



UNIVERSITÉ DE NANTES

Architectures modulaires

Architectures logicielles

M2 MIAGE ISI

Manuel architecture de l'application

Le but de ce document est de présenter notre application, celle-ci possède différents dossiers que nous allons détailler dans les différentes parties de notre document :

- src
- librairies
- ressource
- save

1. Dossier source (*src*):

Le dossier source nommé “src” est le dossier contenant les différentes interfaces, les modèles, les affichages de l’application, etc. Nous allons présenter plus en détail les différents fichiers contenus dans ce dossier.

1.1. Package “appli/interfaces”

Le package “interfaces” contient les différentes interfaces des classes de l’application.

1.2. Package “appli/models”

Le package “models” contient les différents modèles des classes de l’application.

1.3. Package “appli/platform”

Le package “platform” contient deux des classes les plus importantes de l’application que nous allons présenter plus en détail :

- Le fichier **DescripteurPlugin.java** : Cette classe nous permet de représenter le plugin en tant qu’objet java. Il possède les mêmes attributs que ceux présents dans le fichier **plugins.json** à savoir : name, classname, description et dependencies (liste de dépendances).
- Le fichier **Loader.java** : Le fichier Loader est le fichier qui sert à charger les différents plugins de l’application. Le Loader sert de classe principale de l’application et elle charge en autorun les plugins décrits dans **autorun.json**. Il permet également de charger des plugins lorsque l’application est en cours d’exécution. Il apparaît alors des changements d’états.

1.4. Package “plugins/action”

Le package “action” contient la classe **GameManager.java**, celle-ci permet de gérer la sauvegarde et le chargement d’une partie.

1.5. Package “plugins/application”

Le package “application” contient la classe **AppClicker.java**, c’est la classe qui nous permet d’instancier le plugin *FarmerClickerDisplay*. Une évolution possible du jeu serait de créer un autre type de “clicker”, par exemple un fabricant de bière, cette classe nous permettrait alors de basculer entre l’un ou l’autre.

1.6. Package “plugins/displays”

Le package “displays” contient la classe **FarmerClickerDisplay.java**. Cette classe est la Vue de l’application, c’est la classe qui gère l’affichage de notre jeu. C’est au sein de cette classe que sont créés les différents boutons, les différentes progressBar, que sont gérés les différents clics de l’utilisateur.

1.7. Package “plugins/factories”

Le package “factories” contient toutes les fabriques utilisées dans notre application. Ces classes nous permettent d’instancier des objets définis dans nos modèles respectivement :

- **Consumables.java**,
- **ProductsCo2.java**,
- **ProductsSimple.java**

1.8. Package “plugins/monitor”

Le package “monitor” contient la classe **Monitor.java**, cette classe correspond à notre Observable. De la même manière que notre classe **IAppClicker.java**, elle est instancié directement dans notre Loader. Ici nous n’utilisons pas l’interface *java.util.Observer*, qui est déconseillée depuis Java 9, mais nous l’implémentons avec *PropertyChangeListener*. Avec cette implémentation, notre observable conserve une référence à l’instance *PropertyChangeSupport*. Une notification est ensuite envoyée aux observateurs lorsqu’une propriété de la classe est modifiée.

2. Dossier *librairies* :

Le dossier “librairies” contient l’ensemble des librairies utilisées au sein de notre application. Cependant, nous n’avons utilisé qu’une seule librairie pour notre jeu, notre dossier contient seulement le fichier .jar de la librairie Gson.

3. Dossier *ressource* :

Le dossier “ressource” contient deux fichiers que nous allons présenter en détail :

- Le fichier **autorun.json** : C’est un fichier qui liste tous les différents plugins qui seront automatiquement lancés au démarrage de l’application, ainsi que leurs dépendances.
- Le fichier **plugins.json** : C’est le fichier listant tous les différents plugins de l’application. Chaque plugin listé possède différents attributs : name, class, description, dependencies.

4. Dossier *save* :

C’est le dossier où sera stocké le fichier de sauvegarde d’une partie. (Voir le Manuel Utilisateur)