

# MANUEL DEV – Baba is You

## 1- Architecture

Le projet se décompose en **3 package** principaux.

### 1.1 fr.baba.app

Ce package contient une unique classe *Game* qui permet de lancer le jeu et les niveaux en analysant les options données au lancement du programme. C'est d'ailleurs dans cette classe que se trouve le *Main* principal.

### 1.2 fr.baba.engine

Ce package contient l'essentiel du jeu, c'est à dire toute la logique et l'affichage sur fenêtre. Plusieurs packages sont compris dans ce dernier :

- `fr.baba.engine.boardElements` : Ce package donne une représentation des éléments du jeu à travers l'interface *TileObject*. En effet, chaque *TileObject* se voit attribuer un token (de l'enum *Token*) lui donnant sa nature (ex : *sprBaba*, *nounBaba*, *Win*, *Melt* ...). Plus précisément, chaque token se voit attribuer un *Type* (de l'enum *tokenType*) qui est utilisé pour déterminer quel constructeur appeler dans le cadre de l'instanciation d'un nouveau *TileObject*. Par exemple, `new Operator(Token.sprBaba)`; lève une erreur car le type associé à *sprBaba* est *sprite* et non *operator*. De plus, pour associer les nouns aux sprites (*nounBaba* → *sprBaba* par exemple), un attribut (*rep*) de l'enum *Token* est dédié spécialement pour les nouns et contient un *sprite*, permettant ainsi de retrouver le *sprite* associé au *noun*.
- `fr.baba.engine.property` : Ce package contient toutes les manipulations faites sur les propriétés c'est à dire la détection des règles (*PropertyUpdater*), l'application des règles de destruction (*RemoveProperties*), l'application des règles de mouvement (*MoveProperties*). Ces trois classes ne sont pas vouées à être instanciées puisqu'il s'agit en fait d'une pile de fonctions *static*.
- `fr.baba.engine.board` : Ce package contient la classe *Board* qui est basiquement le plateau de jeu, c'est à dire une matrice de cellules, celles-ci contenant un ou plusieurs *TileObject*. Cette classe propose des manipulations basiques sur le plateau de jeu, comme ajouter un objet, supprimer un objet ou transférer un objet. De plus, une seconde matrice permet de stocker des *TileObject* qui seront interprétés comme des objets d'arrière plan. *BoardDisplayer* est une classe auxiliaire qui se charge de proposer l'affichage du plateau de jeu.

Tout ceci est utilisé dans une classe *Level* qui contient la boucle principale du jeu. La classe *LevelLoader* permet quant à elle de créer une instance de *Level* à partir d'un fichier.

### 1.3 fr.baba.utils

Ce package contient un enum *Direction* associant les directions cardinales à des vecteurs ainsi qu'une classe *EventListener* permettant de récupérer les actions de l'utilisateur.

## 2 – Corrections/Améliorations apportées après soutenance

La classe *Board* comportait au moment de la soutenance 800 lignes. Ceci nous a été signalé comme problématique. S'en est suivi la création du package *fr.baba.engine.property* ainsi que la classe *BoardDisplay* afin de remédier à cela. De plus, nous avons ajouté le parser de ligne de commande et le loader de fichier au projet afin de le boucler. La propriété « Boom » et le nom « Bomb » ont été ajoutés. Enfin, une amélioration de la javadoc a eue lieu.

## 3 – Problèmes rencontrés

La seule fonctionnalité non implantée est *--execute mot1 mot2 mot3*. Le format des mots n'étant pas le même que dans le loader de fichier, il aurait fallu créer un autre parser afin d'extraire les règles adéquates, ce qui est faisable mais par faute de temps nous n'avons pas pu le faire. De manière générale, trouver une modélisation convenable des entités de jeu s'est avéré être compliqué.