

# Présentation générale du logiciel 'jfMindMap' (jfMM en abrégé)

## 1) Avant tout signalons une exception dans les conventions de notations

Tout au long de ces explications notre programme est nommé 'jfMindMap', c'est à dire qu'il commence par 2 minuscules 'jf'.

Cependant, veuillez noter une exception : sa classe principale, au lieu de se nommer 'jfMindMap' est appelée : JFMindMap. Cette irrégularité est imposée par la syntaxe du langage Java qui préconise que les classes commencent par une majuscule.

## 2) Pour quels usages jfMindMap est-il conçu ?

### a) Il est à la fois un traitement de texte et un logiciel de présentation

#### Sorte de traitement de texte il permet de :

- Lire, afficher chacun des fichiers '.lsp' constituant la substance de chacun de mes cours à la Philotechnique.
- Présenter les documents sous la forme indentée des programmes structurés : Lisp, C, Python, Java...

Plus précisément, à la manière de la présentation d'un document dans un traitement de texte, il permet d'afficher le plan d'un exposé sous la forme d'un arbre, sous la forme de N paragraphes qui se décomposent en K sous paragraphes. Ainsi le texte de l'exposé à présenter est distribué entre les nœuds de cet l'arbre et ses feuilles terminales.

#### **jfMindMap est un logiciel de présentation, très léger et très simple**

En tant que logiciel de présentation, il affiche des illustrations (photos, dessins, schémas, images) :

Pour cela, en fin de n'importe quelle ligne de l'éditeur jfMM, on écrit un lien vers une illustration,. Ce lien est très visible : sur la droite des explications, il est introduit par le caractères '\$' qui lui-même est entouré d'un cadre vert fluo. Quand on le clique, jfMM escamote le texte arborescent et affiche l'illustration pointée par le lien. Ex : [\\$/jfMindMap/pjVuGlo\\_56-1.png](#).

### b) jfMM est destiné à 2 types d'utilisateurs et chacun d'eux en fait un usage spécifique

De retour chez lui, un auditeur de la Philotechnique peut visualiser son cours à la maison. Il consulte le plan, lit les explications et clique sur les liens qui affichent les illustrations correspondantes.

Pour un conférencier, il permet de planifier une présentation, rédiger un texte en rapport et y lier les illustrations de son propos.

Note : ce fonctionnement en mode 'présentation' n'est pas prévu dans la prestation gratuite que je fournis, pour cette raison, cette utilisation de jfMM est livrée SANS support technique, SANS documentation et SANS garantie du codeur.

## 3): Le problème de l'encodage des fichiers qui diffère entre Linux & Windows

### a) Expliquer à quoi correspond l'encodage des fichiers (file encoding)

Au plus profond de l'ordinateur, les données traitées sont représentées en binaire (0 et 1). Puis vient le code ASCII qui permet de représenter ces données (chiffres, lettres, textes, nombres) en binaire.

Initialement l'ASCII encodait les caractères américain (chiffres, lettres, ponctuation, tabulation...) sur 8 bits. Ensuite il a fallu l'étendre pour coder les caractères internationaux. Ça a donné différents encodages de fichiers sur 16 bits.

Évidemment Windows utilise l'encodage ANSI (cp1252) qui est différent de celui de l'UNICODE de Linux (UTF-8). Et la JVM (Java Virtual Machine) tourne exclusivement avec l'unicod UTF-8. Alors, nativement, les \*.lsp que j'utilise pour les cours sont en UTF-8.

### b) Là où le problème apparaissait :

#### Position du problème :

Quand je fournis un support de cours en PDF, vous tous pouvez le lire car c'est la vocation d'un pdf d'être un outil de communication inter-plateformes.

Quand un utilisateur Linux travaille avec mes \*.lsp, il peut les ouvrir et sauver ses modifications car Linux aussi fonctionne avec l'UTF-8.

**Il en va de même pour un utilisateur Windows qui utilise jfMM pour éditer MES \*.lsp**

En effet, pour Windows, j'ai programmé jfMM afin qu'il puisse ouvrir mes fichiers \*.lsp, les éditer et les sauvegarder (tout ça en UTF-8).

### Reste un problème sous Windows, au cas où un utilisateur veut créer SES PROPRES fichiers \*.lsp

Rappel : ce type de fonctionnement n'est pas prévu dans la prestation gratuite fournie, donc n'est pas soutenu ni documenté.

Mais revenons à cet usager : pour créer ses propres fichiers il doit utiliser un éditeur capable de créer des fichiers en encodage UTF-8. Ce n'est pas un drame car ce genre d'éditeur de texte existe. Mais il reste que quelqu'un qui ne serait pas averti du fonctionnement de mes \*.lsp en mode UTF-8 serait déconcerté par les problèmes qui apparaîtraient à cette occasion !

## 4) Installation du logiciel

### A) Avant toutes choses, il vous faut disposer de Java :

Note : Java est le pendant de C# (C sharp) de Microsoft, mais Java est gratuit pour les particuliers, C# est payant.

Peu importe la version que vous installez : ça peut être : Java1.5, java1.6, 1.7, 1.8...

Comment vérifier si vous disposez de Java ?

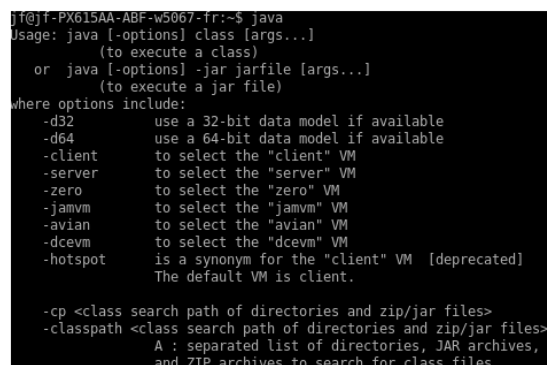
En mode console lancez Java, C'est à dire tapez la commande 'Java' suivie de Entrée :

Si c'est un succès si vous obtenez un long message qui commence comme ceci :

Si c'est un échec si vous recevez un message du genre : "java : commande introuvable".

Pour installer java, reportez-vous à la doc officielle de Sun

Vous n'êtes pas obligés d'installer le gros JDK (Java Développement Kit), le petit JRE (runtime) suffit à jfMM.



```
jf@j1-PX615AA-ABF-w5067-fr:~$ java
Usage: java [-options] class [args...]
           (to execute a class)
 or java [-options] -jar jarfile [args...]
           (to execute a jar file)
where options include:
-d32          use a 32-bit data model if available
-d64          use a 64-bit data model if available
-client      to select the "client" VM
-server      to select the "server" VM
-zero        to select the "zero" VM
-jamvm       to select the "jamvm" VM
-avian       to select the "avian" VM
-dcevm       to select the "dcevm" VM
-hotspot     is a synonym for the "client" VM [deprecated]
              The default VM is client.

-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar files>
           A : separated list of directories, JAR archives,
           and ZIP archives to search for class files.
```

### B) Afin de pouvoir installer jfMM,

Vous devez connaître le mot de passe 'administrateur' car nous copions des fichiers exécutables sur le disque dur.

### C) Vous devez aussi disposer des 3 dossiers fournis :

1) Le dossier '\$' qui contient toutes les images illustrant l'exposé (png, bmp, jpg) :

Votre intervenant (conférencier) vous a fourni ces images classées dans une arborescence, conservez-là sinon l'affichage des liens sera gravement perturbé.

2) Le dossier 'ø19-20' qui contient tous les supports de cours pour l'année scolaire 19-20 :

Pour chacune des leçons de l'année scolaire 19-20, votre intervenant vous fournit les \*.lsp constituant le texte de chacune de ses interventions.

3) Le dossier 'dossierDesClass' qui constitue le code du programme jfMindMap :

Il contient 'JFMindMap.class', le fichier lanceur de jfMM

Note : c'est un exécutable 'Java' qui constitue le point d'entrée de jfMindMap (celui que vous devez lancer pour initialiser le processus).

Il contient les autres fichiers exécutables : 40 autres fichiers avec l'extension 'class'.

### D) Pour installer ce logiciel il faut copier ces 3 dossiers à leur place respective

#### a) Le dossier '\$'

Pour Linux, il doit être impérativement copié dans le dossier '/home/jf'

Pour Windows, il doit être impérativement copié dans le dossier : 'C:/users/jf'

Donc, si ces dossiers n'existent pas sur votre ordinateur, vous devez les créer avant.

#### b) Le dossier 'ø19-20' peut être mis où bon vous semble (vous pouvez même le renommer)

Mais je pense que le mieux serait de le copier dans le dossier qui contient le dossier '\$' afin d'avoir moins à naviguer dans l'arborescence du disque dur !

## c) Le dossier 'dossierDesClass' contient tous les exécutables Java de jfMM, donc :

### Soit vous le copiez à la même hauteur que Java.exe (au-dessus ou au-dessous)

Ces explications sont abstraites, alors voici d'autres façons de dire la même chose : Vous le copiez...

- Dans le dossier qui lui-même contient le dossier 'Java.exe'.

- C'est à dire au même niveau que le JRE (ou le JDK) de Java.

C'est à dire que le plus simple pour installer jfMM est de copier le dossier 'dossierDesClass' dans le dossier où se trouve 'java.exe' (le code de la JVM).

Mais où se trouve 'java.exe', le code de la JVM ? Sous Linux, en ligne de commande, c'est à dire dans une console, on tape : "which java". Et Linux répond : "/usr/bin/java". Sous Windows l'information doit être visible dans le panneau de configuration (la liste des programmes déjà installés).

Expliquer à quoi correspond cette astuce :

En effet, pour lancer JFMindMap, dans une console, on commande au système : java dossierDesClass.JFMindMap

Petite précision sur ce lancement de jfMM : en Java les classes commencent par une majuscule : le point d'entrée de jfMM est JFMindMap et pas jfMindMap. C'est à dire la syntaxe de lancement de jfMM est bien "java dossierDesClass.JFMindMap" et pas "java dossierDesClass.jfMindMap".

Alors le système cherche sur le disque dur où se trouve le fichier 'java.exe' et le lance en lui passant un paramètre : vitAfiArb.MainPr. Par convention ce paramètre indique à Java le point d'entrée du programme (donc ici de 'JFMindMap'). La JVM qui reçoit ce paramètre en déduit donc le point de départ de jfMM, le programme Java à exécuter pour lire le cours et voir les illustrations. Elle va donc à chercher 'JFMindMap' dans le lieu le plus simple pour elle : sous ses pieds, c'est à dire dans le dossier où elle-même se trouve. CQFD.

### Soit vous le copiez là où vous le voulez...

Mais il vous reste à expliquer à votre système (Windows/Linux) où il le trouvera afin qu'il puisse l'exécuter

Sur Linux et les anciens systèmes Windows

Vous devez bricoler le 'path', le 'chemin' de votre système. J'espère que vous voyez ce que je veux dire.

À titre indicatif, sans garantie que ça marche sur votre système, voici un exemple (mais il vous faut le transposer) :

```
path=%path;C:/program files/java/jdk1.6.0./bin;
```

Explication de la ligne au dessus : dans l'ancien path du système, on rajoute le chemin où est installé Java (c'est à dire : C:/program files/java/jdk1.6.0./bin).

Sur les systèmes Windows plus récents... je l'ignore !

## E) Lancement de jfMM

### a) Il se lance au moyen de la commande : "java dossierDesClass.JFMindMap"

### b) Voici quelques explications qui vous aideront à mémoriser cette commande :

'java'

Pour lancer la JVM (Java Virtual Machine) : la machine virtuelle Java qui exécute le code compilé afin d'effectuer le travail.

"dossierDesClass.JFMindMap"

C'est le paramètre qu'on passe à Java, à la JVM. Il se décompose en deux parties :

'JFMindMap'

Car c'est le nom du programme principal du logiciel 'JFMindMap', qu'on demande à la JVM d'exécuter. J'ai obtenu le code compilé 'JFMindMap.class' en compilant le programme principal nommé "JFMindMap.java".

'dossierDesClass'

Car c'est le nom du dossier contenant JFMainPr.class et les autres \*.class, c'est à dire les 40 autres programmes exécutables qui constituent le programme JFMindMap tournant sur la JVM (machine virtuelle Java).

Le point '.' qui sépare ces deux paramètres

En syntaxe Java il signifie que le fichier 'JFMindMap' est dans le dossier 'dossierDesClass'. C'est un peu l'équivalent du slash en syntaxe système.

## 5) Comment me joindre ?

Vers le 18 ou 19 mars 2020, je vais mettre en ligne jfMM et commencera une période de test et de mise au point. Sentez-vous vous donc à l'aise pour m'écrire et me poser des questions qui m'aideront et serviront à tous à

jflucas at free.fr

(remplacer le 'at' par l'arobas @).

Portez-vous bien.

Bien à vous !

Jean-François Lucas.