Projet : simuler un bureau de vote

1 Sujet

L'objectif du projet est de simuler le fonctionnement d'un bureau de vote lors d'une élection[1]. Le président du bureau est le garant des règles. La liste électorale est la liste des personnes inscrites dans un bureau. Seules ces personnes peuvent voter le jour J, au plus une fois. Pour cela, il faut passer dans trois espaces consécutifs :

— La table de décharge est une table sur laquelle sont posés les bulletins de vote regroupés par candidat de l'élection. Dans cet espace, une personne doit prendre au moins deux bulletins de candidats différents et possiblement un bulletin blanc. Soit D_d la durée passée pour effectuer cette tâche. Il ne peut pas y avoir deux personnes en même temps dans cet espace.

— Un bureau comporte au moins un isoloir qui permet à une personne de choisir son bulletin de vote parmi ceux qui ont été pris sur la table de décharge. Les autres bulletins sont jetés à la poubelle. Soit D_i la durée que passe une personne dans cet espace.

— La table de vote est une table où se trouve le président. Dans cet espace, une personne dépose son bulletin dans l'urne et signe la liste d'émargement. Soit D_v la durée que met une personne pour voter. Il ne peut pas y avoir deux personnes en même temps dans cet espace.

La boucle principale du programme incrémente un temps discret $t = 1, 2, 3, \ldots, t_{max}$. Les durées mentionnées ci-avant sont mesurées en unités de temps. À chaque itération, on simule une étape du fonctionnement du bureau de t à $t + 1$. Plus précisément, on fait entrer ou non une personne dans le bureau selon une stratégie à déterminer et on fait évoluer la table de décharge, les isoloirs et la table de vote. Il peut rester des personnes dans le bureau après t_{max}, ainsi il faut continuer la simulation jusqu'à ce que tout le monde en sorte. La fin de l'élection est le moment de publier les résultats du bureau par ordre décroissant du nombre de suffrages obtenus pour chaque candidat, sans oublier le décompte des bulletins blancs et des bulletins nuls.

Complétons le cahier des charges :

— Une personne est définie par un nom, un prénom, un identifiant unique et une sensibilité politique. L'identifiant unique est un entier généré automatiquement. La sensibilité politique est un entier entre 1 et 10 de la gauche vers la droite utilisé pour choisir les candidats.

— Une élection a un nom et une liste de candidats. Un candidat est simplement une personne.

— La liste électorale est une liste de personnes sans doublons triée par ordre alphabétique. L'opération d'insertion et le test permettant de savoir si une personne est inscrite sur la liste doivent s'effectuer en $O(\log(n))$ où n est la taille de la liste.

— La liste d'émargement est une liste d'identifiants de personnes. Le test d'appartenance à une liste doit avoir une efficacité maximum.

— On appelle électeur une personne entrée dans un bureau à laquelle on associe des attributs permettant de gérer le processus de vote : une durée, des bulletins, un choix, *etc.* Par exemple, la durée est assignée à D_v quand l'électeur arrive à la table de vote. Cette durée est décrémentée quand le temps passe de t à $t + 1$. L'électeur doit voter puis sortir de l'espace quand la durée est nulle.

1. Consulter les sites `https://www.vie-publique.fr` et `https://www.interieur.gouv.fr`

Trace d'exécution

```
ELECTION 'Euro 2024'
   Candidat n°0 : X nel 5
   Candidat n°1 : Y rik 2
   Candidat n°2 : Z pat 8

BUREAU n°258
LISTE ELECTORALE
   A bic (0)
   B gad (1)
   C ann (2)
   D pol (3)
   E lam (4)
   F bul (5)
   G yap (6)

PREPARATION DECHARGE
   X nel : 7 bulletins
   Y rik : 7 bulletins
   Z pat : 7 bulletins

OUVERTURE BUREAU n°258
TMAX = 20

T = 1
T = 2
   ENTREE
      1 entre
   DECHARGE
      1 entre
T = 3
T = 4
   ENTREE
      2 entre
T = 5
   DECHARGE
      1 prend Z pat (*)
      1 prend BLANC
T = 6
   ENTREE
      3 entre
   DECHARGE
      1 sort
   ISOLOIR
      1 entre
T = 7
   DECHARGE
      2 entre
T = 8
   ENTREE
      4 entre
T = 9
T = 10
   ENTREE
      5 entre
   DECHARGE
      2 prend X nel
      2 prend Y rik
      2 prend BLANC
T = 11
   DECHARGE
      2 sort
   ISOLOIR
      2 entre
T = 12
   DECHARGE
      3 entre
   ISOLOIR
      1 choisit Z pat (*)
      1 sort
   VOTE
      1 entre
T = 13
T = 14
T = 15
   DECHARGE
      3 prend X nel
      3 prend Y rik
      3 prend Z pat
T = 16
   DECHARGE
      3 sort
   ISOLOIR
      3 entre
   VOTE
      1 vote
T = 17
   DECHARGE
      4 entre
   ISOLOIR
      2 choisit BLANC
      2 sort
   VOTE
      1 sort
   SORTIE
      1 sort
T = 18
   VOTE
      2 entre
T = 19
T = 20
   DECHARGE
      4 prend Y rik

FERMETURE ENTREE

T = 21
   DECHARGE
      4 sort
   ISOLOIR
      4 entre
T = 22
   DECHARGE
      5 entre
   ISOLOIR
      3 choisit Y rik
      3 sort
   VOTE
      2 vote
T = 23
   VOTE
      2 sort
   SORTIE
      2 sort
T = 24
   VOTE
      3 entre
T = 25
   DECHARGE
      5 prend Z pat
T = 26
   DECHARGE
      5 sort
   ISOLOIR
      5 entre
T = 27
   ISOLOIR
      4 choisit Y rik
      4 sort
T = 28
   VOTE
      3 vote
T = 29
   VOTE
      3 sort
   SORTIE
      3 sort
T = 30
   VOTE
      4 entre
T = 31

T = 32
   ISOLOIR
      5 choisit Z pat
      5 sort
T = 33
T = 34
   VOTE
      4 vote
T = 35
   VOTE
      4 sort
   SORTIE
      4 sort
T = 36
   DECHARGE
   ISOLOIR
   VOTE
      5 entre
T = 37
T = 38
T = 39
T = 40
   VOTE
      5 vote
T = 41
   VOTE
      5 sort
   SORTIE
      5 sort

FERMETURE BUREAU n°258
```

Enfin, les résultats sont publiés.

```
BUREAU n°258 : RESULTATS Euro 2024
PRESIDENT : C ann
   nb electeurs : 7
   nb votes : 5
   participation : 71.43%
   abstention : 28.57%
   Y rik : 2 (40%)
   Z pat : 1 (20%)
   X nel : 0 (0%)
   blanc : 1 (20%)
   nul   : 1 (20%)
```

Cette trace a été obtenue avec le paramétrage suivant : $t_{max} = 20$, $D_d = 3$, $D_i = 6$, $D_v = 4$, $p_b = 0,35$ la probabilité de prendre un bulletin blanc, $p_n = 0,15$ la probabilité de rendre un bulletin nul, $n_i = 3$ le nombre d'isoloirs, $d_p = 3$ la distance maximum entre les sensibilités politiques d'un électeur et d'un candidat — un électeur ne peut pas voter pour un candidat trop éloigné.

2 Consignes

Le travail consiste à réfléchir aux structures de données qui doivent être mises en œuvre pour simuler le fonctionnement d'un bureau de vote, à réaliser la conception et la programmation en `C++` d'un logiciel répondant au cahier des charges et à écrire un rapport.

2.1 Organisation et planning

1. Formez un binôme au sein d'un même groupe de TP et inscrivez-vous au plus tard le **vendredi 16 février à 12h** dans le fichier dédié sur `madoc`.
2. Si votre nom n'apparaît pas dans le fichier des inscriptions ou si vous avez changé de groupe de TP, envoyez un mail sans attendre à `Laurent.Granvilliers@univ-nantes.fr`.
3. En cas de difficulté à former un binôme, contactez votre chargé de TP et inscrivez le mot *recherche* dans le fichier des inscriptions en face de votre nom.
4. La date de rendu est fixée au **vendredi 19 avril à 12h**. Déposez sur `madoc` un fichier archive au format `zip` comportant un rapport au format `pdf` et les programmes sources. Pensez à nettoyer les répertoires avant de créer l'archive.

2.2 Le rapport

Le rapport doit être constitué des cinq parties suivantes.

1. Décrivez les différentes classes / structures de données de votre application – un paragraphe par structure expliquant le rôle et le principe général. Illustrez le parcours d'un électeur dans un bureau de vote avec un schéma montrant dans quelles structures de données il va successivement se retrouver.
2. Quelles sont les complexités des opérations de test d'appartenance sur les listes d'émargement et les listes électorales ? Expliquez vos choix de structures de données.
3. Écrivez en pseudo-code la SDA et la SDC choisies pour un isoloir.
4. Écrivez en pseudo-code la SDA et la SDC choisies pour un espace d'isoloirs.
5. Collez des traces d'exécution issues de votre logiciel. Ayez un regard critique sur sa complétude et sa correction. Qu'est-ce qui est implémenté et testé, quels sont les bugs, quelles sont les parties non dévelopées ?

2.3 Absences aux TP

Le projet sera encadré pendant quatre séances de TP. Un appel sera fait à chaque séance. Il ne sera toléré aucune absence non justifiée (au minimum, envoyez un mail à l'autre personne du binôme et à votre chargé de TP).

2.4 Exigences et notation

Les exigences pour une SDC sont les suivantes :

- Au niveau A, la description de la représentation en mémoire est très claire et bien expliquée au moyen de graphiques. Les complexités des algorithmes sont optimales.
- Au niveau B, la description est celle du niveau A et les complexités sont bonnes sans être optimales.
- Au niveau C, la description n'est pas claire mais les complexités sont bonnes.
- Au niveau D, la description est absente ou incohérente ou les complexités sont mauvaises par rapport à ce qui est attendu.

Les exigences pour une SDA sont les suivantes :

- Au niveau A, le rôle de chaque opération ou méthode est bien expliqué. Les signatures sont définies et constituent une bonne interface pour un utilisateur. Les préconditions sont données. On peut utiliser la SDA facilement pour écrire de nouveaux algorithmes.
- Au niveau B, la SDA est globalement bonne mais les explications ne sont pas toujours limpides et l'interface peut être améliorée. La SDA peut quand même être utilisée pour écrire de nouveaux algorithmes.
- Au niveau C, il manque des préconditions, les rôles ne sont pas toujours expliqués clairement et l'interface peut être améliorée.
- Au niveau D, la SDA est inutilisable pour écrire de nouveaux algorithmes. Un utilisateur ne peut pas la comprendre.

Les exigences pour le code `C++` sont les suivantes :

- Au niveau A, le code est clair, bien commenté, bien indenté. Les classes standards de `C++` sont mises en œuvre à bon escient. La compilation ne génère pas de *warnings*. Les méthodes sont élémentaires, le code est bien découpé. Les résultats d'exécution sont corrects, i.e. les algorithmes sont corrects et bien codés.
- Au niveau B, le code est globalement d'un bon niveau mais il y a des faiblesses par rapport au niveau A. Par exemple, certaines méthodes ne sont pas élémentaires. Les résultats d'exécution sont corrects, i.e. les algorithmes sont corrects et bien codés.
- Au niveau C, le code compile mais il est mal commenté ou indenté (résultat d'un développement à la va-vite). Il n'y a pas de plantage à l'exécution mais certains résultats sont incorrects, i.e. certains algorithmes sont faux ou mal codés.
- Au niveau D, le code ne compile pas ou il plante à l'exécution. Le code n'est pas commenté. Il manque des parties du code.

Enfin, un barème est donné ci-dessous à titre indicatif :

- Entre 17 et 20 : niveau au moins B+ en moyenne, niveau A à B dans toutes les catégories ;
- Entre 13.5 et 16.5 : niveau B en moyenne, niveau au moins C+ dans toutes les catégories ;
- Entre 10 et 13 : niveau C+ en moyenne, solution en partie fonctionnelle ;
- Moins de 10 : niveau C ou D en moyenne, projet insuffisant comportant de nombreux manques.