

Sujet de projet informatique – UE 2.4 - Année 2021/2022

Titre : Votes alternatifs

Proposé par : LEMAITRE JérémY

Sujet libre proposé par des élèves : oui

Si oui :

Binôme : SIMPORE Sheick, LEMAITRE JérémY

Groupe : 5

A déposer sur Moodle : <https://moodle.ensta-bretagne.fr/mod/assign/view.php?id=57506>

Description du sujet (1 à 2 pages max)

À l'approche des élections présidentielles, on peut s'interroger sur la pertinence du scrutin uninominal à deux tours. En effet le marquis de Condorcet mathématicien et philosophe du XVIIIème siècle énonça le principe suivant : « si une alternative est préférée à tout autre par une majorité, alors cette alternative doit être élue ». Ainsi l'alternative élue devrait pouvoir remporter tout duel face à une autre alternative. Des études statistiques montrent que ce ne serait pas toujours le cas en politique et que le mode de scrutin y est pour beaucoup.

Or de nombreux modes de scrutins alternatifs existent (scrutin de Condorcet randomisé, jugement majoritaire, méthode de Borda, méthode de Coombs, ...). Bien que ces derniers ne soient pas parfaits, ils semblent mieux répondre au principe énoncé par Nicolas de Condorcet. Ces méthodes étant souvent difficiles à mettre en œuvre à grande échelle, leur utilisation reste souvent limitée. C'est ici que l'informatique pourrait-être un grande aide.

Ainsi, l'objectif de ce projet serait de créer une application simple d'utilisation et qui pourrait être utilisée pour réaliser des votes au sein d'un groupe lorsque le nombre d'alternatives est supérieur à 3. L'application devra permettre de choisir entre plusieurs modes de scrutins alternatifs. L'utilisateur devra pouvoir :

- créer une élection sur un thème donné
- choisir un mode de scrutin parmi ceux proposés
- inviter des électeurs à voter de façon unique.

Afin de s'assurer que l'électeur ne vote pas deux fois, on mettra en place un moyen d'authentification. Le programme devra donc comporter a minima les classes :

- Scrutins → Classe mère qui regroupera tous les modes de scrutin
- Une classe par mode de scrutin → Qui héritera de la classe « Scrutins »
- Election → Caractérise une élection avec notamment le système de vote, le nombre et l'identité des électeurs, la date de fin de vote, ...
- Bulletin → Représente un unique vote par un électeur

Une fois tous les bulletins récoltés, ou après une certaine durée définie par l'utilisateur, l'élection pourra être clôturée automatiquement. L'application devra alors diffuser les résultats à l'ensemble des électeurs et proposer les statistiques relatives au vote. Par ailleurs, il pourrait être intéressant de s'intéresser aux moyens de sécuriser le vote pour assurer que les bulletins soient anonymes et non truqués.

Remarque : Il sera possible de s'inspirer d'implémentations existantes pour afin de réduire le temps de développement. Voir GitHub ou autre.

Figures imposées

- Factorisation du code : au moins trois modules et noms de classes distincts
- Documentation et commentaires du code
- Tests unitaires : (au moins 4 méthodes avec au moins 2 cas testés par méthode)
- Création d'un type d'objet (classe) : il devra contenir au moins deux variables d'instance
- Accès BDD (serveur BDD à valider avec l'encadrant)
- Structure de données dynamique (autre que celles intégrées à Python)
- Héritage au moins entre deux types créés

Partie Interface Homme Machine (IHM)

- PyQt5