

## Descriptif des différentes classes de TurtleKit

### ***CommandEditorPanel***

Classe héritant de PythonEditorPanel (une classe standard Java), permettant d'interagir avec la simulation en utilisant des scripts Python.

### ***Launcher***

Classe abstraite héritant de la classe Agent et permettant de mettre en place la simulation. On dispose donc des méthodes **activate()**, **live()** et **end()** pour l'agent, et de méthodes permettant de contrôler la simulation (changement de la taille, mode « Wrap » ou non, remise à zéro), ainsi que de l'initialiser (**initializeVariables()**).

La classe **HuntLauncher** hérite de cette classe pour la rendre fonctionnelle.

### ***Observer***

Les **Observer** sont des agents permettant d'observer le monde, les individus qui s'y trouvent et leurs propriétés et d'en donner une certaine représentation (graphiques ...).

### ***TopMessage***

Classe vide héritant de la classe **Message** de MadKit. Permet de créer des messages.

### ***TurtleActivator***

L'activateur qui permet d'appeler la prochaine méthode de classe **Turtle** dans la simulation. La prochaine méthode à exécuter est retournée à chaque itération dans une chaîne de caractères.

### ***Viewer***

Un **Viewer** implémenté à partir de la classe **Observer** donne une représentation globale du monde (chaque turtle est représentée par un carré).

### ***LauncherGui***

L'interface graphique permettant de lancer la simulation, l'arrêter, la faire fonctionner en pas à pas, régler la vitesse et afficher les messages de débogage ...

### ***Patch***

La représentation d'une case du monde, sur laquelle on peut déposer un objet, et qui est liée à ses voisins.

## ***PatchVariable***

Un objet qu'on peut déposer dans une case du monde (classe **Patch**) permettant de représenter un taux de phéromones.

## ***Flavor***

Idem **PatchVariable**, avec une implémentation différente.

## ***PythonCommandCenter***

Interpréteur Python.

## ***Turtle***

Une **Turtle** sous forme d'agent, avec quelques actions basiques pour le déplacement, pour calculer des distances, pour marquer notre position...

## ***TurtleEnvironment***

Un agent qui représente le monde, sous forme d'une grille de classe **Patch**. Il permet d'initialiser toutes les cases du monde, et de savoir si il y a une **Turtle** à un endroit donné.

## ***TurtleScheduler***

L'ordonnanceur des **Turtle**.