de points égal au nombre de membres du groupe, à répartir librement entre ses membres.

La répartition devra être validée par l'ensemble du groupe. En cas de litige, l'enseignant se réserve le droit de modifier la répartition.

Exemple :

Pour un groupe de 4 personnes, vous disposez de 4 points au total à répartir.

- Participation égale : chaque membre reçoit 1 point supplémentaire.
- Participation inégale :
  - P1 : 40 % → 1.6 point
  - P2 : 40 % → 1.6 point
  - P3 : 10 % → 0.4 point
  - P4 : 10 % → 0.4 point

2 GitHub

- Le projet peut être accessible publiquement ou privé. S'il est privé, pensez à le rendre public pour faciliter l'évaluation. (**Obligatoire**)
- Chaque personne doit contribuer au Git. Les contributions de chacun seront vérifiées. (**Obligatoire**)
- Lorsqu'une nouvelle feature est implémentée, créez une nouvelle branche. (**Conseillé**)
- Une fois la feature développée et stable (donc testée), mergez-là à la branche principale. (**Fortement conseillé**)
- La branche principale "main" doit toujours être fonctionnelle. (**Très conseillé**)
- Un fichier "requirements.txt" ou "environment.yml". (**Obligatoire**)
- Seule la branche principale sera testée lors de l'évaluation. (**Important**)

3 README.md

Un soin particulier du fichier README sera attendu. C'est la page de garde de votre application. Elle doit être claire, détaillée et un peu catchy pour donner envie de tester votre trvail.
Exemple du GitHub : Ultralytics

- Résumé du projet :
  - Un schéma général du projet (**Obligatoire**)
  - Un UML pour des features principales. Exemple ici (**Bonus**)
  - Un résumé général présentant les différentes features. (**Obligatoire**)
- Tutoriel d'installation :
  - Création d'un environnement virtuel (conda ou venv). (**Obligatoire**)
  - Installation des packages avecle fichier "requirments". (**Obligatoire**)
  - Code minimal pour tester si l'installation s'est bien passée. (**Obligatoire**)
  - Donner la ou les différentes distributions (versions Windows, Mac, Linux) sur lesquelles le projet a été installé avec succès. (**Bonus**)
- Pour chaque feature implémentée :
  - Un petit résumé expliquant les points clés. (**Obligatoire**)
  - Un exemple d'utilisation : (**Obligatoire**)
    - soit un code à copier/coller
    - soit un ligne de commande avec un fichier mis dans un dossier /exemples
  - Une ou plusieurs captures d'écran soignée(s) avec une courte explication. (**Bonus**)
- Au moins une visualisation pertinente. (**Obligatoire**)
  Exemple :
  - Les performances d'un algorithme utilisé (temps d'execution / précision / ablation...)
  - L'analyse d'un joueur, de l'apprentissage d'un algorithme, etc.
  - Une visualisation d'un clustering, d'un descente de gradient, etc.
- Mise en forme générale :
  - La page doit être en anglais. (**Obligatoire**)
  - Utilisez les "Code blocks" de Markdown (bash pour les lignes de commandes, python pour les exemples de codes, latex pour les équations, etc.). Guide ici (**Bonus**)
  - Inspirez-vous de l'existant, soyez créatif mais surtout clair et "user friendly". (**Conseillé**)

4 Considérations pour le code

- Organisez les fichiers comme vu dans le cours. (**Obligatoire**)
- Utilisez au maximum la POO. (**Obligatoire**)
- Respectez au maximum les recommendations PEP 8. (**Bonus**)
- Petit conseil : Lisez tous le PEP 8 avant de commencer à coder. Que tout le groupe partage le même style. (**Conseillé**)
- Le code (variables, fonctions, classes) doit être en anglais. (**Bonus**)
- Les commentaires peuvent être en français ou en anglais. (**Bonus**)
- Prenez du temps à bien nommer vos variables. Si ça vous paraît clair sur le moment, ça le sera moins dans une semaine ou pour un autre contributeur.(**Fortement conseillé**)

5 Bonus

- Ajout du Licence (MIT / Apache / etc.) Guide ici.
- Ajout de test unitaires .