

Octochat
Le chat décentralisé

Alexis GIRAUDET Benjamin SIENTZOFF

26 novembre 2014

Table des matières

1	Utilisation et fonctionnement global	3
1.1	Compilation du programme	3
1.2	Utilisation	3
1.3	Sous le capot	4
2	Patrons de conception	5
2.1	Observer	5
2.2	Factory method	5
2.3	Interpretor	5

Introduction

Ce projet a été réalisé dans le cadre du cours *Objet et développement d'applications* dans lequel M. RICHOUX nous a enseigné l'utilisation des *Design Patterns*. L'ambition de ce projet ne s'arrête pas là, car nous souhaitons poursuivre le développement de notre programme. Le sujet de notre projet est la création d'un client de chat qui n'utilise pas de serveur principal comme c'est le cas pour ce genre d'application réseau. Le fonctionnement est détaillé plus loin.

Octochat, notre programme, est donc un client de chat qui n'a pas besoin de serveur pour fonctionner. Lancer le programme, choisissez un nom d'utilisateur est c'est parti.

1 Utilisation et fonctionnement global

Boost Pour des questions de dépendances et pour faciliter le développement du programme, notamment pour ce qui est du réseau, nous avons choisi d'utiliser la librairie *Boost*. L'utilisation de cette librairie nous permet également d'utiliser le système de compilation associé.

1.1 Compilation du programme

Compilation Notre programme utilise deux bibliothèques à savoir la STL et Boost, plus précisément : Boost.Build : le système de compilation (équivalent de make et des Makefile mais plus portable) Boost.Thread : pour les thread et les mutex Boost.Log : pour les log Boost.Asio : pour les entrées/sorties asynchrones sur le réseau Boost.Serialization : pour sérialiser les données envoyées sur le réseau Boost.System : pour les Smart Pointer et le Lexical Cast

Pour compiler notre programme nous avons donc besoin d'un compilateur (incluant la STL) et d'installer Boost, c'est pourquoi nous avons réalisé un Makefile qui s'occupe d'installer Boost localement et de compiler le programme automatiquement. Une fois la compilation terminée, les executables sont placés dans le dossier build :

- octowatch écoute le réseau
- octoglobalchat propose de chatter avec toutes les paires connectées
- octochat permet de chatter avec des utilisateurs

Remarque Il est possible d'installer Boost avec un gestionnaire de paquets à condition d'avoir les privilèges suffisants.

Pour compiler le projet, on commence par cloner le dépôt, puis on lance la commande make à la racine du projet.

```
$ git clone https://github.com/blasterbug/Octochat.git
$ cd Octochat
$ make
```

Si toutes ces étapes se sont bien passées, vous êtes maintenant en mesure d'utiliser Octochat.

1.2 Utilisation

Pour lancer l'application taper simplement `./octochat`

1.3 Sous le capot

2 Patrons de conception

2.1 Observer

2.2 Factory method

2.3 Interpretor

Conclusion

je conclu