

Rapport Programmation Système

BERNHARDT Dylan & BUNAUX Corentin

Notre application

Notre application est un jeu multijoueur. Le but est le suivant : écrire une phrase plus rapidement que ses adversaires.

Nous voulions programmer quelque chose qui nous plaisait à tous les deux. De nombreux sites web proposent déjà des applications similaires. On retrouve par exemple:

- 10fastfingers.com
- monkeytype.com

Cependant, ces sites n'offrent que la possibilité de jouer seul. Nous avons alors eu l'idée de créer un jeu qui permettait de s'entraîner à la dactylographie avec ses amis.

C'est comme ça que nous est venue l'envie de créer cette application, qu'on a nommée Dactylo Race.

Fonctionnement de l'application

Notre application fonctionne comme la majorité des jeux en ligne. Un serveur fonctionne en continu et accepte les connexions. Les joueurs se connectent et si le nombre maximal de joueurs n'est pas atteint, alors ils peuvent rejoindre la partie.

Chaque joueur qui se connecte est invité à renseigner un pseudonyme qui permet de l'identifier lors du classement final.

Ensuite deux cas peuvent arriver :

- Il est le dernier connecté et sa connexion lance directement la partie.
- Il n'est pas le dernier connecté et il est mis en attente d'autres joueurs.

La partie ne se lance que lorsque le nombre de joueurs (4 par défaut) est atteint.

Chaque joueur voit ensuite un décompte apparaître à son écran qui signifie que la partie va commencer. Au bout du décompte, une phrase apparaît à son écran. Il doit alors la taper le plus vite possible sans faute pour gagner !

Lorsque tous les joueurs ont fini d'écrire correctement la phrase, un classement est affiché sur leurs écrans.

Enfin, ils sont invités à rejouer. S'ils acceptent, ils pourront participer à la prochaine partie. S'ils refusent, le programme s'arrête et d'autres joueurs peuvent prendre sa place pour la prochaine partie.

Structures utilisées en lien avec le cours

L'application fonctionne à partir d'un système de multi-threads.

Le serveur abrite une liste de travailleurs (workers) de taille fixe, et attribue les canaux de communication selon l'arrivée et le départ de joueurs. Il y a autant de workers que le nombre de joueurs maximum.

Les threads sont créés dès la mise en route du serveur, et communiquent avec ce dernier via un socket.

A travers lui, de nombreuses informations circulent (pseudos des joueurs, messages de synchronisation, classement des joueurs en fin de partie, etc...).

Les barrières sont également utilisées dans le but de synchroniser l'affichage de la phrase à écrire en début de jeu, ou encore le classement de fin de partie.

Difficultés rencontrées et solutions

La programmation du jeu ne nous posa aucun souci majeur tant que l'option "rejouer" n'était pas implémentée. Nous avons en effet passé beaucoup de temps à développer cette fonctionnalité.

La gestion des canaux de communication a été traitée de sorte que les canaux se libèrent lorsqu'un joueur quitte la partie. Dans le cas contraire, les informations relatives au joueurs restent inchangées, sauf son rang qui est réinitialisé.

La répétition du jeu s'effectue au moyen d'une boucle while dans le serveur. Côté client, une même boucle while s'assure du bon déroulement du jeu, dont le flag de fin ne change que si l'intéressé fait part de sa volonté de quitter le jeu.

L'utilisation des barrières fut également une des principales difficultés rencontrées. En effet, la synchronisation entre le serveur et les clients ne s'effectuait pas lors des phases d'attente d'adversaires ou lors de l'affichage du classement final.

A faire...

L'application n'est cependant pas optimale puisqu'il reste de nombreuses fonctionnalités que l'on pourrait implémenter avec plus de temps.

On retrouve par exemple la possibilité d'entrer le nombre de joueurs au démarrage du serveur. Ceci permettrait d'avoir un nombre de joueurs variable pour différentes parties.

Un des problèmes qui n'a pas pu être traité est la redondance des pseudonymes. En effet, il est possible d'utiliser plusieurs fois le même "gamertag" dans une même partie. La partie qui nous a été la plus problématique pour parfaire ce manque était la communication sans cesse entre le serveur et le worker (on ne savait pas combien de fois le serveur demandait un nouveau pseudonyme, différent de ceux déjà renseignés par les autres joueurs).

Un autre problème fut la pluralité des phrases proposées à la rédaction. La difficulté ne résidait pas dans l'extraction de plusieurs phrases à partir d'un dictionnaire, mais plutôt l'affichage d'une même phrase entre les workers, qui de plus changeait entre les parties.
