

Groupe 2

L3

Manuel développeur

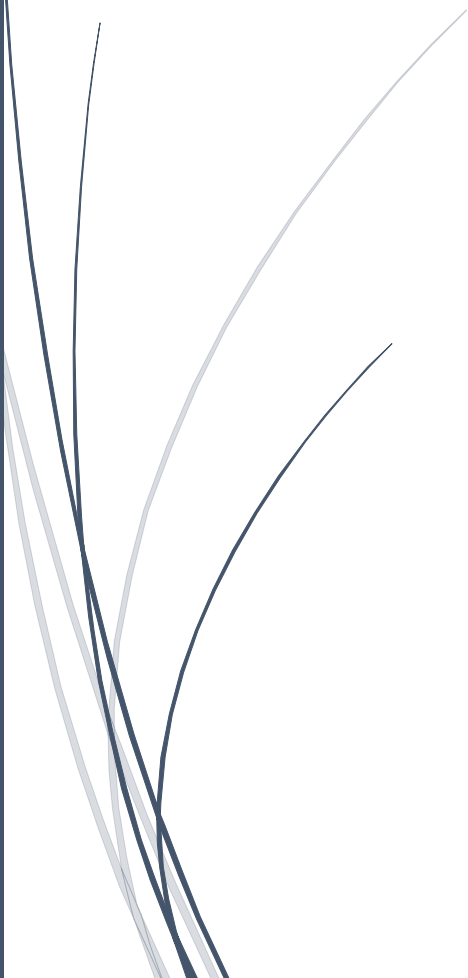
MEMBRES DU GROUPE :

PINTO Bryan

AOUNALLAH SELMA

# JAVA

# Baba Is You



# Architecture

Le projet est divisé en 7 packages :

- fr.umlv.board : il contient la classe Board qui a pour attribut les données d'un niveau. Les données sont stockées dans une Map<String, Set<Element>> nommée dataBoard, « String » représente le type d'un élément (« Wall » par exemple pour les murs) auquel on attribue un set contenant les éléments avec ce type. Les règles sont stockées dans un Set<Rules> nommées rulesOfGame. Les limites du niveau sont aussi stockés dans des attributs.
- fr.umlv.control : il contient la classe et l'enum :
  - Controller qui contient toute la logique du jeu tel que l'application, l'ajout et la suppression des règles. Elle permet de rafraichir la position des éléments et contient la boucle principale du jeu avec la gestion des arguments.
  - Direction, qui représente les directions où le joueur peut se diriger.
- fr.umlv.display : il contient la classe Graphic qui permet l'affichage du niveau grâce à ses méthodes de dessins. Elle prend comme attribut « widthSquare » et « heightSquare » qui représente les dimensions d'une case de jeu, « columnOut » et « lineOut » qui représente les coordonnées du coin haut gauche de la zone de jeu et les attributs fond qui représente le fond de la fenêtre et clear qui représente un rectangle noir pour actualiser l'affichage du niveau.
- fr.umlv.items : il contient toutes les classes en rapport avec les items, tel que (« Baba », « Water », « Is », « Push » etc...) ainsi qu'une classe abstraite « Item » pour factoriser le code en commun de chaque type d'item et une interface « Element » pour regrouper tous les items. L'enum « ObjectConstructor » contient le type des noms et le chemin vers leurs images. Elle est nécessaire pour éviter d'effectuer des new à chaque fois qu'un objet doit être transformé.
- fr.umlv.loader : il permet de charger un niveau avec sa classe levelLoader. Elle contient des méthodes qui vont lire, et traiter les données d'un fichier.
- fr.umlv.player : il contient la classe Player qui contient comme attribut un HashSet<String> nommé players qui regroupe tout les types d'éléments que le joueur peut contrôler.
- fr.umlv.rules : il est composé de la classe Rules et de l'enum GroupsOfElements. La classe contient les attributs left, operator et right qui représente la règle.

L'enum représente tous les groupes d'Element, Nouns contient tous les types de noms, Operators contient tous les types d'opérateur et Properties contient tous les types de propriétés.

## Amélioration après la soutenance

On a ajouté l'enum ObjectConstructor pour enlever les new qui consomment beaucoup niveau performance. L'implémentation des règles et la lecture de fichier de niveaux ont été achevées. On a ajouté les propriétés « Sink », « Melt », « Hot » et « Vanish » et le nom « Karl » puis corrigé le/les problème(s) de certaines propriétés. On a remplacé la plupart des boucles par des Stream. On a créé les fichiers build.xml et actualiser la javadoc.

## Problèmes rencontrés

La plus grosse difficulté à été de choisir une bonne structure pour le projet, on pensait au début à faire une matrice de listes contenant des listes d'éléments, mais on a vite changé de structure car on parcourait toutes les cases à chaque fois même si elles étaient vides et cette structure rendait la tâche plus difficile pour certaines fonctions.

La propriété « Push » n'était pas fonctionnelle tout de suite, nous avons dû réfléchir à beaucoup de méthodes avant d'arriver à notre résultat actuel.