



# **Simulateur de Créatures**

*Projet Génie Logiciel – M1 Informatique*

**Aymen BAYA**

**Daniel Arasu ANBARASU**

**Naïma OUNI**

**Yassine YOUSSEF**

## Installation

```
mvn compile
```

```
java -cp bin:. main.Laucher (sur Unix/Mac)
```

```
java -cp bin;. main.Launcher (sur Windows)
```

Ce projet est compatible avec la version 6 d'Oracle JDK et supérieure.

## Chargement Dynamique

Les fichiers .class suivants qui correspondent aux types de créatures, sont chargés dynamiquement

- BouncingCreature.class
- EnergySearchingCreature.class
- SmartCreature.class
- StupidCreature.class

Lors d'appel "mvn compile", le plugin maven-antrun-plugin, déplace ces fichiers du bin\creatures à pluginsrepo\creatures. Contrairement à bin\creatures, ce dernier répertoire n'est pas sur le classpath, ce qui fait que ces fichiers sont chargés dynamiquement par le URLClassLoader.

Comme les tests unitaires dépendent de ces .class, il faut leur faire sous Eclipse.

## Version de Départ

La version de départ s'agit du [corrigé du TP 5](#).

## La Composition des Comportements

La classe `EnergySearchingCreature` a été rajouté pour modéliser une créature qui recherche des points d'énergie. Plus tard, ce comportement a été composé avec celui de `SmartCreature` (maintient vitesse de direction moyenne de ses voisins) et `BouncingCreature` (rebondit sur les bords), pour que les objets de ces dernières classes puissent aussi rechercher des points d'énergie.

Faut de temps, les patrons de conception, n'ont pas été implémenté pour la composition des comportements, mais nous pensons que le pattern "Strategy" peut être une bonne idée, pour cela, car il permet de changer l'algorithme de la méthode `act()` pendant l'exécution.