Explication diagramme de classe :

La classe Rules est la classe qui permet de créer nos règles pour nos cellules. Une règle est composée d’un tableau de règles de naissance et d’un tableau de règles de survie.

La classe cellule est la base de notre application. Une cellule est composée d’un nextTimeStatus. Le nextTimeStatus est un entier. Il correspond au statut que va avoir la cellule à la prochaine Update. Il peut prendre 3 valeurs différentes : -1 la cellule meurt, 0 la cellule ne change pas de statut, 1 la cellule naît. De plus, on a un booléen pour avoir le statut actuel de la cellule et les coordonnées x et y de la cellule dans la grille. La méthode update transforme la variable nextTimeStatus.

La classe GrilleCellFactory est une classe qui créer un tableau de cellule à partir d’une taille.

La classe Monde est une classe qui prend en paramètre une taille de monde. C’est cette dernière qui va appeler la GrilleCellFactory.

La classe Dieu est le manager de notre application. C’est un singleton. Il possède une instance de dieu, une liste de Rules, une liste de Cellule et une liste traite. C’est lui qui possède l’algorithme de l’application. La méthode evolution() parcourt le tableau pour déterminer le prochain état des cellules. La méthode evolveCell() détermine l’état d’une cellule en fonction des règles du monde. C’est cette dernière méthode qui est appelée dans la méthode evolution() pour toutes les cellules de la grille du monde. La dernière méthode très importante de cette classe est la classe updateCells(). Cette dernière met toutes les cellules traitées à jour. Cette classe hérite de l’interface Notifiable.

La classe saver est la classe qui permet d’enregistrer dans un fichier le monde qui représente un motif. Elle hérite de l’interface ISaver. Cette classe prend en paramètre le monde et un fichier. La méthode save() permet d’écrire les coordonnées de chaque cellule du monde dans le fichier.

La classe stub permet de stocker les motifs de base disponible dans l’application. Pour chaque motif, le stub crée un HashMap avec un nom et l’objet Rules.

Notre classe BoucleDeJeu est la boucle de jeu de notre application. C’est cette boucle qui a chaque tour fait appel à la classe Dieu et permet de premièrement faire la méthode evolution() puis la méthode updateCells().