

Mai 2018

Rapport PC2R - Boggle

Amel Arkoub & Ling-Chun SO

Table des matières

Introduction	2
Le sujet	3
Quelque liberté.....	3
Modification apportée au sujet.....	3
Les extensions	3
Description algorithmique du client autonome	3
Le serveur	4
Le client	5
Fiche d'utilisation du serveur et du client	6
Le serveur	6
Le client	6
Conclusion	7

Introduction

Nous souhaitons créer notre propre jeu de lettres multijoueur en ligne, de type Boggle. Nous avons décidé d'écrire le serveur en C et le client en Java. Ces choix se justifient de la manière suivante. Maîtrisant le C et ayant déjà eu l'occasion d'écrire des serveurs dans ce langage, c'est de façon naturelle que nous optons pour ce langage. Quant au client, Java propose la bibliothèque d'interface graphique JavaFX, qui est facile d'utilisation et dont l'esthétisme final est incontestable.

Le sujet

Le protocole de l'énoncé est respecté.

Quelque liberté

Modification apportée au sujet

Nous prîmes l'initiative de considérer uniquement la vérification immédiate des mots envoyés par le client au serveur. Ainsi, il est inutile de lancer le serveur avec l'option – ***immédiat***, le serveur vérifie toujours que le mot envoyé par le client fait bien partie du dictionnaire.

Les extensions

Toutes les extensions obligatoires, à savoir le chat et la vérification immédiate, sont implantées.

Concernant les extensions facultatives, vous pourrez trouver un client graphique, c'est-à-dire le cube et la représentation de la trajectoire, ainsi qu'un client autonome, tricheur absolu et imbattable, qui vient venter son intelligence incontestable sur le chat. De plus, si vous le souhaitez, les grilles peuvent être générées automatiquement.

Description algorithmique du client autonome

Le serveur

Le processus principal

Une fois le serveur lancé, le processus principal attend sur deux descripteurs de fichier : un correspondant à la socket d'écoute sur le port donné en argument (ou sinon celui par défaut), et l'autre sur un timer.

Quand un client veut se connecter, le processus principal va créer un thread qui s'occupera de la connexion avec ce client.

Avec chaque fois que le timer finit, le processus principal va effectuer une tâche. Voici la séquence de tâches réalisées par le processus :

- 1) Au lancement, le processus envoie le message SESSION/
- 2) Au bout de 10 secondes, le timer envoie un message au processus via le descripteur de fichier pour lui signaler que le temps imparti est écoulé
- 3) Le processus charge une grille et envoie à toutes les connexions établies avec des clients le message TOUR/tirage/
- 4) Au bout de 3 min, le timer envoie un message au processus
- 5) Le processus envoie RFIN/ à tous les clients
- 6) Les étapes 3, 4 et 5 se répètent autant de fois qu'on a de tours par session
- 7) Le processus envoie VAINQUEUR/bilan/
- 8) On retourne à l'étape 1

Le thread s'occupant des échanges avec un client donné

Le thread reçoit les différents messages envoyés par son client, et lui envoie la réponse adéquate.

Le client

Fiche d'utilisation du serveur et du client

Le serveur

Un makefile est disponible dans Boggle_PC2R/Boggle_Serveur.

Pour lancer le serveur avec les options de vous voulez, veuillez entrer les commandes suivantes dans une console, à partir de Boggle_PC2R/Boggle_Serveur :

- make
- ./bin/serveur [-port] *num_port* [-tours] *nb_tours* [-grilles] *grille1 ... grilleN*

On note que chaque option est facultative, et les valeurs par défaut sont :

port=2006, tours=3, et les grilles sont générées automatiquement

Ainsi vous pouvez lancer le serveur en ne faisant que :

- ./bin/serveur

Ou bien une composition de votre choix, comme par exemple :

- ./bin/serveur -tours 10

Si vous optez pour l'option -grilles *grille1 ... grilleN*, sachez que ces grilles seront proposées en boucles.

Le client

Conclusion