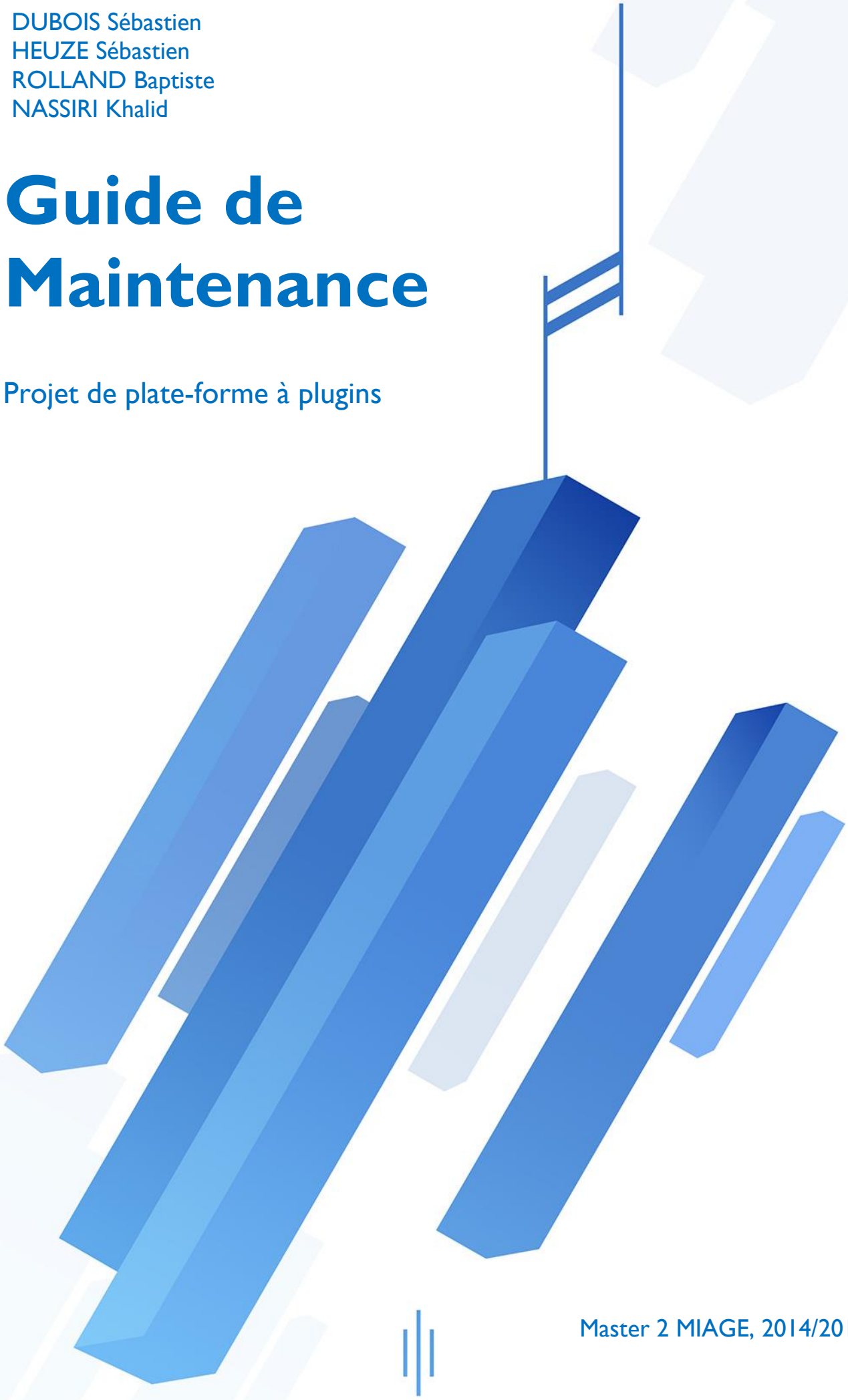
DUBOIS Sébastien
HEUZE Sébastien
ROLLAND Baptiste
NASSIRI Khalid

Guide de Maintenance

Projet de plate-forme à plugins



Master 2 MIAGE, 2014/2015

Table des matières

Préambule

Ce document présente les instructions nécessaires à la maintenance de la plateforme à plugins Age of Mottu. Ce projet a été réalisé dans le cadre de travaux pratiques en architecture logicielle. L'application en l'état, permet de définir un ensemble de plugins autonomes ou dépendants permettant la construction d'applications en décentralisant l'ensemble des développements permettant de remplacer tout ou partie des composants d'une application.

Nous aborderons donc, dans un premier temps, les choix de conception puis les outils employés pour terminer sur les difficultés rencontrées et les perspectives de la plateforme. Toutefois, l'ensemble des sujets abordés dans ce document sont plus détaillés dans le guide de développeur. Il s'agit donc de considérer ce document comme une introduction à ce dernier document.

Choix de conception

La configuration des plugins (le fichier 'config.txt')

Comme présenté dans le guide de développeur, un plugin peut être configuré de façon poussée. Ce projet étant seulement un prototype, nous avons tout de même choisi d'enrichir les informations des plugins.

Nous avons souhaité garder la configuration des plugins dans un fichier plat 'config.txt'. Dans ce fichier, nous avons choisi de structurer une ligne plugin de cette façon :

- La classe du plugin étant appelée (ex : Lanceur)
- L'interface implémentée par le plugin (ex : ILauncherPlugin)
- Le type de Plugin (ex : LANCEUR)
- Le nom du Plugin (ex : PluginLanceur).
  Ce nom représenta aussi le chemin relatif du dossier du plugin.
- L'état du Plugin (ex : true s'il est disponible, false sinon)

Chargement des plugins

Le chargement des plugins se déroule en plusieurs étapes. En premier lieu, au lancement de notre plateforme, la liste des différents plugins disponibles est construite à partir du fichier de configuration.

Au lancement de la plateforme, uniquement le pluginLanceur est chargé, c'est notre pluginPrincipal. Lorsque des manipulations faites par l'utilisateur devront interroger des plugins secondaires, ceux-ci seront chargés (si cela n'est pas déjà fait) et pourront être utilisés par la suite.

Par ailleurs, le but étant de produire une plateforme totalement autonome, si toutefois un développeur souhaite créer d'autres plugins, ceux-ci pourront facilement être chargé via notre plateforme, en ayant au préalable renseigné le plugin dans notre fichier de configuration.

Gestion de l'interopérabilité

Nous n'avons pas de dépendance entre les différents plugins. Le fichier de configuration définit les interfaces imposées pour chacun des plugins dont il dépend et d'autre part, le plugin devant être « casté » pour être utilisé, il doit impérativement respecter ces règles d'implémentation.

Configuration de la plateforme

Une fois un ensemble de plugins déployés sur la plateforme, cette dernière doit être configurable afin de définir les choix d'activation de plugins :

- Quels plugins seront utilisés ?
    - En effet, dans notre fichier de configuration, le dernier attribut représente si nous souhaitons que la plateforme identifie ou non le plugin.

Cette configuration est possible grâce au fichier « config.txt » présent à la racine du projet.

Contrairement à ce que nous avions fait lors des séances de TPs précédentes, nous avons souhaité aller un peu plus loin dans l'implémentation de notre plateforme. Nous avions débuté par instancier les classes via les fichiers .class présents dans le répertoire bin des différents plugins. Par la suite, nous avons préféré travailler avec les .jar des plugins.

Outils employés

Dès les premières étapes de développement, nous avons choisi d'utiliser les outils suivant :

- Github : Pour le développement collaboratif et la gestion des sources.

- Lombok : Nous avions commencé par utiliser lombok, dans le but de nous faciliter dans les développements et surtout la lisibilité du code. https://projectlombok.org/features/index.html pour plus d'informations.
  Par la suite, nous nous sommes rendu compte que lombok posait des problèmes sur les ordinateurs du CIE.

Nous n'avons pas développé de tests unitaires, puisque nous avons jugé que ça ne faisait pas parti du but principal de ce projet. Nous avons préféré notre temps sur l'implémentation la plus générique possible de notre plateforme, ainsi que l'enrichissement de nos plugins. Par exemple, l'interfaçage graphique pour afficher nos données (ici, grille de jeu + ressources …).

Difficultés rencontrées

Durant l'élaboration de la plateforme de démonstration, nous avons rencontré les difficultés suivantes :

- Définir clairement l'architecture et la structure de la plateforme. En effet, il a été difficile pour nous de comprendre comment rendre la plateforme totalement autonome, et de pouvoir gérer divers plugins. Pour cela, il nous a fallu revoir certains principes fondamentaux comme les interfaces.

- Trouver un sujet qui puisse utiliser pleinement les capacités de notre plateforme.

Perspectives

Nous sommes satisfaits des travaux réalisés sur ce projet qui présente, selon nous, une bonne base de travail, l'essentiel des fonctionnalités demandées et nous a permis de comprendre l'usage de bonnes pratiques et le découplage des applications en général.

Le point positif est que nous avons réussi à construire un certain nombre de plugins gravitant autour de notre plateforme, qui permettent de bien montrer l'utilisation de celle-ci, et qui pourra nous servir lors de l'implémentation de futurs projets.

Nous pensons que c'est un très bon exercice pour comprendre les principes d'une plateforme autonome, il est dommage que cet exercice ne nous soit donné qu'en fin de cursus scolaire.