

# Assistant de création de plugin

*Application Tamagotchi*

<b>Rédaction</b>	Bastien Motard
<b>Relecture</b>	Benjamin Dutriaux

## Contexte

Notre application est une plateforme de gestion de plugin. Afin de pouvoir profiter de ses fonctionnalités nous avons réfléchi à la gestion d'un personnage dans le même esprit que le jeu tamagotchi.

### Données

Dans ce projet nous vous proposons d'interagir avec des instances de la classe :

**Person** : String name, String firstName, Boolean dead, Integer calories, Integer sleep, Integer wc.

L'ensemble de ces attributs est accessible pour vous permettre de créer des plugins avec.

### Types de plugins

Nous fournissons une plateforme d'utilisation de plugin. Il n'y a donc que cette dernière qui ne soit pas un plugin. Nous avons défini 4 types de plugins, correspondant aux interfaces qui nous avons réalisées.

**Appli** : Les plugins « appli » sont les plugins des applications. Ils appellent la classe qui générera par exemple une fenêtre permettant de faire appel aux autres plugins.

Un plugin appli doit être composé d'une fonction « main ».

**Builder** : Les plugins « builder » sont les plugins permettant de créer une instance de données Person.

Un plugin builder doit être composé d'une fonction « creation »

**Process** : Les plugins « process » sont ceux qui permettent de modifier les valeurs des attributs de Person.

Un plugin process doit être composé d'une fonction « change (Person p) »

**Display** : Les plugins « display » gèrent l'affichage et la sortie des données.

Un plugin display doit être composé d'une fonction « affichage (Person p) »

## Créer un plugin

### Localisation

Dans « /Tamagotchi/src/Plugin/ » développer votre plugin .java.

## *Application Tamagotchi : Création de plugin*

Dans « /ressources/ » ajouter votre fichier de configuration .txt

### **Exemples fournis**

En accompagnement de notre plateforme nous fournissons différent plugin avec leur configuration.

#### **Exemple de configuration :**

```
Class=Plugin.MangerBurger
Name=Manger Burger
Interface=com.Tamagotchi.IProcess
Methode=change
Autorun=false
```

Class : Le nom de la classe du plugin, de la forme Plugin.NomDeClasse.

Name : Nom du plugin pour d'utilisateur, nom compréhensible.

Interface : IAppli, IBuilder, IDisplay, ou IProcess selon le type de plugin réalisé.

Methode : nom de la méthode appelée, dans notre cas : main, creation, affichage, change, selon le plugin.

Autorun : pris en compte seulement pour les plugins de l'interface IAppli. Une seule de ces application peut être en autorun = true.

#### **Exemple de plugins**

Nous fournissons un plugin applicatif dont la configuration est EcranFenetre.txt, qui est une interface utilisateur sous forme de fenêtre permettant de choisir ses plugins.

Un plugin de type display dont la configuration est displayString.txt qui permet un affichage sous forme de chaîne de caractère.

2 plugin builder, un builderRandom créant un personnage aléatoire et un builderTest qui permet de créer un personnage ayant 3000 dans chaque valeur pour permettre des tests.

Plusieurs plugin de process pour permettre de tester la plateforme avec ses données.