

IUT DE PARIS – DÉPARTEMENT INFORMATIQUE

STRUCTURES DE DONNÉES ET ALGORITHMES FONDAMENTAUX

– PROJET –

1 Présentation du jeu

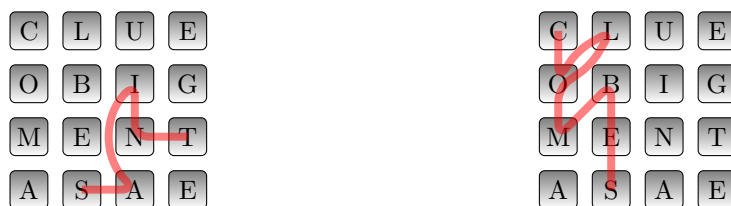
BOGGLE est un jeu de lettres conçu par Alan Turoff et déposé par *Parker Brothers / Hasbro, Inc.* La présentation ci-dessous est partiellement inspirée de WIKIPÉDIA (<http://fr.wikipedia.org/wiki/Boggle>).

Le jeu commence par le mélange d'un *plateau* (carré) de 16 *dés* à 6 faces, généralement en le secouant. Chaque dé possède une *lettre* différente sur chacune de ses faces. Les dés sont rangés sur le plateau 4 par 4, et seule la face supérieure des dés est visible. Après cette opération, un compte à rebours est lancé et tous les joueurs commencent à jouer.

Chaque joueur cherche des mots pouvant être formés à partir de lettres adjacentes du plateau. Ici par "adjacentes", on sous-entend voisines horizontalement, verticalement ou en diagonale. Les mots doivent être de 3 lettres au minimum, peuvent être au singulier et/ou au pluriel, conjugués ou non, mais ne doivent pas utiliser plusieurs fois le même dé pour le même mot. Par exemple, dans le plateau illustré ci-dessous, les mots "LUBIE" et "GITANE" sont effectivement représentés. En effet, chaque lettre composant un de ces mots est adjacente avec la lettre suivante. Le trait bleu permet de visualiser cette liaison entre les lettres.



Par contre, les mots "SAINT" et "COLOMBES" ne sont pas représentés sur ce même plateau. En effet, les lettres 'A' et 'I' ne sont pas adjacentes pour le mot "SAINT" et les deux occurrences de la lettre 'O' du mot "COLOMBES" sont représentées par le même dé. Le trait rouge illustre ces deux cas dans la figure ci-dessous.



Les joueurs écrivent tous les mots qu'ils ont trouvé sur leur feuille personnelle. Après le temps imparti, les joueurs doivent arrêter d'écrire et le jeu entre dans la phase de calcul des points.

Lors de ce calcul, chaque joueur lit à haute voix les mots trouvés. Si deux joueurs ou plus ont trouvé le même mot, il est rayé des listes le contenant. Tous les joueurs doivent vérifier la présence des mots dans la grille. De plus, un dictionnaire est utilisé pour accepter ou refuser un mot (on utilise souvent le dictionnaire du SCRABBLE). Après avoir éliminer les mots communs aux listes des joueurs ainsi que ceux absents de la grille ou du dictionnaire, les points sont attribués suivant le nombre de caractères formant les mots trouvés (voir la table 1). Le gagnant est le joueur ayant le plus grand nombre de points.

2 Format de saisie et d'affichage

Dans le reste de ce document, un *mot* est une suite de caractères contenant au moins un caractère et ne contenant pas de séparateur (*i.e.* un espace, une tabulation ou un retour à la ligne). La longueur des mots ne dépasse pas 30 caractères.

Pour la saisie comme l'affichage, une *liste de mots* est une suite de mots intercalés avec des séparateurs. La liste se termine par le mot spécial * (*i.e.* le caractère étoile).

Longueur du mot	Nombre de points
≤ 2 lettres	0 point
3 ou 4 lettres	1 point
5 lettres	2 points
6 lettres	3 points
7 lettres	5 points
≥ 8 lettres	11 points

Table 1: Nombre de points rapportés par un mot en fonction de sa longueur

Le *nombre de points* associé à une liste de mots est égal à la somme des points associés à chaque mot de la liste.

Une liste de mots est dite *sous forme canonique* si :

- les mots sont triés par ordre alphabétique croissant,
- chaque mot n'apparaît qu'une unique fois et
- chaque mot est suivi par un unique retour à la ligne.

Lorsqu'une *liste de listes de mots* est saisie, la fin de la liste (de listes) est signifiée par une liste vide (*i.e.* une liste dont le premier mot est *). Par exemple, la suite de mots AA BB * CC * * correspond à une liste de deux listes de mots, la première contient les mots AA et BB, et la seconde le mot CC.

Un *plateau* est saisi par une suite de 4 mots de 4 lettres intercalés avec des séparateurs. Par exemple, la suite de mots ABCD EFGH IJKL MNOP correspond au plateau :

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

3 Travail à faire

Plusieurs programmes indépendants vous sont demandés :

1. Lire une liste de mots saisie au clavier et afficher à l'écran le nombre de points qu'elle représente ;
2. Lire une liste de mots saisie au clavier et l'afficher à l'écran sous forme canonique (les mots sont triés, chaque mot n'est présent qu'une fois et les mots sont séparés par un retour à la ligne) ;
3. Lire deux listes de mots saisies au clavier et afficher à l'écran les mots de la seconde liste n'apparaissant pas dans la première ;
4. Lire deux listes de mots saisies au clavier et afficher à l'écran les mots de la seconde liste apparaissant dans la première ; vous pouvez tirer partie du fait que la première liste est saisie sous forme canonique (mots triés et sans répétition, c'est un dictionnaire) ;
5. Lire une liste de listes de mots saisie au clavier et afficher à l'écran la liste de tous les mots apparaissant dans au moins deux de ces listes ;
6. Lire un plateau suivi d'une liste de mots saisis au clavier et afficher à l'écran les mots de la liste étant présents sur la plateau.

Notez que l'ensemble de ces opérations permettent de traiter les coups des joueurs à une partie de BOGGLE. Une fois déterminée la liste des mots joués par plus d'un joueur (5), ces mots peuvent être retirés de la liste de mots de chaque joueur (3). On pourra ensuite réduire ces listes de mots à ceux présents dans le dictionnaire (4) et appartenant au plateau (6) pour enfin afficher le nombre de points de chaque joueur (1). Le programme (2) n'est requis que pour normaliser les affichages (et éventuellement optimiser le chargement d'un dictionnaire (4)).

Notez aussi que si la liste de mots en entrée du dernier programme (6) est un dictionnaire, la liste affichée en sortie correspond au meilleur coup pouvant être joué.

Vous ferez en sorte que ces programmes partagent le plus possible leurs codes. En particulier, le composant permettant de manipuler des listes de mots doit être le même dans tous les programmes mettant en œuvre une liste de mots. Il en va de même pour les composants relatifs aux listes canoniques, au plateau et aux dictionnaires.

Vous n'avez pas le droit d'utiliser les conteneurs de la bibliothèque standard du C++ (tels que `vector`, `stack`, etc). Par contre, vous pouvez employer les conteneur vus en TD/TP (tels que `TabDyn`, `Pile`, etc) et/ou les adapter pour développer vos propres composants.

4 Qui, quoi et quand?

Votre projet doit être fait en binôme. Les groupes de 3 ne seront pas acceptés. Évitez de faire votre projet tout seul (soit vous êtes très fort et des personnes ont besoin de votre aide, soit vous avez des difficultés et il faut vous faire aider).

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est déterminante pour l'évaluation de votre travail. Votre dossier doit être un unique document **pdf** dont la composition est la suivante :

- Une page de garde indiquant le nom et **le groupe** des membres du binôme, l'objet du dossier.
- Une table des matières de l'ensemble du dossier.
- Une brève introduction du projet.
- Le graphe de dépendance des fichiers sources de vos applications. Tous les composants (qu'ils soient réutilisés ou développés) de vos applications devront figurer sur le graphe (*cf.* Cours 4).
- Le listing des tests unitaires de vos programmes (précisez quels tests passent et lesquels échouent).
- Un bilan du projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).
- En annexe, le code complet de vos sources (triez les fichiers selon un ordre logique).

Nous vous rappelons que le critère principal de notation est la structuration de votre code. Votre rapport doit mettre en avant la qualité de celle-ci.

Vous devez déposer une archive **zip** contenant votre rapport ainsi que l'ensemble des fichiers sources de vos applications et vos propres jeux d'essai, le **lundi 4 janvier 2021**.

