

Devoir de Programmation PC2R 2017 - Jeu de Lettres

Description du devoir

But du devoir

Le but du devoir est de réaliser une application clients-serveur permettant à des utilisateurs de jouer à un jeu de lettre multijoueurs (type *Boggle*¹). On fournit à tous les participants connectés la même grille (tirage de lettres d'un plateau de 4x4 lettres). Les participants envoient des mots construits à partir de ces lettres et à la fin du tour, tous les participants marquent des points en fonction de leurs propositions, ensuite le jeu continue pour tous les participants avec une nouvelle grille.

Exigences techniques

L'application doit être réalisée suivant une architecture client/serveur.

La partie client et la partie serveur doivent être chacune écrite dans un langage différent, les deux langages appartenant à l'ensemble {C, Java, OCaml}.

L'utilisation d'autres langages est possible mais soumise à autorisation de la part de l'équipe pédagogique.

Le client et le serveur doivent pouvoir fonctionner indépendamment l'un de l'autre (et pouvoir fonctionner avec le client et le serveur de l'équipe pédagogique). Pour cela ils doivent respecter le protocole décrit plus loin.

Règles du jeu

Déroulement d'une session de jeu

Au moins 1 joueur participe à une session de jeu. Il n'y a, *a priori*, pas de nombre maximum de joueurs.

Une session de jeu comprend plusieurs tours de jeu. Le score d'un joueur persiste au cours d'une session. A chaque tour le serveur propose un tirage de lettres, le même pour tous les participants, selon certaines probabilités.

Un joueur peut rejoindre ou quitter une session à tout moment. Quand un joueur rejoint le jeu au milieu d'un tour, il peut directement participer au tour courant.

Un tour de jeu se déroule ainsi:

- **Début du tour :** Le serveur diffuse à tous les participants l'état courant du plateau de jeu 4x4, c'est-à-dire le contenu des 16 cases qui contiennent toutes une lettre.
- **Phase de recherche :** Un compte à rebours long (3 à 5 min) s'active après cette diffusion, pour permettre aux joueurs de construire des mots à partir de cette grille. A tout moment un joueur peut envoyer un mot au serveur, sous la forme d'une chaîne de caractères contenant ce mot et la trajectoire de lettres pour le construire.
- **Phase de vérification :** Le serveur vérifie que les propositions sont correctes, c'est-à-dire que les trajectoires sont correctes et que le mot proposé existe dans son dictionnaire. S'il y a un même mot proposé par au moins deux joueurs différents, les deux propositions sont considérées comme invalides. Par la suite, il y aura une variante de comptage qui prend en compte le mot proposé en premier. On passe ensuite à la phase de résultat où les scores sont calculés.
- **Phase de résultat:** Les résultats du tour (score de tous les joueurs) sont diffusés aux participants ainsi que les mots proposés par chacun. Après un délai très court (10s), un nouveau tour commence avec un nouveau tirage de lettres.

¹cf. Section "Ressources"

Règles du jeu de lettres

Le plateau du jeu de dimension 4 est une grille carrée de 4 lignes et 4 colonnes contenant 16 cases, dans chacune desquelles se trouve une lettre. Dans le cadre de ce projet les lettres sont en majuscules et non-accentuées. Le but du jeu est de former des mots à partir de lettres adjacentes (horizontalement, verticalement ou en diagonale). Une lettre de la grille ne peut être utilisée qu'une seule fois dans la construction d'un mot. Les mots sont de longueur minimale 3. Toutes les formes de mots sont autorisées : verbe conjugué, singulier, pluriel, féminin, masculin, ... Dans la suite du texte, on parlera de trajectoire pour indiquer la formation d'un mot à partir d'une grille. Une trajectoire correspondra aux différentes positions successives des lettres d'un mot proposé.

Le **tirage** des 4x4 lettres à chaque tour doit comporter une part d'aléatoire. Pour cela le jeu se présente d'une boîte avec 16 dés que l'on secoue pour placer chaque dé sur une case avec une face visible. Dans la version internationale du jeu, les 16 dés sont les suivants :

- E, T, U, K, N, O ;
- E, V, G, T, I, N ;
- D, E, C, A, M, P ;
- I, E, L, R, U, W ;
- E, H, I, F, S, E ;
- R, E, C, A, L, S ;
- E, N, T, D, O, S ;
- O, F, X, R, I, A ;
- N, A, V, E, D, Z ;
- E, I, O, A, T, A ;
- G, L, E, N, Y, U ;
- B, M, A, Q, J, O ;
- T, L, I, B, R, A ;
- S, P, U, L, T, E ;
- A, I, M, S, O, R ;
- E, N, H, R, I, S.

La **grille** suivante correspond à un tirage des dés précédents,



où on peut y trouver les 115 mots suivants.

ADENT DIRENT ILE LETAL RIEN
ADIRE DUE ILET LIE RIENT
ALE DUEGNE IRE LIED TALE
ALENE DUNE JADE LIEN TALED
ATRE EJET JAUNE LIENT TALENT
ATTENUA ELIDA JAUNET LIER TALER
ATTENUE ENJEU JET LIRE TALET
AUNE ENTA JETA NENE TAT
AUNENT ENTAT JETAT NET TATE
AUNER ENTE JEU NUA TATER
DEJA ENTENT JEUDI NUE TEJU
DELA ENTER JEUN NUENT TEL
DELTA ENTRE JEUNE REDU TENGE
DENT ENTRENT JEUNER REDUE TENTA
DENTA ETA JEUNET RENE TENU
DENTAL ETAL JUNTE RENNE TENUE
DENTAT ETAT LATENT RENTA TER
DENUA GENE LATRIE RENTAT TRI
DENUE GENER LATTE RIDA TRIDENT
DENUENT GENET LATTER RIDE TRIE
DER GENT LEI RIDENT TRIENT
DIENE GENTE LENT RIE UNE
DIRE IDE LET RIEL UNTEL

Une **trajectoire** de constitution d'un mot correspond à la position de chaque lettre qui le constitue dans l'ordre de lecture. On note par de lettres les lignes (de A à D, et par des chiffres les colonnes (de 1 à 4). Par exemple, pour le mot "TRIDENT" de 7 lettres de la grille précédente a la trajectoire suivante : C2B1A2A3B2C3D2 où C2 correspond au premier T et D2 au dernier.

Un **mot** est une suite de lettres valides, c'est-à-dire présentes dans un dictionnaire sur le serveur. Ce devoir nécessite donc de mettre à disposition un dictionnaire de mots côté serveur. Il existe des dictionnaires libres qui peuvent être récupérés sur Internet (par exemple le GLAFF <http://redac.univ-tlse2.fr/lexiques/glaff.html>, ou encore <http://www.pallier.org/ressources/dicofr/dicofr.html>) et la création d'un dictionnaire personnel est possible. Une extension facultative proposée (cf. plus bas) est l'écriture d'un serveur qui vérifie les mots dynamiquement sur le Web (plutôt qu'avec un dictionnaire interne).

Le **score** d'un mot doit dépendre de sa longueur. Dans la version classique voici le calcul du score d'un mot :

taille du mot	points
3	1
4	1
5	2
6	3
7	5
8+	11

Si plusieurs joueurs ont trouvé le même mot, celui-ci ne compte pas. Seuls les mots proposés de manière unique entre dans le calcul du score. Dans un tour le score de chaque joueur est augmenté des scores de tous les mots valides qu'il a proposé.

Protocole

Le serveur peut être initialisé avec de nouvelles valeurs pour les compte-à-rebours et éventuellement un fichier cible pour le dictionnaire. Le serveur écoute sur le port 2018 en TCP. Clients et serveur échangent selon un protocole texte. Une commande est composée de chaînes ASCII terminées par des /. La commande elle-même est terminée par un \n. En cas de doute sur la non-ambiguïté ou la complétude du protocole, écrire à l'équipe pédagogique.

Tous les protocoles sont **extensibles** si nécessaire (notamment pour le traitement des extensions). Tous les clients et les serveurs doivent gérer raisonnablement les extensions de protocoles non-reconnues (par exemple, en ignorant les commandes non reconnues). Par exemple, les clients-serveurs doivent être résilients face à une méthode de calcul de score différente entre le client et le serveur (c'est le score du serveur qui prime, évidemment).

Connexion

CONNEXION/user/

(C -> S) Nouvelle connexion d'un client nommé 'user'

BIENVENUE/tirage/scores/

(S -> C) Validation de la connexion de 'user' Envoi du tirage courant et des scores.

CONNECTE/user/

(S -> C) Signalement de la connexion de 'user' aux autres clients.

Déconnexion

SORT/user/

(C -> S) Déconnexion de 'user'.

DECONNEXION/user/

(S -> C) Signalement de la déconnexion de 'user' aux autres clients.

Début d'une session

SESSION/

(S -> C) Début d'une session.

VAINQUEUR/bilan/

(S -> C) Fin de la session courante, scores finaux de la session.

Phase de recherche

TOUR/tirage/

(S -> C) Début d'un nouveau tour, et tirage courant.

TROUVE/mot/trajectoire/

(C -> S) Annonce d'un mot et de sa trajectoire par un joueur.

MVALIDE/mot/

(S -> C) Validation de la proposition d'un mot par le serveur.

MINVALIDE/raison/

(S -> C) Invalidation de la proposition d'un mot le serveur, raison explicite.

RFIN/

(S -> C) Expiration du délai imparti à la réflexion.

Phase de résultat

BILANMOTS/motsproposés/scores/

(S -> C) Bilan du tour, ensemble des mots proposés et validés associés à leur client, scores de tous les joueurs.

Chaînes

- `user` est une chaîne de lettres identifiant un joueur.
- `grille` décrit un tirage de 16 lettres et est composée de 16 caractères comportant les 16 lettres tirées en majuscule. Les quatre premières lettres correspondent à la première ligne, les suivantes à la seconde, ...
- `raison` décrit la raison pour laquelle un placement est refusé. Les chaînes à envoyer sont libres (on peut être aussi explicite que possible) mais doivent commencer par 'POS' (position des lettres invalides), 'DIC' (position des lettres valides mais mot qui n'existe pas dans le dictionnaire)
- `scores` décrit l'état des scores pour la session courante sous la forme `n*user1*score1*user2*score2*...` où `n` est le nombre de tours écoulés dans la session, `useri` parcourt le nom des joueurs et `scorei` les scores.

Tests

Pour faciliter les tests, les commandes de lancement du serveur et du client devront pouvoir utiliser les arguments suivants :

pour le serveur :

- `-port numport` : précisant le numéro de port du serveur
- `-tours n` : indiquant qu'il y aura `n` tours par session
- `-grilles grille1 ... grillen` : indiquant les `n` grilles de tests, chacune sous forme d'une chaîne de caractères de 16 lettres.

pour le client :

- `-serveur hostname` : indiquant le nom du serveur ou son adresse IP
- `-port numport` : précisant le numéro de port du serveur

Extensions

Il est probable que la réalisation d'extension(s) implique un enrichissement du protocole. Il est important que, au maximum, clients et serveurs bénéficiant d'extensions soient compatibles avec ceux n'en bénéficiant pas. En dernier recours, il est possible de rendre deux versions des clients et serveurs ("compatible" et "enrichie").

Extensions nécessaires

Pour être bien évalué, le projet doit implanter les extensions suivantes:

- **Chat:** Le client et les serveurs gèrent un minisalle de clavardage qui permet aux clients de communiquer entre eux à tout moment (quelque soit la phase de jeu)
`ENVOI/message/`
(C -> S) Envoi (public) d'une chaîne de caractère "message" à tous les joueurs.
`PENVOI/user/message/`
(C -> S) Envoi (privé) d'une chaîne de caractère "message" au joueur "user" uniquement.
`RECEPTION/message/`
(S -> C) Réception d'un message public.
`PRECEPTION/message/user/`
(S -> C) Réception d'un message privé de l'utilisateur "user".
- **Vérification immédiate :** Dès la soumission d'un mot, le serveur vérifie qu'il est valable (bonne trajectoire et existant dans le dictionnaire) et l'indique au client. Si un autre joueur propose le même mot, celui est refusé en indiquant une raison qui commence par 'PRI'. On lancera le serveur avec l'option `-immédiat` pour cette vérification immédiate.

Extensions facultative

Pour obtenir la note maximale, le projet doit implanter des extensions supplémentaires parmi les suivantes (ou d'autres extensions non prévues dans ce sujet):

- **Dictionnaire distant** : accès à un dictionnaire sur le web. Le serveur ne stocke pas les mots valides mais fait appel à un serveur web distant (existant ou programmé par les étudiants) qui i) contient un dictionnaire ou ii) teste l'existence d'un mot.
- **Persistance** : système d'authentification pour les clients, mise en mémoire des statistiques des parties précédentes, possibilité de rejoindre une partie quittée précédemment.
- **Journal**: publication régulière des résultats des dernières parties sur une page web.
- **Client autonome** : programmation d'un client tricheur qui calcule automatiquement une solution (voire la meilleure). C'est encore mieux si le tricheur peut se faire passer pour un humain (comportement crédible, messages de chat).
- **Client graphique** : représentation graphique du cube *Boggle* et interface permettant de dessiner sur le cube la trajectoire d'un mot proposé.
- **Variantes**: implantations de variantes (existantes ou inédites) du jeu.

Points d'intérêt

L'évaluation d'un projet s'intéressera (entre autres) aux points suivants:

- **Qualité de la gestion de la concurrence**: le serveur doit gérer plusieurs threads clients qui accèdent de manière sûre et efficace à des ressources partagées.
- **Respect du protocole** et compatibilité avec les clients/serveurs des autres groupes.
- **Qualité du Rapport**: description et justification des choix d'implémentation et manuel du jeu.
- **Jouabilité**: ergonomie de l'interface client, et qualité des entrées/sorties (paramètres, journal) du serveur.

Ressources

- Wikipedia sur le *Scrabble Duplicate*: <https://fr.wikipedia.org/wiki/Boggle>
- Exemple de version Web du jeu: <http://gregfresnel.free.fr/WebMotus/WebGBoggle.php>