

Dec 13, 18 12:09    Annexe.java    Page 1/2

```


/**
 * Pour représenter des types de déplacement
 * (au hasard ou ciblé)
 */

enum MovingMode {RANDOM_MOVE, TARGET_MOVE};

/**
 * Pour modéliser les fonctionnalités d'entités évoluant au cours du temps
 */
interface Updatable
{
    void update(float dt, Environment env);
}

/**
 * Pour modéliser des entités dotées d'une position
 */
abstract class Positionable
{
    private Vector position;

    public Positionable(Vector position) {}

    public Vector getPosition()
    {
        return position;
    }

    public void setPosition()
    {}
}


```

Dec 13, 18 12:09    Annexe.java    Page 2/2

```


/**
 * Pour modéliser l'environnement simulé
 * Vous supposerez qu'un programme principal existe,
 * qu'il crée un objet Environment, et appelle en boucle sa méthode
 * simulate. Cette méthode permet de faire évoluer les entités impliquées
 * au cours du temps (au moyen de méthode update, telle que dictées par
 * l'interface Updatable).
 */
class Environment
{
    // ensemble de brindilles
    private List<WoodChip> woodchips = new ArrayList<>();
    // ensemble de termitières
    private List<Termitary> termitaries = new ArrayList<>();

    // ajoute une termitière à une position donnée
    public void addTermitaryAt(Vector position) {}

    // ajoute une brindille à une position donnée
    public void addWoodChipAt(Vector position) {}

    // supprime une brindille donnée de l'ensemble de brindilles
    public void removeWoodChip(WoodChip woodChip) {}

    // retourne true s'il existe une termitière à la position donnée
    public boolean existTermitaryAt(Vector position) {}

    // retourne une brindille en collision avec une termite
    // suffisamment proche
    public WoodChip getCollidingWoodChip(Termit termit){ return null; }

    // simule les termitières en faisant appel à leur
    // méthode update
    public void simulate(float dt) {}

    // supprime une termite de la termitière à la position donnée
    public void removeTermitFromTermitary(Termit t, Vector TermitaryPosition)
    {}

    // ajoute une termite dans la termitière à la position donnée
    public void addTermitToTermitary(Termit t, Vector TermitaryPosition)
    {}
}


```